ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

**Artisan Technology Group**
(217) 352-9330 | sales@artisantg.com | artisantg.com

**Full-service, independent repair center**
with experienced engineers and technicians on staff.

**We buy your excess, underutilized, and idle equipment**
along with credit for buybacks and trade-ins.

**Custom engineering**
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

**Expert team | Trust guarantee | 100% satisfaction**

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Astronics / Ballard Technology LP1553-3C* at our website: ***Click HERE***

# CoPilot User's Manual

## Version 5 & 6

**Preliminary Rev D**

CoPilot®

Avionics Databus Tools



Ballard TECHNOLOGY

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Copyright © 2008, 2009, 2010, 2012

by




**An Astronics Company**

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

Microsoft® Windows®, Access®, Excel®, Visual Basic® and TerraServer® are registered trademarks of Microsoft Corporation.
All other product names or trademarks are the property of their respective owners.

MA157-20120104

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# License Agreement for CoPilot Software

**IMPORTANT (READ CAREFULLY)**

This document is a legal agreement between you, the end user (LICENSEE), and Ballard Technology, Inc. (BALLARD) for the BALLARD software product identified above (SOFTWARE), which includes associated media, printed materials, and any online or electronic documentation. By installing, copying, or otherwise using the SOFTWARE you are agreeing to become bound by the terms of this agreement, which includes the software license, the Safety Warning, and the version of BALLARD's Limited Warranty (AD195, unless otherwise mutually agreed by written contract) effective when the product was obtained from BALLARD (e.g., at the time of purchase).

If you do not agree to the terms of this agreement, promptly return the product to Ballard Technology. The amount you paid for an unused product will be refunded.

GRANT OF LICENSE: Regardless of how the SOFTWARE was obtained, whether without charge or through a purchase, the SOFTWARE may only be used either in conjunction with a specific BALLARD hardware product or for a brief period of time for evaluation prior to purchase. The software may only be used to control or operate the specific BALLARD hardware product (LICENSED HARDWARE), identified by part number and serial number, specified in the agreement (purchase or other contract) between BALLARD and LICENSEE. Additional copies may be made and may reside on other computers as long as they are used in connection with the LICENSED HARDWARE (e.g., to prepare or analyze data). You may not modify, adapt, translate, reverse engineer, decompile, or disassemble the SOFTWARE.

OWNERSHIP OF SOFTWARE: As the LICENSEE you may own the magnetic or other physical media on which the SOFTWARE is originally or subsequently recorded or fixed; but BALLARD retains title and ownership of the SOFTWARE recorded on the original and all subsequent copies of the SOFTWARE, regardless of the form or media on which it may exist. This license is not a sale of the original SOFTWARE or any copy.

UPGRADES: If the SOFTWARE is an upgrade, then it is considered a part of the original SOFTWARE package and may be used only in connection with the LICENSED HARDWARE.

COPY RESTRICTIONS: The SOFTWARE and accompanying written materials are copyrighted. You may make copies of the SOFTWARE for backup, for use with the LICENSED HARDWARE, or for evaluation as stated in this agreement. Copies may not be sold. Evaluation copies may be freely distributed. You must reproduce and include the copyright notice on all copies, and the full text of this agreement must be readily accessible within all copies.

CONFIDENTIAL INFORMATION: All information, whether in printable or computer readable form, marked as confidential or proprietary to BALLARD (e.g., BALLARD TECHNOLOGY CONFIDENTIAL) is excluded from this license and may only be used as authorized by BALLARD in a separate written document.

TERMINATION: This license is effective until terminated. Without prejudice to any other rights, BALLARD may terminate this license if you fail to comply with the terms and conditions of this agreement.

MISCELLANEOUS: This agreement is governed by the laws of the State of Washington.

**SAFETY WARNING**

Ballard products are of commercial grade and therefore are neither designed, manufactured, nor tested to standards required for use in critical applications where a failure or deficiency of the product may lead to injury, death, or damage to property. Without prior specific approval in writing by the president of Ballard Technology, Inc., Ballard products are not authorized for use in such critical applications.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Contents


## Part I: CoPilot Standard-Basics 21

Part I Overview ..... 21

## Introduction 23

Welcome to CoPilot ..... 23
New in CoPilot 6 ..... 23
CoPilot 6 Basics ..... 25
Start-up Modes ..... 25
Analyzing the Bus ..... 25
The Ribbon Interface ..... 26
Overview ..... 26
Project Tab (Backstage View) ..... 27
Home Tab ..... 27
View Tab ..... 28
1553 Tab ..... 28
429 Tab ..... 28
664 (AFDX) Tab ..... 28
708 Tab ..... 29
Scripting Tab ..... 29
CoPilot Installation Requirements ..... 29
Install and Open CoPilot ..... 30
Supported Protocols ..... 30
MIL-STD-1553 ..... 30
ARINC 429 ..... 31
ARINC 664/AFDX ..... 32
ARINC 708 ..... 32
Serial ..... 32
Discrete Inputs and Outputs (DIO) ..... 32
Other Protocols ..... 34
About Ballard Technology ..... 34
How to Use This Manual ..... 34
Other Resources ..... 35
Quick Start ..... 36
Support and Service ..... 37

## CoPilot Operation 39

CoPilot Operation Overview ..... 39
The CoPilot User Interface ..... 40
The CoPilot Environment ..... 40
Start Page ..... 42
“Project” Backstage View ..... 42
Workspaces ..... 43
Docking Panes ..... 43





Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CoPilot Views ..... 46
Menus & Toolbars ..... 47
CoPilot Options ..... 50
Environment Options ..... 50
Protocol Options ..... 51
MIL-STD-1553 Properties ..... 51
ARINC 429 Properties ..... 52
AFDX (ARINC 664) Properties ..... 53
The CoPilot Project ..... 53
CoPilot Projects Overview ..... 53
Creating New Projects ..... 54
Saving and Opening a Project ..... 55
Project File Management ..... 56
Running a Project ..... 58
CoPilot Conventions ..... 59

**Hardware Interface 61**

CoPilot-Compatible Hardware ..... 61
Licensing CoPilot with Hardware Keys ..... 61
Hardware Installation and Configuration ..... 62
Adding Hardware Devices to CoPilot ..... 63
Hardware Configuration ..... 64
Advanced Hardware Settings ..... 66
Error Injection ..... 66
IRIG Timer ..... 66
Syncs and Triggers ..... 67
Variable Transmit Amplitude ..... 69
Hardware Definition Import & Export ..... 69

**Sequential Monitoring 71**

How to Record and View Sequential Data ..... 71
Recording Databus Traffic ..... 71
Viewing Monitor Records ..... 72
Optimizations for Recording to Disk ..... 72
Recovering Monitor Data ..... 73
Capture Filtering ..... 74
What is Monitor Capture Filtering ..... 74
Sequential Monitor Capture Filters ..... 74
Configuring a Sequential Monitor ..... 75
Sequential Monitor Recording Options ..... 75
Saving and Loading Sequential Monitor Data ..... 75
Analyzing Recorded Data in a Sequential Monitor View ..... 76
Common Sequential Monitor View Features ..... 76
Monitor View Toolbars ..... 76
Time Correlation ..... 78
Display Filters ..... 78
Export Sequential Monitor Data ..... 78
Import Sequential Monitor Data ..... 79
Changing Engineering Unit Interpretations ..... 79
Data Plot ..... 79
Custom Display Settings ..... 80
Optimize Columns ..... 80
Bookmarks ..... 80
Software and Hardware Playback ..... 80



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Engineering Units 83

Engineering Units Overview ..... 83
Create an Engineering Units Definition ..... 84
Engineering Units Value Display ..... 85
Copying Engineering Units Definitions ..... 88
Engineering Units Database Definitions ..... 88
Save Fields and Messages to the Database ..... 88
Database Import & Export ..... 89

## Generate and Edit Data 91

Data Control Overview ..... 91
Introduction ..... 91
Data Editing Tools ..... 91
Data Generating Tools ..... 91
Engineering Units and Raw Values ..... 92
Engineering Unit Editor ..... 92
Raw Editor ..... 93
Data Generator ..... 93
Overview ..... 93
Adding Fields or Labels ..... 94
Accessing Field and Label Properties ..... 94
Item Properties ..... 95
Function Types ..... 96
Building a Data List or Complex Ramp ..... 97
Data Modifier ..... 97

## Standard Displays 101

Display Windows Overview ..... 101
OLE Documents ..... 103
Creating OLE documents ..... 105
Saving OLE Documents ..... 105
Opening OLE Documents ..... 106
Working with OLE Documents ..... 107

## MIL-STD-1553 Protocol 109

MIL-STD-1553 Overview ..... 109
1553 Databus Configuration ..... 110
Overview ..... 110
Hardware Explorer Object Hierarchy ..... 110
Terminal Simulation Assignment (BC, RT, and Monitor Simulation) ..... 111
Pausing Databus Activity ..... 112
MIL-STD-1553 Channel Configuration ..... 112
Advanced Features ..... 113
Mode Codes Configuration ..... 113
Broadcast Mode Configuration ..... 114
Selectable Bus Termination ..... 114
Bus Controller Operation ..... 115
Bus Controller Overview ..... 115
BC Configuration ..... 116
Controlling the BC with Schedules ..... 118
BC Schedule ..... 118
BC Schedule Overview ..... 118
BC Properties for Scheduling ..... 119




Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

User Defined BC Schedules ..... 120
BC Schedule Import and Export ..... 121
BC Schedule Editing User Schedules ..... 122
Scheduling BC Messages ..... 123
Scheduling Subframes ..... 123
Scheduling Gaps ..... 124
Scheduling Branches ..... 125
Scheduling Output Signals (DIO Pulses) ..... 127
Scheduling No Operations ..... 128
Scheduling BC Pauses ..... 128
BC Messages ..... 128
Messages Overview ..... 128
Message Configuration ..... 130
Creating and Editing Messages ..... 133
Message Retries ..... 134
Transmitting BC Messages ..... 135
BC Message Rate ..... 135
Message Data ..... 135
Remote Terminal Operation ..... 136
Remote Terminal Overview ..... 136
Detecting and Adding an RT ..... 137
RT Terminal Simulation Assignment ..... 138
RT Configuration ..... 139
RT Context Menu ..... 140
RT Properties ..... 141
Configurable RT Response Time ..... 142
RT Subaddress Selection ..... 143
RT Mode Code Configuration ..... 144
RT Status Word Configuration ..... 144
RT Configuration Options ..... 146
RT Configurations in the Database ..... 147
SA Configuration ..... 148
SA Context Menu ..... 149
SA Properties ..... 150
SA Data Editing ..... 150
Engineering Unit Fields ..... 151
Data Fields Overview ..... 151
Field Definitions ..... 152
Create New Field ..... 154
Load Field from Database ..... 155
Save Fields to Database ..... 157
Edit Fields ..... 158
Copy Data Field Definitions ..... 159
Field Context Menu ..... 160
Field Properties ..... 160
Field Data ..... 161
Engineering Unit Interpreter Definition ..... 161
BCD Interpreter ..... 163
BNR Interpreter ..... 163
BNR 48-bit Interpreter ..... 164
BNR 64-bit Interpreter ..... 164
CAPS 48-bit Interpreter ..... 164
Custom Script Interpreter ..... 164
DEC Floating-Point 64-bit Interpreter ..... 166
Decimal Interpreter ..... 166
Discrete Interpreter ..... 166



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IEEE 754 32-bit Interpreter....167
IEEE 754 64-bit Interpreter....167
MIL-STD-1750 48-bit Interpreter....167
MIL-STD-1750 64-bit Interpreter....167
Simple Scalar Interpreter....167
Editing Field Data....168
Bus Monitor and Sequential Monitoring....168
Bus Monitor Overview....168
Bus Monitor versus Sequential Monitor....169
Concurrent Monitoring....170
Bus Monitor Configuration....171
Monitor Data Collection Modes....172
Monitor Recording Modes....174
Monitor Capture Filters....175
Saving Monitor Data....178
Saving Monitor Files....178
Saving Segmented Monitor Files....179
Saving Data from Heavily Loaded Buses....180
Monitor View....180
Accessing the Monitor View Window....181
Monitor View Context Menu....183
Monitor View Tool Bar....184
Monitor View Full Display....186
Monitor View Form Display....186
Monitor View User-Defined Displays....188
Monitor View Radix....189
Monitor View Time-Tag....190
Monitor View Display Filters....190
Monitor View Bookmarks and Breakpoints....193
Monitor View Search Options....194
Monitor View Step and Go To Commands....194
Monitor View Recording Controls....195
Monitor View Message Editing....195
Monitor View Data Plotting....196
Monitor View Export Options....197
Monitor View Import Options....199
Monitor View Print Options....200
Viewing MIL-STD-1553 Data....201
Message/Subaddress View....201
Message View Data....202
Message View Fields....203
Engineering View for MIL-STD-1553....203
Engineering View Display Menu....204
Available display columns....205
Drag and Drop to Engineering View....205
Engineering View Column Options....206
Engineering View Display Status Indication....207
Reporting Out of Range and Timeout Conditions....207
Protocol Browser....207
Graphical Displays....208
Error Injection....208
Error Injection Overview....208
Error Injection View....209
Defining the Error....209
Tagging Messages for Error Injection....210
Error Application and Control....211



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MIL-STD-1553 Database ..... 212
Overview ..... 212
Using the database ..... 213
Sharing a database ..... 213
XML database import/export utility ..... 213
MIL-STD-1553 Scripting Events ..... 214

**ARINC 429 Protocol 217**

ARINC 429 Overview ..... 217
ARINC 429 Protocol Summary ..... 217
Hardware Explorer Object Hierarchy ..... 219
ARINC 429 Channels ..... 220
Overview ..... 220
Setting the Equipment ID ..... 220
Auto-Detect Equipment ..... 220
Assigning Equipment IDs ..... 220
Identifying an Unknown Equipment ID ..... 221
Channel Options ..... 222
Channel Speed Settings ..... 222
Monitor Options (Sequential Logging) ..... 223
Channel Options ..... 223
Receive Channel Specific Options ..... 223
Transmit Channel Specific Options ..... 224
Scheduling ..... 225
Default Schedule ..... 226
Software Timer Option with Default Scheduling ..... 227
Free-Running Schedule ..... 228
Custom User Schedules ..... 228
Schedule Import and Export ..... 231
Channel Summary ..... 231
Copying Channel Settings ..... 232
ARINC 429 Messages and Fields ..... 232
ARINC 429 Messages ..... 232
Message Properties and Options ..... 232
Adding Messages to a Channel ..... 236
Message Interpreters ..... 238
Edit Data ..... 239
List Buffering ..... 239
Using SDI Information ..... 240
Fields ..... 240
Engineering Units ..... 241
Helpful Hints ..... 242
Defining Parity ..... 243
Sequential Monitoring for ARINC 429 ..... 244
Overview ..... 244
Configuration ..... 244
Options ..... 244
Capture Filters ..... 245
Monitor Data Collection Modes ..... 245
Monitor recording modes ..... 246
Data collection settings ..... 247
Saving sequential monitor data ..... 247
Analyzing recorded sequential monitor data ..... 247
Monitor View ..... 247
Protocol Browser ..... 251



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Viewing ARINC 429 Data ..... 251
Engineering View with ARINC 429 Data ..... 251
Assigning the Data to Watch ..... 252
Engineering View Display Status Indication ..... 252
Engineering View Display Menu ..... 253
Available display columns ..... 254
Customizing the display ..... 254
Engineering View Column Options ..... 255
Protocol Browser ..... 255
Graphical Displays ..... 256
The CoPilot 429 Database ..... 256
Overview ..... 256
Loading from the database ..... 256
Saving to the database ..... 257
Sharing a database ..... 258
XML database import/export utility ..... 258
Helpful Hints ..... 259
Error Injection ..... 259
ARINC 429 Scripting Events ..... 260

**ARINC 708 Protocol 263**

ARINC 708 Overview ..... 263
ARINC 708 Protocol Summary ..... 263
Hardware Explorer Object Hierarchy ..... 263
CoPilot for 708 Overview ..... 264
Simulating a Weather Radar System in CoPilot ..... 264
Transmitter-Receiver Unit ..... 264
Display Unit ..... 264
Control Panel ..... 265
CoPilot for 708 Simulation and Analysis Tools ..... 266
Data Recording ..... 266
Data Propagation ..... 266
Displaying and Modifying Messages ..... 267
Analyzing Data with Software Playback ..... 267
Monitor and Display Data ..... 269
How to Monitor and Display Data ..... 269
Monitoring Data ..... 269
Displaying Data ..... 270
Viewing the Radar Image ..... 271
Configuring the Radar Screen ..... 272
Specifying Weather Colors ..... 274
Displaying Status, Faults, & Warnings ..... 274
Changing the Scan Area of the Radar Screen ..... 275
Saving and Loading a Radar View Configuration ..... 275
Filtering the Radar Display ..... 276
Displaying Data in Two Radar Screens ..... 276
Viewing the Message and Raw Hex Lists ..... 277
Displaying the 055 Label ..... 279
Stepping through the Message List ..... 279
Searching the Message List ..... 280
Setting and Clearing Bookmarks ..... 280
Configuring the List View Columns ..... 281
Displaying Range Bin Data ..... 281
Record and Reuse Data ..... 282
How to Record and Reuse Data ..... 282



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Recording Data ..... 283
Setting Recording Options ..... 283
Working with Datasets ..... 284
Reusing Data ..... 284
Copying Data ..... 285
Exporting Data ..... 285
Importing Data ..... 285
Modify and Transmit Data ..... 286
How to Modify and Transmit Data ..... 286
Modifying a Message in Engineering Units ..... 287
Modifying a Header in Hexadecimal ..... 288
Transmitting Data ..... 289
Setting Transmit Options ..... 290
Software Playback ..... 290
How to Analyze Data with Software Playback ..... 290
Running Software Playback ..... 291
Setting Software Playback Options ..... 292
Moving the Playback Resume Point ..... 292
Advancing Software Playback in Steps ..... 293
Controlling Software Playback Speed ..... 293
Breakpoints ..... 294
Navigating between Breakpoints ..... 294
Ignoring Breakpoints ..... 295
Setting the Start and End Index ..... 295
Controlling a T-R Unit ..... 296
How to Control a T-R Unit with CoPilot ..... 296
Using CoPilot 429 Virtual Instruments ..... 297

**ARINC 664 / AFDX Protocol 299**

ARINC 664/AFDX Overview ..... 299
ARINC 664/AFDX Protocol Summary ..... 299
Hardware Explorer Object Hierarchy ..... 299
Monitoring AFDX Networks ..... 300
Analyzing Receive Traffic ..... 300
Simulating Transmit Traffic ..... 300
AFDX Views ..... 300
Network View ..... 300
VL View ..... 301
Port View ..... 301
Engineering View with ARINC 664/AFDX ..... 301
ARINC 664/AFDX Hardware Features ..... 301
Card Configuration ..... 301
Card Context Menu ..... 301
Card Properties ..... 302
Card DMA Mode ..... 302
Sequential Monitoring for ARINC 664/AFDX ..... 302
How to Record and View Sequential Data ..... 302
Recording Databus Traffic in the Monitor ..... 302
Viewing Monitor Records ..... 303
Sequential Monitor View Capabilities ..... 303
Using the Monitor Controls ..... 303
Saving the Monitor View ..... 303
Exporting Data ..... 303
Filtering Monitor Recordings ..... 304
What is Monitor Capture Filtering ..... 304



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Sequential Monitor Capture Filters ..... 304
Network Capture Filters ..... 304
VL Capture Filters ..... 304
Port Capture Filters ..... 304
Sequential Monitor Modes ..... 305
How to limit the size of the recorded data file ..... 305
Timed Run Mode ..... 305
Header-Only Setting ..... 305
Monitor Data Collection Modes ..... 305
Monitor recording modes ..... 306
Analysis of the Sequential Monitor Record ..... 307
How to analyze a single record ..... 307
Display Filter Settings ..... 307
Search Options ..... 308
Exporting Sequential Monitor Data ..... 308
CoPilot AFDX Networks ..... 308
Why Are There Three Networks? ..... 308
Differences between Redundant and Independent Networks ..... 308
Displaying Statistics with the Network View ..... 309
Network Error Injection ..... 309
Simulating VLs and Ports ..... 309
How to Simulate VLs and Ports ..... 309
Configuring VLs and Ports ..... 309
Virtual Links ..... 309
Ports ..... 310
Functional Data Sets & Data Sets ..... 310
Displaying VL Statistics with VL View ..... 311
Displaying Port Information with Port View ..... 311
Summary Mode ..... 311
Packet Edit & Display Mode ..... 311
Configuration Mode ..... 312
Error Injection Mode ..... 312
Defining Field Engineering Units ..... 312
Engineering Unit Interpreters ..... 312
Embedding ARINC 429 Engineering Units ..... 313
Viewing Field Engineering Units ..... 313
Displaying Fields with the Port View ..... 313
Displaying Fields with the Engineering View ..... 313
Additional Field Displays ..... 314
Editing Field Engineering Units ..... 314
Editing of Fields in-place ..... 314
Editing Fields indirectly ..... 314
CoPilot Professional for Display and Edit ..... 315
Scheduling for Transmit ..... 315
How to Schedule VLs and Ports for Transmit ..... 315
Transmission Rate ..... 315
Sub-VL Scheduling ..... 315
AFDX Database ..... 316
What is the AFDX database ..... 316
Adding definitions to the AFDX database ..... 316
Loading definitions from the AFDX database ..... 316
Default VL definition for Auto-Detection ..... 316
XML import/export utility ..... 317
Error Injection ..... 318
What is Error Injection ..... 318
Controlling Error Injection ..... 318



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Network Parametric Frame-Time Frequency ..... 319
VL RSN Error Injection ..... 319
Port Error Injection ..... 319
Schedule Errors (Transmit Control) ..... 319
Frame Errors (Packet Control) ..... 319
Network Errors (Bus Control) ..... 320
ARINC 664/AFDX Scripting Events ..... 320

**Part 2: CoPilot Standard-Advanced Topics 323**

Part 2 Overview ..... 323

**Hardware Playback 325**

Hardware Playback Overview ..... 325
Preparing the Data Source for Hardware Playback ..... 327
Limiting 1553 Hardware Playback with Index Points ..... 328
Creating a Hardware Playback Export File ..... 328
Configuring Hardware Playback ..... 330
Configure a MIL-STD-1553 card for Hardware Playback ..... 330
Configure a 429 transmit channel for Hardware Playback ..... 331
Running Hardware Playback ..... 332
Pausing Hardware Playback with Breakpoints ..... 333
Specifying the Hardware Playback Resume Point ..... 334
Advancing Hardware Playback in Steps ..... 334

**Software Playback 337**

Software Playback Overview ..... 337
Software Playback Operation ..... 338
Engage Software Playback ..... 339
Select Software Playback Source ..... 340
Run Software Playback ..... 341
Control the Rate of Software Playback ..... 341
Pause Software Playback with Breakpoints ..... 342
Specify the Software Playback Resume Point ..... 343
Set Software Playback Start and End Index ..... 344
Software Playback Examples ..... 345
Combine Scripting with Software Playback ..... 345
Use Software Playback for Post-Analysis ..... 346
Share Data with Others Using Software Playback ..... 347

**Linking Objects 349**

Links Overview ..... 349
Linking Hardware Explorer Objects ..... 349
Linking Controls and Graphical Objects ..... 349
Creating a Link ..... 350
Viewing Link Status ..... 351
Linkable Properties of Objects ..... 351
Description ..... 351
MIL-STD-1553 Linkable Properties ..... 352
BC Properties ..... 352
BC Message Properties ..... 352
Subaddress Properties ..... 353
ARINC 429 Linkable Properties ..... 353
Channel Properties ..... 353



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Label Properties ..... 354
ARINC 664/AFDX Linkable Properties ..... 354
Network Properties ..... 354
VL (Virtual Link) Properties ..... 355
Port Properties ..... 355
FDS (Functional Data Set) Properties ..... 357
DS (Data Set) Properties ..... 357
Generic Linkable Properties ..... 357
Field Properties ..... 357
Card Properties ..... 357
Sequential Monitor Properties ..... 357
Helpful Hints ..... 358
Changes for Links in CoPilot 5 and 6 ..... 358

**CoPilot Automation 359**

CoPilot Automation Overview ..... 359
The CoPilot Object Model ..... 360
CoPilot Automation Examples ..... 363
Using Third-Party Automation Documents ..... 364

**Part 3: CoPilot Professional 367**

Part 3 Overview ..... 367

**Professional Displays and Controls 369**

Professional Display Windows Overview ..... 369
Professional Display Panes ..... 371
Professional Displays ..... 372
Strip View ..... 372
Strip View Properties ..... 373
Control View ..... 374
Overview ..... 374
Design and Operational Modes ..... 374
Customizing the View ..... 375
Adding Controls to the View ..... 375
The Control Selection Gallery ..... 377
Control Database ..... 379
Linking Controls ..... 380
Customizing a Control ..... 380
Using third-party ActiveX controls ..... 381
Mutually Exclusive Controls ..... 381
Controls and Scripting ..... 382
Quick View ..... 384
Quick View Library of Controls ..... 385
Configuring Quick View Controls ..... 386
Map View ..... 388
Map View Window ..... 389
Map View Context Menu ..... 390
Map View Properties ..... 390
Map View Maps ..... 391
Map View Objects ..... 392
Map View Object Options ..... 393
Map View Views ..... 393
Map View Grids ..... 394
Map View Waypoints ..... 394



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Other Map View Properties....395
TerraServer Maps....396
TerraServer Configuration....397
Scripting Displays....398
Controls....398

## Automated Test Environment (ATE) 399

Overview....399
Python Help....401
ATE Projects....401
Introduction....401
Scripts in Project Explorer....401
Master Script File....402
User Scripts Files....404
Running Scripts....405
Script Threading (Advanced Topic)....405
Script Alias References....406
Built-in Aliases....407
Description....407
The “shell” alias....407
The “atm” alias....407
The “output” alias....407
Automated Test Environment Components....407
Python Script Editor....407
Text Editor Features....408
Find and Replace Functionality....408
IDE Features....409
Syntax highlighting....409
Syntax check....409
Debugger....410
Importing modules....410
Autocomplete....410
Calltips....410
Tab nanny....410
Auto-indent....411
Line numbers....411
Indentation guides....411
Code folding....411
Object Browser....411
Object Alias Assignment Mechanics....412
Using Object Events....415
Python Data Types....416
Python Debugger....416
Overview....416
Using the Debugger....417
Automated Test Manager (ATM)....418
Overview....418
The Automated Test Manager Pane....419
Creating Tests....421
Running Tests....422
Test Output....423
Command Prompt....423
Overview....423
Context Menu Features....424
Output Pane....425



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Output Pane Overview ..... 425
CoPilot Output Stream ..... 425
Python Output Stream ..... 426
ATM Output Stream ..... 426
Other Output Streams ..... 426
Output Pane Toolbar ..... 426
Saving Contents ..... 426
Watch Window Pane ..... 427
Watch Window Pane Overview ..... 427
Details ..... 427
CoPilot Macros ..... 428
Running Macros with the Macro Dialog ..... 429
Accessing Macros to the Macro Toolbar (Version 5) ..... 429

**Project Security 431**

Overview ..... 431
Configuring Project Security ..... 431
Using Project Security ..... 431
Unrestricted Project Access ..... 431
Runtime Project Access ..... 431
Project Passwords ..... 431
Project Security Options ..... 432
Securing Proprietary Python Code ..... 433

**Appendix A: CoPilot Standard and Professional Differences 435**

**Appendix B: Hardware Object Icon Reference 437**

Object Icons ..... 437

**Appendix C: Supported Hardware and Protocols 441**

**Appendix D: Sample Projects 443**

**Appendix E: Control View Library 445**

Controls Library Overview ..... 445
Working with Controls ..... 447
Aircraft Instruments ..... 448
Aircraft Instrument Properties ..... 449
Airspeed Indicator ..... 450
Altimeter ..... 451
Artificial Horizon ..... 451
Automatic Direction Finder (ADF) ..... 452
Climb Rate Indicator (CRI) ..... 453
Compass ..... 453
Course Indicator ..... 454
Heading Indicator ..... 455
Horizontal Situation Indicator (HSI) ..... 455
Omni Bearing Indicator (OBI) ..... 456
Radio Magnetic Indicator (RMI) ..... 457
Turn Coordinator ..... 457
General Purpose Controls ..... 458



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Alphanumeric LED ..... 458
Angular Gauge ..... 459
Data Modifier ..... 460
Knob ..... 460
LED ..... 461
Linear Gauge ..... 461
Percent ..... 462
Selector Knob ..... 462
Slider ..... 463
Strip Chart ..... 463
Toggle ..... 465

**Appendix F: Script Examples 467**

Example Script Library ..... 467
Example Script Details ..... 467
Create a Binary String from a Decimal Number ..... 469
Increment Counter on Value Change ..... 470
Add Random Noise to Data Pattern ..... 471
Generate Data from Monitor Record Count ..... 471
Generate Random Data Values ..... 472
Generate Sine Wave Data Values ..... 472
Create a Simulated Monitor File in Excel ..... 473
Create and Display a Text File ..... 475
Trigger Monitor by Limit Conditions ..... 476
Trigger Monitor by Record Count ..... 477
Save Data to Excel ..... 477
Access a File Open or File Save Dialog ..... 478
Open a Web Page ..... 479
Create a Web Page ..... 479

**Appendix G: Regular Expressions 483**

Purpose ..... 483
Syntax ..... 483

**Appendix H: Legacy CoPilot Support 487**

CoPilot VB Scripting ..... 487
Script Pane ..... 489
Script Objects ..... 490
Script Object Properties and Methods ..... 491
Script Output ..... 493
Script Timers ..... 493
Scripting Principles ..... 495
Script Execution ..... 496
Sample Script Library ..... 497

**Appendix I: Custom 1553 Schedule Examples 501**

Initialization Routines ..... 501
Reusable Subframe Schedules ..... 502
Schedule Swapping and Schedule Modes ..... 503
DIO Controlled Schedule ..... 503
Software Controlled Schedule ..... 503
Status Word Controlled Schedule ..... 504



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

**Appendix J: Revision History** 505

**Glossary of Terms** 507

**Index** 515



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Part I: CoPilot Standard-Basics

## Part I Overview

This section covers the basic topics of CoPilot Standard. All of these basic features and all of the advanced features described in Part 2 of this manual are a part of CoPilot Standard. Part 3 describes the features of the full-featured CoPilot Professional. Included in this section are: general information, hardware operation/configuration, logging sequential monitor data, engineering unit interpretation, display windows, and protocol specific information.

This manual is describes operation of both CoPilot version 5 and CoPilot version 6. Most of the differences between version 5 and version 6 are described in Part 1 of this manual. The *New in CoPilot 6* section on *page 23* provides an overview of the recent CoPilot changes.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Introduction

## Welcome to CoPilot

CoPilot is a Windows-based multi-protocol software system developed by Ballard Technology, Inc. to simplify the simulation and testing of MIL-STD-1553, ARINC 429, ARINC 708, ARINC 664/AFDX and other protocols. The CoPilot environment can simultaneously operate a variety of avionics protocols, and other standard protocols. CoPilot's industry-leading Python scripting support (included with CoPilot Professional) further extends the capabilities by allowing users to control additional hardware and pass information between other applications.

CoPilot is designed to allow users to simulate and test avionics systems efficiently and easily. This comprehensive software package makes the perfect companion to Ballard's powerful hardware interfaces. Some of the interfaces supported by Ballard Technology include Ethernet, PCI, cPCI, PC104, PCMCIA, PMC, USB, VME, ISA (for version 5), PCMCIA (for version 5) and others. Information can be created, modified, and displayed in various radices (including binary, decimal, octal, and hexadecimal) or translated into an easy to read engineering unit format interactively while running. Data can also be monitored and stored in a sequential monitor file for later analysis.

CoPilot is available in two versions: CoPilot Standard and CoPilot Professional. CoPilot Professional includes all the capability of the standard version with the addition of graphical displays, strip charts, Python powered ATE (Automated Test Environment), and more.

This manual is describes operation of both CoPilot version 5 and CoPilot version 6.

## New in CoPilot 6

CoPilot 6 is the culmination of feedback and experience from hundreds of customers looking for a simpler user experience for both the casual and experienced user. CoPilot 6 retains the features of CoPilot 5 but adds improvements in the User Interface (UI), protocol analysis, start-up experience, and ease-of-use. The rest of this "New in CoPilot 6" section contains excerpts from the CoPilot 6 Getting Started Guide which came with your CoPilot software. Please refer to that document for the full text.

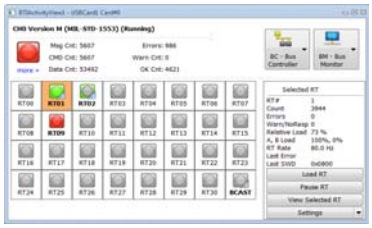
The following are a few of the things to look for in CoPilot 6:



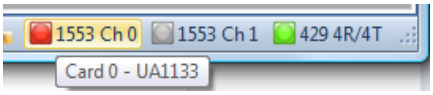
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- New Ribbon (toolbar) interface
- New 

tab on the ribbon shows the Backstage – an improvement to the CoPilot 5 Start Page with additional project management commands, tools and much more
- Protocol Analysis – new displays easily show bus activity with the ability to drill-into details and view status



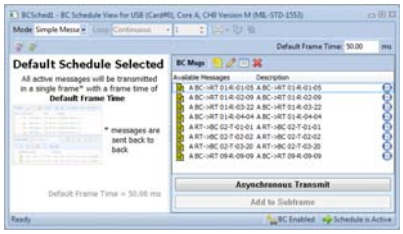
- Increased use of status indicators (including clickable status bar bus overview icons and 'Global Control')



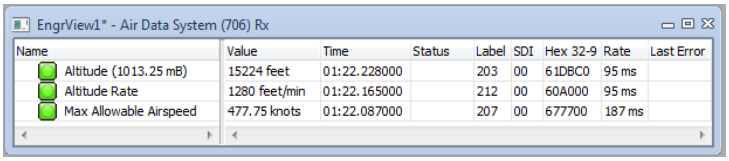

- Automatic creation and organization of buses into individual named workspaces with displays to show activity
- New 1553 Message View
- Quick links on the system menu  to save, start, and stop.
- Easy to use protocol toolbars operate on the selected channel and provide both state information and control of common functions




- Improved 1553 Schedule display with integrated message list




- New 429 Label View auto-updates as labels are added and removed from the channel






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

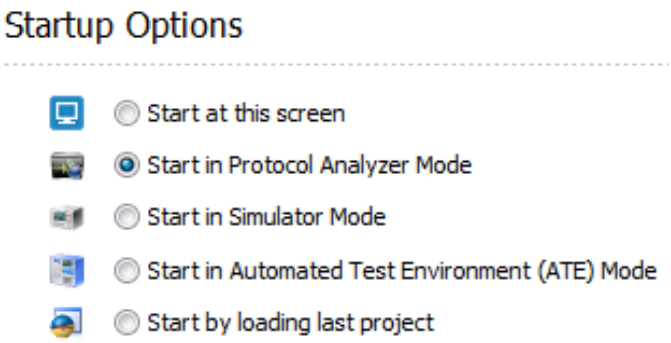
# CoPilot 6 Basics

## *Start-up Modes*

The default CoPilot start-up mode is to begin in Protocol Analyzer mode. The Protocol Analyzer mode automatically detects keyed Ballard hardware then begins monitoring databus activity. The start-up options on the Quick Start tab on the Backstage (

tab on Ribbon) include the following:

- **Start at this screen** – the Backstage screen is shown when CoPilot is started.
- **Start in Protocol Analyzer Mode** – automatically detects keyed Ballard hardware then begins monitoring databus activity.
- **Start in Simulator Mode** – automatically detects keyed Ballard hardware and allows the user to make additional configuration changes before CoPilot is run.
- **Start in Automated Test Environment (ATE) Mode** – automatically detects keyed Ballard hardware and shows various ATE/test scripting displays.
- **Start by loading last project** – loads the most recently used project.

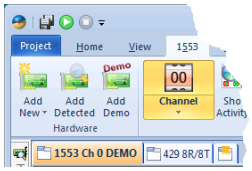



# Analyzing the Bus

Analyzing the bus is now even easier with CoPilot 6. The start-up Protocol Analyzer Quick Start Mode automatically configures installed hardware to monitor and display databus activity. Redesigned Activity Views replace the CoPilot 5 Protocol Browser and display the bus topology with statistics as well as allowing configuration. CoPilot automatically creates a workspace tab and an Activity View for each MIL-STD-1553 channel, ARINC 429 channel group, ARINC 708 channel pair, and ARINC 664 network for all hardware devices added to CoPilot. Additionally, the status of each 1553 channel and 429 channel group is displayed in the status bar; clicking on these icons will display the appropriate activity view. The Global Control pane provides start and stop control with a global status indicator. The Global Control pane is described in the following section with the Home Ribbon tab.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

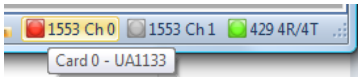



*Worskpace tabs and protocol Ribbon*




*Global Control pane*

**Note:** When working with multiple protocols and workspaces, the easiest way to show the desired Activity View is to click the status icon in the Status Bar.




*Status indicators in Status Bar*

**Status Icon States**

The status icons displayed throughout CoPilot indicate both current and historical state and can also indicate configuration.




**State Colors**


No activity/Not configured


Good status, no errors


Warnings


Errors

**Configuration**


Paused


'S' for Simulated RT

# The Ribbon Interface

The new Ribbon interface introduced in CoPilot 6 provides improved organization of tools and workspaces so you find what you need faster. This intuitive Ribbon interface is a menu that gives you easy one-click access to a comprehensive set of controls.

## *Overview*

The CoPilot Ribbon is designed to help you quickly find the commands that you need to complete a task. Commands are organized in logical groups, which are collected together under tabs. Each tab relates to a type of activity or particular



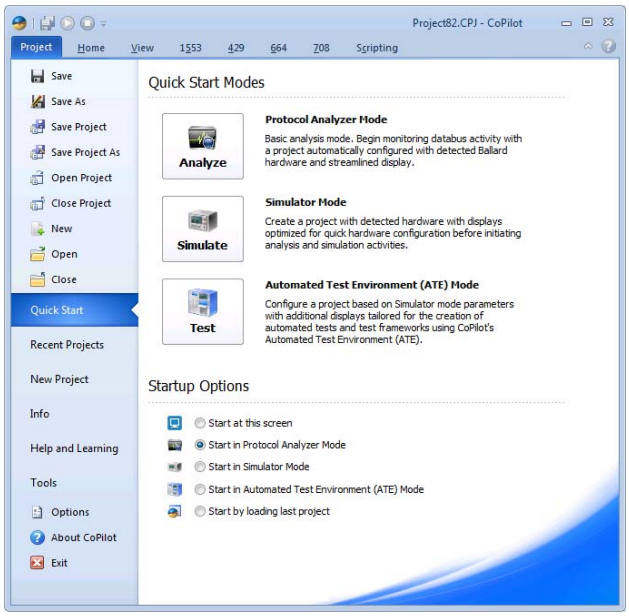
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

avionics protocol, such as Scripting or MIL-STD-1553. The CoPilot Ribbon menu also introduces the new CoPilot Backstage view (the “Project” Ribbon menu tab). The Ribbon replaces the toolbars used in CoPilot 5.

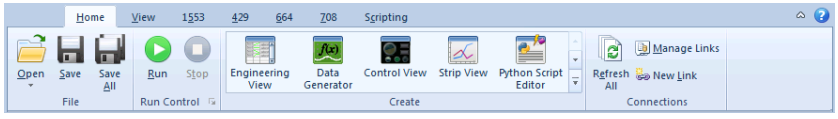
## *Project Tab (Backstage View)*

Selecting the Project Tab of the CoPilot Ribbon shows the Backstage View. The Backstage View is where you manage your project and files, find help and learning resources, access associated tools, control the startup options and set user preferences.

The Backstage View in CoPilot 6 replaces the Start Page along with the project and help menus used in earlier releases of CoPilot.




## *Home Tab*




The Home tab of the CoPilot Ribbon has been designed to make it convenient for you to get to the project features you use the most. If you are familiar with earlier versions of CoPilot, you will find new features here.

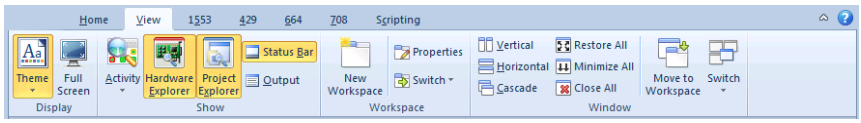
The Home tab includes control groups for file management, run control, creating CoPilot views, and managing connections (links). The Run Control group also has a special icon in the lower right corner which shows the Global Control pane.





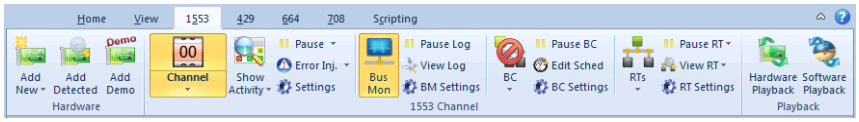

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### View Tab




The View tab of the CoPilot Ribbon is the location for view manipulations commands. The View tab includes control groups for CoPilot environment display options, the Show group for frequently used displays, the Workspace management group, and the Window management group.

### 1553 Tab

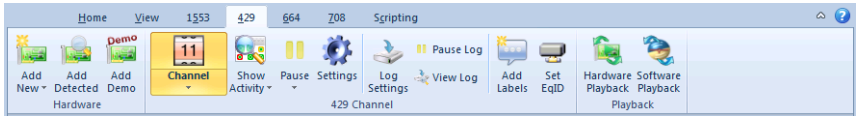



The 1553 tab of the CoPilot Ribbon hosts specialized groups of controls for MIL-STD-1553 operation and analysis. The 1553 tab includes control groups for adding hardware devices, MIL-STD-1553 protocol channel operations, and databus playback.

The 1553 Channel group contains a variety of protocol specific controls. The controls in this group all operate on the currently selected MIL-STD-1553 channel. Use these controls to quickly discover current activity, inspect logged data recordings, and configure databus hardware.




### 429 Tab

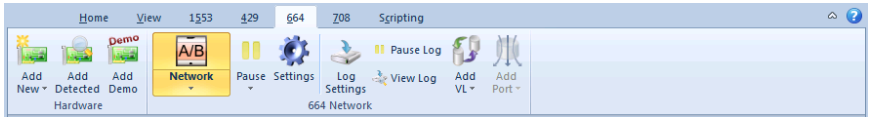



The 429 tab of the CoPilot Ribbon hosts specialized groups of controls for ARINC 429 operation and analysis. The 429 tab includes control groups for adding hardware devices, ARINC 429 protocol channel operations, and databus playback.

The 429 Channel group contains a variety of protocol specific controls. The controls in this group all operate on the currently selected ARINC 429 channel. Use these controls to quickly discover current activity, inspect logged data recordings, and configure databus hardware.




### 664 (AFDX) Tab




The 664 tab of the CoPilot Ribbon hosts specialized groups of controls for ARINC 664 (AFDX) operation and analysis. The 664 tab includes control groups for adding hardware devices and ARINC 664 (AFDX) protocol network operations.

The 664 Network group contains a variety of protocol specific controls. The controls in this group all operate on the currently

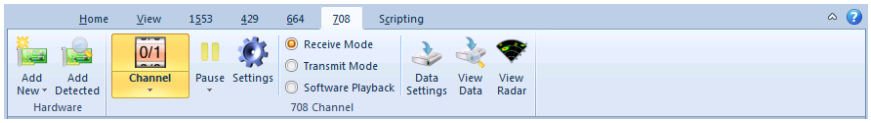





Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

selected ARINC 664 (AFDX) Network. Use these controls to quickly discover current activity, inspect logged data recordings, and configure databus hardware.

### *708 Tab*

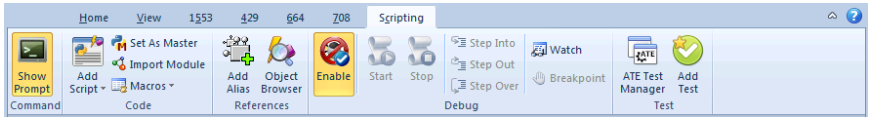



The 708 tab of the CoPilot Ribbon hosts specialized groups of controls for ARINC 708 (Weather Radar) operation and analysis. The 708 tab includes control groups for adding hardware devices and ARINC 708 protocol channel operations.

The 708 Channel group contains a variety of protocol specific controls. The controls in this group all operate on the currently selected ARINC 708 channel pair. Use these controls to quickly discover current activity, inspect logged data recordings, and configure databus hardware.




### *Scripting Tab*




The Scripting tab of the CoPilot Ribbon includes a variety of test and scripting tools. The scripting tab includes control groups for the Command prompt, scripting code files, scripting object references, scripting debug controls, and test using the Automated Test Manager.

# CoPilot Installation Requirements

CoPilot requires the Windows XP operating system or above. At the time of publication of this document, this includes Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, and Windows 7. The minimum system hardware requirements include the following:

#### Suggested System Requirements

- Intel Core 2 Duo, 2 GHz or higher (multi-core) processor or equivalant
- 4 GB of RAM
- 2 GB of available hard disk space (additional free disk space required for data logging)

#### Minimum System Requirements

- 1 GHz 32-bit (x86) or 64-bit (x64) processor
- 2 GB of RAM
- 1 GB of available hard disk space

Note, these minimum system requirements are dependant on actual system architecture and assume no other applications are running. Although not



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

required, the CoPilot software uses multiple threads and can benefit from a multi-core or multi-processor machine. Having a multi-core, multi-processor system is highly recommended.

## 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 5 or earlier) can remain on your computer and the new CoPilot version 6 can be installed alongside this prior version. If necessary, the CoPilot 6 installation will update Copilot 5 to the latest 5.x version for side-by-side compatibility with CoPilot 6.

# Supported Protocols

The CoPilot environment simultaneously supports a variety of avionics protocols. These include MIL-STD-1553, ARINC 429, ARINC 664/AFDX, ARINC 708, serial, discretes, and others. The protocols supported by CoPilot are described below. In addition to the native protocols supported, Python scripting available with CoPilot Professional can add support for additional protocols.

## MIL-STD-1553

### Background

MIL-STD-1553, defined by the U.S. Department of Defense (DOD), has been in use since 1973. MIL-STD-1553 is a standard defining a local area network (LAN) originally developed for and widely used on military aircraft. This digital, command-response, time-division multiplexing network protocol is also used in many other military and commercial applications where fast, positive control is required. The standard defines the handshaking, data formats, and timing requirements of the protocol as well as the electrical characteristics of the bus and the terminals' interface electronics. CoPilot 1553 assumes that the user is familiar with the MIL-STD-1553 protocol.

### MIL-STD-1553B

The MIL-STD-1553 standard was updated and released in 1975 as MIL-STD-1553A through the joint efforts of industry, Air Force, Army, and Navy. MIL-STD-1553B was released in 1978 to correct difficulties encountered in the application of 1553A, and was further refined in 1980 and 1986 with the release of Notice 1 and Notice 2. CoPilot initialization settings are based on 1553B, Notice 2.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# ARINC 429

### ARINC 429

ARINC 429 has been the industry standard databus protocol for commercial aircraft. It is a specification that defines a local area network for transfer of digital data between avionics system elements. The specification describes how an avionics system transmits information over a single twisted and shielded pair of wires (the databus) to all other system elements having need of that information (up to as many as 20 receivers). Bi-directional data flow on a given databus is not permitted.

There are two speeds (approximately 12.5 and 100 thousand bits per second). Words are 32 bits long including a label, parity bit and other fields. The label is an eight bit field that identifies the type of information contained in the word. The 429 specification defines the units, ranges, resolutions, refresh rates, number of significant bits, pad bits, etc. for the words transmitted by the different avionics system elements.

### ARINC 419

ARINC 419 catalogs and compares many of the older avionics databus standards. Protocols for the transmission of digital data were originally defined as part of the standardization of individual equipment characteristics. Since there was no universal databus standard and needs varied, different digital transmission systems were called out in various equipment characteristics, one of them being ARINC 575 Digital Air Data System (DADS).

### ARINC 575

ARINC 575 is an equipment characteristic for a Digital Air Data System (DADS) that provides essential air-data information for displays, autopilots, and other flight controls and instrumentation on commercial and transport type aircraft. ARINC 575 defines a digital databus to distribute information to displays and other systems. This databus was later described in ARINC 419 and became what is now known as ARINC 429. There are only minor differences between the digital databus of ARINC 575 and ARINC 429. The most significant difference is that ARINC 429 reserves bit 32 for parity, while ARINC 575 could use bit 32 for either parity (when BNR) or data (when BCD). This is implemented in CoPilot using hardware features to treat parity as data and with ARINC 575 data interpreters.

### Williamsburg

Williamsburg is one of the many ARINC 429 derivatives supported by Ballard hardware and CoPilot. Williamsburg support is available to CoPilot Professional script users.

Williamsburg is a bit-oriented protocol used to transfer files between systems over ARINC 429 full duplex links. Each LRU has a transmit connection and a receive connection, and implements a control protocol to manage the data transfer. The original block transfer protocol was AIM (ISO Alphabet No. 5). The Williamsburg protocol, introduced in ARINC 429-12, has replaced the AIM protocol (now discontinued).

Files are transferred in blocks known as Link Data Units (LDU), each containing from 3 to 255 words. A system initiates a transfer by sending a Request To Send (RTS). Following the receipt of a Clear To Send (CTS), the source begins the LDU sequence with a Start Of Transmission (SOT) followed by up to 253 data words. The LDU transfer is terminated with an End Of



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Transmission (EOT) containing a 16-bit CRC value calculated from the contents of the LDU. On successful verification of the CRC value, the sink system sends an Acknowledgment (ACK) message. The source may then repeat the process until the entire file is transferred.

Note: Williamsburg samples are included on the CoPilot CD.

## ARINC 664/AFDX

ARINC 664 is a multipart specification that defines an Ethernet data network for the new generation of aircraft installations. Part 7 of ARINC 664 defines a deterministic network known as AFDX, which stands for Avionics Full Duplex Switched Ethernet. IEEE Standard 802.3 (Ethernet) is an integral part of the AFDX specification. The AFDX extensions to IEEE 802.3 address the special requirements (quality of service, reliability, etc.) of an aircraft environment. AFDX is a new standard providing much higher data rates than existing avionics databus protocols. Both Airbus and Boeing are developing aircraft that use AFDX.

## ARINC 708

ARINC 708 describes the characteristics of an airborne pulse Doppler Weather Radar system intended for installation in commercial transport type aircraft. Display data and system mode/status information is transmitted over the Display Databus from the Transmitter-Receiver unit (T-R) to the Control/Display Unit (CDU). Data on the Display Databus consists of 1600 bit messages (words) that are preceded by and followed by a sync. Except for the following sync, bit count, and lack of parity, the signal characteristics on the Display Databus (Manchester encoding) are similar to MIL-STD-1553.

## Serial

In CoPilot, serial communication is over RS422 busses.

### RS422 Serial Communication

RS422 is a Standard interfaces approved by the Electronic Industries Association (EIA), and designed for greater distances and higher Baud rates than RS232. In its simplest form, a pair of converters from RS232 to RS422 (and back again) can be used to form an "RS232 extension cord." Data rates of up to 100K bits / second and distances up to 4000 Ft. can be accommodated with RS422. RS422 is also specified for multi-drop (party-line) applications where only one driver is connected to, and transmits on, a "bus" of up to 10 receivers.

RS422 devices cannot be used to construct a truly multi-point network. A true multi-point network consists of multiple drivers and receivers connected on a single bus, where any node can transmit or receive data.

## Discrete Inputs and Outputs (DIO)

Most Ballard hardware devices have multiple discretes (or DIO – digital input/output, digital I/O lines). The table below shows some information about the discretes in addition to the generation of the device architecture. The 3G hardware and most 4G hardware devices have digital TTL (5V) discretes while 5G hardware devices have avionics level discretes; these two different discrete



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

types are are wired differently. **Consult your hardware manual for the specific capabilities of your hardware and which lines are used with external sync outputs and trigger inputs.**

The discretes are used for external sync outputs, external trigger inputs, signaling script conditions (events), reading/writing digital values, and for controlling 1553 branching. The hardware manual for your hardware device describes correlation between the pin numbers, hardware reference signal, and the usage of these values in the API (driver). The *Scheduling Branches* section on page *125* describes DIO conditions for branching in more detail.

These values are exposed in the CoPilot Automation model and are accessed using scripting and the DIO numbers from the following table. Calling the "IsInputDiscrete()" and "IsOutputDiscrete()" functions determines the existence and direction of each discrete. Some discretes can function as both inputs and outputs. The "ExtDIORd()" and "ExtDIOWr()" are then used to read and write (if possible) the discrete values. The following table lists the supported DIO numbers where the DIO numbers correspond to the *ndionum* parameter used with the IsInputDiscrete, IsOutputDiscrete, ExtDIORd and ExtDIOWr functions on a card object.

| Hardware Product | Generation | Input DIO Numbers | Output DIO Numbers |
|---|---|---|---|
| Box Products (Ethernet, USB, Serial) | | | |
| Avionics BusBox: AB1000 | 5G | 1, 2, 3, 4, 5, 6 | 9, 10, 11, 12 |
| Avionics BusBox: AB2000 | 5G | 1-16 (in/out) | 1-16 (in/out) |
| BUSBox *(v 5 only)* | 3G | 0, 1, 2 | 3, 4, 5(ExtLed), 6(BDOUT) |
| OmniBusBox | 4G | 1, 2, 3 | 5, 6, 7 |
| USB 1553 | 5G | 1-4, 9-12 (in/out) | 1-4, 9-12 (in/out) |
| USB 429 | 5G | 1-4, 9-12 (in/out) | 1-4, 9-12 (in/out) |
| PCI Products | | | |
| LP1553-3 *(v 5 only)* | 3G | 2 (DIN) | 0 (DOUT) |
| LP429-3 *(v 5 only)* | 3G | -- | -- |
| LP1553-5 | 5G | 1-16 (in/out) | 1-16 (in/out) |
| LP429-5 | 5G | 1-16 (in/out) | 1-16 (in/out) |
| OmniBus PCI | 4G | 1, 2, 3 | 5, 6, 7 |
| PCIe Products | | | |
| LE1553-5 | 5G | 1-16 (in/out) | 1-16 (in/out) |
| LE429-5 | 5G | 1-16 (in/out) | 1-16 (in/out) |
| PCMCIA Products | | | |
| CM1553-3 *(v 5 only)* | 3G | -- | 0 (DOUT) |
| CM429-1 *(v 5 only)* | 2G | 1 (DIN) | 0 (DOUT) |
| cPCI Products | | | |
| LC1553-3 *(v 5 only)* | 3G | 2 (DIN) | 0 (DOUT) |
| LC429-3 *(v 5 only)* | 3G | -- | -- |
| OmniBus cPCI | 4G | 1, 2, 3 | 5, 6, 7 |
| ISA Products | | | |
| PC1553-3 *(v 5 only)* | 3G | 2 (DIN) | 0 (DOUT) |
| VME Products | | | |
| OmniBus VME | 4G | 1, 2, 3 | 5, 6, 7 |
| PMC Products | | | |
| OmniBus PMC | 4G | 1, 2, 3 | 5, 6, 7 |

*Table of hardware digital input and outputs (DIO numbers)*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: For hardware that does not support in/out (input/output) DIO, the output DIO could (should) be wired to the input DIO to allow both reading and writing the DIO values (i.e. connecting 1 to 5, 2 to 6, and 3 to 7 for the OmniBusBox). **See the hardware manual for your specific device for additional information.**

## Other Protocols

Please feel free to contact support to find out if your required protocol is already supported or is scheduled to be added. In addition to natively supported protocols, Python Scripting with CoPilot ATE can be used to add support for additional protocols and third-party hardware.

# About Ballard Technology

### Databus Pioneer

Ballard Technology is a leading developer and manufacturer of hardware and software products for the testing of avionics databuses. Providing avionics databus tools and interfaces to the aerospace industry since 1986, Ballard Technology was one of the first companies to realize the potential in using standard computers to host specialized test equipment.

### Ballard Products

Ballard offers an extensive line of hardware and software for MIL-STD-1553, ARINC 429, ARINC 629, ARINC 708, ARINC 717, ARINC 664/AFDX and a variety of specialized databuses. These interfaces enable computers to communicate over an avionics databus network.

Applications for Ballard products include research and development, production and acceptance testing, system simulation, maintenance, flight testing, on-board/in-flight avionics, and as Commercial-Off-The-Shelf (COTS) solutions to flight and other operational requirements. Standard and custom products have also been used in non-aerospace applications. Ballard's capabilities for design, development, and production of electronic hardware and software can extend to any application or industry where quality and reliability are major issues.

### Leader in Customer Satisfaction

By providing quality, easy-to-use products and free customer support, Ballard Technology has won the loyalty of users in the industry, the military, and governments worldwide. Because we listen to their needs, program managers, engineers, and buyers turn to Ballard Technology for cost-effective avionics interface solutions. Special customer requirements have resulted in many of Ballard's standard products.

# How to Use This Manual

### New to CoPilot?

- **General CoPilot principles**— If you are a new CoPilot user, start with *CoPilot Operation* on page *40* to learn the basic CoPilot principles that are true for all protocols.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Protocol-specific tasks**—The *Getting Started Guide* tutorials are a great way to familiarize yourself with CoPilot's protocol-specific functionality. The guides are available for MIL-STD-1553, ARINC 429, and ARINC 664/AFDX. You can work through the lesson without Ballard hardware using the built-in demonstration cards (demo cards).

### Help Conventions

This manual is designed as a reference and a guide to the principles of CoPilot. Users can read only the chapters they need and refer to the rest as required. This manual assumes that the reader is familiar with the MIL-STD-1553 protocol.

This manual has been written to apply to CoPilot version 6 and CoPilot version 5. The *New in CoPilot 6* section on *page 23* provides an overview of the recent CoPilot changes.

The following conventions are used throughout this document:

- **File | Save Project** is a sequence of two menu commands. Click on the File menu and select Save Project.
- Menu selections and buttons are bolded (e.g., **File** and **OK**).
- Special **Notes:** and **Tips:** contain additional information or useful shortcuts.

## Other Resources

The *CoPilot User's Manual* and the various *Getting Started Guides* are available in PDF form on the CoPilot CD.

### User's Manual Help

This manual is the CoPilot help documentation and is accessible from the software's help menu and the start page (version 5) or the Backstage **Help and Learning** tab group. Separate sections of the *CoPilot User's Manual* describe the differences for the major protocols that include: MIL-STD-1553, ARINC 429, ARINC 664/AFDX, and ARINC 708. Use the browse buttons and hyperlinks to navigate through the help document. The document contains an index and is searchable to provide further assistance.

### Getting Started

The *Getting Started* guide provides an overview of the major features and principles of the CoPilot program. The *Getting Started* guide is included with CoPilot on the CoPilot CD.

### Additional Information

In addition to the User's Manual and the Getting Started Guide, the following items may provide assistance:

- Samples and Templates available from the Backstage in CoPilot 6
- The example projects and scripts (on the CoPilot CD)
- The learning video library at www.ballardtech.com/CoPilot (and on the CoPilot CD in the MEDIA directory)



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

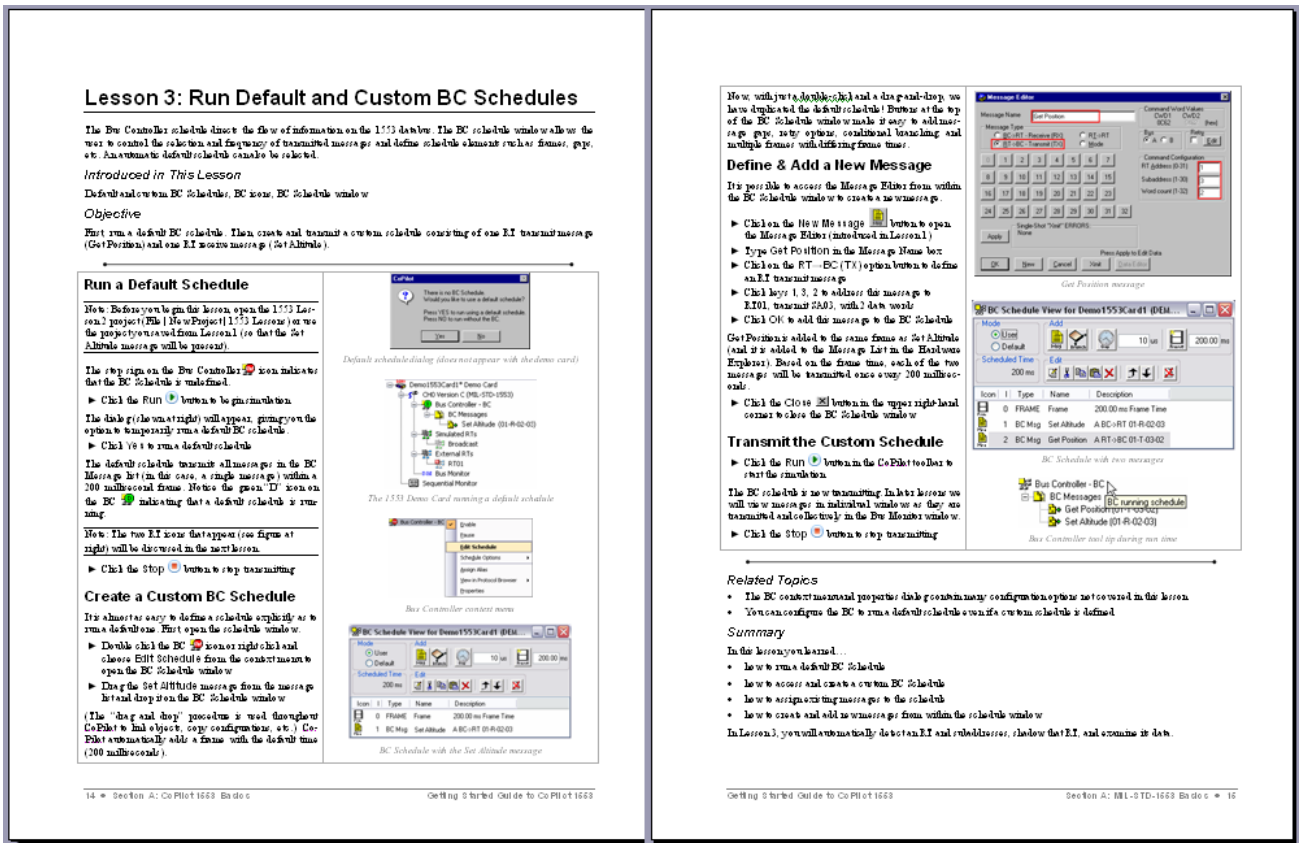
# Quick Start

Are you new to CoPilot? Everything you need to tryout the CoPilot software is right here. By using the prepared lesson projects, the demo card, and the Getting Started guides, you will be taken on a guided tour through many of CoPilot's capabilities. No experience or hardware is required!

### Getting Started Guides

The *Getting Started* guides are designed to help you quickly learn the basics of CoPilot and introduce you to some of the optional features available with CoPilot. The CoPilot 6 Getting Started Guide provides an overview of the software and highlights some of the new features in CoPilot 6.

The CoPilot 5 Getting Started Guides each describe one of three different protocols: MIL-STD-1553, ARINC 429, and ARINC 664/AFDX. The features and principles are presented through brief, two-page lessons, with each lesson building on the previous one. New users can work through all the lessons, and experienced users can jump ahead to learn about a particular feature.




*Sample CoPilot 5 Getting Started Guide*

These full-color tutorials are included with CoPilot documentation. Separate guides are available for each databus protocol. PDF versions are also available on the CoPilot installation CD, but it is suggested that you work with a hard copy if possible.

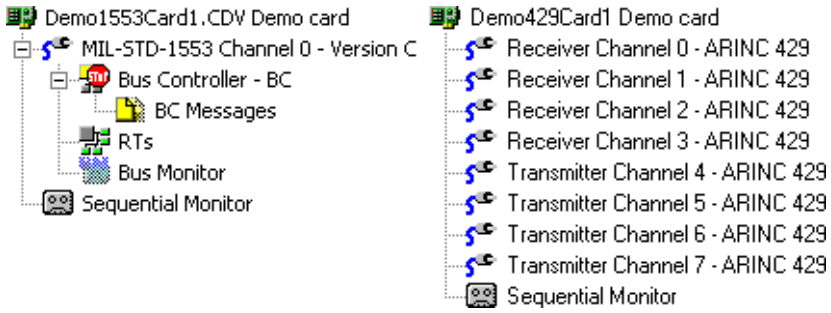


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Demo Cards

The Getting Started lessons are built around a software-generated demonstration hardware or "Demo cards." Using this pseudo-card, you can "try out" many of the features and capabilities of CoPilot without an active databus and without licensed Ballard hardware.

Note: Not all features of real hardware are available with a demo card. Demo cards are software objects for evaluation purposes only and never interact with actual avionics databuses. Scripting events may also exhibit slightly different behavior.




*Sample 1553 and ARINC 429 Demo Cards in the Hardware Explorer*

### Lesson Projects

For each lesson, there is a corresponding lesson project in its own folder. These lesson projects are copied to the CoPilot Projects folder during installation. If you wish to jump ahead to a certain lesson, open the folder for that lesson to obtain the starting point you will need. The Getting Started guides contain detailed instructions on using the lesson folders and the demo card.

# Support and Service

### Contact Ballard

Our experienced sales and customer services staff is available to discuss your requirements for avionics databus tools and interfaces. We invite your questions and comments on any of our products. You may reach us by telephone at (800) 829-1553 or (425) 339-0281, by fax at (425) 339-0915, on the Web at www.ballardtech.com, or through e-mail at support@ballardtech.com.

### Technical Support and Customer Service

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.

### Updates

At Ballard Technology, we take pride in high-quality, reliable products that meet the needs of our customers. Because we are continually improving our products, we periodically release new versions of software and updates to documentation. Please fill out and return the product registration card included with CoPilot so that we can keep you informed of updates, customer services, and new product information.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# CoPilot Operation

## CoPilot Operation Overview

### What Is CoPilot?

CoPilot is an easy-to-use software interface that simplifies and expands user control over receive, transmit, monitoring and analysis of MIL-STD-1553, ARINC 429, ARINC 664/AFDX, ARINC 708 and other databus activity. The software allows you to simultaneously control, monitor, chart, and examine databus activity from multiple hardware and multiple avionics protocols. In conjunction with Ballard avionics interface hardware, CoPilot can be used as…

- A **Bus Analyzer** with powerful tools (such as Software Playback, Monitor View, Activity View/Protocol Browser, etc.) for processing incoming or prerecorded data
- A **Simulator** for imitating real hardware (such as a MIL-STD-1553 Bus Controller or Remote Terminal)
- A **Bus Reader** that automatically detects, translates and displays databus information and automatically adds the appropriate information to the project definition
- A **Monitor** for viewing, recording and displaying all or selected portions of bus traffic
- A **Testing Tool** with features such as error injection, data generation, terminal simulation, engineering unit conversion, hardware playback, and scripting (which can be used to automate test sequences).

### How Does CoPilot Work?

The CoPilot software is a unified environment, capable of simultaneously hosting MIL-STD-1553, ARINC 429, ARINC 664/AFDX, ARINC 708 and other protocols. In CoPilot, you have the advantage of a common environment with features and tools suited to the unique requirements of each avionics protocol. This common interface means that the basic principles of CoPilot operation are universal. The following topics in this section describe:

- The layout of the CoPilot environment with the purpose of each piece
- How to understand, customize, and run CoPilot projects
- How CoPilot conventions and shortcuts can help you use CoPilot more efficiently



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When you are comfortable with the basics of operating CoPilot, you can then design a project around the needs of your specific protocol(s).

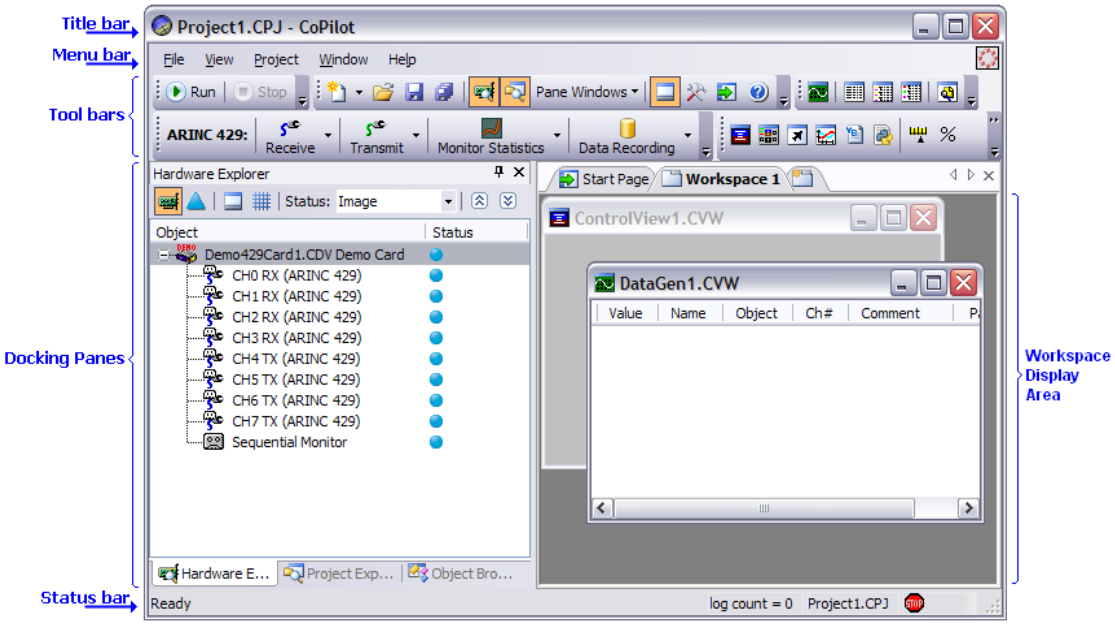
# The CoPilot User Interface

## The CoPilot Environment

### CoPilot Environment Elements

The CoPilot environment (User Interface/UI) consists of an integrated set of windows, ribbon tabs, toolbars, panes, workspaces, and other elements that allow you to create, edit, organize, and run a CoPilot project. The layout of these items can be customized to suit your preferences. Refer to the following images that show the features described in this section.

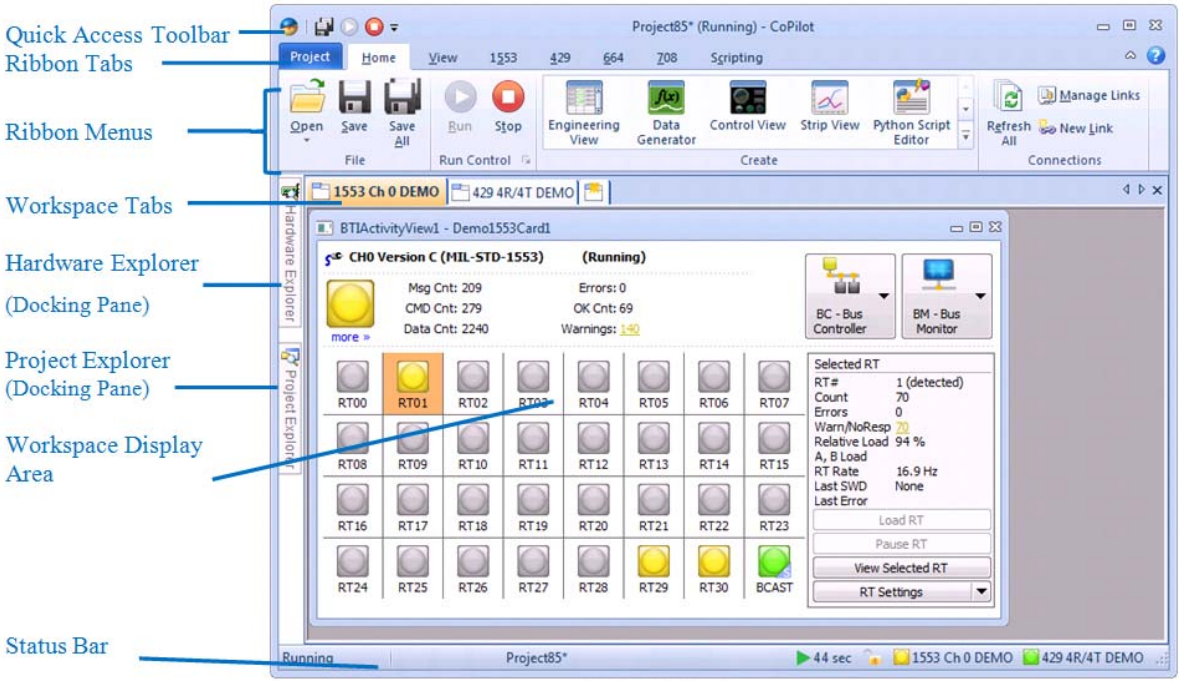
**Note**: *The CoPilot environment is referred to as the CoPilot Shell through the CoPilot automation model*




*CoPilot 5 Environment*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*CoPilot 6 Environment*

- **Title bar**—The title bar displays the filename of the current project and user title (optional; see project properties).
- **Quick Access Toolbar**— The quick access toolbar (version 6) contains controls to save the current project, start and stop CoPilot.
- **Menu Bar**— The menu bar (version 5) 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.
- **Ribbon Tabs**— The CoPilot version 6 Ribbon Tabs appear like a menu bar when the ribbon is hidden. The Ribbon tabs group CoPilot commands into Ribbon Menus. When commands are not available, for example, pressing the run button while CoPilot is already running, the command button is disabled. Moving the mouse over a command in the Ribbon without clicking the mouse will display a tool tip with additional information.
- **Toolbars**—Multiple toolbars contain shortcut buttons for frequently used project commands. To identify a button, position the mouse cursor over a button in the tool bar to view a tool tip display. Toolbars can be hidden, repositioned, and customized to create your own personalized toolbars.
- **Docking Panes**—Dockable windows used for display and configuration that may be moved around or undocked from the CoPilot application to change the display layout. Docking panes include the Hardware Explorer, Project Explorer, Protocol Browser (version 5), Command Line, and many others.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Workspace Tabs and Workspace Display Area**— Various types of view windows, such as the Activity View (version 6) and Data Generator, are hosted in the Workspace Display area. Workspaces are used to group and sort display view windows in tabs of the display area. Use the Window menu (version 5) or the View Ribbon tab (version 6) to tile, cascade, or select/open view windows and organize Workspaces. In version 6, CoPilot automatically creates additional workspaces when hardware is added to a project.
- **Status bar**—The status bar displays status information, the project filename, run state and duration of current (or last) run, and the current log count (summation of all Sequential Monitor counts). The status bar displays a help string when you hover the mouse over a command in one of the project menus or toolbars. CoPilot version 6 adds the current bus activity status to the status bar. Clicking a status icon switches the workspace to view that activity.

## Start Page

### Using the Start Page

The version 5 Start Page provides an easy way to access or create projects, access help documents and tutorials, and display detected hardware information. The Start Page is shown by default, but can be hidden. To access the Start Page, from the **View** menu click **Start Page**. The Start Page includes the following features:

- **Quick Start Tasks**—The quick start panel provides access to protocol analyzer project creation. These projects utilize the auto-detection and monitor capabilities of CoPilot to quickly identify databus activity.
- **Detected Hardware**—The detected hardware panel lists all installed Ballard databus hardware and displays description text identifying the board's CoPilot key (see *Licensing CoPilot with Hardware Keys* on page *61*).
- **Projects**—The projects panel contains controls to open previous and create new projects.
- **Samples and Templates**—The samples and templates panel contains all of the *Getting Started Guide* lesson projects, the installed sample projects, and user created templates. You can add user templates through the **File** | **Save Project As Template** menu command.
- **Common Help Topics**—The common help topics panel contains links to the help documents, tutorials, and other help related items.

## "Project" Backstage View

### Using the Backstage

The version 6 "Project" Backstage replaces the Start Page found in version 5. While the Ribbon contains commands for analyzing and working with databuses, the Backstage view provides commands for operating on project configurations. The backstage is an improved Start Page with more functionality and better orgization. Click the **Project** tab in the Ribbon to view the Backstage.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Workspaces

### Using Workspaces

Workspaces are used to group and sort display view windows in the display area. For example, you could place views used for initialization on a single Workspace, views used during simulation on separate Workspace, and views used for post analysis on another Workspace. Switching between Workspaces is quick and easy using the Workspace tabs.

CoPilot version 6 automatically creates additional workspaces when hardware is added to a project.

### Managing Workspaces

Multiple Workspaces can be created to organize the CoPilot environment and maximize the viewable real-estate.

**Adding a Workspace**—You can add a Workspace through the **Window** | **Workspaces** | **Add New Workspace** menu command, selecting **Add New Workspace** from the default CoPilot context menu, or by pressing the Add New Workspace button on the workspace tab.

**Deleting a Workspace**—A workspace can only be deleted if there is more than one workspaces in the project and the workspace contains no view windows . You can delete a Workspace through the **Delete Workspace** menu item in the Workspace tab context menu or by selecting the workspace tab **Close** button when the Workspace is selected.

**Moving Views between Workspaces**—View windows can be moved from one Workspace to another using the Windows dialog. To view the Windows dialog select **Window** | **Manage Windows…** menu command or select **Manage Windows…** from the default CoPilot context menu or any Workspace context menu in the Project Explorer.

**Selecting a Workspace**—Workspaces can be selected thought the **Window** | **Workspaces** menu (a check mark indicates the current selected workspace), selecting a workspace tab, or by selecting Show from any workspace context menu in the Project Explorer. If using the Project Explorer or Window menu to select a view window, the workspace it is on will be selected automatically.

**Workspace Properties**—Workspace properties consist of a name and a description. The name is displayed in the workspace tab. The description is displayed as the workspace tab tooltip.

# Docking Panes

### Docking Pane Overview

Docking panes are CoPilot environment windows that can be moved around, tiled, or undocked from the CoPilot application to change the display layout.

### Selecting Panes to Display

You can view a pane by selecting it from the **View** or **View** | **Other Windows** menu (open panes are highlighted). Some panes will merge their own menu items into the application menu when selected. To close a pane select the **View** menu again or click the **Close** button located in the pane header.



**Note**: *CoPilot 6 introduced the Ribbon Interface and simplified the the Protocol Browser with the new Activity View. The Hardware Explorer tree references remain for CoPilot 5 compatibility. See New in CoPilot 6.*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Hardware Explorer**—The Hardware Explorer is your view of the hardware associated with the project. The icon in the Hardware Explorer represents state for some objects as shown in the *Appendix B: Hardware Object Icon Reference*. Depending on the databus protocol(s) supported you will see a hierarchical branch structure containing Bus Controllers, Remote Terminals, Equipment, Labels, Networks, VLs, Ports, fields or other items specific to that protocol. The Hardware Explorer is used to specify and display the definitions and structure of the information being communicated over the databus. The context menu and properties associated with items in the Hardware Explorer give you access to board settings, protocol-specific databus settings, and CoPilot features. You can drag Hardware Explorer objects into views to display and manipulate them. For example, dragging a transmit message or field into a Data Generator allows you to view and modify its data.
- **Project Explorer**—The Project Explorer provides a complete listing of all hardware and view windows in a project. View windows are listed even if the window is inactive or hidden (i.e., not visible in the display area). The Project Explorer provides a way to activate, delete, establish focus, or attach comments to project components.

**Note**: *CoPilot 6 introduced the Ribbon Interface and simplified the Protocol Browser with the new Activity View. The Hardware Explorer tree references remain for CoPilot 5 compatibility. See* *New in CoPilot 6**.*

- **Protocol Browser**—The Protocol Browser (version 5) provides a fast overview of what is happening on the MIL-STD-1553 and ARINC 429 databuses. The Protocol Browser allows users to browse into various levels of the protocol to see the activity and statistics. The current and historical states of objects are displayed using color-coded icons. Several display modes are available including small icon, large icon, and list modes.

Note: The Protocol Browser reports statistics from captured Sequential Monitor Records; only data recorded from the Sequential Monitor is used. If the Sequential Monitor is not enabled and collecting data, then statistics in the Protocol Browsers will not be available for that hardware.

- **Command Prompt**—The Command Prompt Pane provides a command line interface for performing automated tasks within CoPilot using its Automation model and Python syntax. See *Automated Test Environment (ATE)* section on page *399* for more details.
- **Output Pane**—The Output Pane displays textual output from CoPilot, Python, and user-generated custom output. See *Automated Test Environment (ATE)* section on page *399* for more details.
- **Object Browser**—The Object Browser shows the various "aliases" available for manipulation via Python scripts and the command line. See *Automated Test Environment (ATE)* section on page *399* for more details.



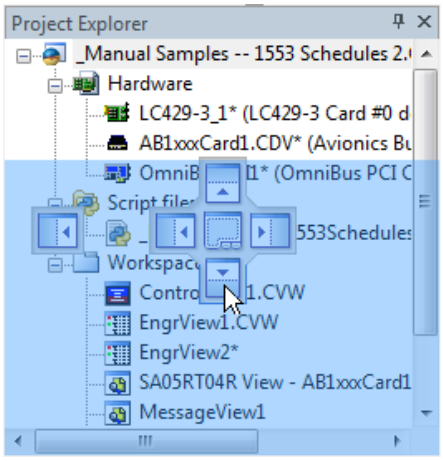
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Automated Test Manager**—The Automated Test Manager Pane provides users the ability to run automated tests using Python scripts with CoPilot ATE. See *Automated Test Environment (ATE)* section on page *399* for more details.
- **Watch Window**—The Watch Window Pane provides the ability to watch (and edit) variables and object properties with CoPilot ATE. See *Automated Test Environment (ATE)* section on page *399* for more details.

### Changing Docking Pane Location

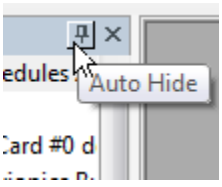
Although all panes have a default location, they can be dragged (or "undocked" from that location) to any position on the computer screen. To “dock” the window into CoPilot again, drag it back into the edge of the CoPilot application window.

Dragging a docking pane over the CoPilot application or over other docking panes will display **Docking Stickers** to visually display where the pane will be docked.




*Example Docking Sticker showing Docking Pane placement*

Docking panes can also be made to **Auto Hide** to minimize screen usage. When a pane is set to Auto Hide, contents of the pane are hidden and the title of the pane is shown in a tab. When the mouse is moved over the tab, the pane is unhidden and the pane is re-hidden when the mouse is moved away from the pane.




*Enabling the Auto Hide feature of a docking pane*



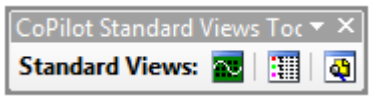
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# CoPilot Views

### View Window Overview

View windows are used to display, organize and modify data. CoPilot categorizes view windows into Standard and Professional views. CoPilot version 5 uses the toolbars described below, however, the CoPilot version 6 commands to add views is found in the **Create** group on the **Home** tab of the Ribbon. See *New in CoPilot 6* on page *23* for additional Ribbon information.

- **Standard Views**—Standard views can be used in any project. An example of a Standard view is the Data Generator. The Data Generator creates a continuous stream of changing data values while the simulation is running. For a complete list of Standard views, see the *Standard Displays* section on page *101*. These standard views can be created from the **Create** group on the **Home** tab with version 6.




*CoPilot Standard Views toolbar (version 5)*

- **Professional Views**—Professional views require the presence of Professional keyed hardware. An example of a Professional view is the Control View. The Control View hosts a variety of ActiveX controls that make up virtual instruments and displays. Entire avionics panels can be recreated virtually. For a complete list of Professional views, see the *Professional Displays and Controls* section on page *369*. These professional views can be created from the **Create** group on the **Home** tab with version 6.




*CoPilot Professional Views toolbar (version 5)*

### Creating Views

There are several ways to create view windows. Many items in the Hardware Explorer have view creation options in their context menu. For example, a field usually has context menu entries for creating Strip chart views and a variety of Quick views. Other items will have entries for viewing that item in a special purpose view (e.g., MIL-STD-1553 BC Schedule Editor can be created from a BC item). In CoPilot 5, there are both Standard and Professional view toolbars. These toolbars contain buttons to create view windows. With CoPilot version 6, views can be created from the **Create** ribbon group on the **Home** tab with version 6. Another way to create views in both version 5 and 6 is from the *New Hardware and Views…* dialog. To open the *New Hardware and Views…* dialog, select **File** | **New** | **New Hardware and Views…** in version 5 or through the **Backstage** in version 6.

### Saving Views

View windows are automatically saved with the project. To save a view outside of a project or to a different location, select the file in the Project Explorer pane then select **File** | **Save *View_Name* As…**



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Loading Views

View windows are automatically loaded with the project they were saved with. To load a view that was not saved with a project, select **File** | **Open** | **Open Hardware or View Files…**

### Enabling Views

Views can be Enabled/Disabled through the context menu of a view item in the Project Explorer pane. Disabled views maintain membership in a project but are not executed during simulation (runtime). To Enable or Disable a view, select **Enable** from the view item's context menu in the Project Explorer pane (a checkmark indicates that the view is enabled).

## Menus & Toolbars

This section applies to CoPilot version 5 only. CoPilot version 6 uses the Ribbon interface. See *New in CoPilot 6* on page *23* for additional Ribbon information.

### Menu & Toolbar overview

The menu bar hosts two kinds of menus: project and view menus. The project menus (File, View, Project, Window, and Help) are used to select and control CoPilot projects and displays. The menus for certain view windows merge with the menu bar when that view is in focus. For more information on the view menu for a specific display, see the section about that view window. The CoPilot Toolbars can be turned on and off, docked in various locations of the application, or floated outside the application window. The icon shown with a menu command will be the default icon displayed on a toolbar button.

### Menu bar

The default menu bar consists of the File, View, Project, Window and Help menus. Keyboard shortcuts can be assigned to menu items on the Keyboard tab of the Customize dialog. To access the Customize dialog, select **Customize…** from the menu/toolbar context menu.

- **File**—The File menu contains controls to create, open, save, import and export hardware, and view windows or entire projects.
- **View**—The View menu contains controls to select, organize and customize what is visible in the application.
- **Project**—The Project menu contains controls to run, stop, run macros, manage links and set project properties.
- **Window**—The Window menu contains controls to manage windows and workspaces.
- **Help**—The Help menu contains controls to view help documents and send feedback.

### CoPilot Toolbars

Commonly used commands can be quickly executed from buttons in a toolbar. Buttons can be identified with tooltips or status bar text (the status bar will display description text when the mouse is moved over a button). The default toolbars are:

- **CoPilot Standard Toolbar**—The CoPilot Standard Toolbar contains a collection of the most common commands



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **CoPilot Project Toolbar**—The CoPilot Project toolbar contains the Run and Stop commands used to control the simulation
- **CoPilot Standard Views Toolbar**—Allows the quick creation of CoPilot Standard views in the currently opened project by clicking the toolbar button associated a particular view
- **CoPilot Professional Views Toolbar**—Allows the quick creation of CoPilot Professional views in the currently opened project by clicking the toolbar button associated a particular view
- **Macro Toolbar**—The macro toolbar is a user customizable toolbar where macro operations are executed upon a button press (see Configuring the Macro Toolbar below)
- **Debugger Toolbar**—Has commands used for the debugging of Python scripts

### Configuring the Macro Toolbar

You can add custom buttons to the Macro Toolbar. Each user-added button links to a Python script method (function). Only methods containing zero parameters are linkable as macros. The Python scripting file can be a part of a project or an unattached file. The Macro Toolbar is environment-based. The contents of the toolbar are always available and will not change between projects.

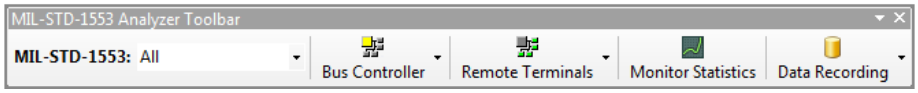
To assign a new macro button:

1. Press **Configure Macro Buttons…** from the Macro Toolbar
2. Press the **New** button (this adds an unnamed item and selects it)
3. Provide a Caption (name) and Tooltip
4. Select the Module (Python file) containing the zero parameter method
5. Select the Method
6. Press the **Apply** button to save the changes

### CoPilot Analyzer Toolbars

Analyzer toolbars are provided to simplify common tasks performed for a specific avionics protocol. Analyzer toolbar commands can be performed on various levels of scope (perform the action on all items in the project or a subset). If the scope was set to **All** and the action was to **Pause BC**, all bus controllers in the project would be set to the pause state.

- **MIL-STD-1553 Analyzer Toolbar**—The MIL-STD-1553 Analyzer toolbar has commands to control the bus controller, remote terminals, and statistic and data recording. All buttons on the MIL-STD-1553 Analyzer toolbar share a single action scope setting.




*MIL-STD-1553 Analyzer toolbar*

- **ARINC 429 Analyzer Toolbar**—The ARINC 429 Analyzer toolbar has commands to control transmit channels, receive channels, statistics and data recording. Each action group (drop down toolbar) has its own action scope setting.



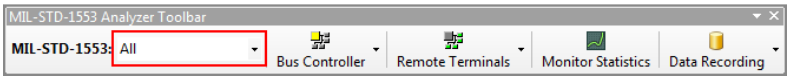
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



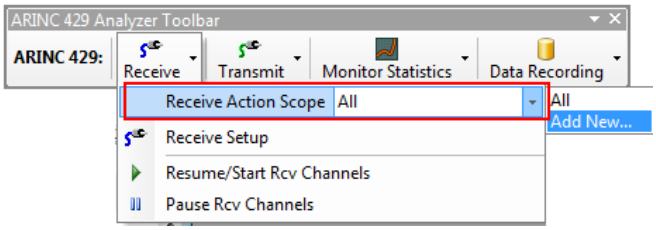

*MIL-STD-1553 Analyzer toolbar*

### Action Scope Analyzer Toolbar

The MIL-STD-1553 Analyzer Toolbar and the ARINC 429 Analyzer Toolbar each have an action scope associated with the toolbar. The action scope selects the scope (or object) the toolbar will operate on if there is ambiguity. For example, if there is more than one MIL-STD-1553 channel, pressing the BC button to configure the BC would be ambiguous and require an additional prompt to select the desired channel to configure if the action scope did not specify one of the channels. To set the action scope, select it from a scope drop down menu or select Add New…. If adding a new scope setting and only a single item is available that new item will be added and selected. Otherwise a list of available items will be provided to choose from. Setting the action scope to All causes every action to be performed on all appropriate items in the project. For example, pressing the Pause button under ARINC 429 Transmit would pause/unpause all ARINC 429 transmit channels in the entire project.




*Action Scope of the MIL-STD-1553 Analyzer toolbar*




*Receive Action Scope of the ARINC 429 Analyzer toolbar*

### Showing and Hiding Toolbars

To show or hide a toolbar, select the toolbar by name from the menu/toolbar context menu (Checked items are shown). Alternatively you could select displayed toolbars from the Toolbars tab on the Customize dialog. To access the Customize dialog select **Customize…** from the menu/toolbar context menu.

### Customizing Toolbars

The custom toolbar editor can be accessed by selecting **Customize…** from the menu/toolbar context menu. To create a custom toolbar, select the **New** button on the Toolbars tab and give the toolbar a name. The new toolbar is added to the toolbar area of the application. You add buttons (commands) by dragging items from the Commands tab onto your new toolbar. Drag an item off the toolbar to remove it. Customize toolbar buttons by using that button's context menu commands.

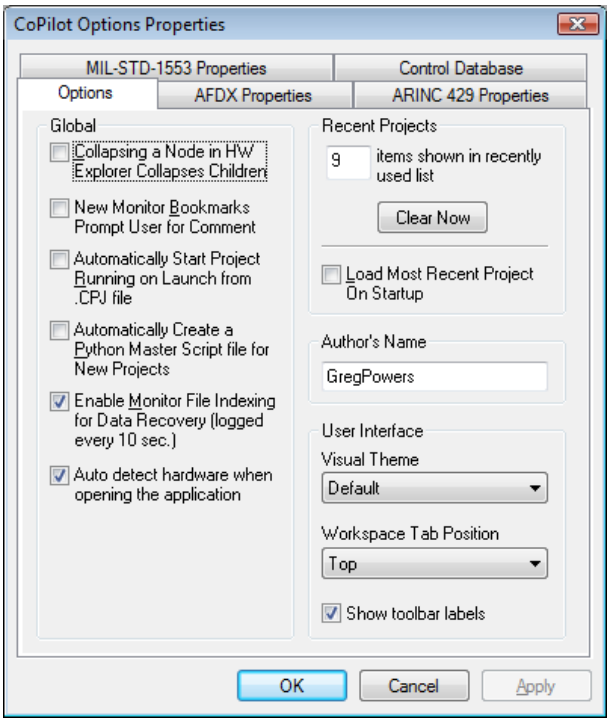


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# CoPilot Options

## Environment Options

The CoPilot options are accessible from the Project menu in the menu bar. The **Project | Option** command opens the CoPilot Options Properties dialog, where you can define global and protocol-specific settings.




### Global Options

The Options tab controls global settings and CoPilot behavior. These options are stored in the host's system registry on a per-user basis.

- **Collapsing a Node in HW Explorer Collapses Children** reduces display clutter in the Hardware Explorer tree by collapsing all nodes of the tree below the object being collapsed. When this option is not selected, any Hardware Explorer tree node will re-expand all levels exactly as they were before it was collapsed.
- **New Monitor Bookmarks Prompt User for Comment** will give the user the opportunity to add a user comment when new bookmarks are inserted in a sequential monitor view record.
- **Automatically Start Project Running on Launch from .CPJ file** will set CoPilot into the running state when a project file (.CPJ) is opened from outside of CoPilot (such as double-clicking the project file). If this option is not selected, double clicking a .CPJ file will open the project in edit mode without running it.
- **Automatically Create a Python Master Script for New Projects** will automatically include a Python master script file when new CoPilot projects are created. The script code will include stub code for project and user events.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Enable Monitor File Indexing for Data Recovery** will periodically write the information of the current monitor file(s) while recording (every 10 seconds). This index information, along with the temporary monitor data files, can then be used to recover the logged data in the event of power failure or system crash.
- **Auto Detect Hardware when Opening the Application** will search for all available hardware when the CoPilot application is launched and output the results in the detected hardware list of the Start Page (version 5 only).

#### Recent Projects Options

The recent projects settings in the Options tab control the Most Recently Used (MRU) project history list and if the last used project is automatically loaded each time the CoPilot application is started. These options are stored in the host system registry on a per-user basis.

- **Items shown in recently used list** specifies the number of recent projects to remember and keep in the "MRU" list.
- **Clear Now** resets the stored history of recently loaded projects.
- **Load Most Recent Project On Startup** automatically opens the last project used by CoPilot the next time the CoPilot application is launched.

#### Author's Name

The author name is used as the default author of projects.

#### User Interface Options

The user interface settings in the Options tab control the visual elements for your CoPilot application.

- **Visual Theme** determines the look of the various visual elements of your application, such as windows, icons, fonts, and colors.
- **Workspace Tab Position** determines the location of the workspace tabs relative to the workspace area.
- **Show Toolbar Labels** adds a toolbar description to each CoPilot toolbar when this option is set.

## Protocol Options

Options in the protocol tabs establish how CoPilot processes or presents protocol-specific information. The default database for each protocol appears in the database window (see figure below). You can create, rename, and maintain custom databases. To change to a different database, click the browse button and select a different .MDB file. The settings on these tabs initialize all newly created messages, labels, data and fields on all channels on all boards.

### *MIL-STD-1553 Properties*

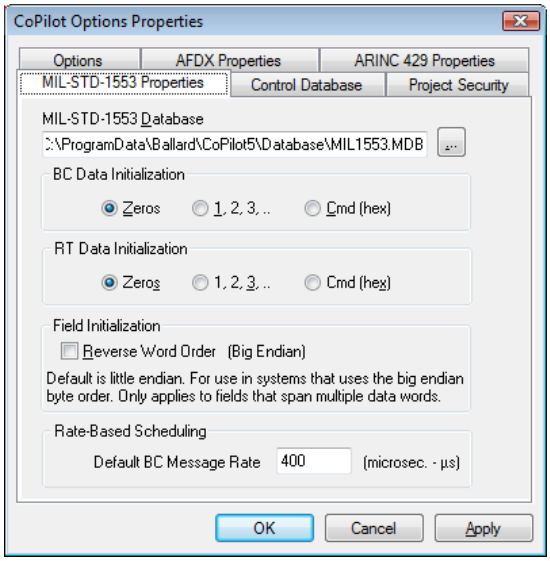
The MIL-STD-1553 tab allows you to select the MIL-STD-1553 database to use and set default initialization values. Initialization of BC or RT data sets the initial data values to zeros, incremental numbers, or command word values. The default endian ordering for multi-word MIL-STD-1553 fields is little endian assuming the least significant words are transmitted first. Enable the 'reverse



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

word order' option to default the encoding and decoding of data fields in big endian format where the most significant (high byte) data is transmitted first. The default BC message rate sets the initial rate (in microseconds) for newly created BC messages. If the BC is not using rate-based scheduling, the BC message rates are ignored.

Note: Newly created BC messages are initialized using the 'Default BC Message Rate' configured in the MIL-STD-1553 Properties tab of the CoPilot Options. This default value can be changed and it will update messages that have already been created. However, until any of the properties (even properties other than the transmission rate) on the property page are changed, the rate will no longer be updated by the default rate.

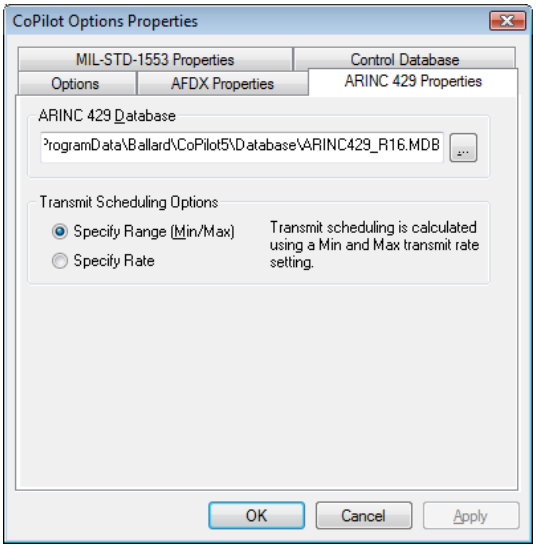



## *ARINC 429 Properties*

The ARINC 429 tab allows you to link to the CoPilot ARINC 429 database. The transmit scheduling options set the ARINC 429 transmit messages to use either the min/max range transmit interval or the transmission rate. When specifying a rate, the tolerance can be adjusted to set the maximum allowable transmit range.

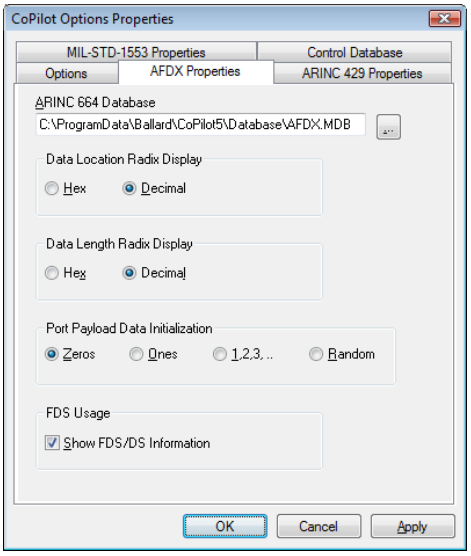


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




### *AFDX (ARINC 664) Properties*

The AFDX tab allows you to initialize Port Payload data as zeros, ones, incremental numbers, or random values for the AFDX and ARINC 664 protocol. Data length and location text displays can be shown in hex or decimal radix. To decode port payload data using FDS/DS specifications, choose the Show FDS/DS Information option.




# The CoPilot Project

## CoPilot Projects Overview

#### What Is a Project?

The project is a container that manages hardware interfaces, view windows, data connections, and additional display settings. Projects can be saved, closed, and reopened. Running a project activates the configured hardware, views, and other



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CoPilot features for interaction with the avionics databuses. See *Running a Project* on page *58* for additional information.

A project is composed of hardware and view components, and additional display settings. The components in a project contain configuration, data definitions, and display settings used to collect, manipulate, and analyze data from a variety of avionics databus protocols. When projects are saved, hardware devices and views are saved in separate files in the project directory and are referenced by the project file. For more information, see *Project File Management* on page *56*.

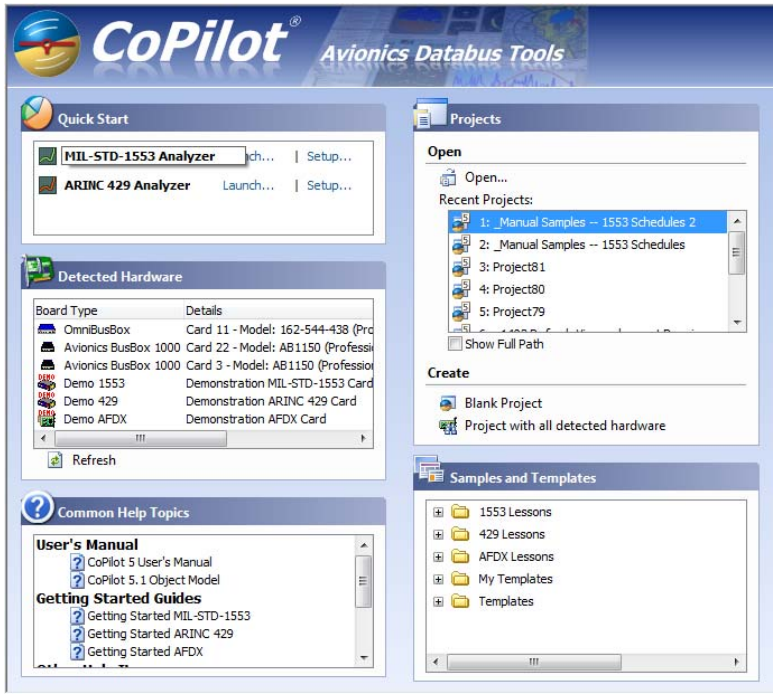
Project Security features can be set to limit users' abilities to change a project without a project's password. For example, project security settings can disable hardware configuration changes but allow data changes. Tests can then be executed by a technician without having to worry about configuration changes. Additional information for *Project Security* can be found on page *431*.

Note: only one project may be opened in the CoPilot application at a time.

## Creating New Projects

### Creating Projects from the Start Page

The primary function of the Start Page is to create and access projects. You can create a new project from any of the following Start Page panels. Note, In CoPilot version 6, the Start Page has been replaced by the Backstage. See "Project" Backstage View on page 42 for additional version 6 Backstage information.




**Quick Start Panel**—Quick Start tasks create a new project with detected hardware or demonstration hardware. These projects are not required to be saved. They can be regenerated over and over (or saved to preserve user settings).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

**Projects Panel**—From this panel you create a new blank project or a new project with all detected hardware installed.

**Detected Hardware Panel**—Double-Click on a detected hardware item to add the hardware to the current project or create a new project with this single hardware item installed.

**Samples and Templates Panel**—Selecting an item from the Samples and Templates panel creates a new project as a copy of template with all devices and view windows.

### Creating Projects using The New Project Dialog

In addition to creating projects from the Start Page (version 5) or Backstage (version 6), the New Project dialog allows you to create a new project from a template (See Samples and Templates above). To Access the New Project Dialog, select **File** | **New** | **New Project** in version 5 or the **New Project** tab group on the **Backstage** in version 6.

### Defaults Project Name and Location

The default name for a project folder is "Project***N***" where ***N*** is the lowest number available in the CoPilot directory (e.g., if Project1.CPJ and Project3.CPJ are listed in the CoPilot project directory; the default name will be Project2.CPJ). The default path is …My Documents\My CoPilot Projects\. The default names for hardware and view files in the project folder follow the same system of defaulting to the lowest number available. You can rename any of these files at any time.

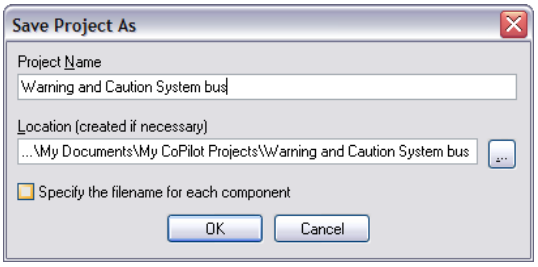
## Saving and Opening a Project

### Save the Current Project

Although projects could be saved under a default name (Project*N*), a descriptive name will make it easier to identify.

To save a project with a unique filename:

1. Choose **Save Project As…** from the File menu (version 5) or **Backstage** (version 6)
2. In the Save Project As dialog, you can specify a new filename (see figure below)
3. (Optional) You can also create a new project folder or rename the existing folder
4. Click **OK** to save the new information




As you continue to work with and modify the project, you can save the current state at any time with the **File** | **Save Project** command.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Open a Project

Once a project has been created and saved, it can be reloaded at any time. A project folder can be distributed and opened on any machine with CoPilot software installed.

To open a saved project:

1. Choose **File | Open | Open Project** from the File menu (version 5) or **Open | Open Project** from the **Home** ribbon tab (version 6)
2. Browse to a project folder and select the .CPJ file (a project can only have one .CPJ file)
3. Click **OK** to load the selected project

Before the project is loaded, the current project will be closed. If the current project has unsaved elements, you will be prompted to save these first.

---

Note: a list of recently used projects in also available on the Start Page (version 5) or **Recent Projects** tab group of the **Backstage** (version 6)

---

## Project File Management

### Project Membership and File Types

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 an ARINC 429 CoPilot 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 **Hardware Playback files (.CPB)** contain information extracted from a Sequential Monitor file to be retransmitted through the ARINC 429 board to the databus

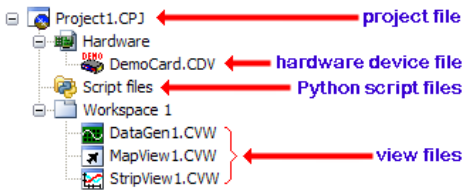
Optional **Python script files (.PY)** contain Python script code to perform the specified operations and extend the functionality of CoPilot

Optional **Monitor Data files (.MON)** are created for logging sequential monitor activity

---

Note: Monitor Data files are located below a sequential monitor object in the Hardware Explorer.

---






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

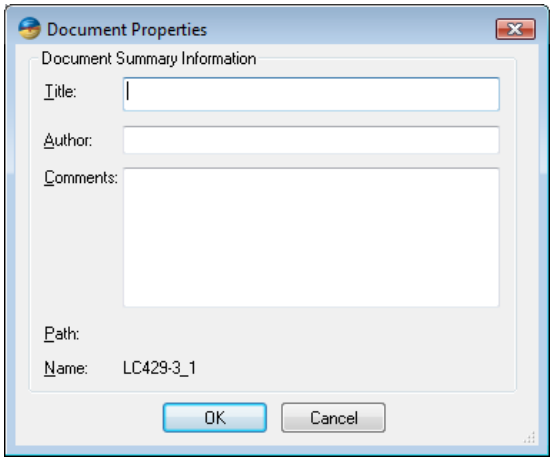
When a new CoPilot project is first saved, a project folder is created on the hard drive and the initial project file (.CPJ) is placed in that folder. As device and display components are defined and saved, they are added to the project folder.

### Sharing Files between Projects

Device and view files can be shared among CoPilot projects. You can browse to a project folder and copy and paste component files into another project folder. Alternatively, you can open a component in the current project from any project folder. When you save the current project, the filename and path of the "guest component" will be recorded in the project file (to view the filename and path, right click on the component in the Project Explorer pane and choose Properties).

### File (Document) Properties

To access document properties, select **Summary** from the context menu of an item in the Project Explorer pane. The title, author, and comment fields correspond to columns in the Windows® File Explorer® window. You can explicitly display these columns if they are hidden. Even if they are hidden, often the title, author, and other document information will be displayed in the "web view" area. You can use this information to assist you in organizing the CoPilot files on your hard drive and tracking files that you share with other CoPilot users.




*Example Document Properties*

- **Title** allows you to append a custom title to an object.
    - For a **hardware device**, the title is appended to the name in the Project Explorer pane (but not the Hardware Explorer pane).
    - For a **view** window, the title is appended to the view name in the Project Explorer pane and also appears in the title bar of the view window.
- **Author** allows you to identify the project author.
- **Comment** allows you to add comments and notes about the selected file. To view the comments in a pop-up display, position your mouse over the object in the Project Explorer pane.
- **Filename and path** display the path and name of the selected file (if it has been saved to file). This may be useful if you are using files from



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

other project folders. If you are going to share a project with another CoPilot user, you will typically copy all the files referenced in the project into a single project folder.

### Renaming Files

You can assign a custom name to a CoPilot project, hardware, and view files.

To name a component (board or view):

1. Select the component and choose **Save As…** from the File menu
2. In the **Save As** dialog, you can specify a new filename and even browse to a different project folder
3. Click **OK** to save the new information

Although the component is simply renamed in the project, a new copy is created and the original (with the old name) still exists in the project folder.

---

Warning: If you browse to a component file directly on your hard drive and rename it, the project will not be able to open it since the component links are specified in the .CPJ file. You will need to add the component to the project again.

---

You can rename a project file (and its folder) with the 'Save Project As…' command (see *Saving and Opening a Project*).

## Running a Project

CoPilot supports two basic operational modes: Edit mode (stopped) and Simulation mode (running). When a CoPilot project is activated through the Run command (**Project** | **Run**), configurations defined in the Hardware Explorer are loaded onto the hardware and CoPilot display views are activated. If the start project running on launch of .CPJ file is enabled in the CoPilot options, then double-clicking a project file will also start CoPilot running. CoPilot reverts to Edit mode through the Stop command (**Project** | **Stop**).

- **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 command, enabled 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, users can modify data, pause and restart channels, add and delete items from view windows, and perform many other operations. However, some configuration and setup options are only available in Edit mode.

### Using automation to control a project

Automation can be used to run, stop, load, save and close CoPilot projects. You can access these automation features using the command line pane, Python scripts, macros or by using an external automation capable application.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# CoPilot Conventions

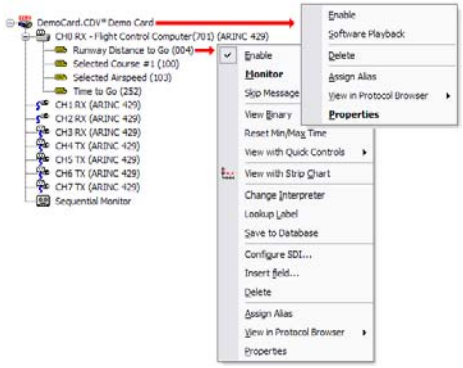
There are certain consistent principles used throughout the CoPilot environment. Understanding these principles will help you use CoPilot effectively.

### 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 Workspace Display Area. Equipment definitions, fields, and other items 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.

### 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 figure).




### Default Commands

The default command in each context menu is bolded (see previous figure). Simply double-clicking the item (without opening the context menu) will perform the bold action 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.

### Property Pages

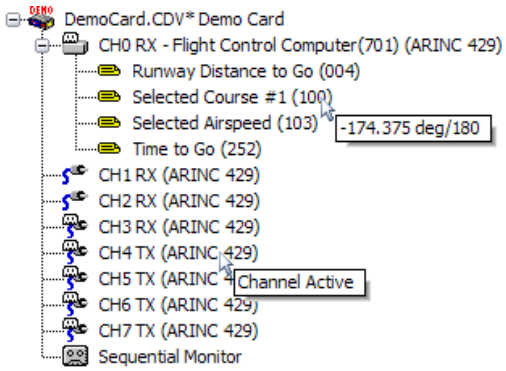
Most objects in both the Hardware Explorer and the display pane can be configured and customized through Properties dialogs. These are usually opened by a command in the context menu of the object (see the Properties commands in the context menus above). The Properties windows are often divided into tabbed subsections. Many options in a Properties dialog are duplicated in the associated context menu.

### Tool Tips

Objects in the Hardware Explorer pane implement tool tips. 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).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*Examples of tooltips in the Hardware Explorer*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Hardware Interface

## CoPilot-Compatible Hardware

### Ballard Hardware Products

The CoPilot software system is compatible with a variety of Ballard Technology, Inc. interface devices. These devices provide a consistent interface to Ballard's family of Ethernet, PMC, PCI, PC104, cPCI, USB, PCMCIA, ISA and other products. CoPilot operates in conjunction with many Ballard interface devices. A full list of the supported products and avionics protocols is found in *Appendix C: Supported Hardware and Protocols*. See the following section for additional keying information.

Note: some features in CoPilot are hardware limited. When features have limitations or exclusions, they are noted throughout this document.

## Licensing CoPilot with Hardware Keys

The licensing of hardware with CoPilot is controlled with 'hardware license keys.' To run hardware devices with CoPilot and access avionics databuses, a valid license must be installed. CoPilot can be installed and used on multiple computers for pre-run setup and for post-analysis without hardware. The hardware key information is displayed on a card's property page and in the detected hardware panel of the start page (version 5 only).

There are two kinds of keys:

- **Evaluation Key—**An evaluation key on the interface allows full use of all CoPilot features for a limited time. In the final two weeks of the evaluation period, a notification dialog will appear whenever a simulation is run. The time remaining for an evaluation key is displayed with the key information.
- **Production Key—**A production key allows CoPilot to operate through the keyed board indefinitely. These permanent keys can be purchased for CoPilot Standard or CoPilot Professional.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Installing a License Key

A CoPilot key can be installed on a board at time of purchase or added at a later time. Purchasing a "CoPilot System" (software and hardware together) provides significant saving over buying components separately.

To add a CoPilot Professional or CoPilot Standard key to an existing Ballard interface, contact a Ballard Technology Customer Support Engineer at (800) 829-1553 or support@ballardtech.com.

### Evaluating and Using CoPilot without Hardware

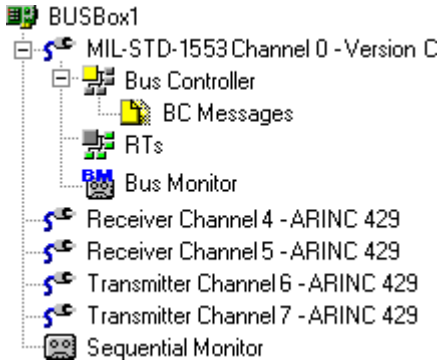
CoPilot software may be installed on any number of computers since licensing is tied to hardware interface devices and not a user or a computer. Even without a keyed board, users can still design projects, configure objects, access windows, and perform other pre-simulation tasks (in Edit mode). CoPilot includes demonstration hardware devices for several protocols with the ability to configure devices and analyze previously recorded data. The Getting Started Guides for MIL-STD-1553, ARINC 429, and ARINC 664/AFDX use the demonstration hardware to walk users through the CoPilot software. With these 'demo cards,' users can explore and evaluate the majority of the features of CoPilot without the need for actual hardware.

# Hardware Installation and Configuration

This section describes the process of adding and configuring hardware interfaces for CoPilot at the card level (the highest level in the Hardware Explorer tree). The installation of drivers and hardware into a computer is described in the *Hardware Manual* that came with the hardware.

This section does not describe any protocol specific configuration (such as channels or messages), but does highlight hardware settings available to some products (such as the OmniBus product line). Refer to the protocol section of interest (e.g.MIL-STD-1553 Protocol or ARINC 429 Protocol) for details about a particular protocol.

The Hardware Explorer pane manages the device(s) added to the current project. The Hardware Explorer provides a framework for the system configuration. Depending on the hardware type, some hardware devices contain cores that group one or more channels together. These channels can be either the same protocol or multiple protocols.




*Multi-protocol device in the Hardware Explorer Tree*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Hardware Devices in the Project Explorer

Added hardware devices appear in both the Hardware Explorer tree and in Project Explorer. The Project Explorer context menu for hardware devices allows commands such as showing, deleting, or viewing summary information. Additional Project Explorer information can be found in the *Docking Panes* section on page *43*.

## Adding Hardware Devices to CoPilot

CoPilot provides simple and powerful methods for adding hardware interfaces. Selected hardware is added to the Hardware Explorer tree and defines how the hardware will be used and how information received through the hardware can be displayed.

### Hardware Auto Detection

The Start Page's Detected Hardware panel lists all detected hardware installed in the host computer. Additionally, projects can be created selecting a detected hardware device. There are multiple options to add new auto-detected hardware to a project that include:

**Option 1**—(version 5 only) Individual hardware components can be added to the current project by double-clicking a hardware item listed in the Start Page Detected Hardware panel (if no project is open one will be created).

**Option 2**—In version 5, Detected hardware can be added to a project by selecting **File** | **New** | **Search for Hardware** from the menu bar. This command adds all detected hardware that is not currently a part of the project. In version 6, new hardware is available from the protocol **Ribbon** tabs in the **Hardware** group.

**Option 3**—You can create a new project containing all detected hardware by selecting the *Project with all detected hardware* link found on the Start Page Projects panel (version 5) or *Search Hardware Project* button of the **New Project** tab group on the **Backstage** (version 6).

### Manually Add Hardware

In addition to auto-detecting hardware, hardware can be added to CoPilot manually. This is typically done when adding additional hardware to an existing project. Manually adding hardware is also required when creating new projects that will be run on another host computer or when hardware is not connected to the computer at the time of project creation.

To manually add a hardware to a project:

1. In CoPilot 5, Select **File** | **New** | **New Hardware and Views…** to access the New Hardware and Views dialog box and select the appropriate hardware tab (for example, the ARINC 429 Hardware tab). In version 6, new hardware is available from the protocol **Ribbon** tabs in the **Hardware** group.
2. Select the hardware of your choice and click **OK** to open the board configuration dialog or wizard
3. Specify the card model, card number, or other options



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The card configuration dialogs and wizards simplify the specific parameters of each card. The following Card Configuration section describes these setting.

Note: CoPilot displays additional information for the devices present in the system and minimizes additional configuration requirements.

### Import Hardware from an XML definition

See *Hardware Definition Import & Export* on page *69*.

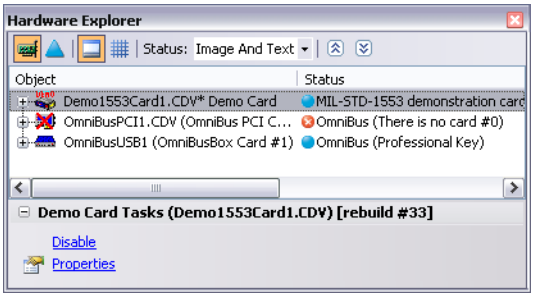
## Hardware Configuration

Hardware devices contain many properties that can be set using property pages. These settings are available for each level of the hardware in the Hardware Explorer. For example, the speed property of a channel would be configured from the property page of that channel. Context Menus and Hardware Explorer Tasks provide a faster option to changing common properties.

### Hardware Property Pages

Hardware options are located in the Card Configuration tab of the hardware property page. Properties for a Card in the Hardware/Project Explorer are equivalent to the hardware-level properties for the particular hardware device. Commonly configured properties are also available for configuration through the card's context menu.

The state of the enable property is reflected in the Hardware Explorer. Each item can individually enable and disable hardware devices. This allows a hardware configuration to exist without actually running the hardware if the hardware is not enabled. The icon of an inactive hardware is grayed out in the Hardware Explorer. When the hardware has an invalid key or if the hardware cannot be accessed, the hardware icon is marked with a red X to show there is an error (see figure below).




*Hardware Explorer with active, inactive and unkeyed hardware*

Depending on your hardware, some of the following properties may not be available.

- **Enable** setting toggles the enable/disable state of the object
- **Card Number** allows the user to specify the unique number identifying each installed Ballard Technology hardware device (crucial when multiple boards are installed). This number matches the numbers assigned to BTIDriver devices with the BTITST32.EXE application.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- The **card type** and **model/level** designations are displayed but cannot be modified. These parameters indicate the hardware type and the level of functionality (if the hardware has more than one level of functionality). Additional information on the models/levels is found in *Appendix C: Supported Hardware and Protocols*.
- The **Key Information** tab states whether or not the card contains a CoPilot key and if so, the type (i.e., evaluation or production key), CoPilot release supported (e.g., CoPilot 4.0), and version contains information about the CoPilot key used with this board, the version of CoPilot, whether CoPilot 1553 may be used with the board, and which features may be accessed.
- The **Card Information** tab displays board statistics such as firmware version and release date.

Depending on your hardware, the other tabs may be present:

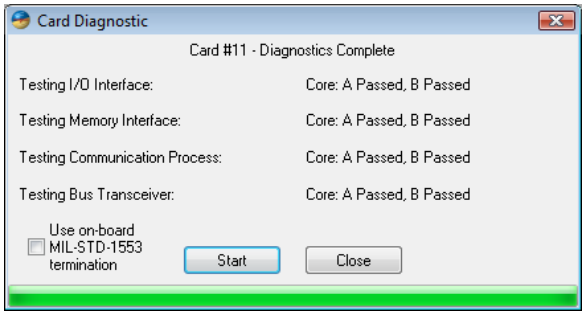
- The **TSA** (Terminal Simulation Assignment) and **Mode Code** tabs configure the 1553 bus.
- The **IRIG Settings** tab allows the user to configure the source and initialize the IRIG timer for compatible hardware.
- For more information on the **Playback** configuration tabs, see *Hardware Playback* and *Software Playback*.
- The **Card Pinouts** tab (not available for all hardware products) provides a labeled picture of the board connector with the pin assignments and signal names for quick reference. Refer to the hardware user manual for detailed information on pinouts.

### Self Test Hardware Diagnostics

The hardware self-test should be run after hardware is installed into a computer to verify there are no errors. These BIT (built-in tests) are used to verify that the hardware is working properly and communicating with the databus.

To open the Card Diagnostic dialog:

1. Right click on the card in the Hardware Explorer
2. Choose **Diagnostic** from the context menu
3. Press the **Start** button to begin testing




*Card diagnostics output*

The Diagnostic command invokes a sequence of I/O, board memory, communication, and transmission tests. If additional steps are required, such as termination settings, the diagnostic test will prompt for the necessary



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

information. Upon successful completion, each test provides a list of specific errors or displays a “Passed” message.

# Advanced Hardware Settings

Features in the advanced hardware settings section include advanced features and those features that not all hardware devices implement.

## Error Injection

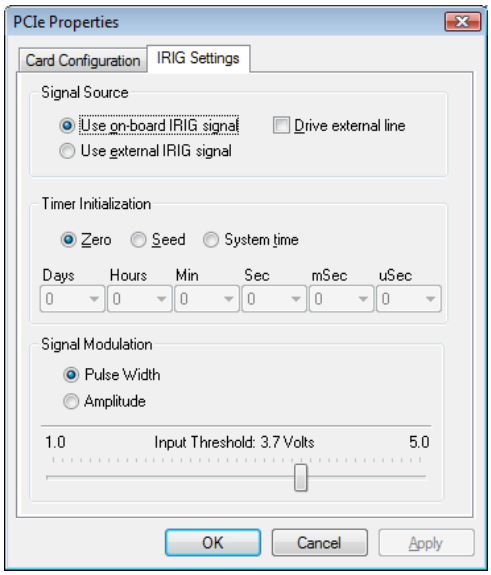
Error injection is used to create non-standard scenarios on the databus. These error conditions are useful to determine how systems respond in the presence of particular errors. Refer to the error injection section for the protocol you are using for detailed information on this topic.

## IRIG Timer

Time-tags on IRIG-capable hardware utilize an on-board 64-bit IRIG timer. At the time of publication of this manual, Omnibus, Avionics BusBox, USB (non-BUSBox), 4G and 5G products support IRIG. The IRIG specification is published by the Inter-Range Instrumentation Group and is commonly used for synchronizing equipment within a system. IRIG time ranges from days down to microseconds.

The IRIG timer can be seeded from the system clock, a seed value, a zero value, or from an external IRIG source. The IRIG configuration settings are available from the card property page for IRIG capable hardware devices.

Some hardware also supports AM (amplitude modulated) IRIG signaling inputs. Driving an external line with Amplitude Signal Modulation is NOT allowed: when **Drive external line** is enabled, the signal modulation is automatically set to **Pulse Width**. The **Input Threshold** sets the input voltage level when using amplitude signal modulation. See you hardware manual to determine if your hardware supports Amplitude Signal Modulation for IRIG inputs.




*Sample property page for IRIG configuration*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To configure the IRIG timer:

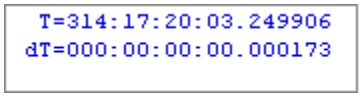
1. Right click on an card in the Hardware Explorer and choose Properties from the context menu
2. Select the IRIG Settings tab and choose a signal source
   - Select **on-board** to generate an IRIG signal for the selected hardware
     - Selecting **Drive external line** causes the internally generated IRIG values to be sent to the IRIG pins on the board connector
   - Select **external** to cause the IRIG timer to listen for and synchronize to an external IRIG signal
3. Select a Timer Initialization option if IRIG time is being generated on-board
   - Choose **System time** to seed the timer from the host computer clock
   - Select **Zero** to start the timer counting up from zero
   - Click **Seed** to specify the exact starting time using drop-down listboxes for each IRIG field (see figure above)

The timer begins counting when simulation mode is engaged through the CoPilot Run button. IRIG values are displayed in time-tags in the format show below:

ddd:hh:mm:ss.ssssss

Note: The IRIG timer signals use IRIG-B format.

The figure below shows an IRIG time-tag from a Monitor View record. The IRIG timer was initialized with system time, so the day field represents the 314th day of the year (i.e., November 11).




## Syncs and Triggers

### Overview

Sync and trigger lines allow the communication between the Ballard Technology hardware device and other systems or devices. Syncs and Triggers share the use of some of the hardware's DIO lines. The sync lines are used to output synchronization pulses, while the trigger lines are used as inputs from external devices to signal particular events to the hardware. For example, synchronization pulses can be sent when a message is received or an external trigger can be used to conditionally send different messages in the schedule.

The number and type of sync and trigger lines available is dependent on the specific Ballard Hardware used. If available, the hardware pinouts from the card property page show the available input (EXTRIG/DIN/CDIN/DIO) and output lines (SYNCOUT/DOUT/BDOUT/DOUT). The *Discrete Inputs and Outputs (DIO)* section on page *32* lists the supported DIO lines. Otherwise, **refer to**



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

**your hardware manual for additional information, pinouts and configuration of syncs and triggers**.

Ballard Technology 4G products and later (including OmniBus, Avionics BusBox, USB [non-BUSBox], and other 5G product lines) support multiple input and output discrete. Some of these discretes can be used for sync and trigger lines. Each core has three trigger (input) lines and three sync (output) lines. For these hardware devices, the Sync Output Line Selection and Trigger Input Line Selection must be configured to select which, if any, digital input or output lines will be used. Each line can be individually configured for active high or active low. Output synchronization pulses and input triggers can use one or more of these lines. For example, a trigger may be defined as a particular state of only one input or a particular combination of two or three inputs. Other triggers and syncs may use the same or different combinations of these lines.

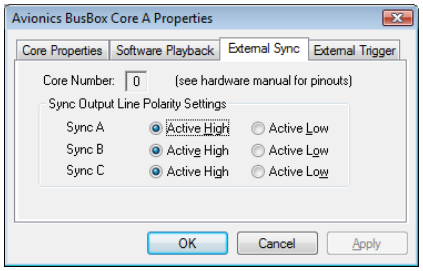
Using syncs and triggers is then done by enabling the object for either sync and/or trigger (such as a message) and then selecting the desired lines to use (if more than one line exists for the hardware being configured). If an object is capable of generating sync pulses and external triggers, the section in the manual for those objects (such as an ARINC 429 message) will contain additional information.

### Configuring the Polarity of Sync Lines

The polarity of the sync lines on each core are configurable to either active high or active low.

To configure the polarity of sync lines on a core:

1. Right click on the core icon in the Hardware Explorer tree and select **Properties** to open the Core Properties dialog
2. In the External Sync, select the polarity for each line (see figure)
3. Click OK to apply the configuration and close the dialog



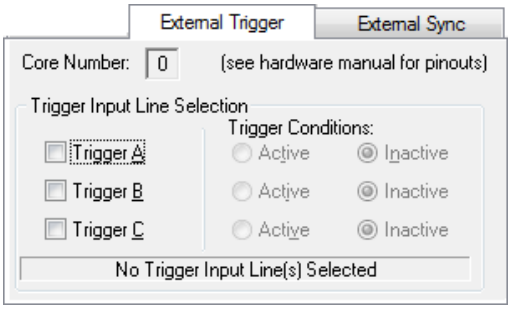

*External sync output line polarity settings*

### Triggern Input Line Selection

The Trigger Input Line Selection is used to select which input trigger lines are driven when an object (such as a message) is configured for a trigger input. This setting is found on the External Trigger tab of the object's property page as shown below. In addition to configuring this trigger condition, the trigger must be enabled on the object you wish to use the trigger (i.e. waiting for a trigger condition before transmitting a message).



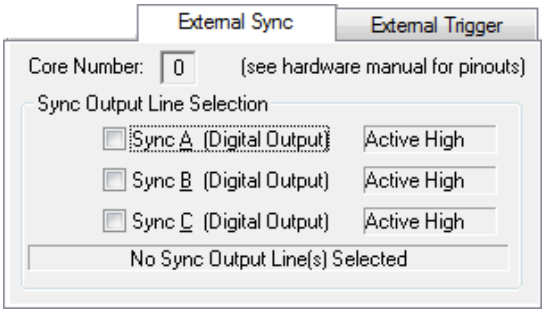
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*External trigger input configuration*

### Sync Output Line Selection

The Sync Output Line Selection is used to select which output sync lines are driven when an output synchronization event occurs. This setting is found on the External Sync tab of the object's (such as a message, channel, BC, etc.) property page as shown below.




*External sync output configuration*

## Variable Transmit Amplitude

Mil-Std-1553 level D hardware and Parametric ARINC 429 products can be configured to modify the transmit amplitude. The parametric amplitude setting can range from 0 to 100% (where 100 is the normal full-scale amplitude).

Note: The amplitude can be modulated in simulation mode, while the card is running.

# Hardware Definition Import & Export

Hardware definition (channels, labels, messages, ports, fields, interpreters, etc…) can be saved and loaded using an XML file format. Using this file format, hardware configurations can be defined or modified by outside applications and then imported into CoPilot. A similar file format is used for engineering unit import/export with database definitions. CoPilot supports other types of importing/exporting – see those sections for additional information.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Export hardware definitions

Export definitions include card, core, channel and sequential monitor settings. The channel definition includes the protocol specific settings and definitions required to re-import the definition. Specific data values are not a part of the import/export file.

To export a hardware definition:

1. Select the card object in the Hardware Explorer pane
2. Select **Export <card object name>…** from the File menu
3. Specify a file name and press the **Save** button

### Import a hardware definition

Specific data values are not a part of the import/export file.

To import a hardware definition:

1. Select **Import Hardware…** from the File menu
2. Browse to the XML file containing the hardware definition
3. Select the XML definition file and press the **Open** button



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Sequential Monitoring

The CoPilot Sequential Monitor records a sequence of data in a file so that bus traffic can be examined later. With respect to MIL-STD-1553, the Sequential Monitor is important to distinguish from the Bus Monitor. A Bus Monitor in MIL-STD-1553 is anything on the bus that passively listens to traffic on the bus, whereas the CoPilot Sequential Monitor is a software object responsible for recording the sequence of bus data to a file for later review. A time-tag is attached to each record in the sequential file. Users may capture data selectively using filters or, with a Monitor All command, users can specify that all data will be recorded in the sequential file. This section of the manual covers the key concepts and common functionality of Sequential Monitoring for MIL-STD-1553, ARINC 429 and AFDX. For details on configuration, filtering, and protocol-specific options refer to Sequential Monitoring under the protocol section of your choice.

Note: This section of the manual focuses on the MIL-STD-1553, ARINC 429 and AFDX protocol sequential monitoring. The ARINC 708 protocol, although treated similarly, is different enough to have its own section of the manual. Sequential monitoring for ARINC 708 is discussed with the rest of the ARINC 708 protocol.

## How to Record and View Sequential Data

### Recording Databus Traffic

CoPilot makes it easy to record databus traffic (see *ARINC 708 Protocol* section for recording ARINC 708 data). The quickest way to collect data is by selecting the Monitor All option in the Sequential Monitor. This option records all data on the databus. Users may also limit the recording by enabling capture filters at various levels of the protocol.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Check the Model of your hardware to see if concurrent recording of sequential monitor data while simulating is supported.

| Protocol | Concurrent Recording Support |
|---|---|
| MIL-STD-1553 | Multi-Function, Model C or higher |
| ARINC 429 | All |
| AFDX / ARINC 664 | Model B |

To monitor data:

Right click on the Sequential Monitor icon in the Hardware Explorer and check Enable in the context menu.

1. Then check the Monitor All option also from the Sequential Monitor context menu
2. Click the CoPilot Run button to begin recording
3. Click the CoPilot Stop button to stop recording and return to Edit mode

CoPilot collects the monitored data with time-tags, activity information, and detected errors into MonData files. A new MonData file is created each time CoPilot is run, unless the sequential monitor is configured to overwrite or append data. A record count with the time/date is displayed next to the MonData files in the Hardware Explorer as well as a total summation of all recordings in the status bar.

## Viewing Monitor Records

The recorded databus traffic is viewed in a Monitor View window (see *ARINC 708 Protocol* section for viewing ARINC 708 records). The Monitor View window displays data and time-tag information in several formats.

To open a Monitor View:

Right click the 'MonData' item of interest (expand the Sequential Monitor object if necessary) in the Hardware Explorer, then select 'View Monitor Data'. If there are no MonData files below a sequential monitor, then right-click the sequential monitor and select 'Create View' from the context menu.

## Optimizations for Recording to Disk

Recording large amounts of data from one or more sequential monitors can require significant system resources and disk space. Several steps can be taken to optimize the system for data logging. The following are some suggestions that can be done to improve performance.

- Limit the amount of data that is captured by configuring capture filters (see *Capture Filtering* on page *74*) and using delta and interval mode for ARINC 429 (see *Data collection settings* on page *247*).
- If using an OmniBusBox device, ensure it is connected via Ethernet for optimal performance.
- Close as many other background applications as possible (such as Outlook, Acrobat, etc.) – especially those that are resource intensive.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Disable any anti-virus software from scanning the temporary files, such as monitor data and index files, as they are being recorded. A typical path is **C:\Documents and Settings*username*\My Documents\My CoPilot Temp Data** and the system temporary directory.
- Limit how often the hardware is queried to service the current values of channels and messages by configuring channels to use the Message Sample Rate for 1553 and Label Sample Rate for 429 from the channel properties.
- Remove unused or unneeded views from the CoPilot project. These needlessly take up valuable resources (delete them from the project tree, don't just close them).
- Close Sequential Monitor Views – they do not need to be visible while data is collecting.
- If a Sequential Monitor View is displayed while recording data, remove the columns in the view that are not needed.
- Limit the number of configured labels on ARINC 429 channels. When the receive channels or the Sequential Monitor is configured to monitor all, the traffic is recorded even if receive labels are not defined in the Hardware Explorer.
- Limit the number of configured External RTs and SA (especially receive SAs that cannot edit data value) on MIL-STD-1553 RTs. When channels or the Sequential Monitor is configured to monitor all, the traffic is recorded even if these items are not defined in the Hardware Explorer.
- Turn off the following:
  - Transmit Equipment ID for ARINC 429 transmit channels
  - Auto-detect equipment for ARINC 429 receive channels
  - Auto-detect labels for ARINC 429 receive channels
  - Auto-detection of RTs from the External RT properties.
  - Auto-detection of SAs from the RT properties.
- Keep other activity on the computer running the project to a minimum.
- Keep activity in CoPilot to a minimum. Things such as configuring the monitor view display filters while running could use up valuable resources.
- Delete broken and unnecessary links (from Project -> Link Status).
- Configure as few fields as possible.

For extremely critical recording applications, additional performance can also be achieved by not configuring interpreters, and by not configuring SAs on RTs.

After making changes to the project, be sure to save so the project doesn't need to be reconfigured. The above suggestions will all help to increase CoPilot performance.

## Recovering Monitor Data

When logging data with the Sequential Monitor, data is written to file on the host PC. If the host computer loses power or the system crashes, then the data



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

file is never closed and the amount of data in the file is unknown. CoPilot can optionally periodically index the amount of valid data recorded in a temporary index file to allow for data recovery. The compromise for indexing comes at the expense of performance. This option, 'Enable Monitor File Indexing for Data Recovery,' is configured in the CoPilot Options available from the **Project | Option** menu.

The path used for temporary CoPilot files is in the temporary system directory or the 'My CoPilot Temp Data' directory typically located at **C:\Documents and Settings*username*\My Documents\My CoPilot Temp Data**. The temporary index file names generally have the format 'CPIxxxx_yyy(y)_MONINDX.TMP' where xxx is a unique value and yyy(y) is the protocol. An example index file would be 'CPI1F2E_1553_MONINDX.TMP.' The temporary data files generally have the format 'COPxxxx_yyy(y)_MONDATn.TMP' where n is the order of the temp data files.

To recover data after an event such as a power loss:

1. Open a project to recover the data into
2. Select the **File | New | Recover Data** from the File menu
3. Select the temporary monitor index file to recover

The data is then recovered into a new sequential monitor view.

Note: The temporary monitor data files referenced by the index file must all be located in the same directory to perform a restore. Once data is restored, the temporary files will be removed and restore cannot be performed again for the same files.

# Capture Filtering

## What is Monitor Capture Filtering

Capture filters limit the amount of data that is collected into the sequential record. Capture filters can be set at various levels of the protocol. Enabling the recording of all traffic for a particular level overrides the settings at the lower levels in the Hardware Explorer tree. If capture filters are not set, then the filters of the lower levels are used.

## Sequential Monitor Capture Filters

When the Monitor All option is enabled on the Sequential Monitor, CoPilot records all databus traffic, and ignores all other filter settings. If the Monitor All option is disabled, capture filters for the entire card can be configured in the Sequential Monitor property pages. For protocol specific details regarding Sequential Monitor Capture Filters refer the protocol sections.

In contrast to capture filtering, display filtering limits the visual display of data and not the contents of the data. Display filtering is described in more detail later in this chapter.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Configuring a Sequential Monitor

The *Analyzing Recorded Data* section below describes how to configure the display of logged data. Refer to the protocol sections of this manual for details on configuring the Sequential Monitor to log data for your protocol.

# Sequential Monitor Recording Options

The Sequential Monitor has options to filter and limit the amount of data to be recorded for minimizing file size and improving performance. This can be done through timed runs, which capture databus traffic for a specified time period then pauses collection, or by setting Capture Filters. Refer to the protocol sections for details regarding options for the Sequential Monitor for your protocol.

# Saving and Loading Sequential Monitor Data

### Preserving Sequential Data

The Sequential Monitor Data (MonData) files contains a recorded log of the traffic that it has seen on the data bus, minus anything that has been filtered via various capture filter settings in CoPilot. The recorded data can then be analyzed in a Monitor View. Monitoring for AFDX is slightly different, where the data is actually saved directly into a Monitor View file as opposed to a MonData file. It is important to remember to save this data in order to allow it to be reviewed at a later time. Monitor Views are displays for viewing, filtering, or analyzing data and are preserved with the project. For this reason, the default option in CoPilot is to create a new file every time a simulation is started, which guarantees that the data will be tagged for saving when the project is closed.

### Sequential Monitor Views Components

CoPilot automatically numbers each Monitor View or MonData file as they are created in the project. For example, the names for Monitor Views follow the pattern MonView*N*.CVW (where *N* is the first unused number in the project folder). The user can change the name however, through the File menu in version 5 or from the Backstage in version 6. The Project Explorer lists the Monitor Views.

Each Monitor View is saved to the project folder as a separate file. A user can have a Monitor View for each set of monitored data (MonData) in the current project. Previously saved Monitor Views can be added to the project using the File | Open Hardware or Views… command.

### File Menu Options

Sequential Monitor views are loaded and saved through the File menu or CoPilot toolbar (version 5) or the **Project** tab of the Ribbon (version 6). All Active Monitor files are saved simultaneously with the Save Project command. Monitor files may be saved selectively with the Save button  or File | Save command.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Analyzing Recorded Data in a Sequential Monitor View

## Common Sequential Monitor View Features

### Summary

This section will discuss the features that are common to Sequential Monitor Views for MIL-STD-1553, ARINC 429 and AFDX. For protocol specific functionality refer to the Monitor View section of that protocol. The ARINC 708 sequential monitor view is described in the ARINC 708 section.

Note: You can view/analyze previously recorded data while simultaneously logging and analyzing live data recordings.

### *Monitor View Toolbars*

#### Overview

The Monitor View windows have their own toolbars. These toolbars can be customized, hidden, moved or floated outside the window.

#### Standard Toolbar

The Standard Toolbar contains a collection of the most common commands. These include controls for:

- **Pause**—Pause controls the pause state of a connected Sequential Monitor.
- **Display Settings (Mode/Radix/Time)**—Display Settings control the display of data records in the list. For protocol specific Display Settings refer to the Monitor View section of that protocol.
- **Auto Scroll**—Select this option to scroll to the bottom of the record list while collecting data.
- **Split**—Select this option to show the data in two vertically aligned lists.
- **Status bar**—The status bar displays status information, the view filename, and the current log count. The status bar displays a help string when you hover the mouse over a command in one of the toolbars.
- **Print Preview**—Open a preview window displaying what a print operation would look like. The print preview operation may take several seconds to generate the display. You can cancel a preview operation at any time by pressing the cancel button on the preview progress dialog displayed during processing.
- **Print**—Printing from a monitor view will print the data record list as it is configured. All column settings and order will be printed as they are displayed in the view window.






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Display Filter Toolbar

The display filter toolbar is used to configure the monitor view display filtering. Display filtering shows or hides the selected records from the display. In addition to configuring the display filter with the filter toolbar, the various protocol specific sequential monitor views have context menu options available by right-clicking on a record.



### Goto Toolbar

These controls assist in locating records by index or to simply single step advance the selection forward or back.




The time-tag correlation source and sink feature allows for the synchronization of data log entries displayed in Monitor Views between multiple busses, boards, or even databus protocols. The **Time Correlation Source** sends the time synchronization message to all other monitor views. The **Time Correlation Sink** option updates monitor views with this option enabled to show and highlight the appropriate record.

This feature is described in further detail in the following *Time-tag Correlation* section.

### Search Toolbar

Monitor search features can be engaged on a completed file or while data is being collected. Search criteria are expressed through a data editor and navigation is controlled through direction keys in the Search toolbar. To learn more about the search criteria available for your protocol refer to the Monitor View section of that protocol.



### Playback Toolbar

Playback controls are used when the Monitor View is configured as a Software Playback data source. See *Software Playback*.

- **Ignore Breakpoints**—When set all breakpoints are ignored
- **Go to next breakpoint**—Searches forward to find the next breakpoint
- **Go to previous breakpoint**—Searches reverse to find the next breakpoint
- **Seek to playback begin**—Highlights and displays the playback start location
- **Seek to playback end**—Highlights and displays the playback end location
- **Speed**—See *Software Playback*





Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Time Correlation

The time correlation source and sink feature allows for the synchronization of data log entries displayed in Monitor Views between multiple busses, boards, or even databus protocols. The time-tags of records are used for finding the nearest record.

- **Time Correlation Source**—When set, changes to the current record selection cause a time-synchronization message (containing the record time-tag) to be sent to all other monitor views in the project. Changing the selection, using the goto commands, and auto-scrolling will all send messages that monitor views can sink.
- **Time Correlation Sink**—When set and a time-synchronization message is received, the view will update the selection and scroll to make visible the record nearest the time of the time-synchronization message.

A sequential monitor view can be configured as either a source or sink, both source and sink, or with none of the time correlation options. In addition, if multiple views are configured as sources, then changing the selection on any of those views will send a time correlation notification message.

A shell script method (**TimeSyncViews**) is also available to generate time-synchronization messages from scripting. This script method sends a time-tag value to all monitor views in the project. Any view set as a Sink will select and scroll to the record nearest the transmitted time-sync time-tag.

Similar to the time correlation feature, the ARINC 429 **Compare Record** feature allows side-by-side comparison of data records in the same sequential monitor view. The compare record feature is described in more detail in the *Monitor View* section on page *247*.

Note: Multi-item selection (multi-select) is not possible when the **Time Correlation Source** option is enabled for the Monitor View; only one record may be selected at a time.

## Display Filters

Display filters limit the displayed data to only the data records that match a set of user defined criteria. Display filters can be specified to limit data down to single source or to eliminate a source. Criteria can be based on addressing and/or data values. To learn more about display filters available for your protocol refer to the Monitor View section of that protocol.



## Export Sequential Monitor Data

Export allows recorded data to be exported to file for use by another application. The exported data columns and display radices are based on the current view settings. The exported record set is based on the current display filter criteria. Export can also be limited to only include selected records (highlighted list items) or the filtered items (items not shown will not be exported) by changing the 'Export Range' settings. Even though the RAW data export format does not export engineering unit information, engineering units may be applied later as described in the following *Changing Engineering Unit Interpretations* section.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: Only the RAW Data export option is the only data format that can be re-imported into a Monitor View (See Import).

When exporting to a Microsoft Excel® .XLS file, some data (such as when multiple fields are defined per message) can contain more than one line of text. Be sure that the application viewing the data is configured to expand rows to view multiple lines or do so manually.

## *Import Sequential Monitor Data*

Import allows previously exported .RAW monitor files to overwrite or append the current Monitor View records.

The RAW file format is a delimited ASCII file containing raw, uninterrupted, data values. Each data value in the file is a hexadecimal number. The information in the file contains the same information as the data in MonData and Monitor View files, but is human readable and does not contain any engineering unit interpretations. Although the imported RAW field does not inherently contain engineering unit interpretations, interpretations can easily be applied to the imported data. The following section describes the steps to assign engineering units to imported data.

The exported data in the RAW file may be modified or created before importing provided that new file matches the proper file format.

## *Changing Engineering Unit Interpretations*

The engineering unit interpreters defined for the hardware being recorded by the Sequential Monitor are stored with the data in the Sequential Monitor Data Files. The engineering unit interpretation of data may be changed by applying the currently configured engineering unit interpreters from a core (or card for 3G and earlier hardware devices). Reapplying the engineering units does not change the stored data; only the display of the data in the Sequential Monitor View changes. Applications of this feature include adding engineering unit definitions to imported data or to data initially recorded without any engineering units.

To change engineering units for sequential monitor data, first display the data in a sequential monitor view. Drag a MIL-STD-1553 channel onto the monitor view displaying the data you want to reapply with the 1553 interpreters defined on that channel. Drag an ARINC 429 core onto the monitor view displaying the data you want to reapply with the 429 interpreters defined on that core. The channel/core dropped on the monitor view does not have to match the hardware configuration that the data was recorded with. For example, if the data was recorded without any engineering units defined; later, engineering units can be defined for a channel/core then reapplied to the stored data. Note that the reapplication of interpreters can be done more than once.

## *Data Plot*

The plot data command command creates a DataPlot strip chart showing all recorded data points of selected values recorded in a sequential monitor. A data plot can only be created while the CoPilot simulation is not running. The DataPlot and Strip Views both share the same display setting; See the section on Strip Views for additional information. The data plot is a CoPilot Professional feature.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### *Custom Display Settings*

The 1553 and 429 Monitor View windows have the option of displaying the records in a predefined layout or using custom display settings. The AFDX monitor always uses custom display settings. To customize the display set the Mode to a custom selection and select **Configure View** from the view context menu. This will open a dialog listing all of the available columns for display. To save the selected configuration of columns as the named custom mode configuration, rename the configuration (from the default names: CustomUser1 or CustomUser2) as desired and press the **Save as Default** button. Select the columns of interest and press **OK**. Additional settings include time display, radix display, auto scroll, and many other settings. Once the display is configured as you normally want it, can press the **Save as Default** button (or select **Set as Default** in the context menu) to preserve these settings to the computer registry to be used as the default. Then, any newly created monitor views will use the updated settings.

### *Optimize Columns*

The optimize columns command expands or contracts the width of each displayed column to fit its contents. This setting does not affect the following display modes for the MIL-STD-1553 Sequential Monitor View: Form, Statistics.

### *Bookmarks*

Users can tag individual data records using bookmarks. Bookmarks are identified by a bookmark icon in the record index column. Bookmarks can contain an optional comment that is displayed in the tooltip of the bookmark icon or accessed from the record item context menu. Bookmarks can be used as search criteria as well as exported with the record.

## Software and Hardware Playback

#### Summary

A Monitor View can be set as a playback data source. Because of this several playback features are a part of the view. For more information about Software or Hardware Playback see the *Software Playback* and *Hardware Playback* sections of this manual.

#### Start and End Indexes

By default, the entire file contents will be used for playback and continue indefinitely. Alternatively, users can replay a portion of the playback data by setting Start and End Indexes. Setting the start and end index defines a subset of the monitor record. When the playback simulation is run, only that subset is replayed.

#### Playing the data backwards

If only the end index is set, playback will begin from that point and replay backwards.

#### Breakpoint

A breakpoint causes software playback to halt. Breakpoints may be applied or removed at any time, even when software playback mode is turned off. However, breakpoints perform no function outside of software playback. When



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

the software playback simulation is started, it will enter a paused state when it encounters a breakpoint. To continue from that point, press the Monitor Pause button to resume the playback. To continue from a different breakpoint, use the Go To Previous Breakpoint and Go To Next Breakpoint buttons, then resume the playback.

### Speed Control

The Speed Control controls the relative playback speed for software playback. When the playback is slowed, changing data values are more easily detected than under normal run conditions.

### Single Step

In software playback mode, the step commands cause the next or previous record to be replayed when stopped at a breakpoint.

### Set As Next Message

Use the Set as Next Message command to specify a new resume location in the file when stopped at a breakpoint.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Engineering Units

## Engineering Units Overview

Engineering units, or engineering unit interpreters, translate data from messages and fields into human readable values. Interpretations can vary from string to floating point values to custom interpretations. Although some protocols have their own specialized interpreter types, the basic concept of converting raw data into engineering unit values remains the same. An interpreter does not change the value of the data it is interpreting. However, an interpreter can provide an editor specialized to a particular type of data interpretation (e.g., a graphical slider for scalar values). CoPilot ships with a predefined ARINC 429 database of engineering unit interpretations. You can define a custom database in CoPilot to hold your definitions, such as those from an ICD (interface control document). A complete list of the available engineering unit interpreters is found in the protocol specific sections.

#### Predefined Engineering Unit Conversion Databases

CoPilot ships with databases for each protocol that supports engineering unit conversions. The ARINC 429 specification contains a predefined database of engineering unit interpretations and these have been incorporated into the standard ARINC 429 database for CoPilot. The ARINC 429 specification standardizes engineering unit definitions to for transfer of data between instruments. This is in contrast to MIL-STD-1553, ARINC 664/AFDX, and other protocols which do not specify the formatting of data in specific messages. For example, in MIL-STD-1553, the data definitions in each message (as well as the RT address) is left to system designers.

#### Engineering Unit Definitions

CoPilot allows the user to quickly and easily define all the parameters of a message and its data field(s). This definition is automatically propagated throughout the CoPilot project, which can be saved and reopened with the engineering units translation and specifications intact.

Messages and fields can be saved to a database. Then the database definitions can be shared with other users or used to load engineering unit definitions in different projects. Once engineering units are defined, data can be viewed and manipulated in engineering units in various displays and controls. Thus, CoPilot allows the user to translate their ICD or other set of data definitions into easy-to-use graphics and displays that correspond directly to aircraft controls or other real-world components.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CoPilot automates the defining of message parameters, including the following:

- Name
- Interpreter/Signal Type (BNR, Discrete, etc.)
- Location and Length (start/end location, and number of bits)
- Units of Measurement
- Range (Min/Max Value)
- Resolution

# Create an Engineering Units Definition

Engineering unit interpretations are generally configured for fields. However, for ARINC 429, interpreters may be defined on messages (labels) and fields. This section provides a general overview for defining engineering units.

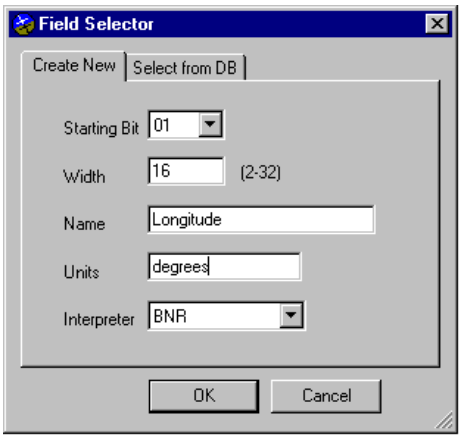
### Create Message

An engineering unit interpreter is defined for fields and for ARINC 429 messages; so the first step is to create a message. This is either a BC message or a subaddress for MIL-STD-1553, a message on an ARINC 429 channel, or an ARINC 664/AFDX port. Since interpreters can be defined for ARINC 429 messages, you can skip down to the "defining the interpreter" section below if you are not creating a field. Sections specific to each protocols describe the process of creating messages in detail.

### Create a Field and Select Interpreter

This section describes the insertion of a data field with engineering unit interpretation within a message. Sections specific to each protocols describe the process of creating messages; here the focus is on the engineering unit definition.

Insert a field from the message context menu in the Hardware Explorer tree. Here the location of the field, name, and units are defined. The interpreter type is selected from a drop list of available interpreters.



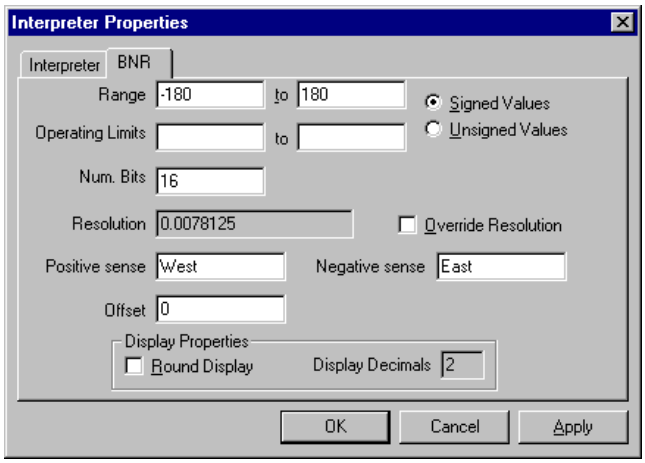

The interpreter you choose depends on the how the data is encoded/decoded (e.g., 2's complement, unsigned numeric, discrete bit, etc.). Once an interpreter is selected, the properties of the interpreter can be defined.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Defining the Interpreter

As a field (or ARINC 429 message) is being inserted, an interpreter must be selected. Before the insertion is complete, a property page to configure the interpreter is displayed. The image below illustrates a property page for a scalar (or floating-point) interpreter. The resolution 'scales' the binary value.

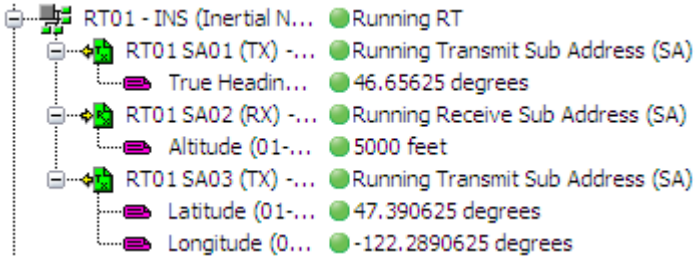



# Engineering Units Value Display

Once the raw data in a field is associated with an interpreter and translated into engineering units, the data may be displayed and manipulated in engineering units everywhere else in the CoPilot environment. The following are examples of some of the ways engineering units can be used in CoPilot. See the *Generate and Edit Data* section for additional data generation and editing information.

### Hardware Explorer Tree

CoPilot's interactive Hardware Explorer tree also uses the defined engineering units. The status column in the Hardware Explorer tree displays the current engineering value as shown in the image below.



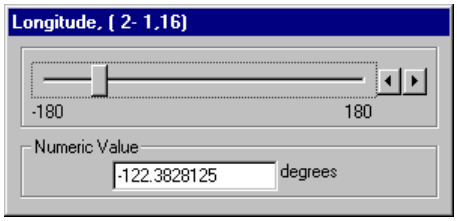

### Engineering Unit Editor

Double-clicking the field (or ARINC 429 message) item in the Hardware Explorer tree brings up the engineering unit editor if the value can be edited (see figure). Values for ARINC 429 transmit channels, ARINC 664/AFDX transmit VLs, MIL-STD-1553 transmit RTs/SAs, and MIL-STD-1553 receive BC Messages are editable. If the value cannot be edited, then the raw view radix display will be shown to see the current value in engineering units and various radix formats.



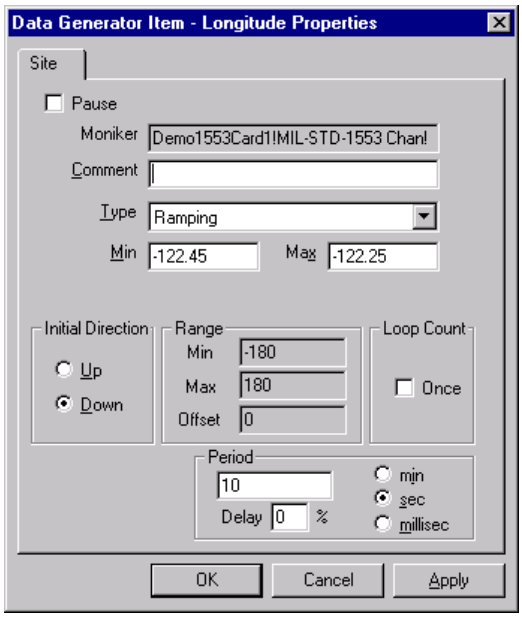
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The engineering unit value editor displays the range, current engineering units data value, and units. The name (and location) is displayed in the title bar. To change the data dynamically, slide the bar with the mouse, use the buttons on the right for smaller increments, or type a value in the text box.




### Data Generator

The Data Generator can be used to automatically modify data values. Drag a field or message into a Data Generator View and configure the data generator type.




For example, the ramping data generator type updates the values as it ramps up and down between the min and max every period.

### Map View

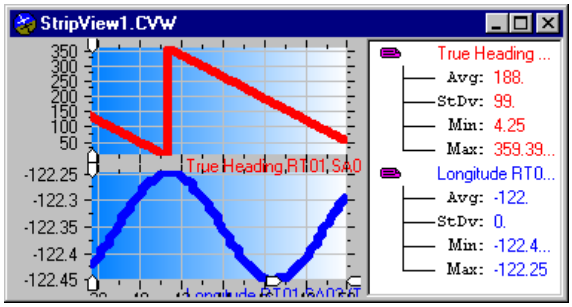
Drag a field or message, such as a Longitude value, from the Hardware Explorer tree onto one of the properties of a Map View window to source the data. Data from the field/message controls the map property. The name of the field/message and the current value are displayed.

### Strip View

When dragged into a Strip View window, the name is displayed (if the caption option is toggled on). Notice that the range of the y-axis automatically adjusts to the engineering units range of the displayed field. When an engineering value is added onto the Strip View window, the range automatically adjusts to the interpreter's minimum and maximum values.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

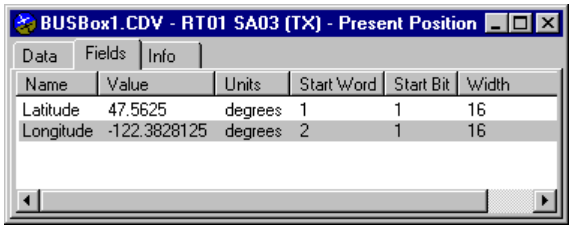



### Engineering View

Multi-protocol view for watching data change with engineering units. Additional columns can be added to display more information Display columns include name, value, individual bits, hex, octal, binary, status, errors and others. See the *Engineering View for MIL-STD-1553* section for more information about using the Engineering View with 1553. See the *Engineering View with ARINC 429* section for more information about using the Engineering View with 429. See the *Engineering View with ARINC 664/AFDX* section for more information about using the Engineering View with AFDX.

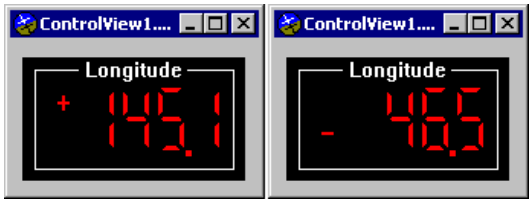
### Message/SA View

Double click either the BC message or RT subaddress containing our Longitude data field. The Message/SA View will open. The message or subaddress name appears in the title bar of the window. Click on the Fields tab to view our Longitude field.




### Control View

To further enhance the real-world presentation of engineering unit data, it can be displayed with graphical controls. The Control View provides the ability to customize a display with multiple controls. For example, a simple LED display of a few engineering units values are shown below. Quick Views provide a simple alternative to the Control View where a single display type is already selected and engineering values can be dropped on the view from the Hardware Explorer tree.




*Two examples of LED controls in a Control View*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

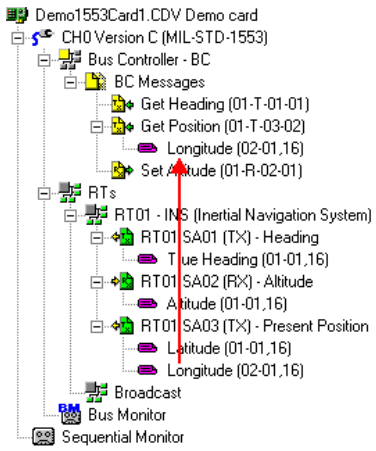
# Copying Engineering Units Definitions

Once a data field or message is defined, the field and engineering unit interpreter information can be quickly and easily duplicated to other objects in the Hardware Explorer tree. This section describes the drag and drop operation; the following section describes the use of the database for saving and retrieving engineering unit interpreter definitions.

### Drag and Drop a Defined Field

Once a field is defined, the field can be dropped onto another message to copy the field. Simply drag a defined field from one message and drop it on another in the Hardware Explorer tree. This will copy the units definitions of the data field are transmitted.

Note: Drag and drop operations for item in the Hardware Explorer tree are limited to the same protocol. For example, you cannot drop a MIL-STD-1553 field onto an ARINC 429 message.




*Drag and drop of a field between messages of the same protocol*

# Engineering Units Database Definitions

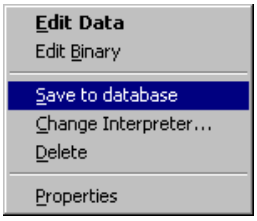
## Save Fields and Messages to the Database

CoPilot allows field and message engineering unit definitions to be reused by saving them to a database. Protocols that do engineering unit conversions each have a protocol specific database as described in the *User Interface Options* section on page *51*.

Right click on the field/message and select the Save to database command as shown in the following image.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




A verification dialog window will open asking you to verify that you want this field saved to the database. Select OK to approve this action.

Note: If a message or field already exists, the user will be prompted to overwrite this information in the database. For this reason, do not forget to make backup copies of your database.

## Database Import & Export

Included with CoPilot is a separate BTIMDB.EXE utility for database importing and exporting MIL-STD-1553, ARINC 429, and AFDX database definitions from an XML format. More information about this BTIMDB.EXE utility is described in specific protocol sections. The MIL-STD-1553 protocol section *XML database import/export utility* can be found on page *213*. The ARINC 429 protocol section *XML database import/export utility* is on page *258. F*or AFDX, see *XML import/export utility* on page *317*.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Generate and Edit Data

## Data Control Overview

### Introduction

CoPilot provides several methods for modifying and manipulating data transmitted by Ballard Interface Hardware. Precisely defining data values bit by bit or applying user-friendly "engineering unit" definitions to data fields (or 429 labels) can both be done with ease. CoPilot also provides provisions for automatically generating data, through several different methods.

### Data Editing Tools

Editing data in CoPilot can be done in several ways:

- Engineering Units Editor—a convenient way to handle data in "human terms."
- Raw Editor—a special editor that allows message data to be entered in binary, octal, or hexadecimal radices.
- Data Editor (1553 only)—a tool for setting the value of data words in a message or subaddress. Please see the 1553 Section for details.
- Port View (AFDX only)—a tool for setting the value of data within a port. Please see the AFDX Section for details.
- Command prompt (CoPilot Professional)—an ATE pane used to type in commands (e.g. Field1 = 101.05).
- Watch Window (CoPilot Professional)—an ATE pane used to view and edit variables and object properties.

### Data Generating Tools

CoPilot also makes it possibly to generate data automatically through several methods:

- Data Generator—an engine for creating dynamic streams of data in a variety of waveforms and patterns



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Scripts—Python and Visual Basic scripts can provide methods for modifying field values on-demand or when upon particular event(s) (see Automated Test Environment (ATE) for more details)
- Data Modifier—a control (hosted in the CoPilot Professional Control View) that receives, modifies (by user-defined criteria), then re-sources data

## Engineering Units and Raw Values

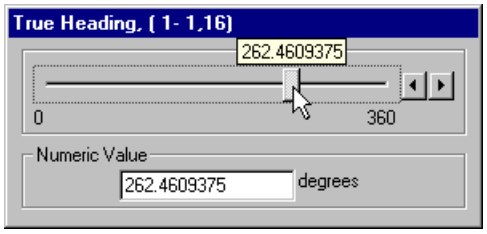
Data can be edited or viewed in CoPilot in a raw format (binary, hex, octal) such as through the Raw Editor) as well as using Engineering units (through the Engineering Units Editor). In some cases, it makes the most sense to modify data in its raw state, while in others it may be more useful to edit using a human-readable value.

Data above the field level (such as the subaddress in 1553 or the port in AFDX) is often only viewable/editable in its raw binary format, since no data definitions are made at this level. Through the use of ICDs, users can define "fields" or "labels" (in the case of 429) to provide decoding instructions for particular sequences of bits within these messages. CoPilot can then decode these bits and display them in their human-readable form (engineering value).

# Engineering Unit Editor

## Description

The most basic and easiest way to edit data using engineering values is through the Engineering Unit Editor. The Editor exists as a modeless dialog, and clicking away from the Editor will cause it to disappear, making editing particularly efficient.




## Using the Engineering Unit Editor

Double clicking on a field or label (429) within the Hardware Explorer (or, optionally right clicking and selecting "Edit Data") will bring up the Engineering Unit Editor. The title bar will show the name of the field, and some numbers that indicate its location and position within the message.

Note: If the (**Shift**) key is held down while double clicking, the Raw Editor for editing values in binary, octal and hexadecimal is shown instead.

The Engineering Unit Editor consists of a slider and a text box. Moving the slider will change the value within the text box. The maximum ranges of values possible for the field are specified by the interpreter's "Range Min" and "Range Max" properties. As the value changes from moving the slider, the transmitted



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

value held by the field is updated constantly. Thus, it is possible to simulate a sliding range of values just by moving the slider around.

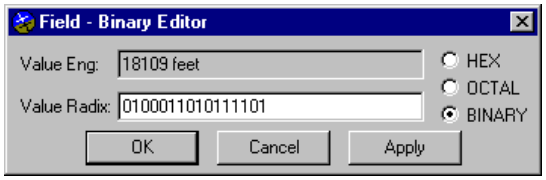
The value can also be changed by clicking the arrows next to the slider, or by simply entering a new value manually into the text box. Hitting enter will apply the newest value and close the Engineering Unit Editor.

Note: There is no “OK” button on the Engineering Unit Editor. Click outside the dialog box to close the Engineering Unit Editor.

# Raw Editor

## Description

The Raw Editor allows the user to modify data in a field in any of three radices while viewing the value in engineering units (when available). Values can be defined through the Raw Editor before or during Simulation mode.




## Using the Raw Editor

Double-clicking while holding the (**Shift)** key on a field or label (429) within the Hardware Explorer (or, optionally right clicking and selecting “Edit Raw”) will bring up the Raw Editor. The title bar will show the name of the field, and some numbers that indicate its location and position within the message.

The current value in your current radix as well as the engineering value are displayed in the top and bottom text boxes. Changing the current radix displayed in the bottom text box is accomplished by selecting one of the different radices from the radio buttons on the right. Note that changing the radix will show the value in the bottom text box in the new radix. Then, the value may be changed by changing the value in the bottom text box. To apply the new value to the field and see its engineering value, press **Apply**. Pressing **OK** will save the current value and close the dialog.

# Data Generator

## Overview

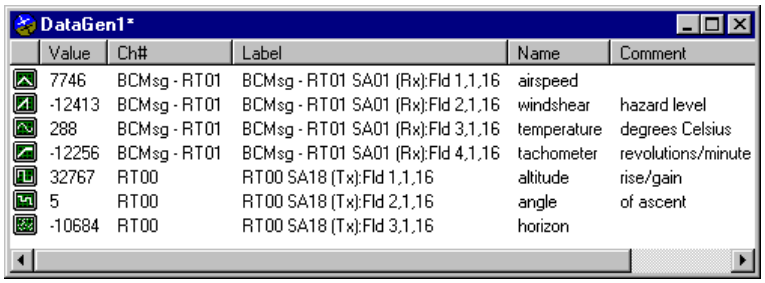
### Description

The CoPilot Data Generator creates a continuous stream of changing data values while the simulation is running. Data Generator views are special-purpose containers for fields and labels (429) that are responsible for transmitting data values onto the databus. Values are based on the user-defined range, cycle time, direction, and the Data Generator function selected. Multiple Data Generator views may be included in a CoPilot project.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Depending on computer speed and other processes running, the Data Generator may increment data values several times between field transmissions (or in some cases, a field can be transmitted several times between updates from the Data Generator).




### Accessing

The Data Generator may be accessed from **File | New Hardware or View** and then by selecting the "Standard Views" tab and then "Data Generator View." Optionally, you may click the Data Generator icon on the toolbar to create the view directly (version 5) or the **Create** group of the **Home** tab of the ribbon (version 6).

## Adding Fields or Labels

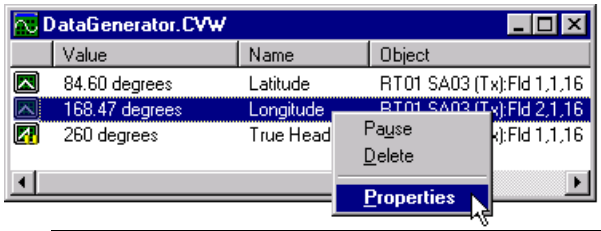
In order to make use of the Data Generator's functionality, a field or label (429) must be dragged from the Hardware Explorer and dropped into the view.

Tip: Dragging an object such as a channel into the Data Generator will automatically add all (transmit) fields that branch beneath it to the view window.

An item can only be dragged into a Data Generator once. Dragging the item or its parent into the view again will only replace the already present item rather than creating a duplicate.

## Accessing Field and Label Properties

Once an item has been added to the Data Generator, it can be modified through its context menu:




| Name | Description |
|---|---|
| Pause | Halts data generation for that item |
| Delete | Removes the item from the Data Generator window |
| Properties | Opens the Data Generator Properties window, where the Data Generation settings for that item can be configured |

Note: right-clicking inside the view but not on one of its fields brings up a different context menu:

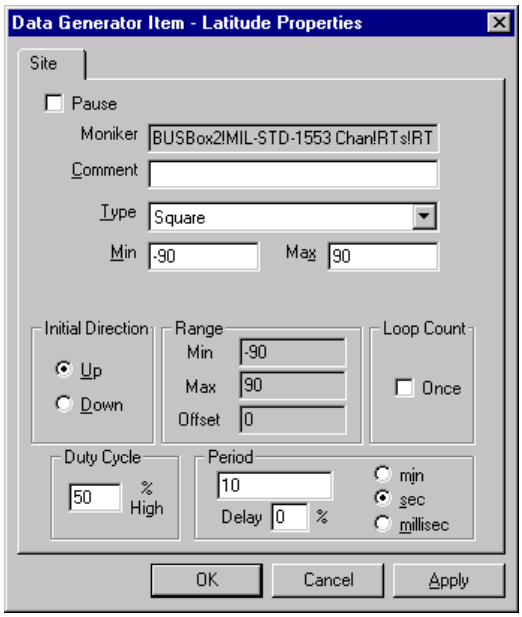


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| Name | Description |
|---|---|
| Pause All | Halts data generation for all items in the Data Generator window |
| Unpause All | Resumes data generation for all items in the Data Generator window |
| Delete All | Removes all items from the Data Generator window |

## Item Properties

Each object assigned to a Data Generator view is defined by customizable properties. Double click on the object in the Data Generator view to access the properties dialog box.

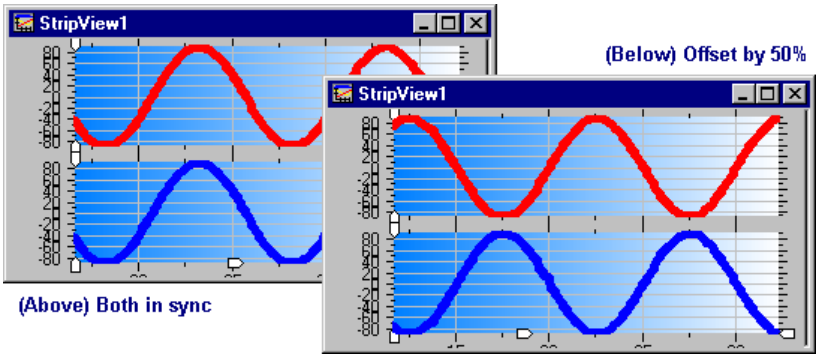



Here are the various properties available via this property page:

- **Pause**—Allows halting/resuming of data generation for the item.
- **Type**—There are eight data patterns: ramping, sawtooth, sine, approach, square, data list, random, and complex ramp. For a detailed description of each type, see Data Generator *Function Types*.
- **Min/Max**—The Min and Max boundaries can be specified up to the full available range (as displayed in the Range frame)
- **Range**—The range values are informational only, reporting the min/max and offset values specified for that item
- **Direction**—The Up/Down option buttons define the initial direction of Sine, Sawtooth, Ramp, Square, and Approach functions (selected in the Type box). For the Data List type, Up = Forwards and Down = Backwards through the list.
- **Loop**—When the Once box is checked in the Loop Count frame, the selected data pattern will execute once, then halt.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Duty Cycle**—The duty cycle applies to the Square wave only and defines the percentage of data transmitted at the maximum value. Square values alternate between a minimum and maximum level at specified intervals.
- **Period**—The period defines the full cycle in minutes, seconds, or milliseconds. For the Approach pattern, this is the time it takes for the approach value to change between the min and max value to arrive at the steady-state value.
- **Delay**—This setting offsets the waveform in time by the specified percentage. For example, if two fields in the Data Generator have the same configuration and starting value, their patterns will be in sync (left Strip View below). If the second item is delayed by 50%, it starts at the halfway point of its pattern and the two items will be in exact opposition or 180 degrees out of phase (right Strip View below).
- 


## Function Types

The eight CoPilot Data Generator functions are described below:

- **Ramping** values increase gradually from the minimum to the maximum value in half the interval time and decrease gradually to the minimum value during the second half of the interval.
- **Sawtooth** values increase gradually from the minimum to the maximum value then reset to the minimum value and repeat the process
- **Sine** creates sinusoidal data between the minimum and maximum values.
- **Approach** values increase gradually from the minimum to the maximum value over the full interval then continue transmitting that same value.
- **Square** values alternate between a minimum and maximum level at specified intervals. The duty cycle is associated with the square wave (only) and sets the percentage of data transmitted at the maximum value.
- **Data List** transmits a sequence of specific values defined by the user. Data list and relative time values may be entered manually or copied from a delimited file (for example, you can copy and paste a column of data from Excel and it will be automatically reformatted).
- **Random** values between the upper and lower range are continuously assigned during the run.

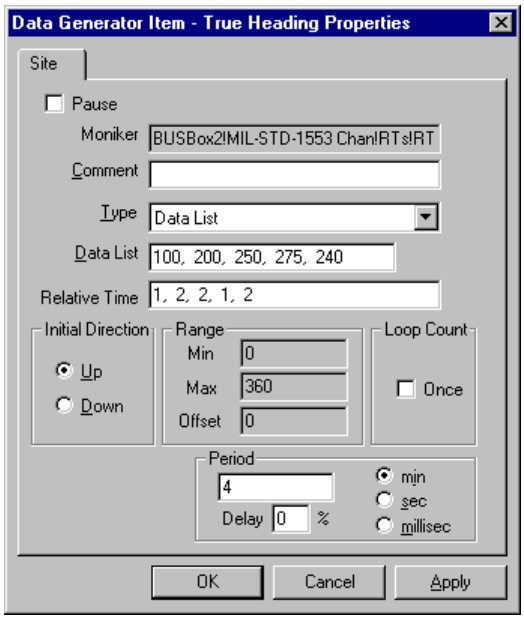


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Complex Ramp** creates a sequence of linearly interpolated values between specified data points. A second data sequence can be specified to define the relative time intervals between data points. The number of data values must match the number of relative time values if any relative time values are used.

## Building a Data List or Complex Ramp

Both the Data List and Complex Ramp types in the Data Generator use the Data List and Relative Time fields to control the stream of data.




Data list and relative time values may be entered manually through the keyboard by typing a sequence of engineering values in the Data List text box separated by blanks or commas. Alternatively, values may be copied from a delimited file. For example, you can copy and paste a column of data from Excel and it will be automatically reformatted.

When transmitting, data values are placed in the on-board message buffer at even intervals over the specified period. In the example above, values of 100, 200, 250, 275, and 240 will be loaded into the message buffer. If the Relative Time textbox was blank, during the 4-minute interval, each value would be transmitted for 48 seconds. Based on the relative times in the figure above, the first and fourth values will be transmitted for 30 seconds and the second, third, and fifth values will be transmitted for 60 seconds. The number of data values and relative time values must match.

For the Data List type, the Up/Down option buttons in the Initial Direction frame control whether the generator moves forwards through the list (Up) or backwards (Down).

# Data Modifier

### Description

The Data Modifier Control allows users to configure fields or labels (429) for value scaling. It is an ActiveX component that takes an input value from one



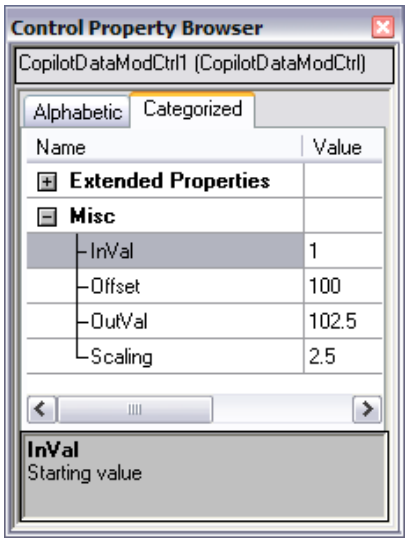
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

object, performs a linear translation, and sets an output that can be linked to another object. The Data Modifier Control allows for both a scaling factor as well as a numeric offset.

### Accessing

1. First, a Control View must be created from **File | New Hardware or View** and selecting it from the "Professional Views" tab.
2. Right-click the view and select **Design Mode**.
3. Click the left mouse button on the **Data Modifier** button on the controls palette to select it.
4. Drag the mouse across onto the Control View window to place the control in the window.

### Configuring




First, you must configure the scaling and offset on the Data Modifier:

1. Select the Data Modifier control to access its properties in the Control Property Browser
2. Type the multiplier and offset values in their respective textboxes

   *(For example: Output Value = 2.5 x Input Value + 100)*

Then, the input field must be linked to the input of the Data Modifier:

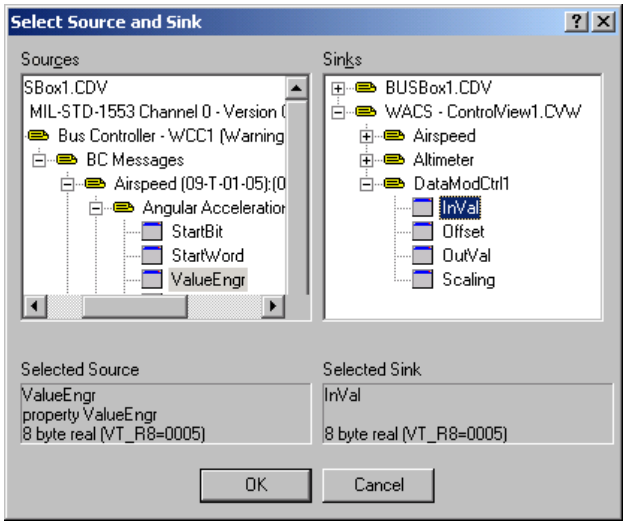
1. Click Project | New Link to link to identify and link the input value
2. Within the Source pane on the left, expand and click to select the input property from the input field or label (e.g. ValueEngr)
3. Within the Sink pane on the right, expand and select the control property receiving the input value (InVal)

Finally, the output of the Data Modifier must be linked to an output field:

1. Click **Project | New Link** to link to identify and link the input value
2. Select the OutVal property from the Data Modifier in the Source Pane
3. Link it to the output field by selecting an appropriate field or label property (e.g. ValueEngr) in the Sink Pane



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*Linking an engineering value to the data modifier control*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Standard Displays

## Display Windows Overview

### Display Window Introduction

Display windows are containers to view, manipulate, or analyze avionics data. For example, the Monitor View display allows you to view, filter, search, and analyze recorded messages.

These displays can be created, opened, and saved via the **File** menu, the New , Open , and Save  buttons in the CoPilot standard toolbar or created from one of the buttons on the CoPilot Standard Views toolbar (version 5). In version 6, these operations are available from the **Backstage** and the **Home** ribbon tab. Once a display is created, it becomes a part of the project it was created in until it is physically deleted from the project. Deleting the underlying file will not remove it from the project, but will cause an error the next time the project is opened, indicating the file is missing.

Note: Display window 'Views' are created in on a workspace tab in the 'Workspace Display Area.' In contrast to Views, the docking panes can either be floated outside of the CoPilot application (i.e. on another monitor) or docked to one side of the CoPilot application. See *Docking Panes* on page *43* for information about docking panes usage and for specifics about the various available docking panes.

Creating new displays is an important part of managing a useful project. Understanding the various purposes and uses for each of the available Display windows can be facilitated by reading their documentation in this manual. The table in the following subsection lists all available Display windows, as well as where their documentation can be accessed in this manual.

CoPilot 5 hosts two toolbars specifically for the creation of new Display windows, known as the **CoPilot Professional Views Toolbar** and the **CoPilot Standard Views Toolbar**. These toolbars can be shown and hidden via the toolbar context menu (see *Menus & Toolbars* on page *47*). In CoPilot 6, these new displays are created via the **Create** group on the **Home** tab of the ribbon.

### Summary of CoPilot Standard Display Windows

The table below summarizes the characteristics of the view windows available in the CoPilot Standard Version. Each view is named and described. Additional information includes whether the view can be saved to file and the section that



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

describes the use of that view. For a table of the displays only available in CoPilot Professional, see the Professional Displays and Controls section.

| Standard Display Views | | | | |
|---|---|---|---|---|
| **View Name** | **Description** | **Type** | **File** | **See Section…** |
| Data Generator | Generate a customized stream of changing data values | Standard | Yes | Generate and Edit Data |
| Monitor View | View, save, filter, search, format, export, etc. recorded messages | Standard | Yes | Sequential Monitoring |
| ARINC 429 View | View is now converted to an Engineering View | Standard | Yes | ARINC 429 Protocol |
| MIL-STD-1553 View | View is now converted to an Engineering View | Standard | Yes | MIL-STD-1553 Protocol |
| Engineering View | View ports/fields in columnar format in engineering units | Standard | Yes | ARINC 429, ARINC 664 / AFDX, and MIL-STD-1553 Protocols |
| BC Schedule View | View and define a BC schedule with subframes, messages, gaps, retries, etc. | Standard | No† | MIL-STD-1553 Protocol |
| Error Injection View | View, define, and control 1553 error injection | Standard | No | MIL-STD-1553 Protocol |
| Message/SA View | View and edit 1553 messages and subaddresses | Standard | No | MIL-STD-1553 Protocol |
| Port View | View and Edit AFDX Ports, functional data sets, data sets, and fields | Standard | No | ARINC 664 / AFDX Protocol |
| VL View | View statistics and MIB information for AFDX Virtual Links (VLs) | Standard | No | ARINC 664 / AFDX Protocol |
| Network View | View cumulative statistics MIB information for AFDX Networks | Standard | No | ARINC 664 / AFDX Protocol |
| Radar View | View, filter, search, edit, export, etc. recorded messages | Standard | No | ARINC 708 Protocol |
| 429 Schedule View | View a channel schedule | Standard | No | ARINC 429 Protocol |

† The display view can create export files which can be modified and imported.

### Managing Display Windows

The display and placement of view windows can be controlled through buttons in the view's title bar, through the Project Explorer tree, and through the Window menu. Views that can be saved to file can be saved, opened, and named from the File menu.

- **Title bar buttons**—Use the button in the upper-right corner of the view to minimize, maximize, or close the window.

---

Note: Closing the window hides it from view but the display is still active and serviced during the simulation. To prevent a view from running, make it inactive



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

or delete it through the Project Explorer. Also note that fileless views are deleted when they are closed.

- **Project Explorer commands**—Use the view's context menu in the Project Explorer to set the active/inactive state, show (or restore) the view in the display area, move the view to a different workspace, delete it from the project, or access summary information (see *Project File Management* for more information). Some views may have special context menu commands (such as the Python Script Editor).
- **Window menu**—Use the Window menu to select or arrange open views. For complete information, see page *47*.
- **File menu**—(version 5) Use the File menu to save, open, close, and name views. In version 6, use the **Backstage** tab.
- **Workspace Tabs**—Views can be moved between various workspaces, which serve to logically organize views into groups that can be viewed separately. See *Workspaces* on page *43*.

### Other Display Window Concepts

- **Default Names**—All view components are given a default name by CoPilot when they are created.
- **Saving**—Most views can be saved to file and named by the user through the **File | Save As** command.
- **Fileless Views**—Some views do not save to file. These views can be recreated from their object in the Hardware Explorer tree.
- **Third Party Components**—In addition to the CoPilot views, you can open selected applications and third-party component windows within CoPilot. See the following section for more details.
- **Menu Merging**—(version 5) Some views have unique menus which are merged with the standard CoPilot menus when selected. These menus offer special commands unique to that particular view. Please see the view's documentation for details.

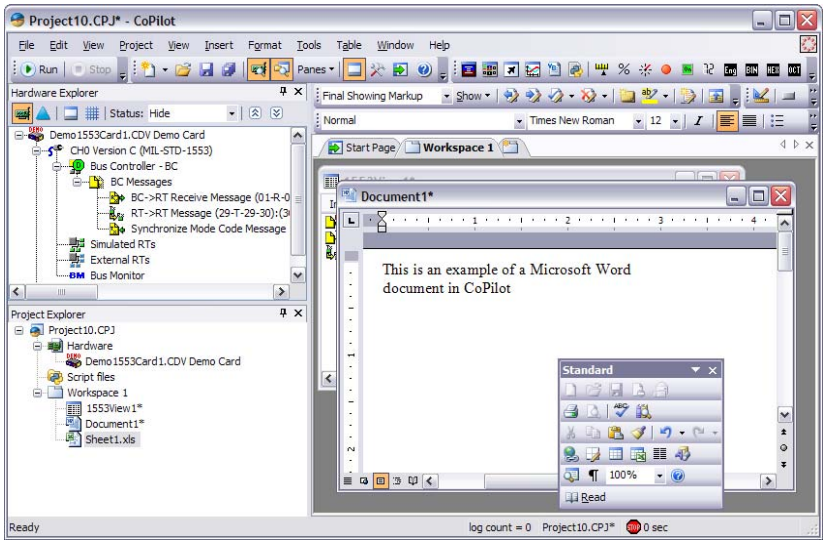
# OLE Documents

### Hosting External Document Applications

In addition to CoPilot view components described in the previous section, you can open OLE documents within CoPilot, such as a Word® document or an Excel® worksheet (see figure).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*Word Document (shown here using CoPilot version 5)*

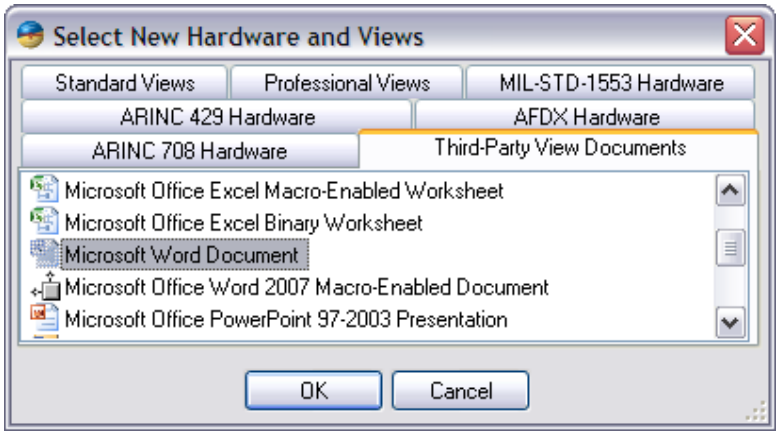
### Benefits of Third-Party File Hosting

CoPilot incorporates the resources of other programs by hosting them in its workspace, providing the user new flexibility and convenience. Applications for third-party file hosting include:

- Recording the instructions for a test procedure right in the actual CoPilot project
- Comparing a spreadsheet of data to changing data values in CoPilot
- Keeping notes or other records within individual CoPilot projects
- Using scripting to output results to a file (such as Word or Excel), and then view and save that file in CoPilot

### Adding External Documents

CoPilot can host third-party applications with built-in "OLE Documents" capability. CoPilot automatically detects the OLE Documents—compliant applications on the host computer—and lists them in the Add New Hardware and Views dialog (see figure below).






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Creating OLE documents

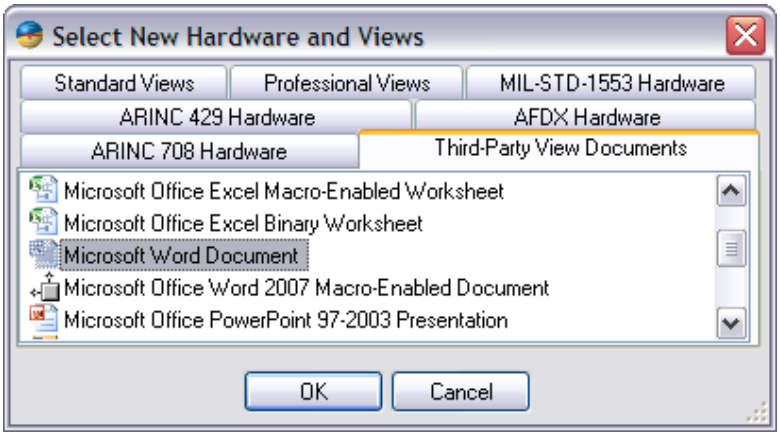
CoPilot can host OLE Documents–compliant application files as components in a CoPilot project. These OLE documents can be created, saved, and manipulated just like other CoPilot components. These applications are selected through the New Hardware or Views dialog.

To create an OLE Document:

1. Select the **File | New | New Hardware** or View command from the File menu to open the New Hardware and View dialog
2. In the Third-Party View Documents tab, select from the list of available file types (see figure)

Note: CoPilot automatically detects the available file types on the host computer and lists them in this tab.

3. Click **OK** to open the selected component in CoPilot




The file will open in the CoPilot display area and it will be listed in the Project Explorer. A temporary file name assigned by CoPilot will appear in its title bar, but you can also save and reopen it with a unique file name.

## Saving OLE Documents

CoPilot manages the saving, closing, etc. of the third-party files that it hosts. Therefore, commands such as Save and Open are not available in the menus or tool bars of the OLE document.

To save an OLE Document:

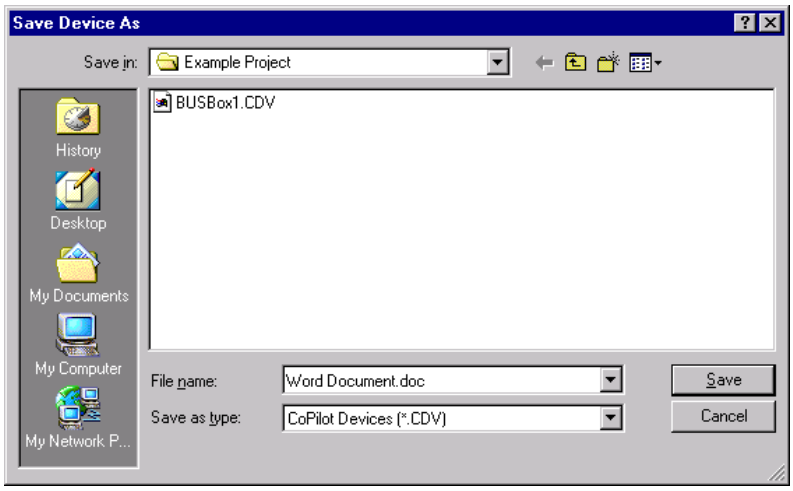
1. Click on the component icon in the Project Explorer to select it and choose the **Save As…** command in the CoPilot File menu
2. The name of the component will be supplied by this menu. For example, the command might read: “Save Document1 As.”
3. In the Save Device As dialog, enter the file name and extension in the “File name:” field (see figure below). Be sure to use the correct extension for that file type.

Note: You cannot choose the file type from the “Save as type:” listbox; you must enter it manually as part of the file name.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4. (Optional) Browse to the location where you wish to save the file if you want to save the application file outside the project folder.
5. Click **Save** to save the component and close the dialog




The new file name and extension will appear in the Project Explorer and in the title bar of the component.

Note: For a newly created OLE document, you cannot use the **File | Save** command because CoPilot does not know what file extension to use. After you have saved the component initially with a specific extension, you can then use the Save command for subsequent saves.

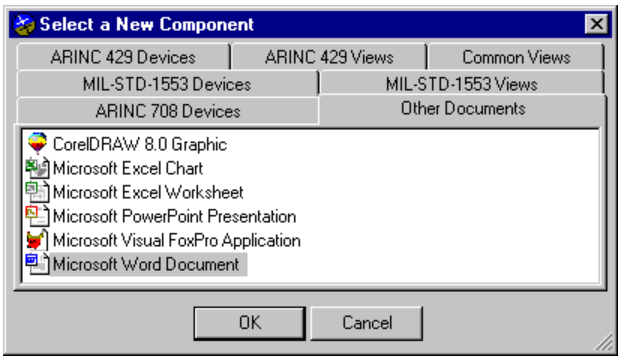
## Opening OLE Documents

Whether the file was created in CoPilot or in the source application, CoPilot can browse to and open a third-party file in the CoPilot workspace.

To open a third-party file, the application must be installed on the host computer and must be designed to be hosted (known as “OLE Documents” technology).

To identify available file types:

1. Select File | New | New Hardware or View to open the New Hardware or View dialog
2. Click the Third-Party View Documents tab to display a list of the file types that can be hosted in CoPilot (see figure below)

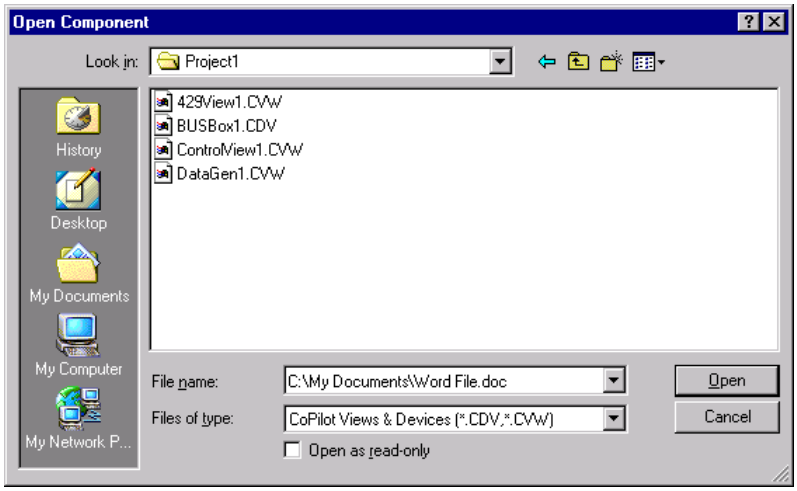





Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To open an OLE Document:

1. Select File | Open | Open Hardware or View Files to access the Open dialog
2. Enter the exact path and file name of the file you wish to open (see example in figure below)




*You cannot browse to a third-party file through this dialog because it filters out all file types except for CoPilot .CDV or .CVW files.*

3. Click Open to load the selected file in CoPilot

The file will open in the CoPilot display area and it will be listed in the Project Explorer.

## Working with OLE Documents

Working with an OLE document in CoPilot is very similar to working with that same file in its source application. Most of the menus and commands for an OLE document file are also available in CoPilot. However, because CoPilot controls window placement and file functions (such as Save and Open) for the components it hosts, the equivalent commands in the hosted file are unavailable.

### Controlling an OLE Document with CoPilot

CoPilot manages file functions for hosted documents and controls the display of third-party windows.

- **File functions**—CoPilot handles file management functions such as New, Open, and Save through the CoPilot File menu. The equivalent commands in the hosted file are disabled (for example, the Save button in a Word tool bar is grayed out).
- **Window menu**—You can use the standard CoPilot menu commands such as tile and cascade to position third-party windows.
- **Title bar buttons**—The minimize and maximize/restore buttons function just as they do for other windows.

  *Warning: The X (close) button deletes the file from the project; any unsaved changes will be lost. In previous versions of CoPilot, the close button was used to hide the window.*

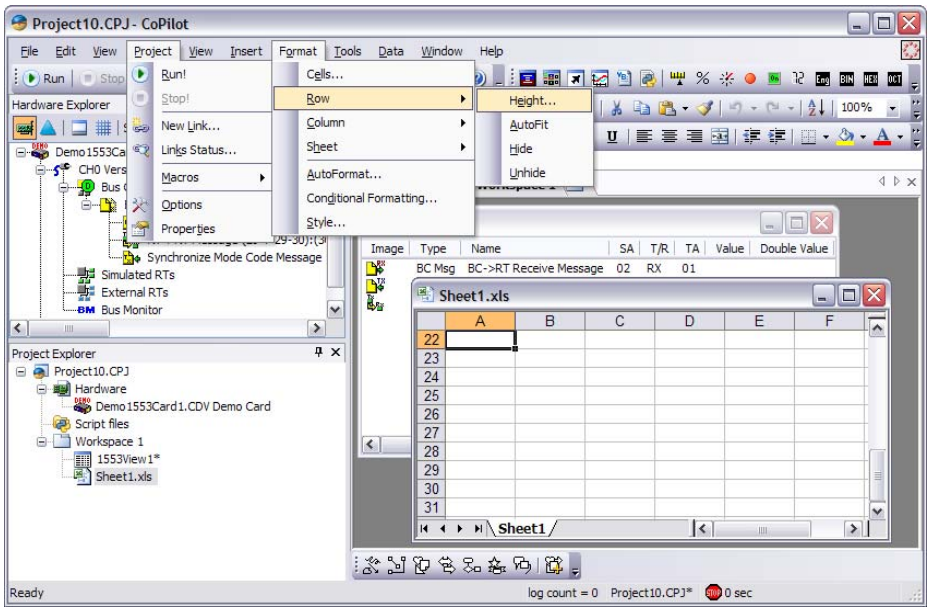


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Project Explorer**—The Show, Delete, and Summary menu commands all function the same for third-party files as for CoPilot views.
- **Workspaces**—Third-party windows can be hosted in different project workspaces just like normal CoPilot views.

### Native Functionality

When a file is hosted in CoPilot, its menus and tool bars merge with the CoPilot workspace. Occasionally, there will be duplicate menus (for example, two Help menus). With the exception of the functions that CoPilot manages (described in the subsection above), all of the native functionality of the source application is available. In the figure below, you can see a native menu such as Excel's Format menu in the same menu bar with CoPilot's Project menu. Notice also the Excel tool bars that are added to the workspace below the CoPilot tool bar.




If you create or modify a third-party file in CoPilot, you can open it later in its own program with the same results as if you had created and saved it with the source application originally. CoPilot does not leave "fingerprints" on any file it hosts.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# MIL-STD-1553 Protocol

## MIL-STD-1553 Overview

This sections covers: installing hardware, configuration of the MIL-STD-1553 databus (Card, BC, RTs, SAs, BM, and fields with engineering units), data logging to the sequential monitor, 1553 displays, and error injection. Refer to *Hardware Installation and Configuration* on page 62 for additional information.

Note: the default CoPilot initialization settings are based on the 1553B, Notice 2 specification and allow the user to simulate all aspects of MIL-STD-1553 behavior. These default MIL-STD-1553 settings can be changed with CoPilot.

#### MIL-STD-1553 Tasks

In CoPilot 5, the MIL-STD-1553 Analyzer Toolbar provides quick links to performing the most common 1553 tasks - from configuring Bus Controller schedules to analyzing the bus. Additionally, the task list for objects selected in the Hardware Explorer shows a list of common tasks for the object(s) selected. Many of these operations are available in CoPilot 6 via the **1553** ribbon tab.

#### Hardware Level Capability

Ballard Technology MIL-STD-1553 cards can be ordered with varying levels of MIL-STD-1553 bus capabilities. These levels have been referred to as models in some of our previous hardware products. As an example, the PCMCIA board could be ordered as a CM1553-3B4 or a CM1553-3C to correspond to level B4 or level C functionality. These levels are summarized in *Appendix C: Supported Hardware and Protocols* on page *441*.

The configuration of the hardware level is done to configure cards in CoPilot when the hardware is not currently installed, or to change the configuration for use with another hardware device of a different level of functionality (but still the same hardware type). Additionally, hardware may be configured to a lower level than it actually possesses to operate with restricted functionality. Attempting to run hardware configured to a level higher than the actual capability of the device will not work. The level of 1553 functionality (or Version) is configured independently for each 1553 channel. When using detected hardware, this setting does not require any changes.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# 1553 Databus Configuration

## Overview

The configuration of the MIL-STD-1553 databus is done through the use of the Hardware Explorer tree. The Hardware Explorer tree contains the object hierarchy ranging from the 1553 card device down to fields. Objects are configured by modifying an object's properties. Object properties are accessed from property pages, context menus, Hardware Explorer tasks, and with script.

This part of the manual describes the MIL-STD-1553 specific configuration differences. The 'Hardware Configuration' section on page *64* describes common hardware configuration settings. Included in this common section is a description of the use of the enable setting to enable/disable an object.

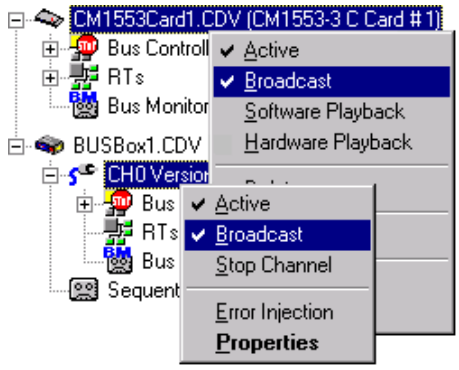
## Hardware Explorer Object Hierarchy

> **Note**: *CoPilot 6 introduced the Ribbon Interface and simplified the the Protocol Browser with the new Activity View. The Hardware Explorer tree references remain for CoPilot 5 compatibility. See New in CoPilot 6.*

MIL-STD-1553 hardware is either single-bus or multi-bus capable. For Ballard single-bus 1553 hardware, the BC, RT, and Bus Monitor branches directly under the board icon in the Hardware Explorer (as shown with the CM1553-3 card in the figure below). Multi-bus capable Ballard hardware, such as the BUSBox® can host multiple buses and protocols. The 1553 channels are organized by channel numbers.

Many of Ballard Technology's products, including the OmniBus® family of boards, support several avionics protocols and multiple cores. The flexible, multi-core architecture of OmniBus hardware allows more protocols on a single board for increased ease of use and reduced cost. Each core on the OmniBus board is represented in the CoPilot Hardware Explorer tree by a separate branch, and branches are defined by one or more avionics protocols and channels. Channels are then grouped together below each core.

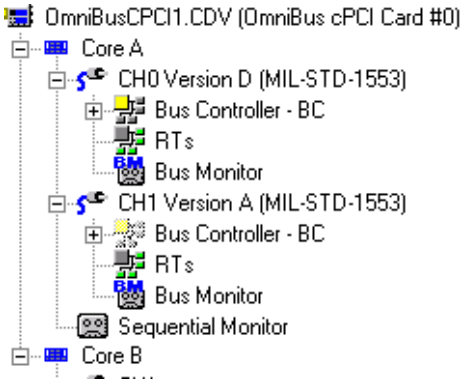
The figure below shows the hierarchy of a single-channel CM1553-3, a single-core multi-channel BUSBox, and a multi-core OmniBusBox.




*Example CM1553-3 and BUSBox devices*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

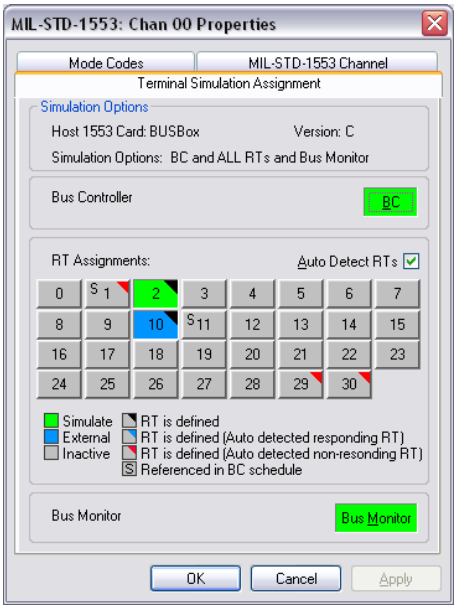



*Example multi-core OmniBus device*

- *CM1553-3, PC1553-3, and LP/LC1553-3* hardware configures the 1553 bus options from the 'card' level context menu and properties.
- BUSBox, OmniBus, Avionics BusBox, USB and other multiple bus capable hardware configure the 1553 bus options from the 'channel' level context menu and properties.

## Terminal Simulation Assignment (BC, RT, and Monitor Simulation)

The TSA (Terminal Simulation Assignment) tab allows the user to activate the BC, RTs, and the Bus Monitor. The Simulation Options frame displays the available resources. For example, a 1553 level B4 board could simulate a BC and four Remote Terminals. Users cannot define a more complex simulation through the TSA window than their 1553 board can support.




*MIL-STD-1553 channel terminal simulation assignments*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To access the TSA window:

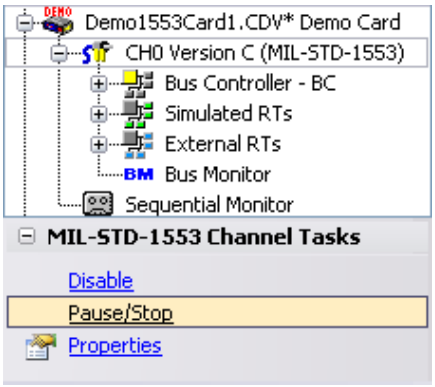
1. Right click on a 1553 channel (or the card for 1553 devices without channels) in the Hardware Explorer tree and select Properties
2. Click the **TSA** tab, specify terminals as desired, and click **OK**

Click any of the buttons to activate the bus controller, monitor, or remote terminals. The button will appear green and remain depressed to indicate that the object it represents is activated. Click OK to accept this configuration and close the window. Selected RTs will appear in the Hardware Explorer. To apply the current choices without closing the TSA window, click the Apply button.

For all MIL-STD-1553 hardware, this tab is also available in the RTs Properties window (accessible from the RTs context menu in the Hardware Explorer). Additional information about terminal simulation assignment can be found in the following *RT Terminal Simulation Assignment* section.

## Pausing Databus Activity

The BC, channels, RTs and other items can be paused to temporarily stop databus activity for that object. A 1553 databus (channel) may be paused and resumed while the simulation is running. To pause the channel, select the channel then click the pause/stop command in the task list. The channel context menu and scripting can also be used to pause/unpause the channel. A pause symbol will appear on the channel icon (see figure). The channel will remain in a paused state until this option is cleared (even if the simulation is stopped and restarted). Similarly, pausing the BC stops the active schedule from sending BC messages and pausing an RT is like virtually disconnecting the RT from the bus.



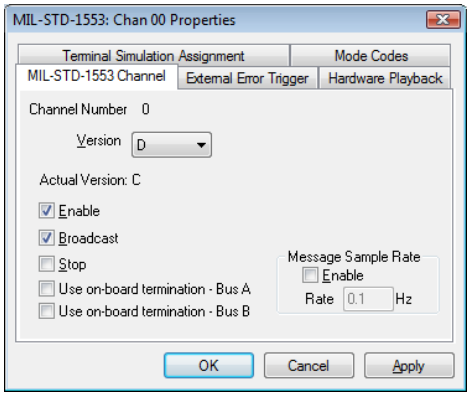

*MIL-STD-1553 channel tasks: pause command*

## MIL-STD-1553 Channel Configuration

A MIL-STD-1553 Channel can be configured from the property page. The available properties, such as on-board termination, are hardware dependent. Only the settings applicable the hardware being configured will be shown.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




### Stop/Pause Channel

When the channel's stop/pause state is set, the channel is placed into a pause state and interaction with the databus is suspended.

### Sample Rate

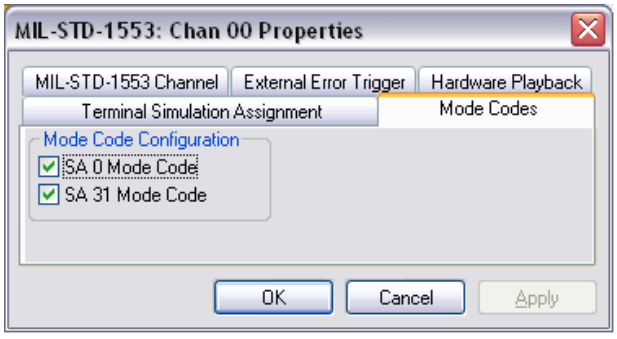
When enabled, the Message Sample Rate is limited to the specified rate (in Hz). This setting is useful when the project contains a large amount of Messages. By limiting the Hardware Explorer object update rate you are freeing the system up to do other work. Sequential Logging functions are not affected by this setting.

## Advanced Features

This section describes the MIL-STD-1553 specific advanced hardware features. Other common hardware features, such as IRIG timing, synchronization output, trigger input, and parametric transmit amplitude modulation, are described in the Advanced Hardware Settings on page *66*. MIL-STD-1553 error injection is described in detail in a later section on page *208*.

### *Mode Codes Configuration*

The Mode Code settings in CoPilot 1553 are based on definitions established within MIL-STD-1553, Notice 2.




*Example of the Mode Code configuration options*

Mode codes legalized within this menu apply to all RTs simulated on the 1553 databus by CoPilot unless a particular RT is configured to override this setting. Broadcast mode codes are globally enabled through a separate option described in the following section.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To change the mode code settings:

1. Right click on the 1553 channel (or the card for hardware devices without channels) in the Hardware Explorer tree and select **Properties**
2. In the **Mode Codes** tab, click to clear or select subaddress 0 and 31 for modes codes
3. Click **OK** to save your settings and close the dialog

## Broadcast Mode Configuration

Broadcast mode is enabled by default and uses RT address 31 for broadcast messages. The broadcast terminal will not respond to broadcast commands when broadcast is configured. Broadcast can be disabled or enabled individually for each 1553 channel. If broadcast is disabled, then the RT 31 terminal can act as a normal RT and respond to messages or inject errors by responding to broadcast messages. Received broadcast messages appear beneath the Broadcast RT in the Hardware Explorer tree as shown in the figure below.




To clear the broadcast setting and use RT31 as a standard terminal:

1. Right click on the 1553 channel (or the card for hardware devices without channels) and choose **Properties** from the context menu (see figure)
2. Click on the **Card** tab (or Channel tab) and clear the **Broadcast** checkmark
3. Press the **OK** button to accept the changes and close the dialog

Broadcast mode will be disabled and RT 31 will be configured as a disabled remote terminal. To simulate RT31, it must then be enabled.

## Selectable Bus Termination

MIL-STD-1553 buses on many products, including OmniBus and Avionics BusBox hardware, can be configured via CoPilot to use on-board bus termination. This feature is intended for use a direct-coupled connection to the bus and switches a 75-ohm termination resistor across the direct-coupled terminals. Refer to the hardware manual for additional information about direct and transformer coupling options.

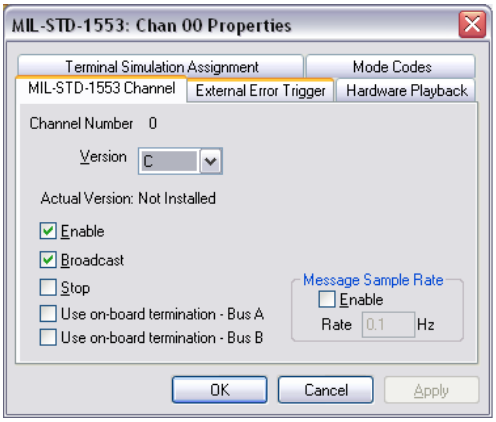
**Warning!** On-board bus termination should *not* be selected when transformer-coupled to the bus. Transformer coupling required external couplers and terminators for proper bus termination. Consult your hardware manual to see if your device supports this feature.

To configure a 1553 channel with internal termination:

1. Right click on the 1553 channel icon in the Hardware Explorer tree and choose Properties
2. In the Channel Properties window, select the MIL-STD-1553 Channel tab (see figure below)
3. Click the checkbox for the Use on-board termination option
4. Click OK to close the dialog and apply the configuration



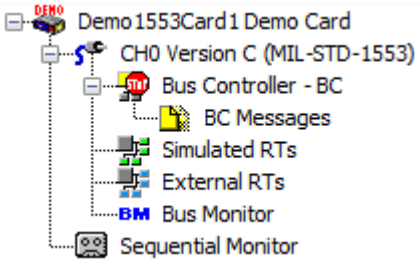
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




# Bus Controller Operation

## Bus Controller Overview

The Bus Controller (or BC for short) directs the information flow on a MIL-STD-1553 databus. The content and pace of messages transmitted on the 1553 bus is determined by the Bus Controller schedule.




The Bus Controller message list holds a reference to created BC messages. The schedule references BC messages and contains timing information.

### Bus Scheduling

A BC schedule is defined by a sequence of commands that direct Remote Terminals to transmit or receive information. In simplest form, the Bus Controller Schedule is a collection of messages to be transmitted by the Bus Controller within a specified period.

CoPilot provides various scheduling modes. The Default scheduling mode builds a schedule containing all defined messages in a single frame specified by the BC's default frame time setting. The Rate-Based scheduling mode transmits messages using an assigned transmit rate (in Hertz or times per second). The Bus Control will build a schedule using these rates. The custom user scheduling allows users to create a sequence of subframes containing messages, gaps and branches. Multiple user schedules can be created; however, only one may be active at a time. See *Controlling the BC with Schedules* on page *118* for additional information.

### Bus Controller Messages

The commands used in the BC schedules are referred to as BC Messages. In addition to using messages in schedules, they can also be used asynchronous

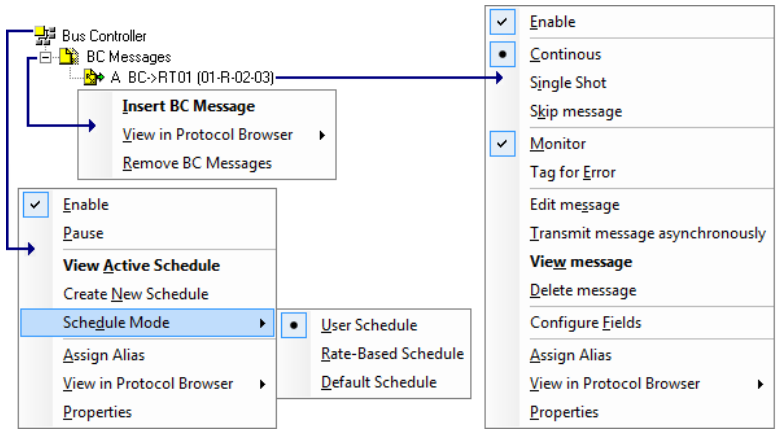


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

transmission and data interpretation. Transmit messages instruct an RT to transmit data to the BC while receive messages send data to an RT. Fields can be defined on messages to interpret data into engineering units.

### Context Menu, Data, and Other Properties

Messages appear in the Message List in the Hardware Explorer tree as they are defined. As with all other objects in the tree, they are defined by properties. Right clicking the mouse on any object in the tree brings up a Context Menu and Property list.

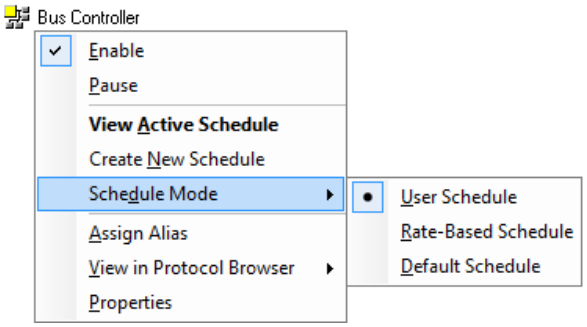



Data that originates in the BC are defined based on conventions selected in the Project | Options window or they may be uniquely defined through the Data Editor.

## *BC Configuration*

### BC Context Menu

To configure the Bus Controller, right click on the BC icon in the Hardware Explorer tree to open the BC context menu.




- **Enable** toggles the enabled/disable status of the BC
- **Pause/Resume** pauses or resumes the BC schedule when the simulation is running
- **View Active Schedule** opens the BC schedule editor window for the current active user schedule
- **Create New Schedule** adds a new custom user schedule to the BC
- **Schedule Mode** opens a submenu of schedule commands (see *Controlling the BC with Schedules* on page *118*)



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- User Schedule informs the BC to load and run the active custom user schedule
- Rate-Based Schedule informs the BC to load and run a schedule based on transmit rates set at each BC message
- Default Schedule informs the BC to load and run a single frame containing all active BC messages

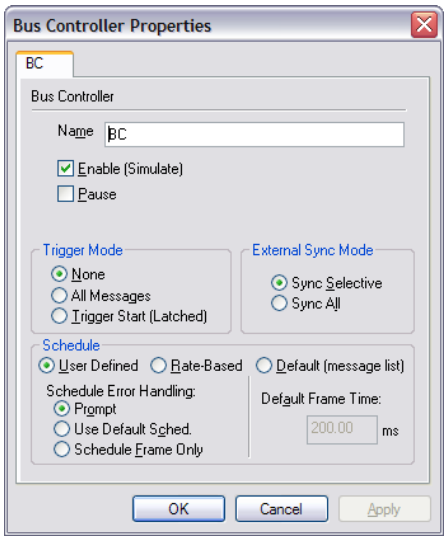
- **Assign Alias** associates a globally unique string name with this object for use in scripting
- **View in Protocol Browser** in version 5 open one of the Protocol Browser panes and zooms into this protocol level (or as close as possible)
- **Properties** opens the BC Properties dialog box

The BC pause property may be set before or after a simulation is initiated with the Run button. The pause halts the BC simulation but does not affect the other configured terminals.

### BC Properties

The BC properties window allows the user to configure the BC directly. By default, the BC is enabled and unnamed. To name the BC, type in the Name box. Pause will pause the BC from sending commands and resume the schedule from where it left off when unpaused.

Note: Pausing a BC for demonstration hardware (demo card) has no effect.




### Triggering with External Triggers

If your 1553 board is capable of external triggering, CoPilot can support external trigger input. When the All Messages option is selected, each message in the schedule will wait to be transmitted until an external signal pulses. The external trigger steps through the messages in the schedule. When Trigger Start is selected, the BC schedule will wait for a single external signal before beginning the schedule for normal transmission.

The Trigger Input Line Selection must be set on the External Trigger tab for devices with more than one possible trigger line if triggering is enabled.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### Output Synchronization with External Syncs

When Sync Selective is chosen (the default), only those messages in the BC schedule explicitly tagged for synchronization will trigger the external sync pulse. To tag a message in the BC schedule for sync, right click on the message and choose Properties from the context menu. Click the check box labeled "Sync (Output)."

When Sync All is chosen in the BC properties page, an external pulse is active for the duration of each message in the schedule.

The Sync Output Line Selection must be set on the External Sync tab for devices with more than one possible sync line if sync is enabled.

#### Schedule Options

BC schedules can be either custom user defined schedules, rate-based schedules or the default schedule. The default schedule is automatically created based on the contents of the active BC messages all scheduled in a single frame set to the default frame time. For a complete discussion of BC Schedules, see the next section: "BC Schedule." Examples of some of the advanced schedule concepts are found in *Appendix I: Custom 1553 Schedule* on page *501*.

### *Controlling the BC with Schedules*

BC Schedule controls Bus Controller operations. The run state of a BC schedule can be changed while running. The BC can be paused and resumed via user control.

The BC schedule is defined by a sequence of commands that direct Remote Terminals to transmit or receive information. Scheduled messages are grouped into one or more subframes and each subframe is processed in a time interval known as subframe time. Subframes may also include gaps, branches, and other commands.

If custom user defined BC Schedule is undefined or there are no active messages to transmit, a Stop symbol will appear on the BC  icon in the Hardware Explorer. To define a schedule for the BC, you can choose to run a custom schedule, create a Rate-Based schedule or create a default schedule. The following section describes BC Schedules in more details.

## BC Schedule

### *BC Schedule Overview*

The BC Schedule controls Bus Controller operations. The BC schedule is defined by a sequence of commands that direct Remote Terminals to transmit or receive information. To define a schedule for the BC, you can choose to run a custom schedule, a rate-based schedule or a default schedule.

The default schedule mode transmits all active messages using the Default Frame Time value set in the BC properties. The frame time is automatically extended by the hardware as the frame is transmitting if frames are not large enough to accommodate the messages scheduled within the frame.

Custom schedules group messages into one or more subframes and each subframe is processed in a time interval known as a frame-time. Subframes may also include gaps and branches. If the BC is set to user schedule mode and no custom user schedule is defined, a Stop symbol will appear on the BC  icon



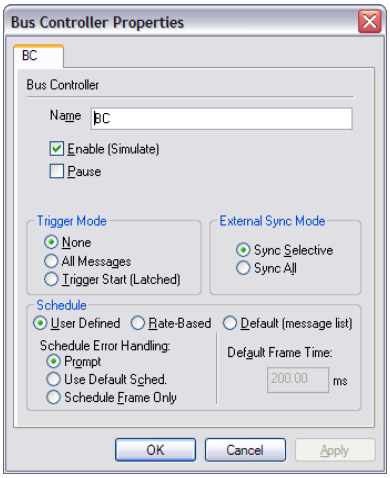
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

in the Hardware Explorer. Multiple user schedule may be created to allow changing between different schedules (while the simulation is not running).

When a BC message has the skip option set, every instance of that BC messages in the schedule is skipped, regardless of the BC Schedule mode. Likewise, if a message is set to single-shot, the message sets the skip option after the message is transmitted. See *Appendix I: Custom 1553 Schedule Examples* on page *501* for specific examples of BC schedules.

Note: CoPilot automatically adds an additional 20 µs (.02 milliseconds) frame time to the end of schedules to allow asynchronous transmissions. This 20 µs value may be subtracted from the time of the last frame in the schedule to preserve exact timing (i.e. a 200 ms frame could be changed to 199.98 ms).

## BC Properties for Scheduling




### Selecting the Schedule Mode of the BC

Bus Controllers can be configured to run the default schedule, rate-based schedule or a custom user schedule. When the default schedule is selected, CoPilot automatically builds and executes a BC schedule based on the active BC messages. A default schedule consists of one instance of each message in the BC message list in a single frame using the default frame time. The schedule mode can also be set from a BC Schedule View.

In this mode, a green “default” symbol on the BC icon gives visual notification that the BC is set to run a default schedule.

Note: If you run a default schedule without any active messages in the message List, then a single empty frame will be scheduled.

When the rate-based BC schedule is selected, CoPilot automatically builds and executes a BC schedule based on the transmission rates specified for each BC message. In this mode, a green “rate-based” symbol on the BC icon gives visual notification that the BC is set to run a rate-based schedule.



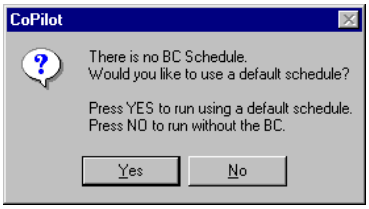
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: Newly created BC messages are initialized using the 'Default BC Message Rate' configured in the MIL-STD-1553 Properties tab of the CoPilot Options. This default value can be changed and it will update messages that have already been created. However, once until any of the properties (even properties other than the transmission rate) on the property page are changed, the rate will no longer be updated by the default rate.

#### Schedule Error Handling

Schedule error handling settings are available when the user defined schedule is selected. An error will occur if the User Schedule is specified but no schedule has been defined (in the BC Schedule Window) or if there is less than one active, scheduled message. The error handling options are located in the BC Properties dialog (see figure at top of this section).

- **Prompt** will open a dialog (shown figure) giving you the option to choose the default schedule or to run without the BC
- **Use Default Schedule** will switch automatically to the default schedule without and user input
- **Schedule Frame Only** will run the BC Schedule as a single empty frame




*BC Schedule error handling prompt*

## *User Defined BC Schedules*

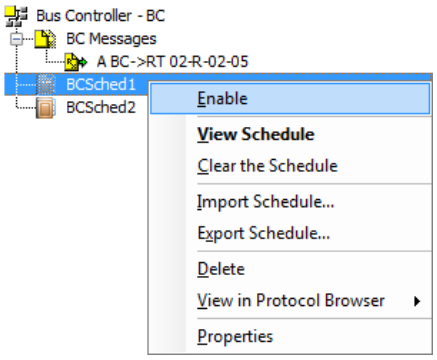
A Bus Controller can have multiple user schedules defined. When the BC schedule mode is set to user defined schedule, the active user's schedule is executed. Although multiple user schedules may be created, only one user schedule can be active at any given time. The "Edit Schedule" command from the 1553 Ribbon (version 6) will edit the active schedule. The user defined schedules are configured and edited using BC Schedule Editor Views.

#### Selecting the User Schedule

Only one schedule may be run at a given time. If multiple schedules are created, only the one enabled schedule will be used when CoPilot is run. The image below shows the context menu to select 'BCSched1' for use by enabling it.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

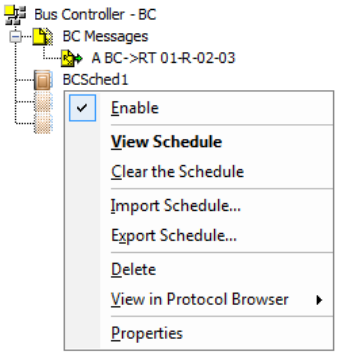



*Context menu to enable a User Schedule*

Note: When the BC schedule mode is set to either the Rate-Based or Default schedule modes, the user schedules are ignored.

## *BC Schedule Import and Export*

Importing and exporting user defined schedules can be used to share/reuse schedules between projects, load programmatically created schedules (created outside the CoPilot environment) and for creating reports. The user defined schedules are saved in XML file format. When a schedule is saved the XML file also includes all BC message definitions.




### Import Schedule (Load)

.To import a schedule select **Import Schedule** from a BC schedule icon in the Hardware Explorer. After selecting an XML file containing a user schedule all BC message definitions in the file will be added to the BC message list and the contents of the user schedule will be replaced by the schedule contained in the file.

### Export Schedule (Save)

To export a schedule, select **Export Schedule** from a BC schedule icon in the Hardware Explorer. Name the schedule and save the resulting XML file on your computer. This file will contain all current BC message definitions along with the contents of the user schedule. The exported file is formatted to provide a printable report of the schedule contents.



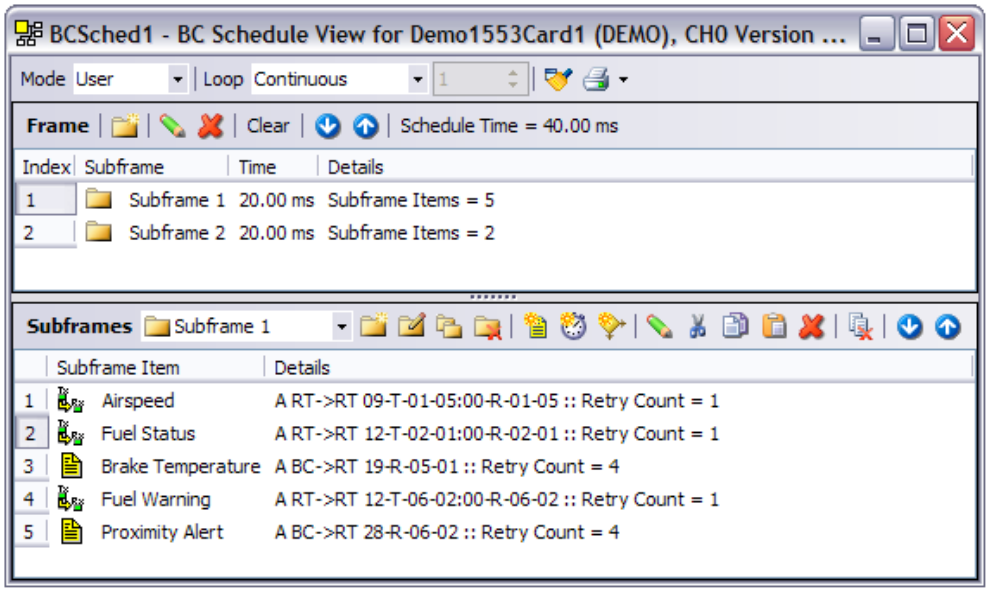
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## BC Schedule Editing User Schedules

### BC Schedule View Overview

The BC Schedule View is accessed through the Bus Controller button in the MIL-STD-1553 toolbar (version 5), **1553** ribbon tab (version 6), or from the BC context menu in the Hardware Explorer. If user schedule is selected and a schedule has not been defined, the BC icon will include a stop image. This section describes the use of the BC Schedule View to create custom user schedules.

User schedules can range from a single message subframe to complex with multiple subframe schedules with differing subframe times. The use of gaps can control the timing between messages in the same subframe (minor frame). The use of branching (both conditional and unconditional) allows for capabilities such as initialization routines that are only executed once, schedule swapping, and conditionally executing different parts of the schedule.




*MIL-STD-1553 Schedule Editor editing a user schedule*

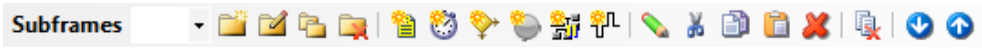
Note: Double-clicking a subframe in the frame list will select that subframe to display the contents of the subframe below. Alternately, select the subframe from the subframe drop list.

### Editing the Schedule

Items in the schedule can be added, edited and deleted through toolbar buttons, context menu commands, drag and drop operations, or keystrokes.




*BC Schedule View frame toolbar*




*BC Schedule View subframe toolbar*

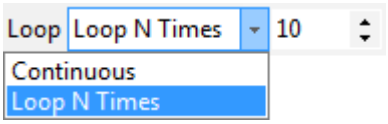


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Placing the mouse over the toolbar buttons will show the tooltip describing the functionality of each of the toolbar buttons.

#### Schedule Loop Settings

Users may specify how many times the schedule should loop. By default, the Continuous option is selected and the Schedule will run indefinitely, continuously looping until halted by an external trigger or by the user pressing the Stop button.




Setting the schedule Loop option and specifying the number of loops will cause the schedule to automatically pause after the specified numbers of loops have been completed.

## Scheduling BC Messages

BC Messages may be added to the schedule by dragging them from the BC Messages list in the Hardware Explorer tree and dropping them in the schedule window. Pressing the new message button will open the "Add BC Message" dialog window. From this dialog you can select a predefined message (from the BC Message list) or press the Create New button to create a new message then add it to the schedule. Messages are always added to a subframe; if a subframe is not already created, one will be automatically created when the first message is added. See the *BC Messages* section on page *128* for additional BC Message information.

## Scheduling Subframes

#### Schedule Subframes Overview

Subframes manage the timing for BC schedules. Often times the terms "schedule", "frame" and "major frame" are used interchangeably and as are the terms "subframe" and "minor frame." In simplest terms, a subframe is a minimum amount of time allocated for executing the messages, gaps, and other schedule objects within the subframe. The schedule may contain many subframes (minor frames) with each assigned a different subframe time. Subframes are defined and edited in the lower portion of the schedule editor window.

#### Schedule Subframe Time

Each call to a subframe in the schedule (frame) has a unique subframe time. The largest subframe time allowed is 1,310 ms and the smallest subframe time is 0.02 ms (200 microseconds). The required time needed to transmit an individual message takes between 70 and 700 microseconds. If the accumulated time for processing the messages in the subframe is longer than the specified subframe time, then the subframe will be extended by finishing the current subframe then immediately beginning processing the next subframe. To edit the subframe time, select the subframe in the upper part of the window and press the edit button.

Each subframe contains a seed time. The seed time defaults to the current BC default frame time. To change the seed time, open the "Subframe Settings" dialog by pressing the edit subframe button. The seed time is used each time you add a call to a subframe in the schedule (frame).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The total time for the schedule (or major frame time.) is the accumulation of all subframe call times and is displayed as the schedule time on the right side of the Frame Edit Toolbar.




*Display of the total scheduled time*

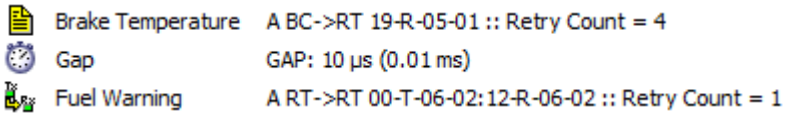
### Subframe Settings

Subframe settings consist of a name, subframe time and a subframe color. The subframe name is shown in the subframe list located on the lower portion of the edit window as well as on each line of a call to a subframe in the upper portion of the window. When you change the name in one place it is also changed in the other. The same goes for the subframe color. The subframe color is an optional setting used to help graphically identify subframe entries in the schedule (or major frame). The subframe time is discussed above.

## *Scheduling Gaps*

### Schedule Gaps Overview

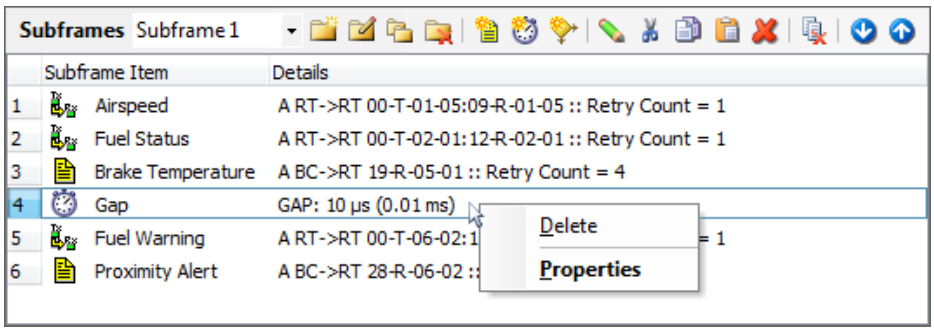
Messages added to a user schedule are transmitted in rapid succession (back-to-back) followed by a gap to finish the frame. Users may space apart messages by placing gaps between messages as the image below shows.




*Example schedule gap placed between two messages*

To add a gap to a subframe, click the add gap button in the subframe edit toolbar (the gap will be inserted below the selected line). Gaps can be added between the sizes of 10 to 10,000 microseconds.

Note: You may create gap values larger than 10,000 microseconds (10ms) by inserting multiple gaps in succession. However, it is suggested that rather than using large gap values, subframes are used instead.



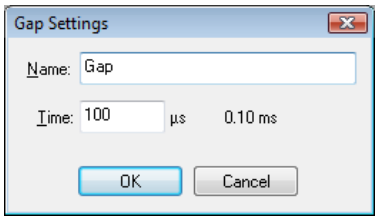

*Schedule gap context menu to access the properties or delete the item*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Editing Schedule Gaps

To modify a gap, right click anywhere on its line in the subframe edit widow open its context menu and choose Properties to open the gap settings window (or click the Edit button). Type the new name or edited gap time as desired.




*Configuring the gap properties*

Note: The Time text box is in microseconds (µs), but the schedule timing is in units of milliseconds (ms). The millisecond conversion is displayed next to the Time text box for your convenience.

## *Scheduling Branches*

### Schedule Branches Overview

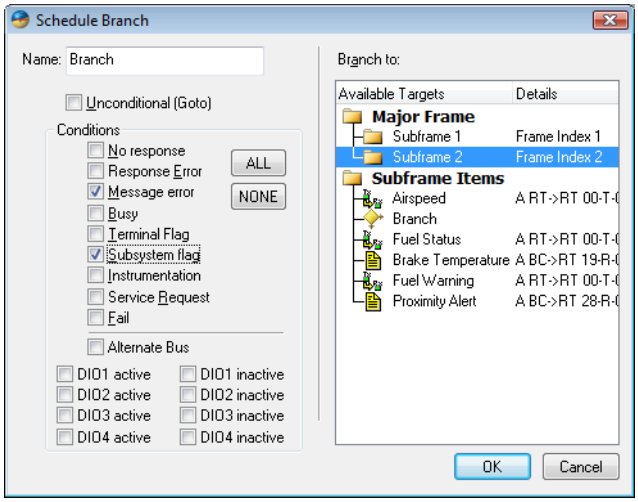
Branches are used to jump to another location in the schedule. Branches can be either conditional or unconditional branches. If you choose a conditional branch, the condition(s) apply to the message immediately prior to the branch item in the BC Schedule. Branches in the schedule allow users to skip sections of the schedule, create initialization sequences, and to create multiple virtual schedules operating in different user-defined schedule 'modes.' Examples of specific uses of schedule branching are found in *Appendix I: Custom 1553 Schedule* on page *501*.

### Schedule Branch Conditions

The unconditional branch (effectively a goto) always jumps to the 'branch to" item as the next schedule item to process. For conditional branches, the response to the previous message is used. The schedule will jump to the branch to item when **any** of the branch conditions are met; conditions are ORed together. The No Response condition is triggered when a status word is not received. The Fail condition is set when all retries for the message failed. The 'branch to' location may consist of either another item in the same subframe or to another subframe call in the schedule. Details of discrete DIO (digital input/output) branching is described later is this section.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

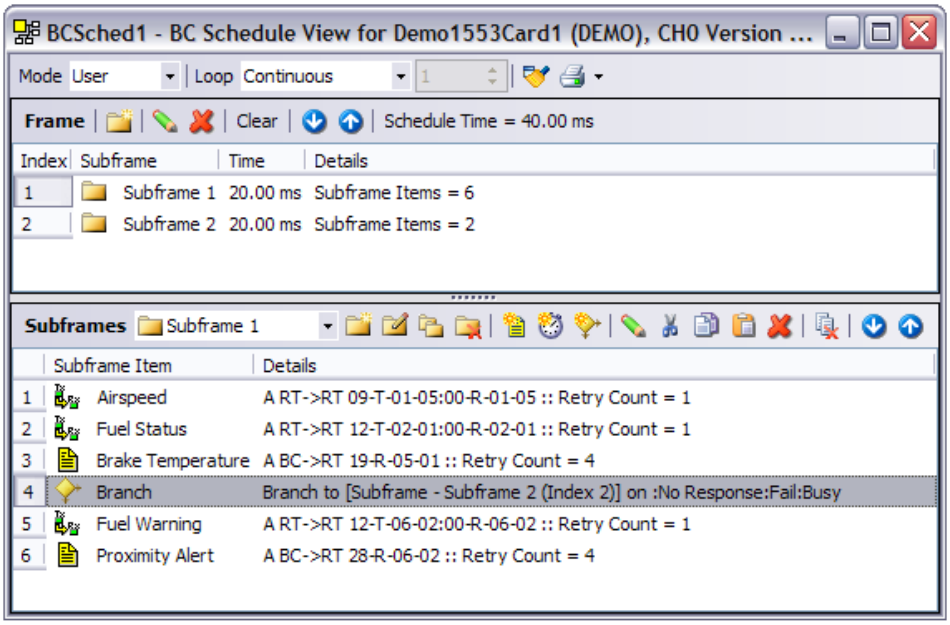



*Adding a branch to a MIL-STD-1553 schedule*

Note: A conditional branch operates based on status the selected DIO lines or on the previously sent message. For clarity in the schedule, place the conditional branch immediately following the message to branch on.

### Schedule Branch to Location

When the branching condition occurs, control is passed to a specified position in the Schedule. In the example below, the branch entry located at subframe index 3 jumps to Subframe 2 (at Frame index 2) if there is a message error for the Brake Temperature message (at subframe index 2). When the branch condition is met, the Fuel Status, Fuel Warning, and the Proximity Alert messages are skipped. If the Brake Temperature message does not have a message error, the branch condition is not met and the branch is not taken; the Fuel Status message at index 4 is sent.






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: If a branch becomes broken due to a deleted or moved item the item image will change to a conflict/error icon. Any entries in the 'major frame' will also show this conflict icon until the branch is fixed.

#### DIO Branch Conditions

Custom MIL-STD-1553 schedules may conditionally branch based on the state of one or more DIO (digital input/output) values. These signals can be used by external hardware, change branch conditions in the schedule, and be used by CoPilot ATE scripts.

**Consult your hardware manual for the specific capabilities of your hardware.** DIO schedule branching is supported by 4G and 5G hardware devices. DIO branching is not supported by the PC1553-3, the Lx1553-3, the CM1553-3 or the BUSBox. The following table lists the supported DIO numbers where the DIO numbers correspond to the *ndionum* parameter used with the ExtDIORd and ExtDIOWr functions on a card. Input DIO can be read using the ExtDIORd method and output DIO can be written using the ExtDIOWr method. DIO that is input only can only be read and DIO that is output only can only be written. The DIO of 5G hardware devices support both ExtDIORd and ExtDIOWr on the same DIO – these signal lines function as BOTH input and output. Additional discrete information can be found in the *Discrete Inputs and Outputs (DIO)* section on page *32*.

| Hardware Supporting 1553 DIO Branching | DIO Number for Branching | Input DIO Numbers | Output DIO Numbers |
|---|---|---|---|
| Box Products (Ethernet, USB, Serial) | | | |
| Avionics BusBox: AB1000 | 1, 2, 3, 4 | 1, 2, 3, 4, 5, 6 | 9, 10, 11, 12 |
| Avionics BusBox: AB2000 | 1, 2, 3, 4 | 1-16 (in/out) | 1-16 (in/out) |
| OmniBusBox | 1, 2, 3 | 1, 2, 3 | 5, 6, 7 |
| USB 1553 | 1, 2, 3 | 1-4 (in/out) | 9-12 (in/out) |
| PCI Products | | | |
| OmniBus PCI | 1, 2, 3 | 1, 2, 3 | 5, 6, 7 |
| LP1553-5 | 1, 2, 3 | 1-16 (in/out) | 1-16 (in/out) |
| PCIe Products | | | |
| LE1553-5 | 1, 2, 3 | 1-16 (in/out) | 1-16 (in/out) |
| cPCI Products | | | |
| OmniBus cPCI | 1, 2, 3 | 1, 2, 3 | 5, 6, 7 |
| VME Products | | | |
| OmniBus VME | 1, 2, 3 | 1, 2, 3 | 5, 6, 7 |
| PMC Products | | | |
| OmniBus PMC | 1, 2, 3 | 1, 2, 3 | 5, 6, 7 |

Note: Only the AB1000 and AB2000 support MIL-STD-1553 conditional branching on DIO4. For hardware that does not support dual-mode in/out (input/output) DIO, the output DIO can be wired to the input DIO to allow both reading and writing the DIO values (i.e. connecting 1 to 5, 2 to 6, and 3 to 7 for the OmniBusBox). See the hardware manual for your specific device for additional information.

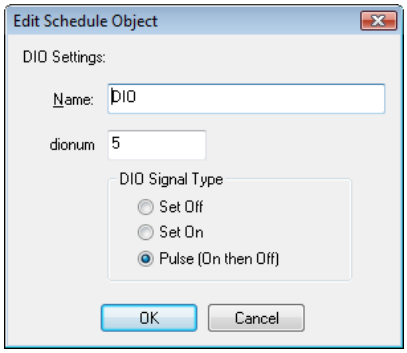
## *Scheduling Output Signals (DIO Pulses)*

Output DIO pulses in a BC schedule will set the selected output discrete value to either 'On', 'Off', or 'Pulse'. The pulse setting internally schedules a pair of



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

schedule opcodes that pulse the specified dionum to the 'On' state followed by the 'Off' state. These signals can be used by external hardware, change branch conditions in the schedule, and be used by CoPilot ATE scripts. Refer to the DIO section on page 32 or the hardware manual of the device for additional DIO information.




*Setting the Properties for a Schedule DIO Pulse*

To add a DIO output signal pulse to a subframe, click the add DIO pulse button in the subframe edit toolbar (the DIO pulse will be inserted below the selected line). Set the DIO number to the desired output dionum (must be valid for the hardware in use). Then set the DIO signal type as desired. Entering an invalid DIO number displays an error message.

### *Scheduling No Operations*

No Operations can be added to a schedule to serve as placeholders. To add a No Operation to a subframe, click the add no operation button in the subframe edit toolbar. There are no properties to configure for no operation items.

### *Scheduling BC Pauses*

Scheduling a BC Pause in the schedule will suspend the operation of the BC when the pause opcode is encountered. To add a BC Pause to a subframe, click the add BC pause button in the subframe edit toolbar (the BC Pause will be inserted below the selected line). There are no properties to configure for no operation items.

## BC Messages

### *Messages Overview*

The BC Schedule is composed of a sequence of commands from the BC that direct Remote Terminals to transmit or receive information. These commands are referred to as Messages. These messages in CoPilot include the command word, associated data words, configuration options, and status information. Messages are created using the Message Editor window. New messages may be created from the BC Message object in the Hardware Explorer or from the BC Schedule View by clicking the new message button.

Note: Messages created in the Schedule View window are also added to the BC Message list in the Hardware Explorer tree as they are created.

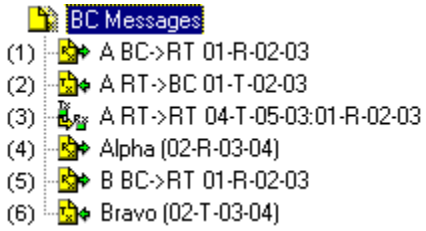


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Messages on the MIL-STD-1553 databus include the command words, status words, and data words. Messages contain up to 32 data words.

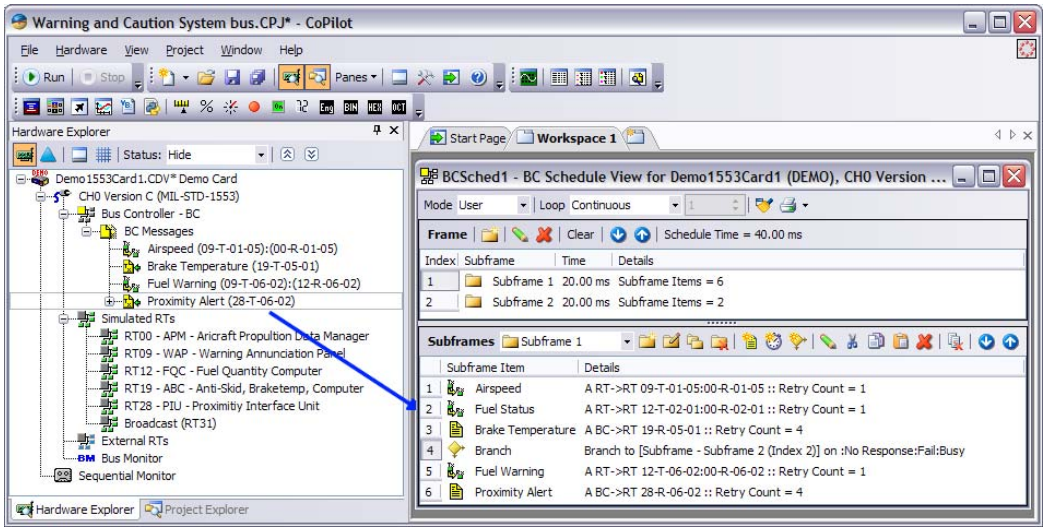
### BC Messages List

All BC messages are listed under the BC Messages icon in alphabetic order. Notice, in the following figure that the command words on lines 1 and 5 are the same, but because the fifth item is configured to transmit on Bus B, it appears later in the list.




### Drag and Drop Messages to the Schedule

The quickest way to place messages in the BC schedule window is to drag and drop them from the BC Message List in the Hardware Explorer tree. If a subframe is not already present in the schedule window, an unnamed subframe of the default subframe time is added automatically when the first message is assigned.




When dragging a message to the subframe portion of the schedule editor window the drop location highlights to indicate that the new item will be added below this entry. If no items in the schedule are highlighted, the new message is added to the bottom of the list.

When dragging a message to the frame portion of the schedule editor window you must drop the message on a highlighted subframe entry. The message is then appended to the end of that subframe list.

### Messages Assigned to Schedules and Displays

Messages listed in the Hardware Explorer tree can be assigned to BC schedules and display pane windows through drag and drop methods. Users may also define engineering units to fields of the message. Fields are simply groups of data bits assigned to an interpreter. Interpreters may define anything from simple



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

decimal or discrete strings, to engineering units such as BNR, BCD and other custom interpretations. Once interpreted, the "engineering unit" fields can be assigned and viewed throughout CoPilot.

#### Message Naming

BC messages can be named explicitly by the user through the Name textbox in the Message Editor. If no name is entered, a name based on the bus, direction, and address is created. For example, "A BC->RT 01-R02-03" corresponds to a receive message on bus A sending three data words from the BC to RT01, SA02.

#### Duplicate Messages in the BC Messages List

CoPilot allows the creation of multiple messages with identical command words (RT, SA, T/R, and word count). Each message on the BC maintains its own data structures and configuration options. However, external devices connected to on the bus see the messages as identical. Some applications when creating duplicate message include changing the number of data words, scheduling a particular instance of a message for error injection, triggering on a specific message location in the schedule (when the message is in the schedule multiple times), and asynchronously sending different data values.

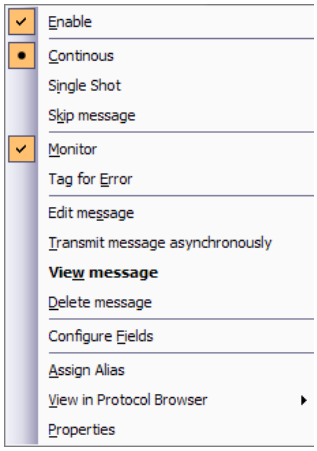
Once a duplicate message is created, make sure it is named properly to avoid future confusion since unnamed duplicate messages will have identical names. Then configuration of each message is done as if they are entirely different messages.

## *Message Configuration*

BC messages properties are configured using the context menu and the message property page. Additionally, scripting, automation, and linking can modify an object's properties and data values.

#### BC Message Context Menu

The context menu for BC messages contains settings and commands to use the message. Right click a message to access its context menu. Many of these choices are also available in the message properties page.




- **Enable** turns a message on (default) or off. Icons for inactive messages are dimmed.
- **Continuous** is the default state for normal transmission
- **Single Shot** causes the message to be transmitted a single time, then set to skip
- **Skip** causes this message to be paused or ignored during transmission (if it is in the BC Schedule)
- **Monitor** defines whether individual messages should be monitored. Global monitoring declaration can also be made through Monitor Properties
- **Tag for Error** links error injection criteria to this message (available on multi-level, level C and level D hardware)

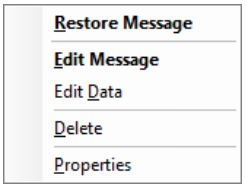


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Edit message** opens the Message Editor
- **Transmit message** asynchronously sends a single transmission of this message, even if the simulation is not currently running
- **View Message** opens the Message View window
- **Delete message** removes the message from the tree
- **Configure Fields** opens the Add & Edit fields dialog, which allows you to define, modify, and organize data fields within this message
- **Properties** opens the message properties window

### BC Message Context Menu from the BC Schedule

Messages in a BC Schedule can be edited and configured from within the schedule. To modify a message from the schedule window, right click on the message to open the schedule message context menu.



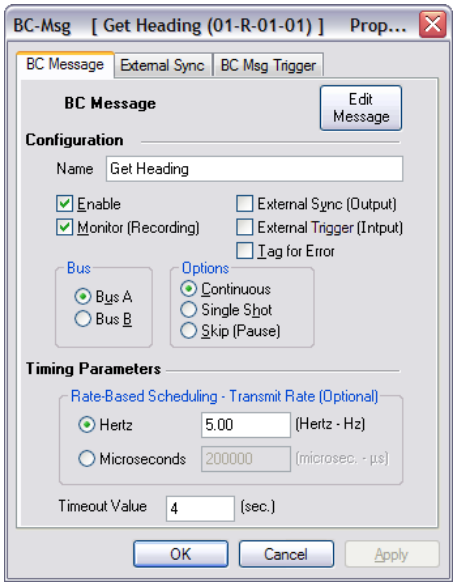

- **Restore Message** restores a previously deleted BC message to the Hardware Explorer tree.
- **Edit Message** opens the Message Editor window
- **Edit Data** opens the Data Editor window
- **Delete** deletes this instance of this message in the schedule. Other instances in the schedule or message list are not affected.
- **Properties** opens the message properties window

### Message Properties

The properties window for BC schedule messages allows the configuration of the message options. Editing the actual message or the message is not done though the message properties



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*BC Message Properties*

- **Name** allows the message name to be modified
- **Edit Message** opens the message editor dialog for this message
- **Enable** toggles the enable/disable status of the message (enable is default)
- **Monitor** tags this message for monitoring (all messages tagged by default)
- **External Sync (Output)** tags this message to trigger a sync pulse for the duration of its transmission. However, the signal will only be pulsed if the Sync Selective option is chosen in the BC properties window. The Sync Output Line Selection must be set on the External Sync tab for devices with more than one possible sync line if sync is enabled.
- **External Trigger (Input)** tags this message to wait for a trigger before transmitting the message. The Trigger Input Line Selection must be set on the Trigger tab for devices with more than one possible trigger line if triggering is enabled.
- **Tag for Error** tags this message as a candidate for error injection
- **Bus** allows the user to choose whether the message is transmitted on the primary bus (Bus A, default) or the secondary bus (Bus B). This option can be changed while the bus is running
- **Options** control the transmit conditions for this message (only applies if the message is scheduled)
    - **Continuous** transmits the message with no restrictions
    - **Single Shot** allows the message to be transmitted once, then it switches to a Skip condition
    - **Skip** tags this message to be paused or ignored during transmission



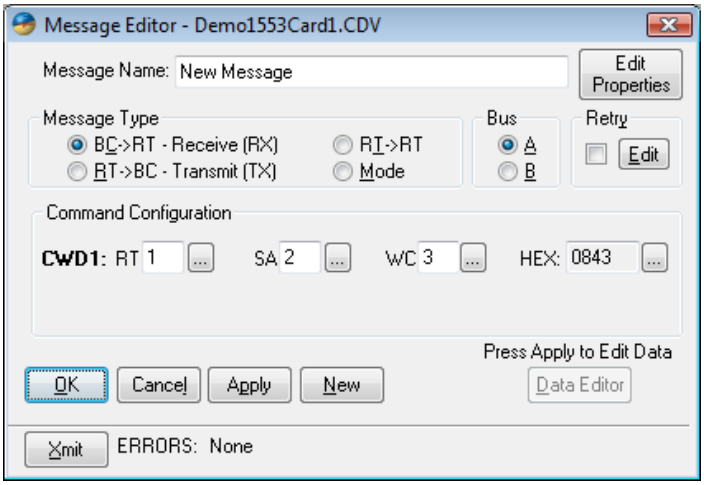
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Rate-Based Scheduling** allows the user to set a transmission rate to be used in Rate-Based scheduling (schedule mode set at the BC)
- **Timeout Value** tests for inactivity (during a run). A timeout condition is reported in the Engineering View window and the Hardware Explorer tree with a red "TO" symbol on the message icon

Note: Changing the options for a message makes the setting change of all instances of that message in a schedule, even if it is referenced in the schedule multiple times. The message options to skip and single-shot a BC message have no effect on demonstration hardware (demo cards).

## *Creating and Editing Messages*

BC commands are defined and modified with the BC Message Editor. Users can access the Message Editor from the BC Schedule or the Hardware Explorer tree.

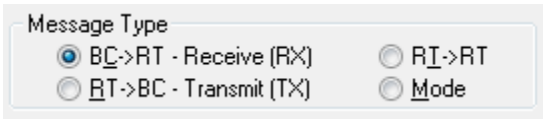



Message properties include message type, address and word count, bus designation, Retry options, and data.

### Message Definition

To create a new message through the Message Editor:

1. (Optional) Type a name in the Message Name textbox
2. Click one of the option buttons in the Message Type frame. The Message Editor window options change to reflect the selected message type.




3. Type in the RT, SA, and WC fields. Alternatively, you may use the pop-up decimal keypad located next to each cell or enter a hexadecimal value.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




4. (Optional) Click the A or B option buttons to select the alternate Bus and/or click on the Retry check box or Edit button to change the retry settings
5. (Optional). Click the Data Editor button to bring up the Data Editor
6. Click OK to close the BC Message Editor or the New button to define another message. (Clicking the New button allows you to construct additional messages without closing and reopening this window.)

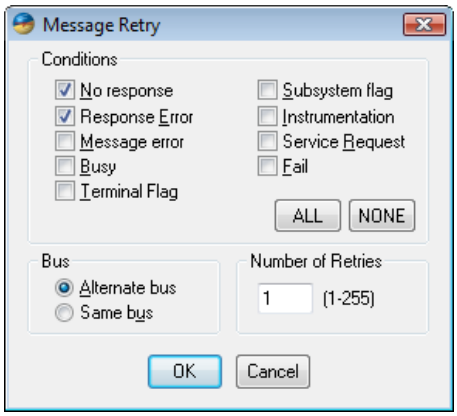
## *Message Retries*

Normally, messages in a BC Schedule are processed once, and then the schedule advances to schedule entry. When the end of a schedule is reached, the sequence continues again from the top of the schedule, continuing until the CoPilot simulation is stopped. If a retry option is attached to messages, the messages will be re-transmitted a specified number of times if the configured message conditions exist.

A message retry is associated with a particular message; if the message is in the schedule multiple times, then the retry settings for that message apply to each reference to the message in the schedule.

### Assign Retry Criteria to a Message

The retry configuration sets the conditions for retrying the message. The status of the sent message is used to determine if **any** of the retry conditions are met. The retry conditions are ORed together; so meeting any of the conditions will cause a retry to occur. Configuring the retry to transmit using the alternate bus attempts to get a valid response using the redundant bus. Retry for a BC message if configured from the Message Editor.




*BC Message retry settings*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Transmitting BC Messages

BC Messages, or commands, can be transmitted asynchronously (one message at a time) or by scheduling messages.

### Asynchronous Messages Transmission

If a keyed 1553 hardware device is properly defined and installed, BC messages may be transmitted on the bus. BC Messages can be transmitted asynchronously while running whether or not the messages are scheduled and they may also be transmitted while CoPilot is stopped (not running). The Message Editor exposes this functionality with the Xmit 

button to asynchronously transmit the defined message. Asynchronously transmitting a message can also be done from BC message objects in the Hardware Explorer tree or via scripting. Each single message transmission is confirmed with a "Message Sent" dialog box. Additionally, the message context menu allows the transmission of messages asynchronously. A summary of errors is displayed if errors occur. Advanced applications use the "TransmitMsg()" command with ATE (Automated Test Environment) to transmit messages via script events.

Note: Asynchronous message transmission for demonstration hardware (demo card) has no effect.

### Scheduled Message Transmission

Scheduled BC messages are transmitted periodically according to the schedule. A detailed description of scheduling BC Messages is found in the *Controlling the BC with Schedules* section on page *118*.

## BC Message Rate

The BC message rate is used when the BC is configured to use rate-based scheduling. The BC message rate can be configured as either a frequency value in hertz or as a period in microseconds. Newly created BC messages are initialized with the global 'Default BC Message rate' in the MIL-STD-1553 Properties tab of the CoPilot Options. See the *Controlling the BC with Schedules* section on page *118* for additional information.

Note: As long as NONE the BC message's properties are modified in the message property page, then any changes to the global BC Message transmit rate will update the message rate.

## Message Data

MIL-STD-1553 messages may contain up to 32 data words. The number of data words associated with a message is determined by the word count of the command word. Receive messages send data from the BC to an RT/SA while transmit messages instruct the RT/SA to send data to the RT. Data Fields are assigned to one or more bits of the message for engineering unit interpretation.

### Data Editor

Data for receive messages can be defined through the Data Editor window. The Data Editor can be accessed from the Message/Subaddress View window or the Message Editor window (click the **Data Editor** button). To open the Message/Subaddress View, right click on the message and choose **View**

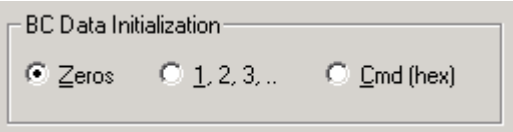


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

**Message** from the context menu. When the Message/Subaddress View window is opened, the Data Editor is contained in the Data tab.

#### Data Initialization

If data words are not defined through the Data Editor prior to running the simulation, CoPilot sets the data values in accordance with choices registered within the CoPilot Options dialog box in the MIL-STD-1553 Properties tab.




#### Data Fields

Individual words, bit strings, and word segments within a message can be defined as data fields. Once declared, fields can be named and associated with an interpreter, thus permitting data to be defined, translated, and viewed in an "engineering unit" format. Data in fields can be manipulated by the Raw Editor, Engineering Unit Editor, Data Generator, and through scripting. Details about MIL-STD-1553 engineering unit interpretations are found in the *Engineering Unit Fields* section on page *151*.

# Remote Terminal Operation

## Remote Terminal Overview

The MIL-STD-1553 specification allows for up to 32 Remote Terminals, or RT for short, on a 1553 bus. Up to 32 transmit and 32 receive Subaddresses, or SA for short, can be defined for each RT. The remote terminal responds to commands from the Bus Controller. Remote terminals are either simulated or external. A simulated RT is a remote terminal that the hardware device will generate status words responses to BC commands. An external RT is shadowed by the hardware to monitor the command, status, and data words of the external device without ever responding on the databus.

The BC can direct a simulated RT to:

- Receive data and respond with a status word
- Transmit data (to the BC or another RT) and send a status word
- Receive and respond to mode codes (data is optional)
- Respond to a bus-wide Broadcast command

CoPilot allows the user to simulate all aspects of RT behavior as defined by MIL-STD-1553 and to shadow monitor external RTs.

Note: Ballard Technology MIL-STD-1553 hardware is available in various levels of capability supporting from one terminal (configured as either BC, RT, or Monitor) to up to 32 terminals with concurrent monitoring and error injection.

#### RT Initialization

CoPilot is configured to auto-detect external RT activity by default. Newly configured simulated RTs are initialized with the following settings:

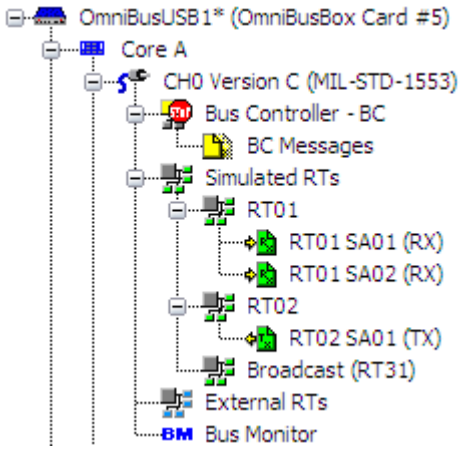


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Configured terminals are active
- Data is initialized according to the RT Data Initialization settings in the CoPilot MIL-STD-Properties
- Auto-detection of subaddresses is enabled
- All subaddresses (receive and transmit) are legalized
- All mode codes are legalized
- All subaddresses will be monitored
- Broadcast is enabled and RT 31 is reserved for broadcast

### RT and SA Hierarchy

Simulated remote terminals are grouped in the Simulated RTs node under a MIL-STD-1553 channel in the Hardware Explorer tree. External RTs are likewise represented in an External RTs node in the Hardware Explorer tree under a MIL-STD-1553 channel. The hardware will generate status word responses for simulated RTs, but act as a non-responding shadow monitor for the external RTs.




RTs and their subaddresses are configured from properties windows accessed through context menus. Right clicking on any object in the Hardware Explorer tree accesses these menus. Remote terminals and subaddresses each have their own Properties dialog boxes.

## Detecting and Adding an RT

In order to add an RT to the CoPilot project and configure it, it must be present in the Hardware Explorer tree. The user can add an RT to the Hardware Explorer tree by selecting it for simulation or shadow mode, or the user can create an “inactive” RT. CoPilot can add an RT to the Hardware Explorer tree automatically when it detects a message to it on the bus. Once an RT has been added to the Hardware Explorer tree from one of these two sources, it remains until explicitly deleted by the user.

### Automatic RT Detection

RT auto-detection is enabled by default. If CoPilot detects activity to or from an RT when the simulation in run, the RT is added to the Hardware Explorer tree (if not already present) and marked as “responded” (blue R symbol) or “no response” (red N symbol). See the figure below for an illustration of these two



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RT icons in the Hardware Explorer tree. The response designations identify the state of the RT when it was first added to the tree. These designations are not updated if the RT state changes during another run (e.g., it starts responding).

RT01 **detected (valid response)**
RT02 **detected (no response)**
RT03 **simulated**
RT04 **external**
RT05 **inactive**

### Manual Selection

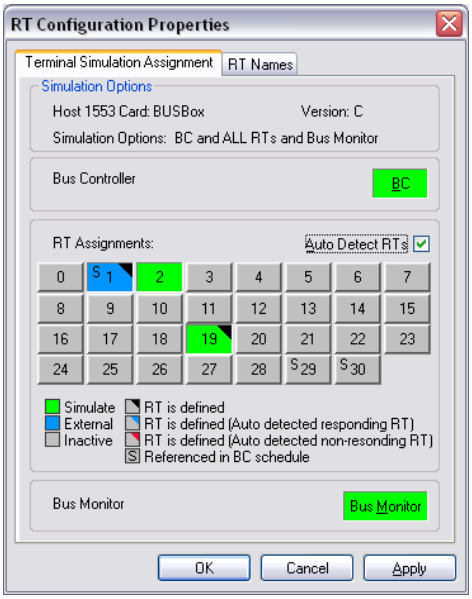
Any RT (including auto-detected RTs) can be selected for simulation, external RT shadow mode, or added to the tree in an inactive state. The Hardware Explorer tree icons for these three states are shown in the figure above.

Simulated and External RT shadow mode configurations are specified through the Terminal Simulation Assignment (TSA) window or through the context menu of individual RTs.

## *RT Terminal Simulation Assignment*

### The TSA Window

The TSA (Terminal Simulation Assignment) window (accessed from the RTs context menu) allows the user to activate the BC, Monitor, and selected RTs. The options available with the specific 1553 board are shown in the Simulation Options frame. For example, a B4 board could simulate four remote terminals or a BC and three remote terminals. Users cannot define a more complex simulation through the TSA window than their 1553 board can support. The BC and Bus Monitor can also be selected for simulation through their respective context menus.




The TSA window is also appears when channel/card properties is selected.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### RT Assignments

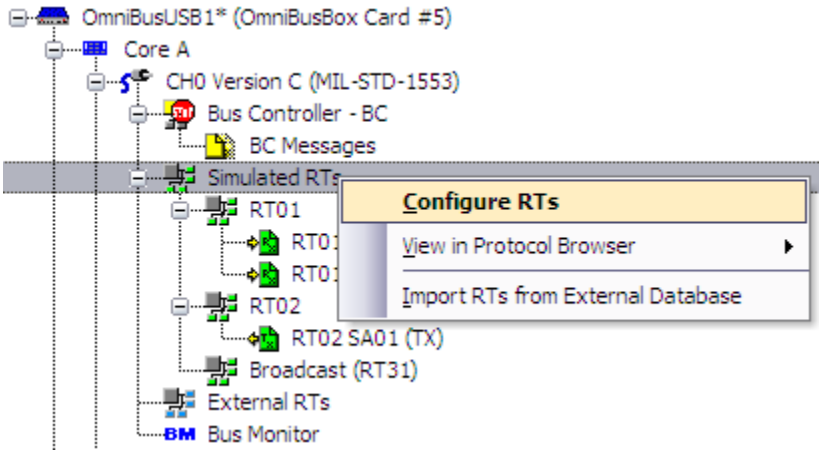
The buttons in the Simulated RTs frame indicate RT mode, detected state, and presence in the BC schedule. The legend in the terminal simulation assignment window describes these different states as shown in the figure above.

- **"S" Symbol**—indicates that this RT is addressed in the BC Schedule
- **Black Triangle**—indicates that this RT is visible in the Hardware Explorer tree
- **Blue Triangle**—indicates that this RT is visible in the Hardware Explorer tree because it was detected as an external RT with a valid response
- **Red Triangle**—indicates that this RT is visible in the Hardware Explorer tree because a BC command word addressed to it was detected but this external RT did not respond
- **Blue Button**—indicates the users selected this external RT to be "shadowed" by CoPilot
- **Green Button**—indicates the user selected this RT (or BC or monitor) to be simulated by CoPilot

## RT Configuration

### Adding an RT

To add a remote terminal for simulation or external shadow mode, double click the Simulated RTs or External RTs icon in the Hardware Explorer tree or right click and choose Configure RTs from the context menu to open the Terminal Simulation Assignment (TSA) window as shown in the figure below.




- **Configure RTs** opens the RT Configuration Properties window, also known as the TSA (Terminal Simulation Assignment) window
- **Auto-Detect RTs** toggles automatic detection of RTs during Simulation mode (applies to External RTs only)
- **Delete Detected RTs** removes the auto-detected External RTs (applies to External RTs only)

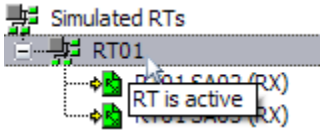


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Import RTs from External Database** allows the importing of RT definitions from dataMars *.BLS and *.ELS files (tested with dataMars version 2.0 and earlier)

### Configuring an RT

Once an RT is selected for either simulation or shadow mode, the RT can be configured through its context menu and a multi-tab properties dialog (see following sections). Since the Auto-detect subaddress option is selected by default, CoPilot automatically adds subaddress information to the tree when CoPilot is run based on the BC messages detected.




### Reusing an RT Configuration

Individual RT definitions can be saved to and loaded from the 1553 database. When you save or load an RT configuration from the database, all of it subaddresses and data fields are included.

### Save RT to Database

Another way to preserve and reuse engineering units definitions is to save an entire RT to the 1553 database. When the RT profile is loaded from the database, all the defined SAs and data fields appear automatically in the Hardware Explorer tree. To extract certain SAs or fields from a saved RT, it could be temporarily loaded on a dummy RT and the desired SAs and fields could be dragged and dropped to new locations.

Alternately, an RT could be loaded from the database and then modified as needed. The modified RT could be saved with a new name.

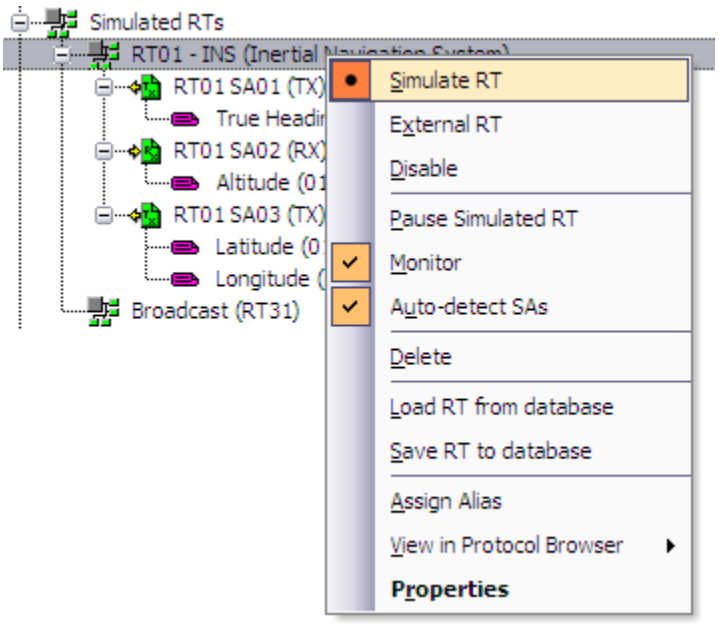
To save an RT to the database, right click on the RT icon and choose Save RT to database. To add an predefined RT from the database, follow the same process, but choose the Load RT from database command.

## *RT Context Menu*

To configure a remote terminal, right click on its icon in the Hardware Explorer tree to access its context menu (see figure below).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



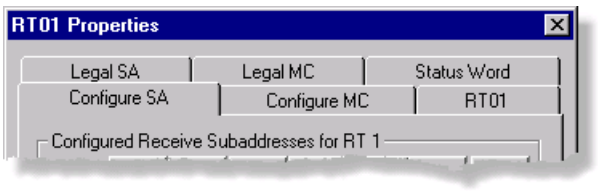

- **Simulate RT** configures this RT for simulation
- **External RT** instructs CoPilot to processes interactions of an external RT at this terminal address, but not respond
- **Disable** "turns off" this RT without removing it from the Hardware Explorer tree
- **Pause Simulated RT** allows a simulated RT to be paused, preventing it from responding to commands
- **Monitor** tags this remote terminal for selective monitor capture filtering
- **Auto-detect SAs** controls the auto-detection of subaddresses when the simulation is running
- **Delete** removes this RT from the tree
- **Load RT from database** loads an RT from the database
- **Save RT to database** adds the selected RT and its subaddresses and fields to the database
- **Assign Alias** allows the creation of an alias reference to this RT, accessible from Python in CoPilot ATE
- **View in Protocol Browser** in version 5 shows this RT in a Protocol Browser pane (1 of 4)
- **Properties** opens the Properties window for this RT

### *RT Properties*

The RT Properties window allows users to define each RT in detail. The user can enable or disable transmit and receive subaddresses, configure/legalize mode codes, set RT options, and set unique status word values for each RT. The following sections discuss individual tabs within the Properties window.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




## *Configurable RT Response Time*

At the time of publication of this manual, some MIL-STD-1553 hardware products support configurable RT response times. This feature allows the response time of unique Remote Terminals on 1553 channels. At the time of publication of this manual, level D, level M and higher products support this feature; however, consult the hardware manual to determine if this feature is supported with your hardware. The response time is measured from the mid-bit zero crossing of the parity bit to the mid-bit zero crossing of the status word. The default time is 9 microseconds. The minimum response time according to MIL-STD-1553B is 7.7 microseconds and the maximum acceptable response time is 14.0 microseconds. Ballard hardware capable of configurable response times also support the 7.7 microseconds. The maximum configurable response time is configurable to values much larger than the 14 microseconds limit in order to simulate improperly responding hardware. The response time of an RT can be modified to inject errors. Commands that do not have a RT response within the BC's allotted time should be flagged with a no response.

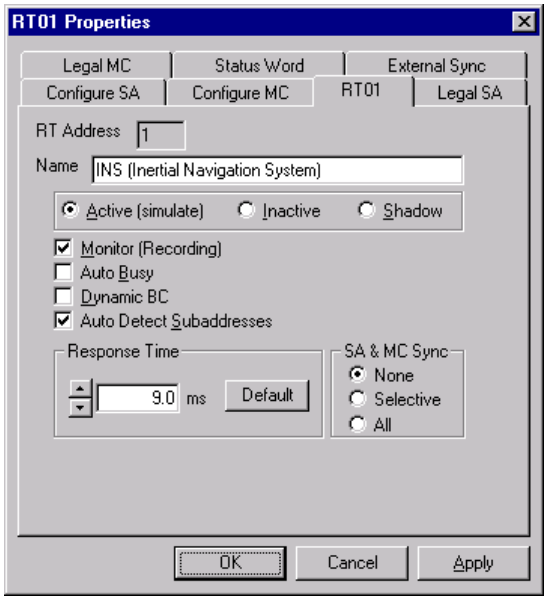
The response time of an RT is configured from the RT property page; the setting is not shown if the hardware is not capable of configuring the response time. Setting a response time greater than 14.0 microseconds can be used to generate errors as described in the Error Injection section for the MIL-STD-1553 protocol.

To configure the response time of an individual RT:

1. Right click on the RT icon in the Hardware Explorer tree and choose **Properties**
2. In the RT Properties window, select the RT## tab (see figure below)
3. In the Response Time frame, click the arrows or type in a new response time in microseconds (use the Default button to restore the 9.0 µs default)
4. Click **OK** to close the dialog and apply the configuration



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

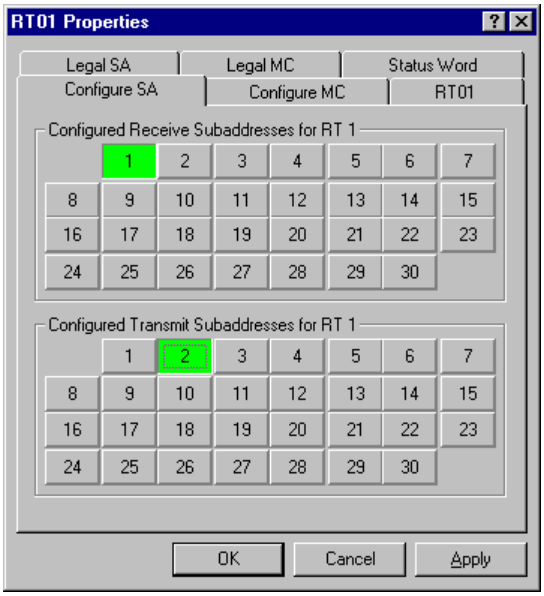



Note: Exact response time may vary depending on several factors, such as where on the bus it is measured, analog and digital delays in the on-board circuits, and uncertainty due to the board's 100-nanosecond sampling time.

## *RT Subaddress Selection*

### Configure Subaddresses

The "Configure SA" tab in the RT Properties window allows the user to specify individual subaddresses for transmit and receive. All subaddress are legalized by default (i.e., all subaddresses are available to be configured or auto-detected).




Clicking a button toggles the legalization status. When the simulation is initiated, a 64-word message buffer is established on the 1553 board for each



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

legalized subaddress. Legalized or enabled subaddresses are green and depressed.

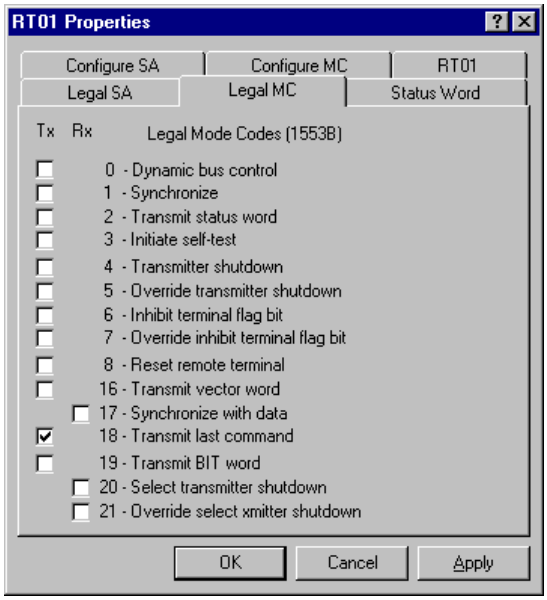
It is often quicker to have CoPilot auto-detect subaddresses. If a simulation (with a valid BC Schedule) is run without configuring specific SAs here, they will be auto-detected and appear below their corresponding RT as they come up in the Schedule. Auto-detection can be disabled from the context menu of individual RTs.

#### Legal Subaddresses

The user can illegalize certain subaddresses (i.e., render them unavailable for use). This is done under the "Legal SA" tab. Simply click on a legal SA (button will be depressed and green) to illegalize it (the button will become raised and gray). Disabling a subaddress in the hardware explorer tree is equivalent to illegalizing the subaddress.

## RT Mode Code Configuration

The "Configure MC" tab in the RT Properties window allows the user to configure individual mode codes for transmit and receive. Like SAs, all mode codes are legal by default. To configure specific mode codes, select the appropriate box and a check mark will appear. Click again to clear the check mark. Values associated with mode code data words are defined through the Data Editor. The "Legal MC" tab allows specific mode codes to be disabled (legalized/illegalized).



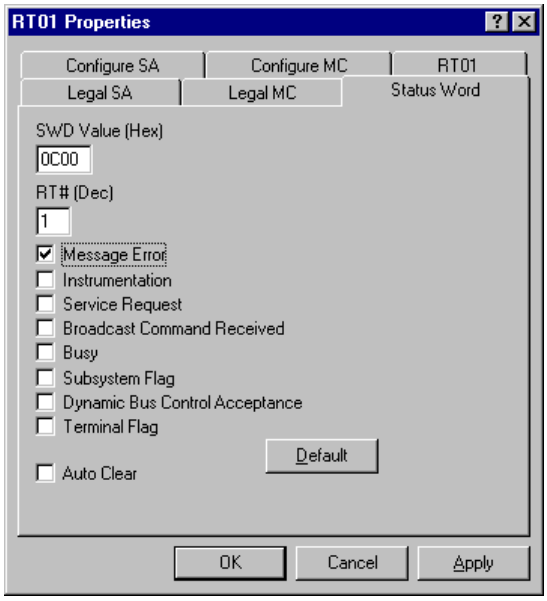

*Mode Code legalization for an RT*

## RT Status Word Configuration

When simulating an RT, the Ballard 1553 board constructs a status word based on the circumstances of each message. The first 5 bits of the 16-bit status word are reserved for the RT's terminal address. In normal operation, status bits are set to zero.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




The following conditions are represented by status bits:

- **Message Error:** This bit is set to 1 by several error conditions defined under MIL-STD-1553B. In most cases, the transmission of the status word is suppressed.
- **Instrumentation:** The purpose of the Instrumentation bit is to differentiate between status and command words, which are otherwise differentiated only by their position in a message. If used, this reduces the number of subaddresses from 31 to 15, since the instrumentation bit uses the first bit of the subaddress. This bit is always set to zero under MIL-STD-1553B.
- **Service Request:** This bit is set to 1 when the RT wants the BC to service it in some manner.
- **Broadcast Command Received:** This bit is set to 1 any time a valid broadcast command (RT address 31) has been received. RTs do not transmit status words in response to broadcast commands.
- **Busy:** This bit is set to 1 when a functional RT cannot transfer data to or from its subsystem in response to the BC command. When the RT is busy, only the status word is transmitted.
- **Subsystem Flag:** An optional status word bit used by a Remote Terminal to alert the Bus Controller that a fault is in the subsystem. It also tells the Bus Controller that the data being transmitted may be invalid.
- **Dynamic Bus Control Acceptance:** The operation of a databus system in which designated terminals are offered control of the databus, i.e., they become a BC when the terminal offering control relinquishes control. This is used only by RTs that can accept Dynamic Bus Control. Both Notice 1 and Notice 2 prohibit dynamic Bus Control.
- **Terminal Flag:** This bit indicates some fault condition in the RT. Typically, this bit is set upon a failure in a built-in test, which might have been requested by a command to initiate self-test mode.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Since the status word is a property of the RT, it applies to all transmit and receive subaddresses of that RT.

To set individual status bits:

- Click the check boxes in the Status Word tab for the bits you wish to set (notice that the status word in the SWD Value (Hex) box changes to reflect the bits that have been set)

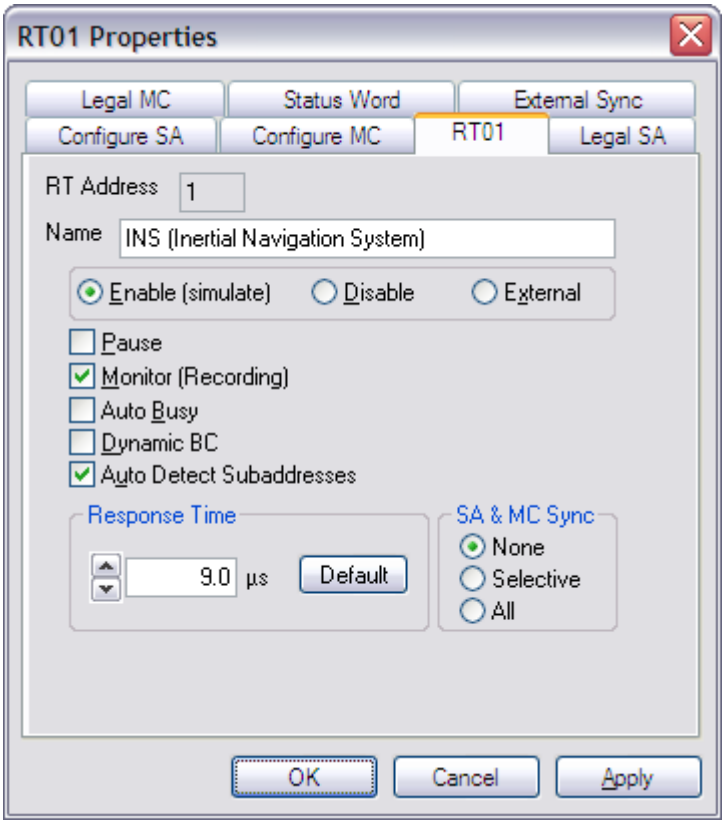
Alternately, a specific status word (in hex) can be entered in the SWD Value box and the appropriate check marks will appear below

- (Optional) Click the **Auto Clear** check box to clear the status mask after a single use. If this option is not selected, the status mask is defined until further notice
- (Optional) Click the **Default** button to reset the check boxes to the default
- Chose **OK** to accept all current settings and close the dialog box

Tip: Apply the selected bits at any time by clicking the Apply button.

## *RT Configuration Options*

The Options tab in RT Properties allows the user to name an RT and specify settings. Although the active, auto-detect and shadow options can also be set through the RT context menu, this is the only place the RT can be named or other options set.






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Name:** The name entered here will appear in the Hardware Explorer tree
- **Pause:** Suspend the remote terminal's activity
- **Monitor:** All traffic to this RT is captured and logged if the Bus Monitor is enabled
- **Enable/Disable/External:** Selects the simulation state for this RT
- **Auto Busy:** Automatically sets the busy bit of the status word after each command word
- **Dynamic BC:** Configures this RT to respond to Dynamic Bus Control mode codes
- **Auto Detect Subaddresses:** Controls the auto-detection of subaddresses when the simulation is running
- **Response Time:** Specifies a response time in microseconds for this RT (4G and 5G hardware only) - the default value is 9 µs
- **SA & MC Sync:** Configures the RT to generate a sync signal on all SA and MC responses or selectively on tagged SAs and MCs (4G and 5G hardware only)

## *RT Configurations in the Database*

By saving an RT configuration to the database, it can be accessed from other CoPilot projects. The following RT properties are preserved in the database:

- RT name
- Subaddresses/mode codes
- Data fields

All other RT properties, such as status word options, Enable/Disable/Shadow state, etc. are defined in the Hardware Explorer tree and are not modified when a definition is loaded from the database.

### Saving to the Database

An RT must be given a name by the user before it can be saved to the database. (To name an RT, double click on it, select the RT# tab, and type in the Name field.)

To save an RT definition to the database:

1. Right click on the RT in the Hardware Explorer tree and choose Save RT to database from the context menu
2. Click OK in the Database Save message box to approve the saving of a new definition or overwriting an old definition

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

### Loading from the Database

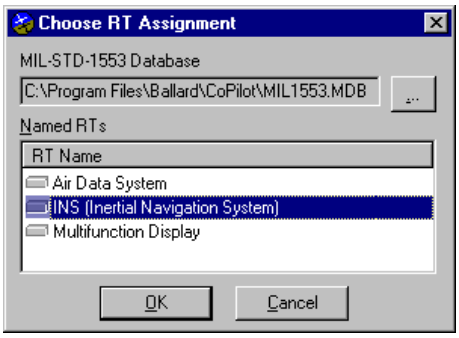
An RT must be present in the Hardware Explorer tree to accept the database definition.



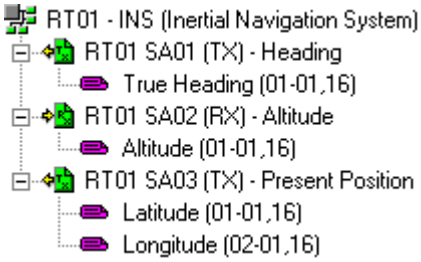
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To load an RT configuration from the database:

1. Right click on an RT in the Hardware Explorer tree and choose Load RT from database to open the Choose RT Assignment dialog (see figure below)
2. Click an RT configuration to select it and click **OK** to close the dialog and apply the definition




The specified configuration is copied to the Hardware Explorer tree (as shown below).



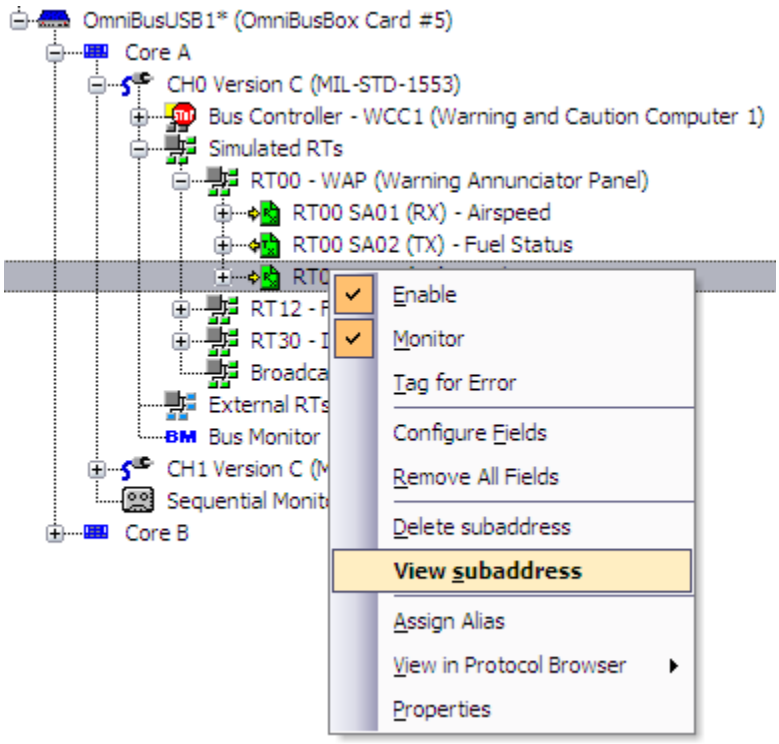

Warning: Loading an RT configuration from the database will overwrite the current name, SAs, and data fields of the selected RT.

## SA Configuration

When a new 1553 board is added to the CoPilot project, all subaddresses are enabled or legalized by default. You can change the status of each subaddress through the subaddress buttons in the subaddress tabs of the RT Properties window.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



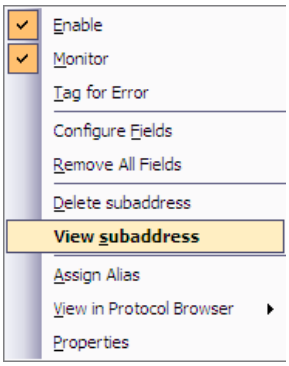

Once an RT is selected for simulation or external RT shadow mode, CoPilot automatically detects subaddresses for that RT (when the simulation is run) and expands the Hardware Explorer tree accordingly.

Subaddresses are configured through context menus and Properties pages. SAs can be renamed, and engineering units can be assigned to message data. This is accomplished through interpreters associated with data fields.

The Data Editor provides a way to define SA data and to invoke error injection in data words (if your 1553 board supports error injection).

## *SA Context Menu*

Right click on a subaddress in the Hardware Explorer tree to access its context menu.




- **Enable** enables or disables this SA
- **Monitor** tags this individual subaddress for capture by the Bus Monitor
- **Tag for Error** marks this subaddress as a candidate for errors (error injection must be enabled)
- **Configure Fields** opens a dialog that allows you to define, modify, and organize data fields within this message
- **Remove All Fields** clears all data fields from this subaddress
- **Delete subaddress** removes this subaddress from the Hardware Explorer tree (but does not illegalize it)

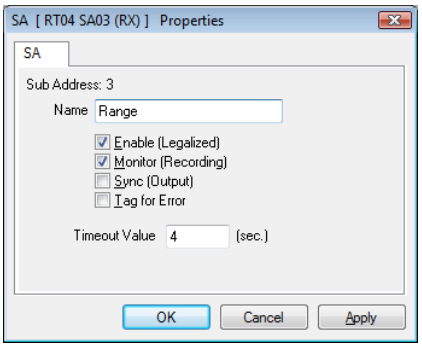


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **View subaddress** opens the Message/Subaddress View window
- **Assign Alias** allows the creation of an alias reference to this SA, accessible from Python in CoPilot ATE
- **View in Protocol Browser** in version 5 shows this SA in a Protocol Browser pane (1 of 4)
- **Properties** opens the properties page for this SA

### *SA Properties*

The subaddress properties page allows the user to define the properties of individual subaddresses. To access this dialog box, right click on a subaddress and choose Properties from the context menu.




*Subaddress property page*

- **Name:** The name entered here will appear in the Hardware Explorer tree, appended to the default name
- **Enable:** This command enables or disables this subaddress. This command only applies if the SA is legalized (all SAs are legalized by default but may be explicitly illegalized by the user).
- **Monitor:** Tags the individual subaddress for capture by the Bus Monitor (monitoring all channel traffic can be selected from the Bus Monitor context menu)
- **Sync:** Generates a sync pulse (not transmitted on the data bus) when a message is sent, received, or responded to with a status word. The External Sync options of the RT configure the selected sync lines.
- **Tag for Error:** Tags this subaddress as a candidate for error injection (error injection must be enabled)

## SA Data Editing

#### Data Editor

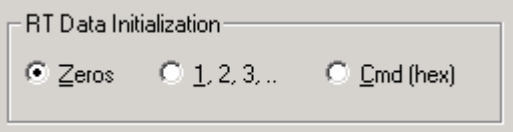
Data for transmit subaddresses can be defined thought the Data Editor window. The Data Editor can be accessed from the Message/Subaddress View window. To open the Message/Subaddress View, right click on the subaddress and choose View Subaddress from the context menu. When the Message/Subaddress View window is opened, the Data Editor is contained in the Data Tab.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### Data Initialization

Data initialization of all subaddresses (even undefined subaddresses) using the RT data initialization setting in the CoPilot Options for MIL-STD-1553 (from the Project menu). The current data initialization setting value is used each time the simulation is started. Once data in an SA is modified (including editing a SA's field), then the data initialization setting is no longer used for that SA.




#### Data Fields

Individual words, bit strings, and word segments within a message can be defined as a field with engineering unit interpretations. Once declared, fields can be named and associated with an interpreter, thus permitting data to be defined, translated, and viewed in an "engineering unit" format. Data in fields can be manipulated by the Raw Editor, Engineering Unit Editor, Data Generator, and through scripting.

# Engineering Unit Fields

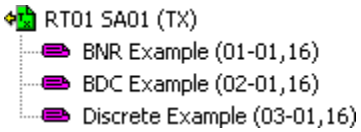
Although fields and engineering units are described earlier in this manual, this section highlights the specifics for MIL-STD-1553 fields and engineering unit interpreters.

## Data Fields Overview

#### Engineering Unit Translation

Individual words, bit strings, and word segments within a message can be defined as data fields. The *Engineering Units* section on page *82* provides an overview to engineering units, while this section focuses on MIL-STD-1553 specifics. Once declared, fields can be named and associated with an engineering unit interpreter, thus permitting data to be defined, translated, and viewed in an "engineering unit" format. Thus, data can be assigned meaning within the context of the CoPilot simulation. Fields can be defined on both BC Messages and SA's.

In the figure below, three fields are shown. Each is identified by the user-defined name and the word and bit sequence within the message. Typically, each field would be identified by a meaningful name such as "*altitude*" or "*ground speed.*"




*Multiple fields defined below a SA message*

#### Data Interpreter Types

MIL-STD-1553 fields can be assigned different data interpretations. These formats change how data is interpreted. CoPilot uses the interpreters associated



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

with fields to display and edit data values. Newly created fields in CoPilot 1553 default to the BNR (binary) interpreter. It is possible to change the interpreter of a fields using CoPilot. The MIL-STD-1553 interpreter types are described in detail in the following Interpreter Definition section.

## Field Definitions

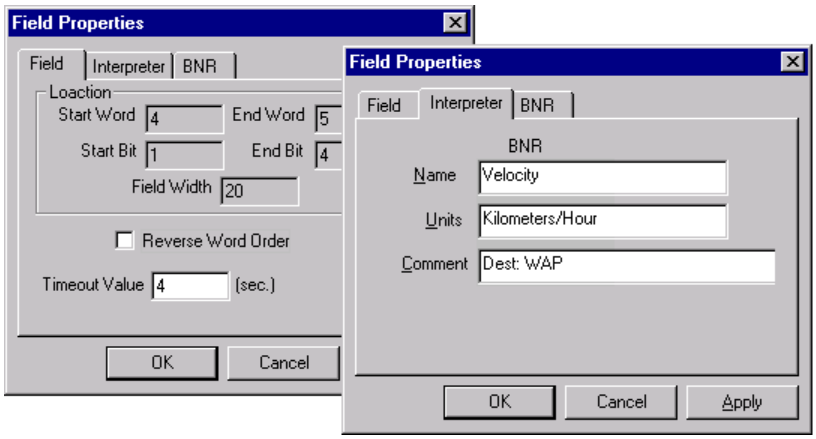
When a new 1553 field is created in CoPilot, it is defined by its name, description, location in the message, and interpreter. Field properties are typically defined and attached to a 1553 message through the Add & Edit Fields dialog, but they can be viewed and modified through the Field Properties dialog.

Once a field and its interpreter are defined, the field will appear in the Hardware Explorer tree. It can then be displayed through Message/Subaddress Views, Engineering Views, Control Views, Map Views, and Strip View windows. Field data can be modified through the Raw Editor, Engineering Unit Editor, Data Generator, or through scripting routines. Field properties are accessed from the context menu of individual fields.

### View Field Properties

To view the definition of an existing field:

1. Right -click on the field in the Hardware Explorer tree
2. Select **Properties** from the context menu




The field location, description and interpreter specifications are displayed and modified through a three-tabbed properties window:

### Field Location

A field location is defined by a starting word and starting bit. The location frame of the Field Properties window also shows the end word and bit (determined by the field width). As long as the start and end locations are within the host 1553 message (e.g., the field in a 4-word message cannot start or end in word 5 or beyond), 1553 fields can begin at any bit position and may traverse data word boundaries.

The selected field width is chosen to accommodate the targeted data range, scale, resolution, and interpreter type. The range for various interpreters are shown in the table:

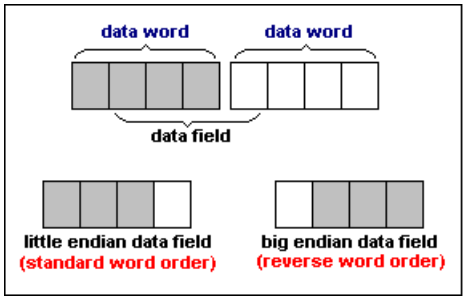


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| Interpreter | Width |
|---|---|
| BCD | 1-32 bits |
| BNR | 2-32 |
| BNR 48-bit | 48 only |
| BNR 64-bit | 64 |
| CAPS 48-bit | 48 |
| Custom Script | 1-32 |
| DEC Floating Point 64-bit | 64 |
| Decimal | 1-32 |
| Discrete | 1-16 |
| IEEE 754 32-bit | 32 |
| IEEE 754 64-bit | 64 |
| MILSTD1750 48-bit | 48 |
| MILSTD1750 64-bit | 64 |
| Simple Scalar | 1-32 |

### Reverse Word Order

The Reverse Word Order option appears in the field properties dialog for fields that span more than one data word. The default word order is little endian, however, if you are communicating with a system that uses the big endian order, setting this option allows you to reverse the word order of an individual fields to be compliant with the system. Setting the reverse word order in the global CoPilot MIL-STD-1553 options (from the project->options menu) initializes this setting for all newly created fields. When selected, this option reverses how the data field is interpreted and displayed (see figure below).



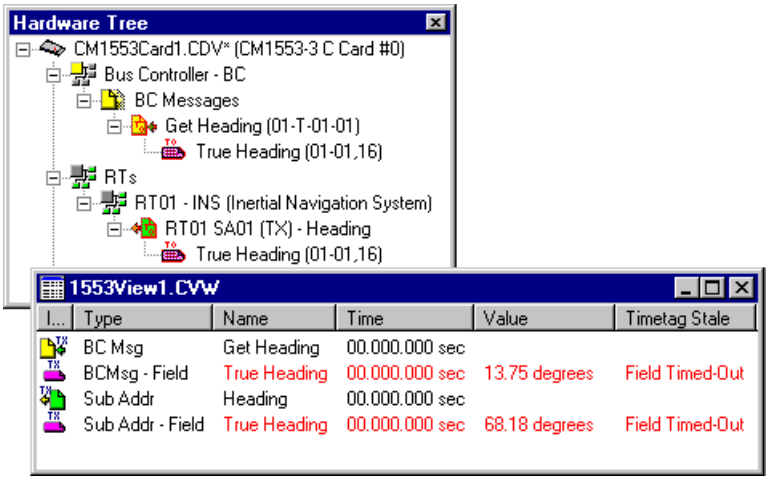

Note: Little endian is a format in which data is stored or transmitted least significant (low byte) first and is the default format used by CoPilot. Big endian format is most significant (high byte) first.

### Timeout Value

The timeout value tests for inactivity (during a run). Timeout conditions are reported in Engineering View windows (see figure below) and the Hardware Explorer tree with a red “TO” symbol on the field icon on its message/subaddress.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




#### Interpreter

The interpreter type is displayed above the Name field in the Field Properties dialog above. The default interpreter is BNR (binary). You may also select BCD, Custom Script, Decimal, Discrete, IEEE 32-bit, and Simple Scalar.

#### Name

A default name is provided when a field is created or imported. Since the field name is used in the Hardware Explorer tree and all displays, a more descriptive name will be useful.

#### Units

This defines the units of measurement such as “feet” or “degrees.” These units are added to the interpreted value to create “engineering units” and are displayed in many windows and objects in CoPilot.

### *Create New Field*

The Create New tab in the Field Selector dialog (launched from the Add & Edit Fields window) allows the user to define a new data field. Fields may be inserted in BC messages or RT subaddresses.

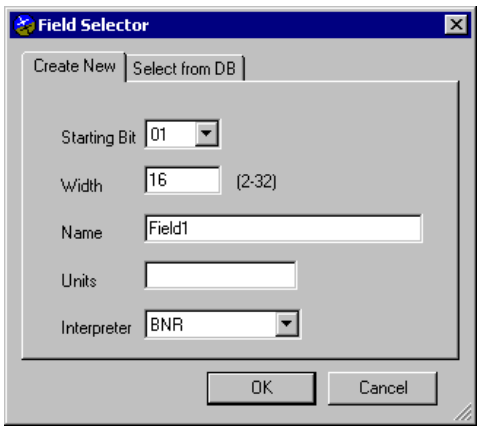
To insert a data field:

1. Right click on a message or subaddress in the Hardware Explorer tree and choose **Configure Fields** from the context menu to open the Add & Edit Fields dialog (see figure)
2. Select a start word by clicking its listbox
3. Click the **Add** button to open the Field Selector dialog

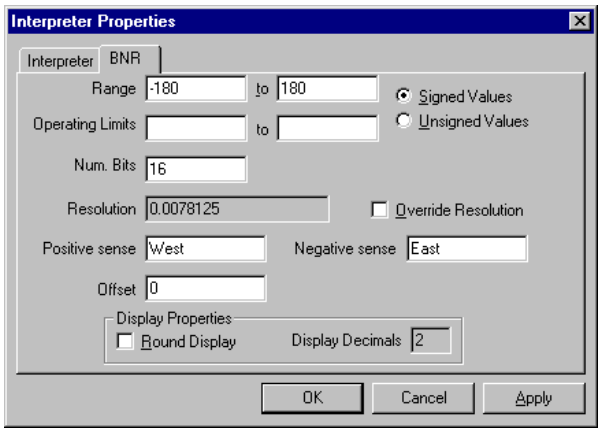
   The Field Selector dialog defaults to the Create New tab (see figure). You could also load a field from the database by clicking the Select from DB tab.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




4. Define the starting bit, width, name, units of measurement, and interpreter type and click **OK** to launch the Interpreter properties window (name and units are pre-supplied, based on what you entered in the Field Selector dialog)




5. Click the interpreter type tab (BNR type in the figure above), define the interpreter characteristic, and click OK to return to the Configure Fields dialog (where you can continue to add and edit fields)
6. When you are done adding and editing fields, press the OK button to close the Configure Fields dialog

#### Extended Fields

Data fields may extend over more than one data word. For example, a 32-bit BNR field that starts in bit 8 will extend across parts of three data words.

Note: Fields may overlap one another provided they have different start word and start bit combinations.

## *Load Field from Database*

#### Selecting a Field from the Database

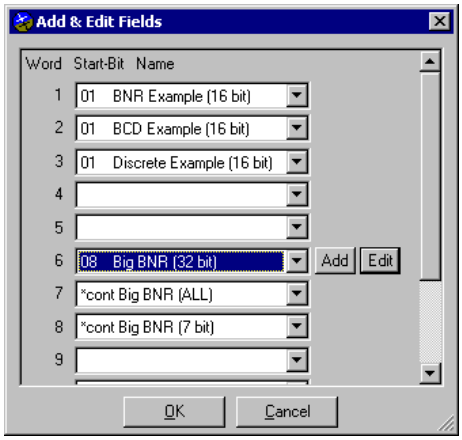
In the Select from DB tab in the Field Selector dialog (launched from the Configure Fields dialog) you can browse a database for a previously defined field.



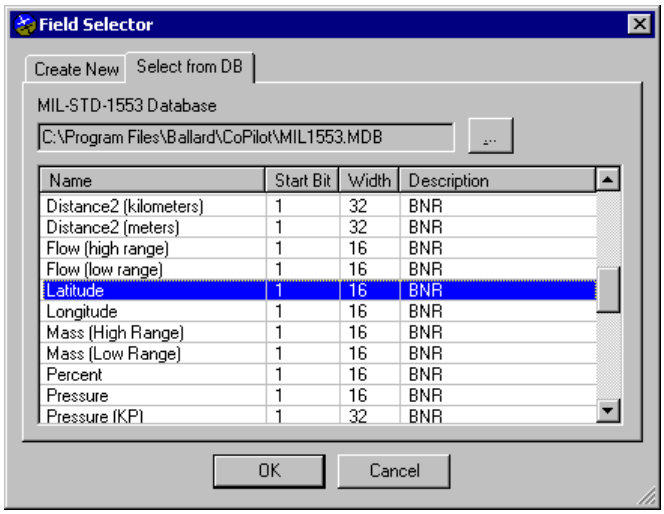
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To insert a field from the database:

1. Right click on a message or subaddress in the Hardware Explorer tree and choose **Configure Fields** from the context menu to open the Add & Edit Fields dialog (see figure below)




2. Select a start word by clicking its listbox
3. Click the **Add** button to open the Field Selector dialog
4. Click the **Select from DB** tab to access the list of saved data fields

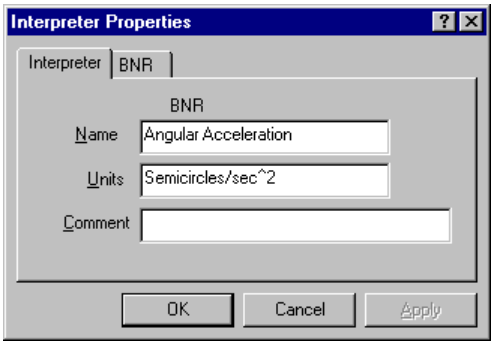



   If you wish to change the database listed, press the browse button and select the new database.

5. Select the field you wish to load and press **OK** to launch the Interpreter Properties dialog
6. (Optional) Enter a new name, resolution, field width, etc. if desired



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




7. Click **OK** to accept the current definition

You can define and save data field(s) to the database, thus customizing it for your use.

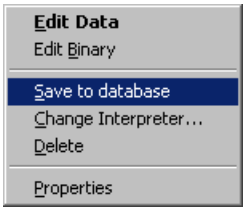
Tip: In addition to loading individual fields, you can also load an entire RT from the database (complete with subaddresses and data fields).

## *Save Fields to Database*

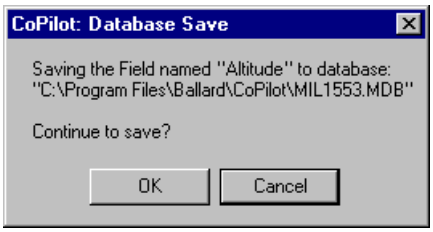
CoPilot allows field definitions to be reused by saving them to a database.

To save a field to the database:

1. Right click on the field (in the Hardware Explorer tree) and select **Save to database** from the context menu




2. Click **OK** to confirm the saving or overwriting of this named field (see figure below)




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

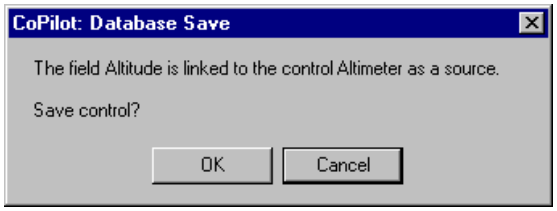
It is recommended that you create a unique name for a modified database field so that the previous version is not lost.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### Save with Control

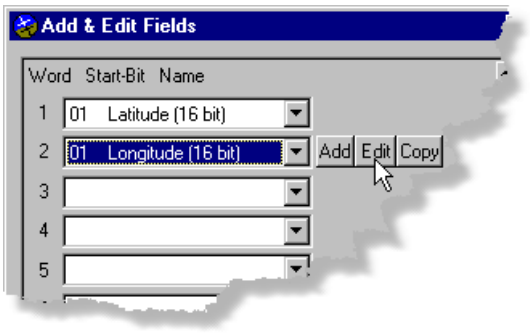
If the field is linked to a control in the Control View window (as a source or as sink), a second dialog will appear with the option to save the link to that control to the database with the field.



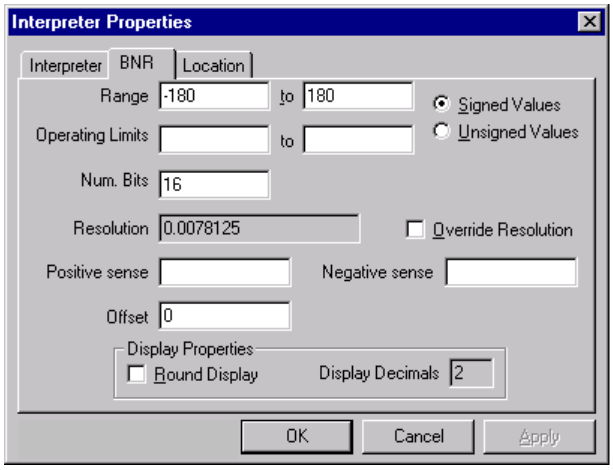

As a result, dragging that imported field into a Control View window will cause the saved control to appear, rather than the default control. For example, if you saved an altimeter control that is linked to an altitude field, dragging that field (altitude) into Control View will produce an altimeter control.

### *Edit Fields*

To edit a field from within the Configure fields dialog, select its start word listbox and press the Edit button (see figure below) to open the Interpreter Properties dialog (see second figure below).




From here, you can edit the properties associated with the field (including its start bit within the start word in the Location tab).




To access this dialog from the Hardware Explorer tree instead of from the Add & Edit Fields dialog, right click on the field in the Hardware Explorer tree and select Properties from the context menu.

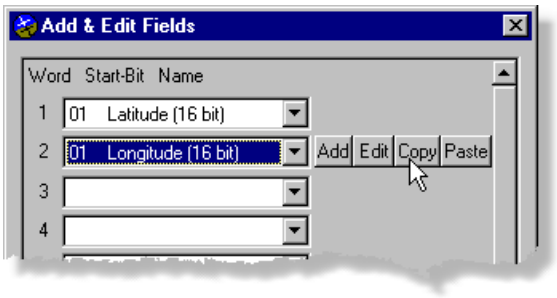


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## *Copy Data Field Definitions*

### Copy within a Message or Subaddress

You can easily copy and paste a field from within the Add & Edit Fields dialog from one data word to another within the same message or subaddress (see figure below).



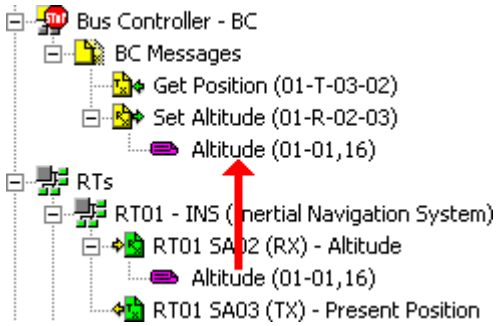

Because you are given the opportunity to modify the field before pasting it, you can use this feature to create a slightly different field based on the original.

To copy a field in the Add & Edit Fields dialog:

1. Click on the word that contains the starting bit for the field you wish to copy and click the **Edit** button
2. Click to select the word location where you wish to place the copy and click the **Paste** button to launch the Field Selector dialog (the fields in the Create New tab are filled with the specifications of the copied field)
3. Modify any of the fields if you wish and click **OK** to open the Interpreter Properties dialog (once again, it is filled out with the properties of the copied field)
4. Modify any of the fields if you wish, click **OK** to close to close the Interpreter Properties dialog, then click **OK** to close the Add & Edit Fields dialog and insert the copied field into its new location

### Copy to a Different Message or Subaddress

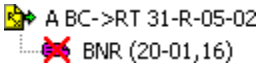
Within a CoPilot project, field definitions can be quickly propagated to other locations in the Hardware Explorer tree through a drag and drop procedure. To copy a field definition to a new location, click on the field and drag it onto a BC message or RT subaddress.




If the target location does not have the word or starting bit or bit field available to host the new field, a warning dialog will appear and the copied field will be marked with an X icon.



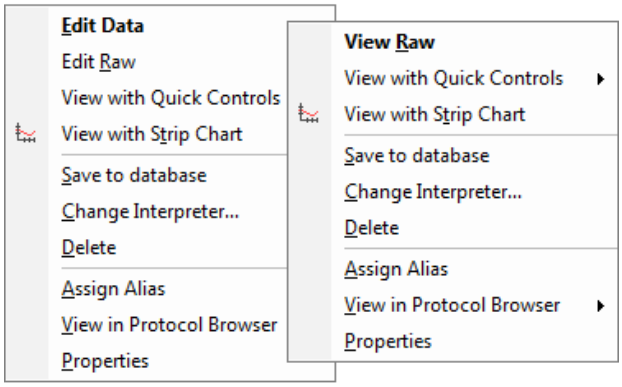
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




Note: Dragging a field onto a message or subaddress that already contains a data field in that location will overwrite the previous field.

## *Field Context Menu*

Field interpreters, data, and properties may be modified through context menus. Right click on a field in the Hardware Explorer tree to access its context menu.




- **Edit Data** opens the Engineering Unit Editor
- **View/Edit Raw** provides access to the Raw Editor for editing and viewing
- **View with Quick Controls** expands a submenu of Quick View virtual controls (CoPilot Professional only)
- **View with Strip Chart** launches a Strip View window displaying field values (CoPilot Professional only)
- **Save to database** saves the data field with its name, interpreter, and other properties to the database.
- **Change Interpreter…** invokes the Select New Field Interpreter dialog
- **Delete** removes the field from the Hardware Explorer tree
- **Properties** opens the Field Properties window

## *Field Properties*

The Field Properties page allows the user to define the properties of individual fields. To access this dialog box, right click on a field and choose Properties from the context menu.

### Field Tab

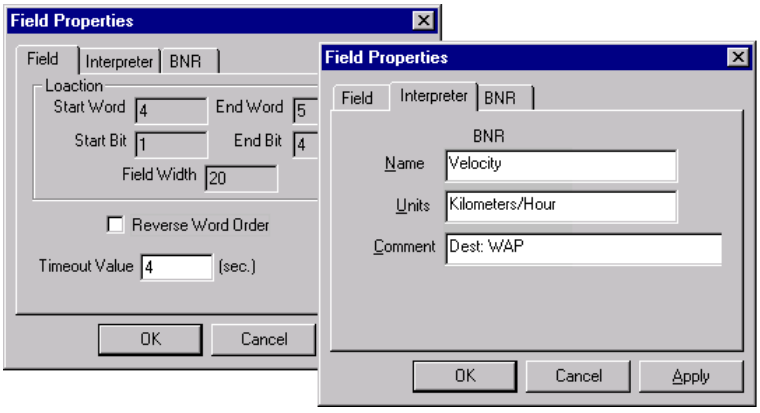
The Field tab displays the field statistics. It contains the same options as the Create New Field dialog box. Some text boxes are dimmed, indicating that these values cannot be modified.

### Interpreter Tab

Custom name, units of measurement, and comments may be added here. This tab contains the same options as the Interpreter Properties dialog.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




#### Interpreter Type Tab

The final tab will be the interpreter type (BCD, BNR, Decimal, Discrete, IEEE 32-bit, or Simple Scalar). The options in this tab depend on which interpreter is selected.

### *Field Data*

#### Data Word and Data Fields

Field data will affect the value of the data word or words in which it resides. In CoPilot, values may be set for the data word as a whole, or fields may be inserted into the data word. When a field is inserted in a data word or words, its value overwrites the previous word values for the width of the field (unused bits are padded with zeros). If the value of a data word is modified after a field has been defined, it will overwrite the field data. The data word bits reflect the most recently set value. Understanding this principle will prevent data conflicts.

- The global data initialization choice (zeros, incremental, or hexadecimal command words) and the Data Editor modify data words as a whole.
- The Data Generator, Raw(radix) Editor, Engineering Unit Editor, Data Modifier control, and scripting routines can modify data fields within data words.

## Engineering Unit Interpreter Definition

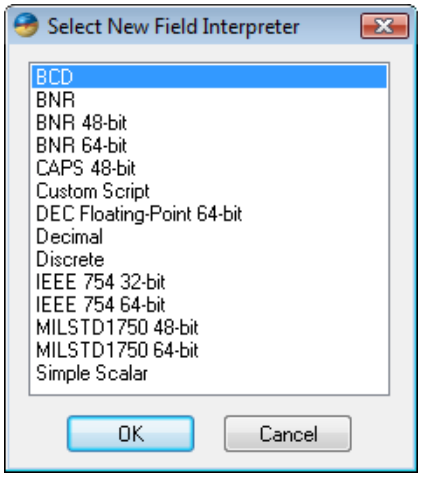
An interpreter translates the ones and zeros that come across the bus into meaningful values. Data may be encoded to represent alphanumeric characters, for example, or include a sign bit to indicate a certain code or condition. The interpreter is an important step in creating engineering units for data values.

#### Interpreter Assignment

The interpreter is chosen and its properties defined when adding a new field to the Hardware Explorer tree. The interpreter for a data field may be changed at any time. Right click on a field in the Hardware Explorer tree and choose **Change Interpreter…** from the context menu to open the Select New Field Interpreter dialog.



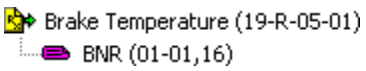
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



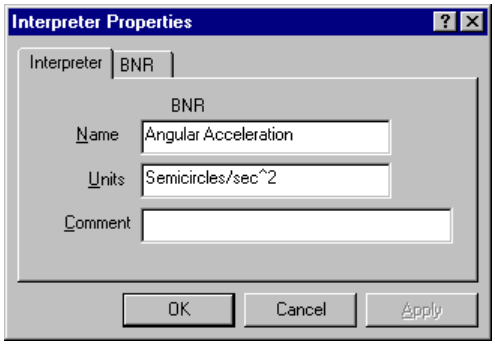

Select a new interpreter from the list and click **OK**. The Interpreter Properties dialog will open so you can configure and define the new interpreter.

### Interpreter Properties

The Interpreter tab (in the Interpreter Properties or Field Properties dialogs) is where the data field is named and the units of measurement are defined. Typically, each field would be identified by a meaningful name (e.g., altitude, ground speed, etc.). The field is listed in the tree by the interpreter name and the field statistics (start word, bit, and width).




The units of measurement associated with the data can be typed in the Units text box. Comments pertaining to the data field may be added as well (e.g., “plus equals north”).




### Interpreter Types

The tab immediately following the Interpreter tab bears the name of the selected interpreter (e.g., BNR or Discrete). This tab contains all of the settings unique to that interpreter. There are multiple MIL-STD-1553 interpreters to choose from:

- BCD
- BNR
- BNR 48-bit
- BNR 64-bit



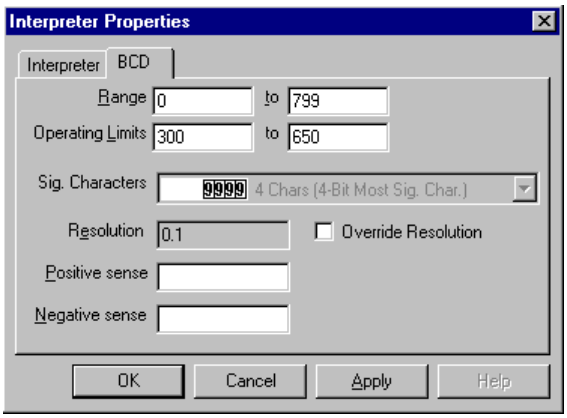
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- CAPS 48-bit
- Custom Script
- DEC Floating-Point 64-bit
- Decimal
- Discrete
- IEEE 754 32-bit
- IEEE 754 64-bit
- MILSTD1750 48-bit
- MILSTD1750 64-bit
- Simple Scalar

In the following sections, each interpreter is discussed in detail.

## BCD Interpreter

The BCD (Binary Coded Decimal) interpreter allows the user to adjust the range of the data, to establish the BCD character rules, and define other optional parameters. Resolution values are generated automatically from the upper and lower ranges unless Override Resolution is checked. Operating limits are used to provide a visual indication and status display in an Engineering View if data is outside the specified operating range.



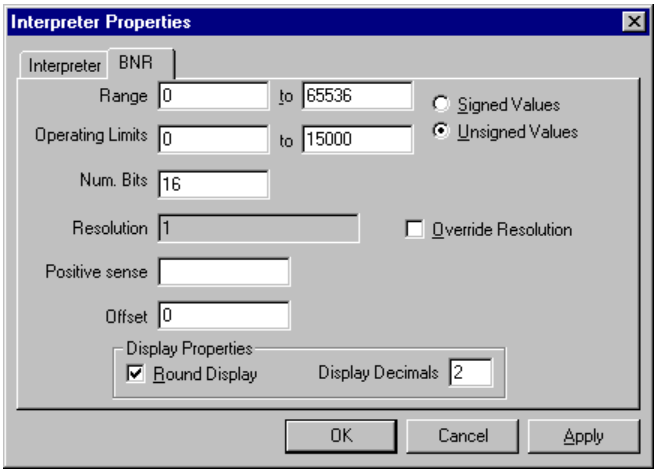

*An example of BCD data is date, time, or frequency values.*

## BNR Interpreter

When a new field is created, BNR (binary) is the default interpreter. BNR is a widely used binary format that is compatible with many types of data. The BNR interpreter defaults to a 2's complement format (sign bit indicates positive/negative). If you wish to encode a positive number, select the Unsigned Values option button. CoPilot automatically sets the low range to 0 (zero) and recalculates the resolution.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




The BNR interpreter allows the user to adjust the range of the data and the number of significant digits in the data field. Resolution values are generated automatically from the upper and lower ranges and number of bits, unless Override Resolution is checked. Operating limits are used to provide a visual indication in an Engineering View window if data is outside the specified operating range. The value of the least significant bit is shown in the Resolution text box.

## *BNR 48-bit Interpreter*

Same as a BNR interpreter with a fixed size of 48-bits.

## *BNR 64-bit Interpreter*

Same as a BNR interpreter with a fixed size of 64-bits.

## *CAPS 48-bit Interpreter*

The CAPS interpreter is a floating point interpreter. The length of a field with a CAPS interpreter must be 48 bits. The lower 8 bits are a 2's complement exponent (excess 128) and the upper 40 bits are a 2's complement mantissa (including the msb sign bit). As with other scalar interpreters, operating limits and rounding can be configured.

## *Custom Script Interpreter*

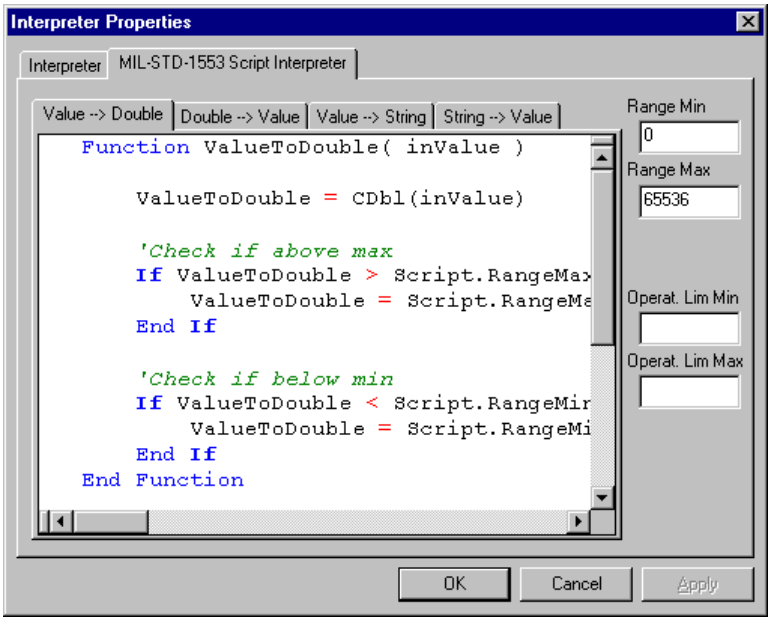
### Custom Interpreter

When the user's requirements go beyond the scope of the other CoPilot interpreters, the Script interpreter allows the user to create a customized data interpreter for their specific application. The custom script interpreter requires the use of VBScripting.

The default script interpreter code (contained in four function-panes) type casts between raw binary data to other data types (for example, double) and validates the results against the supplied range (users must supply the range and operating limits). Users can define custom conversions by modifying the script.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




### Modifying the Script Interpreter

Global variables and helper functions for custom conversions may be defined on any of the four script interpreter tabs. All functions and global variables can be called from and used by any of the Script Interpreter function tabs.

---

Note: The four predefined functions (ValueToDouble, DoubleToValue, ValueToString, and StringToValue) can be added to or replaced, but they cannot be renamed or deleted.

---

Four properties may be accessed by the script interpreter object:

- RangeMax
- RangeMin
- OperatLimMax
- OperatLimMin

These range and operating limits properties are addressed with the following syntax: Script.*Property* (where *Property* is one of the four listed above, e.g., RangeMax).

---

Note: The minimum and maximum values for range and operating limits are calculated against the double value.

---

For more information on the principles and guidelines for scripting in CoPilot, please refer to the section on Scripting and the Script View window.

### Example Interpreter Scripts

There are example interpreter scripts included with CoPilot. To locate the interpreter scripts folder (copied during installation), browse to the CoPilot Projects folder (default location is C:\UserName\My Documents\My CoPilot Projects\Samples\Scripts) or retrieve them from the CoPilot CD. Check the CoPilot scripts page



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

(http://www.ballardtech.com/pages/CopScripting/CopScripting.aspx) on Ballard's website for more example scripts.

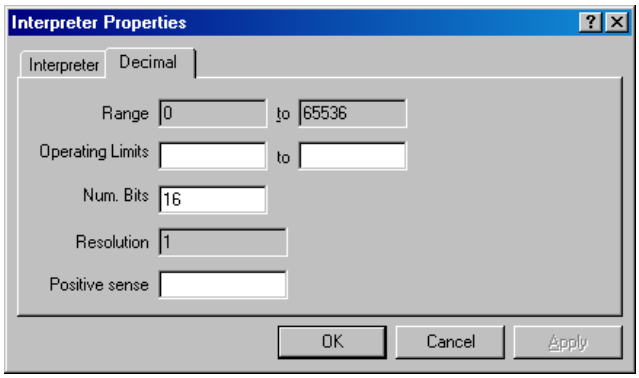
To use an example script interpreter, open the sample script, copy the first section (for Value→Double), and paste it into the Value→Double tab (overwrite the default contents). Repeat this process to replace the code in the remaining tabs.

### DEC Floating-Point 64-bit Interpreter

The DEC interpreter is a floating point interpreter. The length of a field with a DEC interpreter must be 64 bits. The exponent is an 8 bits 2's complement value (excess 128) and the 2's complement mantissa is a 56 bits long (including the msb sign bit). As with other scalar interpreters, operating limits and rounding can be configured.

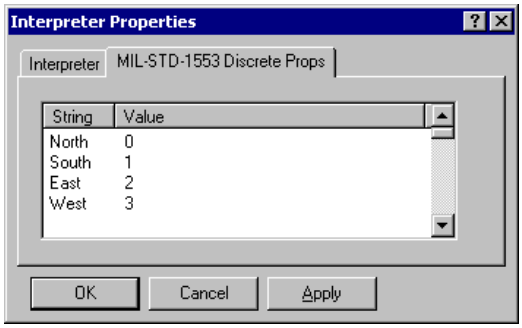
### Decimal Interpreter

The decimal interpreter is a simplified positive value only BNR interpreter. The range is calculated for you using the number of bits in the field and a fixed resolution of 1 (one).




### Discrete Interpreter

A discrete is a single binary bit whose state of one or zero has a specified meaning. Discrete bits in 1553 data fields are used to represent characters or modes. The discrete interpreter represents the discrete field as strings. For example, a two-bit discrete may be translated into east, west, north, and south. A unique text string represents each possible value. The Discrete property tab contains a list of possible values (determined by the number of bits in the field) and a display string for each one.




Users can define individual outcomes by clicking on a line and typing in a text string.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### IEEE 754 32-bit Interpreter

The IEEE interpreter is a floating-point representation based on the IEEE standard. The single-precision 32-bit IEEE 754 interpreters have the following formats:

- 32-bit - [1sign][8exponent][23mantissa]
- 64-bit - [1sign][11exponent][52mantissa]

The length of a field using an IEEE interpreter must be either 32-bits for single-precision (spanning 2 data words) or 64-bits for double-precision (spanning 4 data words).

### IEEE 754 64-bit Interpreter

Same as IEEE 754 32-bit interpreter with a fixed size of 64-bits.

### MIL-STD-1750 48-bit Interpreter

MIL-STD-1750 is a microprocessor standard meant to be used in all airborne computers and weapons systems. This standard included definitions for floating point values. CoPilot implements 48 bit and 64 bit MIL-STD-1750 interpreters. For a 48-bit MIL-STD-1553, the mantissa is a 40 bit 2's complement value (including the msb sign bit) with exponent is an 8 bits 2's complement value. The format of the 48-bit floating point formats are:

- 32-bit - [1sign][23mantissa][8exponent]
- 48-bit - [1sign][23mantissa][8exponent][16extended exponent]
- 64-bit - [1sign][23mantissa][8exponent][32extended exponent]

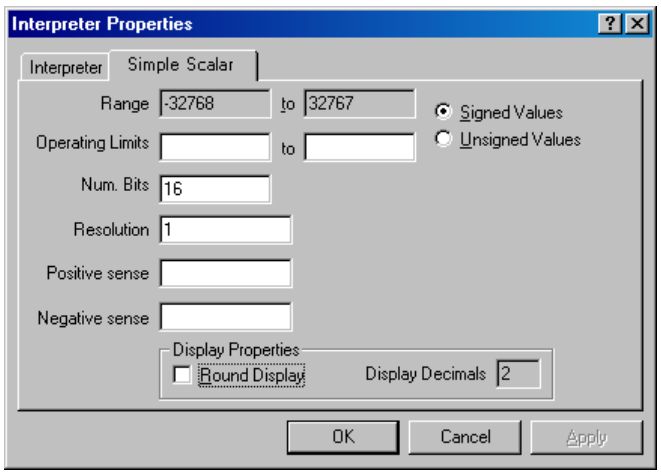
Conversion between a 48-bit or 64-bit value only loses extra precision.

### MIL-STD-1750 64-bit Interpreter

Same as a MIL-STD-1750 48-bit interpreter with a fixed size of 64-bits.

### Simple Scalar Interpreter

The Simple Scalar interpreter is a simplified BNR interpreter. The only difference between the simple scalar interpreter and the BNR interpreter is that the simple scalar range values are automatically calculated for you based on the number of bits and resolution.






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Editing Field Data

### Edit Interpreted Data (Engineering Units)

To edit interpreted data select Edit Data from either a Field context menu. A data editor window will open containing controls to change the value graphically (usually a slider control) or type in a value directly. To close the data editor window click the mouse anywhere outside the editor. The data editor window can also we opened from linked items in some view windows (e.g., 1553View). For more information on editing data see *Generate and Edit Data* on page *91*.

### Edit Raw

Messages can also be edited in Binary, Octal or Hexadecimal formats using the Binary Editor. To open the raw value editor select Edit Raw from the message context menu. To change the binary value double click on a binary bit (1 or 0) to toggle that bit value. Alternatively, use the alternate radix edit controls to view and edit the value in octal or hexadecimal.

### Edit data from Scripting

Data throughout CoPilot can be edited in a variety of ways. In addition to editing the data directly or sourcing data with a data generator, scripting can be used to edit data. The data values could be conditionally set based on the values of other parameters (such as setting a discrete when a parameter is above a limit) or automatically step through values for a simulation. The ATE Command Prompt and Watch Window panes can also be used to quickly set parameters.

# Bus Monitor and Sequential Monitoring

## Bus Monitor Overview

### Monitor with any Ballard 1553 Board

The CoPilot Bus Monitor is used to capture and review bus traffic on a MIL-STD-1553 databus. Any of Ballard's 1553 boards can be used as an independent Bus Monitor. With a Ballard 1553-3 level C or higher board, users can simulate a Bus Controller, up to 31 RTs, and monitor all bus activity concurrently (see "CoPilot-Compatible Boards" on page *61*).




The Bus Monitor preserves a sequence of 1553 messages in a file so that bus traffic can be examined later. A time-tag is attached to each record placed in the file. With a single Monitor All command, users can specify that all data will be recorded in the monitor file or they may, with message, RT, or subaddress filters, create a highly selective record.

### Activating the Monitor

If the monitor is not configured or is inactive, the monitor icon  is dimmed. The monitor, like other 1553 terminals, is configured through the Hardware



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Explorer tree. Monitor criteria are uploaded to the Ballard 1553 board when the run commences and data is placed in a monitor cache.

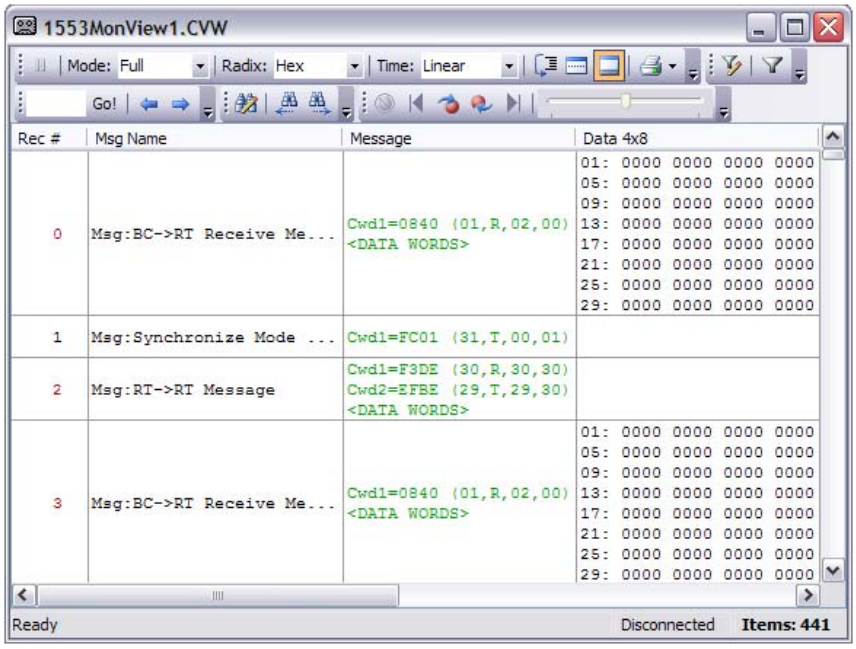
To monitor all bus traffic:

1. Right click the dimmed monitor icon in the tree to access the monitor context menu
2. Select the **Enable** command to activate the Monitor
3. Select **Monitor All** to monitor all messages
4. Click the CoPilot **Run** button

When the monitor is active, the record count is displayed in the Bus Monitor (or Sequential Monitor) branch of the Hardware Explorer tree.

### The Display of Monitor Records

MIL-STD-1553 bus traffic is recorded using the Bus Monitor into a Sequential Monitor. The recorded history of the current capture file (or previous capture files) can be simultaneously viewed using a Sequential Monitor View (see the general section on page 69). The viewing window can be opened before, during or after the 1553 hardware is set in the running state.




The display filters, search commands, and formatting options available in the 1553 Monitor View window simplify the examination of captured data. The monitored file can be printed or exported to a tab-delimited file.

## *Bus Monitor versus Sequential Monitor*

### Do You Have a Sequential Monitor?

Several Ballard Technology, Inc. products are capable of hosting multiple avionics protocols and multiple 1553 buses (channels), and therefore include a Sequential Monitor in the Hardware Explorer tree in addition to the Bus Monitor (BM). If you have a 3G hardware device or older, the Sequential

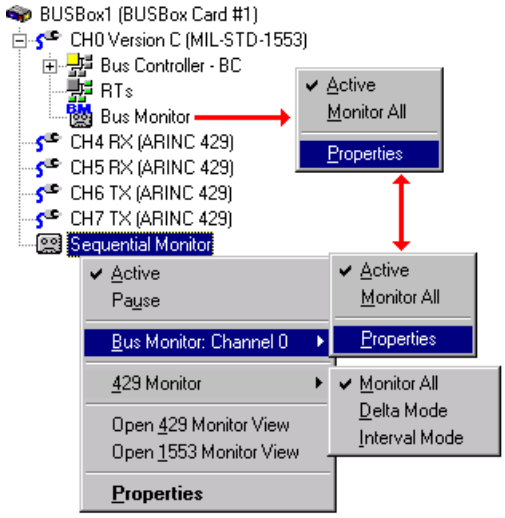


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Monitor does not exist in the hardware tree. The BM and the logging controls are available from the CoPilot version 6 1553 Ribbon tab.

### What Is a Sequential Monitor?

The Sequential Monitor is a high-level software tool capable of monitoring multiple databus channels. Its icon in the hardware does not represent a physical terminal (the Bus Monitor icon does represents a 1553 terminal, whether real or simulated by CoPilot). When two or more protocols are serviced through a single Sequential Monitor, the record is displayed through multiple Monitor Views to accommodate the differences in data content. For example, in the figure below, the Sequential Monitor is collecting data from ARINC 429 and MIL-STD-1553 buses, and the context menu has commands to open both a 429 Monitor View and a 1553 Monitor View.




### How Does the Bus Monitor Relate to the Sequential Monitor?

If you have both monitors, the Bus Monitor controls the flow of 1553 data into the Sequential Monitor. For example, if you deactivate the Bus Monitor, the Sequential Monitor will not collect and 1553 data. Therefore, options such as capture filters are located in the Bus Monitor properties window. For convenience, the Bus Monitor menu and properties dialog are accessible from the Sequential Monitor in a submenu (see figure above).

## *Concurrent Monitoring*

When the Bus Monitor and Bus Controller are simulated concurrently on a 1553-3C board, the Monitor record is constructed from internal message buffers associated with BC commands. This approach is used to give priority to processing the BC schedule.

Although all the message records are recorded in the Bus Monitor file, it is important to understand how the process works.

### Special Considerations

- **Bus Connection—**The presence of valid messages from external terminals demonstrates that the board is connected to the 1553 bus and the transceiver is working properly. Since data associated with the BC or RTs simulated by CoPilot are written directly to the Monitor from



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

internal buffers, these messages will appear in the Bus Monitor record even if the 1553 board is not connected to the bus.

- **Error Detection—**the decoder on the 1553 board detects errors. Therefore, all errors originating with external terminals are accurately recorded. Error injection and detection between simulated terminals is not shown since these messages are not processed through the decoder.

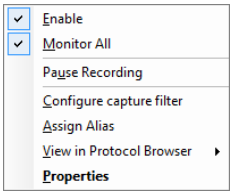
Note: Only boards with level C and higher 1553 functionality are capable of concurrent monitoring.

## Bus Monitor Configuration

The CoPilot 1553 Bus Monitor services are controlled through the monitor context menu. Although the functions are the same, there are slight differences in organization between the context menu for single and multi-protocol boards (see *Bus Monitor versus Sequential Monitor* on page *169*).

### Bus Monitor Context Menu

Ballard Technology 1553 boards (that do not contain a Sequential Monitor) display the 1553 monitor functions in the Hardware Explorer tree in the form of a Bus Monitor icon. Right click the Bus Monitor object in the tree to access the monitor context menu.




*Menu for a Bus Monitor where no Sequential Monitor is present*

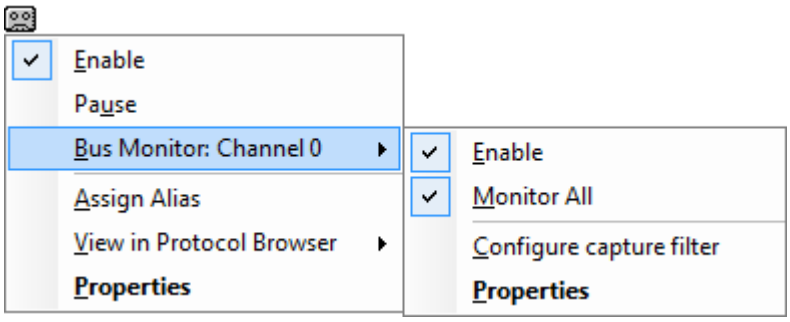
- **Enable** turns the Bus Monitor on or off
- **Monitor All** overrides monitor selections at the message and RT level and causes all messages to be monitored
- **Pause** places monitor in a pause condition. Pause may be set before or during the run
- **Configure capture filter** allows the user to set recording filters used to limit the recording size
- **Assign Alias** associates a string name with this object for use in scripting
- **View in Protocol Browser** in version 5 opens this object in one of the Protocol Browser pane windows
- **Properties** provides access to Bus Monitor properties

### Sequential Monitor Context Menu

Since the Sequential Monitor represents the collection of data across all channels (for example, 1553 and 429), the menu commands are organized accordingly. Commands that affect the Sequential Monitor as a whole are in the main menu, commands that affect the flow of 1553 data are contained in a Bus Monitor submenu (and 429 monitoring commands are in a submenu if 429 channels are present).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




- **Enable** turns the Sequential Monitor on and off
- **Pause** halts monitor activity until unpaused (collection from all channels of all protocols are paused together)
- **Bus Monitor** opens a submenu with the 1553 Bus Monitor context menu commands (see first figure above)
- **429 Monitor** opens a submenu with commands that affect the collection of ARINC 429 data only
- **Assign Alias** associates a string name with this object for use in scripting
- **View in Protocol Browser** in version 5 opens this object in one of the Protocol Browser pane windows
- **Properties** opens the Sequential Monitor properties page

## *Monitor Data Collection Modes*

### Where the Data Goes

When CoPilot collects data for a Sequential Monitor, data is collected using MonData files as described above. As records are collected, the count displayed in the status column of Hardware Explorer is updated to reflect the count of items recorded to each currently connected MonData file and the sequential monitor item displays the total count for the current run. The records in MonData files can then be viewed and analyzed using a 1553 Monitor View. The 1553 Monitor Views can be saved, reopened, modified, and shared with other projects.

### Monitor Collection Modes

That default response can be changed by the user by selecting a different Data Collection mode:

- **Append to Current**—New data will be added to the monitor buffer each time the CoPilot simulation is run, thus accumulating until the monitor site is saved or overwritten
- **Create New Recording**—A 1553 monitor data (MonData) file will be created each time CoPilot is run to capture the data until the simulation is stopped
- **Overwrite with Prompt**— A message will appear (with the number of records currently in the monitor buffer) asking for confirmation to run the card and overwrite those records (see figure) each time the project is run (simulation started). If accepted, previously recorded data in the



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

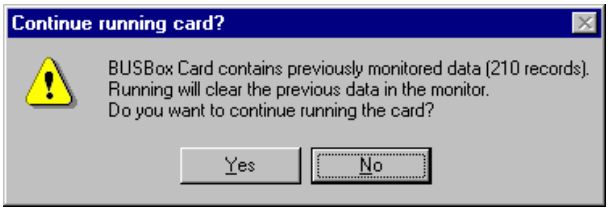
monitor capture are erased. The new data overwrites the records in the 'connected' monitor data log (MonData1) each time CoPilot is run.

| Comparison of Data Collection Modes | | | |
|---|---|---|---|
| | **Overwrite with Prompt** | **Create New Recording** | **Append to Current** |
| *Description* | Overwrite existing records (on input from user) | A new MonData file is created | Add new records to existing file |
| *User-input dialog* | Yes | No | No |
| *Monitor buffer in hardware tree* | Records are erased and counter is reset to 0 | Creates new (active) MonData file and counter set to 0 | Records are pre-served and counter increments |
| *Monitor View window* | Uses active Monitor View window (if present) | Monitor Views are unaffected | Uses active Monitor View window (if present) |

*Table of Sequential Monitor data collection modes*

**Warning**: Overwritten data cannot be recovered.

In the default mode, a new view is created each time to prevent the possibility of data loss. If overwriting the data, then the following dialog will appear when the CoPilot Run button is pressed (if the monitor buffer contains records from a previous run). The dialog is intended to minimize the accidental loss of data.



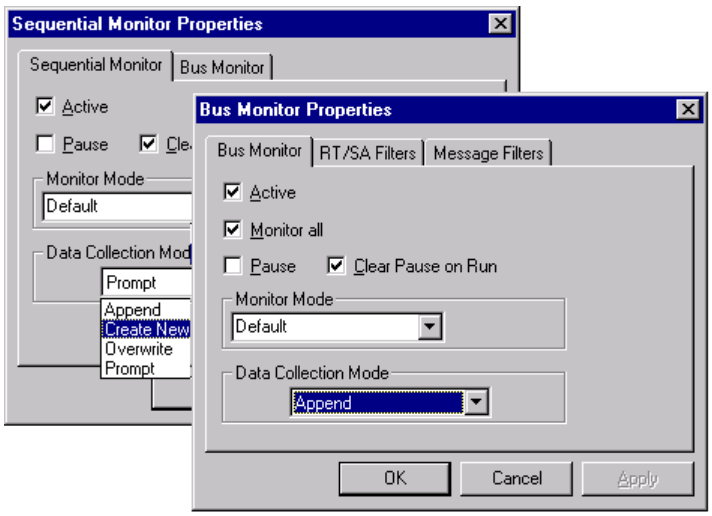

*Prompt to overwrite previously recorded data when in overwrite mode*

To change the monitor collection mode:

1. Right click on the Bus Monitor (or Sequential Monitor) icon in the Hardware Explorer tree and select **Properties** from the context menu
2. In the Data Collection Mode frame, click the down arrow to expand the list and select a data collection mode (see figure)
3. Click **OK** to apply the changes and close the Properties dialog



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




### Sequential Monitor Considerations

The specified Data Collection mode affects all channels being recorded by the Sequential Monitor. For example, if a Sequential Monitor is connected to a hardware device monitoring both ARINC 429 and MIL-STD-1553 channels, the record count is the total number of records for both.

## *Monitor Recording Modes*

The monitor can be configured to run continuously or for a specified length of time then is automatically set into the pause state. (If you have a Sequential Monitor, the recording mode affects all channels and protocols being recorded by the Sequential Monitor.) Un-pausing (resuming) the Sequential Monitor while running will again collect data for the specified time then pause again.

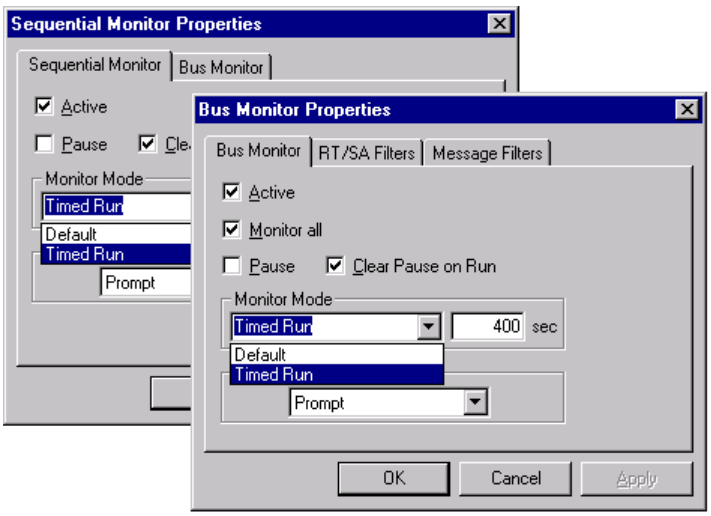
- **Default**—The monitor will run continuously when CoPilot is in simulation mode
- **Timed Run**—The monitor will run until the specified time period is reached. This option can be useful when using scripting technology to only collect data around a triggered event such as a value out of range.

To change the monitor recording mode:

1. Right click on the Bus Monitor (or Sequential Monitor) icon in the Hardware Explorer tree and select **Properties** from the context menu
2. In the Monitor Mode frame, click the down arrow to expand the list and choose **Default** mode or **Timed Run**
   - If you chose Timed Run, enter the number of seconds in the text box provided (see figure)
3. Click **OK** to apply the selected mode and close the window



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




Note: The Pause and Clear Pause on Run options can explicitly pause and restart the monitor independent of the recording mode.

## *Monitor Capture Filters*

Bus Monitor can record all databus traffic. Recording heavily loaded buss can result in large log files with tens of thousands of records in just a seconds. The more heavily loaded the bus(es) are, the more host processing is required to capture a record of the entire history. However, capture filters can limit the monitor file to targeted data samples, thus reducing data overload and facilitating data analysis.

Note: Enabling capture filtering limits the data that is recorded. Unlike display filters where records are hidden, messages that are filtered out are not recorded.

### Many Ways to Filter

There are several ways to limit the size of the monitor record by targeting the data you wish to collect or capturing bus traffic at controlled times.

- Tag individual messages/SA in the Hardware Explorer tree for capture
- Specify RTs, SAs, mode codes, or messages for monitoring in the Bus Monitor properties window
- Control the recording session from the Monitor View window
- Configure the Bus Monitor for a timed run so that recording halts after a specified period of time
- Toggle monitor recording with trigger conditions using Scripting (CoPilot Professional only)

### Capture Filters versus Display Filters

CoPilot can filter data before or after capture to the Bus Monitor. Capture filters and display filters are defined in exactly the same way; the difference is in the application. Capture filters are useful for limiting the size and scope of monitor files. Display filters restrict the display of messages in the Monitor View window. They do not affect the content of the monitor file.

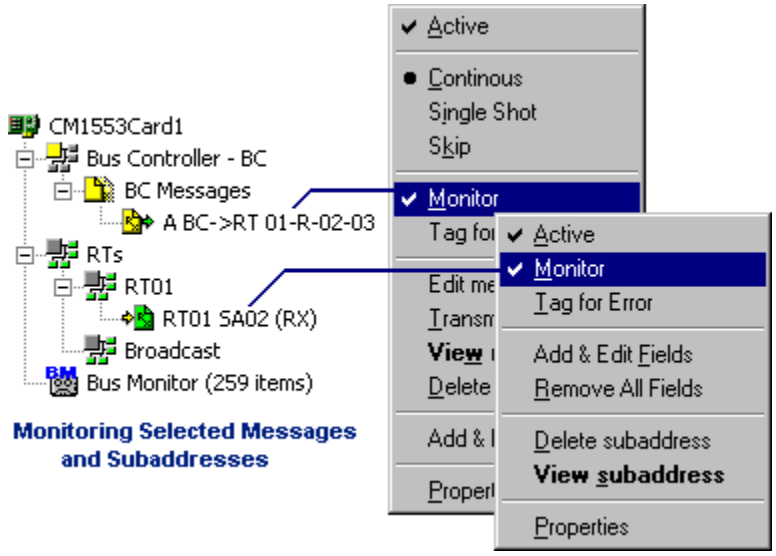


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Capture Individual Messages and Subaddresses

Capture filters can be specified through the 1553 Bus Monitor properties dialog or through individual items in the Hardware Explorer tree.

If the Monitor All option is not selected in the Bus Monitor properties, messages can be filtered selectively through the context menu of each BC message and RT subaddress.




To tag a message or subaddress for selective monitoring, right click the object to access the context menu, and then select the Monitor option.

### Capture all Messages

Individual filter conditions can be overruled through the monitor context menu.

- If Monitor All is selected, monitor specifications at the message and RT subaddress level are ignored and all messages on the 1553 bus will be recorded.
- If Monitor All is not selected, monitor capture is based on the monitor selection of individual messages and subaddresses.

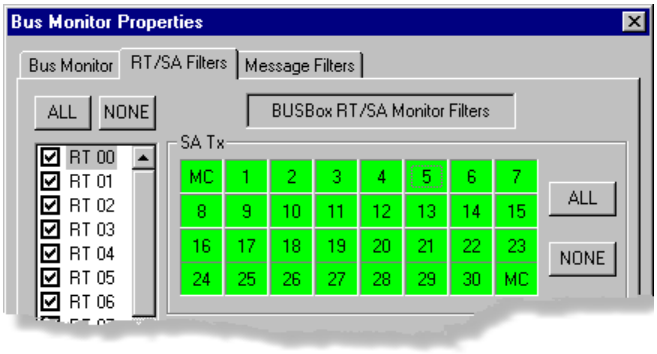
### Remote Terminal and Subaddress Filtering

If the Monitor All option in not selected, the filtering options on the RT/SA Filters tab of the Bus Monitor properties page become available. The capture of 1553 bus traffic can be limited to selected remote terminals and subaddresses. All RTs and SAs are selected by default.

To filter out an RT, clear its check mark from the list box. To filter out certain transmit or receive SAs, highlight the RT in the list box, then click the key for the SA you wish to filter out. The key will become raised and gray to indicate that it is deselected. You can select or deselect all receive or transmit SAs with the ALL and NONE buttons.

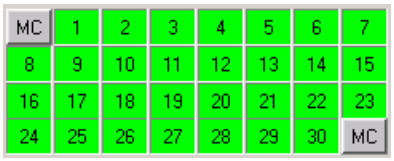


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

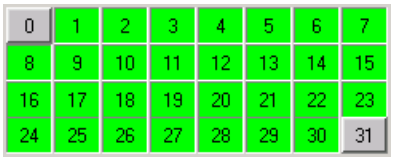



### Mode Code Filtering

If both SA 0 and SA 31 mode codes are legalized in the properties page for the 1553 card (or channel), then those subaddress numbers are reserved for mode codes. Clicking on one mode code key deselects both.

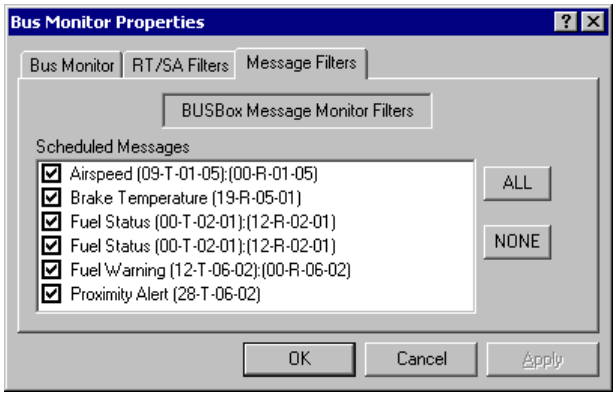



If mode codes are illegalized, SA 0 and 31 revert to normal subaddresses, and may be selected or deselected individually.




### Message Filtering

The 1553 Bus Monitor can also ignore selected messages. This applies to messages in the BC message list in the Hardware Explorer tree. To filter out a message, clear its check mark. You can select or deselect all messages with the ALL and NONE buttons.




To exclude a message from an external terminal on the bus, filter out the SA that is receiving the message, or recreate the message in the BC message list and tag it for exclusion in this dialog.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: If monitor all is toggled on, these filter definition windows are unavailable.

#### Filter through Recording Controls

Recording controls in a 1553 Monitor View window (that is linked to the active monitor) provide a convenient way to capture data samples at strategic moments in the 1553 session. The Run, Stop, and Pause controls are part of the 1553 Monitor View window and can directly control the Bus Monitor and the monitor record. By pausing and resuming the Bus Monitor, a segment of bus activity is not captured to file. This helps eliminate data overload and can set the stage for the segmentation of monitor records, to better facilitate data analysis.

#### Controlling the Recording Session

To control the recording of bus traffic, launch a 1553 Monitor View window from the Bus Monitor (or Sequential Monitor if it is present). The 1553 Monitor View title bar will name the board that it is actively monitoring. If you open another view, you will break the link. If you wish to reconnect a the monitor to an open 1553 Monitor View window, drag and drop the monitor icon onto the view window (this action will erase any data currently in the view window).

Once you have a linked Monitor View, you can control the collection of data using buttons in the Monitor View tool bar:

- The **Pause** button halts the recording of data with an opportunity to resume. Click the Pause button a second time to restart the monitor. The additional data will be appended to the same file.

## Saving Monitor Data

In normal operation, the CoPilot monitor collects data on an active bus when the Run command is executed. Monitor data is extracted directly from the active databus and is saved to a temporary host file, whether or not the Monitor View display window is visible. If the simulation is stopped, then restarted by pressing the Run button, this data may be overwritten (depending on the specified Data Collection mode). There are two ways of preserving monitored data:

- Open the 1553 Monitor View window, and save it to file. When it is reopened later, all of the records are preserved.
- Open the 1553 Monitor View window and export the data as a text or ASCII file to another application, such as Excel.

### *Saving Monitor Files*

#### Managing Monitor Files

All display windows are project components. Like Engineering Views, Control Views, and Data Generator Views, each Monitor View is defined in a separate file and placed in the project folder. CoPilot automatically names monitor files as they are created. Names follow the pattern “1553MonViewN.CVW” (where N is the first unused number in the project folder). The user can change the name, however, through File menu options.

Users can retain as many monitor files in the folder as they wish. All currently open monitor files are listed under the Windows menu and in the Project Explorer. Other monitor files in the project can be opened using the File | Open Hardware or View… command.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### File Menu Options

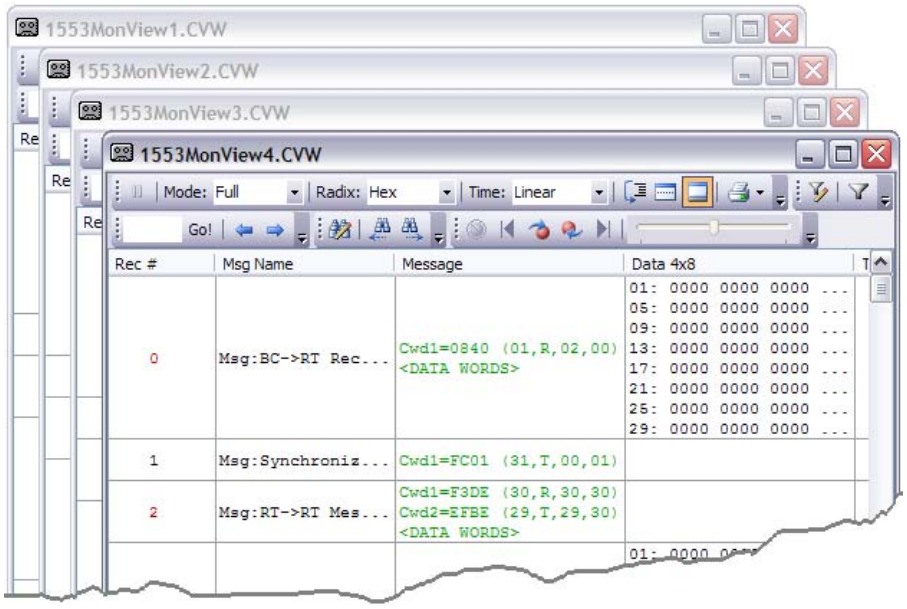
Monitor Views can be accessed and saved through the File menu or CoPilot tool bar. Active Monitor files may be saved simultaneously through the Save Project command. Monitor files can also be saved selectively through the Save button or File | Save command.

## *Saving Segmented Monitor Files*

### Create Multiple Viewing Windows

The Monitor View window, like other windows in the display pane, is a container. When the bus is active and the Monitor is recording, incoming and outgoing messages that match the filter criteria will be placed in that open display container.

Double clicking the Bus Monitor BM branch on the Hardware Explorer tree activates a Monitor View window. If the Monitor View window is open and the bus is active, CoPilot processes that as a request for a new Monitor View window. When the user responds to the confirmation request, the Bus Monitor link with the active Monitor View window is broken and new messages are routed to the new container.




Consequently, by double clicking the Monitor branch when the bus is active, Monitor data can be separated into a sequence of Monitor files. Thus, data related to special situation can be viewed and saved in separate files.

### Combine with Recording Controls

This segmenting technique works well in combination with the Pause command. Use pause to halt a data-recording segment, and then create a new Monitor View by double clicking the Bus Monitor icon. Recording resumes immediately and each message sequence is preserved in the separate files and view windows.

### Save Segments Selectively

All active Monitor Views created in a session are listed in the Window menu and may be saved simultaneously with other project files through the Save



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Project command. Alternatively, useful segments can be saved individually through the Save command. If you attempt to close a Monitor View window through the Close button without previously saving the data, a warning message will appear.

### *Saving Data from Heavily Loaded Buses*

#### Monitor Data Can Accumulate Quickly

It takes 70 microseconds to send a one-word MIL-STD-1553 message with status word. Thirty-two word messages take 690 microseconds. At that rate, you would accumulate from 86 and 857 thousand records per minute. At this rate monitor files can quickly become large.

#### Problems With Heavy Databus Loading

The two principle challenges associated with monitoring heavily loaded MIL-STD-1553 databuses are (1) establishing a focus on meaningful data and (2) data transfer capacity between the Ballard 1553 board and the host file. Too much data usually makes data analysis and discovery more difficult. With CoPilot 1553, users can reduce the time required to process the monitor file through terminal, message, and subaddress capture and display filters.

Data transfer from the MIL-STD-1553 board to host memory can be a bigger challenge since the transfer from the board to the host is controlled by the operating system, which is subject to unpredictable interruptions of other applications. To minimize unexpected interruptions from the operating system, users should close all other applications when monitoring a heavily loaded bus.

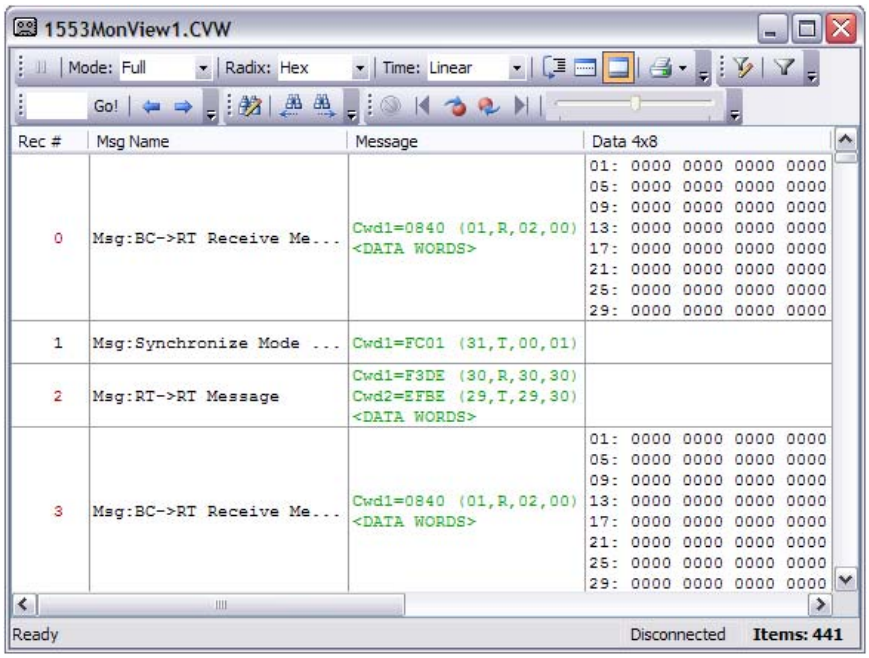
## Monitor View

#### Monitor Display Window

Bus traffic is viewed in the 1553 Monitor View window, which exists in the display pane. The Monitor View window supports several display formats, a variety of data radices, and various time-tag options. You can filter the display of data and easily move through the monitor records using navigation controls. The viewing window can be activated before or after the 1553 board is set in operation. While the simulation is running, the Monitor can be stopped, paused, and run (independent of the bus) with controls in the Monitor View window. After a run, individual messages may be modified.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




#### Using 1553 Monitor View Windows

More than one Monitor View window may be open at one time, but only one window at a time can be linked to the data from a specific MIL-STD-1553 bus. The title bar at the top of the window identifies which card and which channel the Monitor View is receiving bus traffic from. Each view may be individually saved to file.

#### Analyzing Monitor Data

> **Note**: *CoPilot 6 introduced the Ribbon Interface and simplified the Protocol Browser with the new Activity View. See New in CoPilot 6.*

CoPilot's Monitor View can be used to display data from a previously recorded monitor file or display messages as they are being received. In both cases, monitor files can be analyzed using powerful display, search, filter, print, and export options. Statistics from the data captured is displayed in the Protocol Browser in version 5 or the Activity View in version 6.

#### Data Source for Playback

The data source for hardware or software playback is a Monitor View file. When a Monitor View window is selected as a playback data source and playback mode is engaged, the controls in the Playback frame become active, and some of the other Monitor View buttons take on playback functionality. The following sections only address the Monitor View functions available in CoPilot edit mode and simulation mode. Please refer to the sections on hardware playback and software playback to learn more about the Monitor View functions available in those modes.

## *Accessing the Monitor View Window*

#### Creating a New Window

The 1553 Monitor View window is accessed from the monitor icon in the Hardware Explorer tree.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

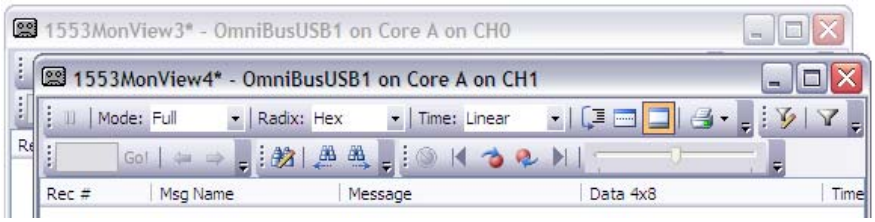
To open a new 1553 Monitor View:

- If you have a Sequential Monitor—Right click on the Sequential Monitor icon in the Hardware Explorer tree and select **Open 1553 Monitor View** from the context menu
- If you have only a Bus Monitor—Right click on the Bus Monitor icon in the Hardware Explorer tree and select **Open** from the context menu

The view window will open in the display area and appear in the Project Explorer. It can be saved to file and reopened in any CoPilot project.

### Linked to the Monitor

When a new view is opened from a monitor in the Hardware Explorer tree, it is "linked" to that monitor and the name of the board is displayed in the title bar (see figure). Because of this link, data is added to the view window as it is collected by the monitor and the recording controls in the Monitor View window directly control the monitor in the Hardware Explorer tree.

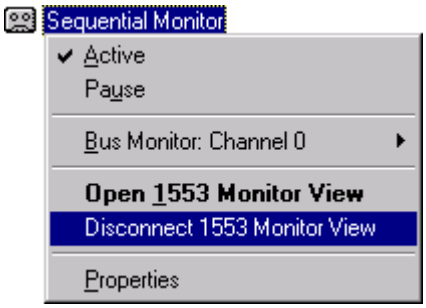



### Independent (Unlinked) Views

As soon as the link is broken between a 1553 Monitor View window and the monitor in the Hardware Explorer tree, data is no longer added, the buttons no longer control the monitor, and the "Monitoring" notice is removed from the title bar.

To disconnect a view from its monitor:

- Open another Monitor View window from the same monitor, which becomes the linked view, and the link to the first Monitor View is disconnected.
- Also, if you have configured the data collection mode of the monitor for Create New View, a new 1553 Monitor View will be created every time CoPilot enters simulation mode, thereby disconnecting the link with the previous view.
- Alternatively, right click on the monitor icon and choose **Disconnect 1553 Monitor View** from the context menu (see figure)






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To reconnect a monitor to a view window:

- Drag the monitor icon from the Hardware Explorer tree onto the 1553 Monitor View window (if there are records in the window, you will be prompted to confirm this action, which will overwrite all existing records)

## *Monitor View Context Menu*

Right click in the column area of the Monitor View window to access the Monitor View context menu. Alternatively, you can access this menu from the CoPilot menu bar (this menu merges with the CoPilot menu bar when the 1553 Monitor View window has focus). Each of the commands described briefly below are covered in more detail in later sections.

**Monitor View Menu**



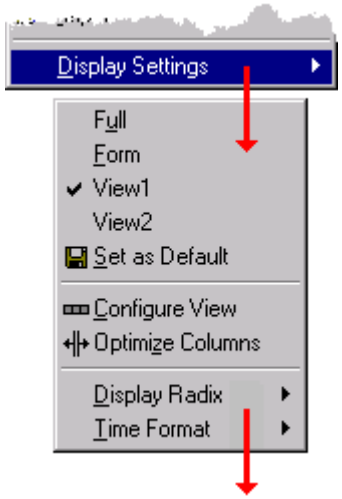

- **Bookmark Record** tags the selected monitor record with a bookmark icon
- **Edit Bookmark Comment** allows a text comment to be associated with a bookmarked record
- **Breakpoint Record** tags the selected monitor record with a breakpoint icon
- **Set Start/End Index** establishes the index for software or hardware playback
- **Set as Next Message** establishes the resume point when software or hardware playback is paused
- **Edit Message** opens the Monitor Record Editor
- **Plot Data** creates a strip chart that maps all recorded data points for each selected field
- **Print** all or selected records
- **Export** allows you to export all or selected records or fields to the file format of your choice
- **Import** allows previously exported .RAW monitor files to overwrite or append the current Monitor View records

Several submenus are nested within the main menu. Point to a submenu name with the mouse to expand it.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Display Settings Submenu




- **Full** displays all message information in a multi-line display, including all data words in the chosen radix
- **Form** displays messages in fixed format so that changes in command word, error conditions, or data are easily detected
- **CustomUser1/CustomUser2** can be configured by the user (using the Configure View command below)
- **Set as Default** saves the currently selected view (from the four above) as the default display format when a new Monitor View is created
- **Configure View** allows CustomUser1 and CustomUser2 to be customized by the user
- **Optimize Columns** expands or contracts the width of each column to fit its contents

### Display Radix Submenu

- **Binary** displays data in a binary radix
- **Octal** displays data in an octal radix
- **Hexadecimal** displays data in a hexadecimal radix
- **Decimal** displays data in a decimal radix

### Time Format Submenu

- **Linear** displays time-tags sequentially
- **Delta** displays the increment between time-tags
- **Linear (sec)** displays time-tags sequentially in units of seconds
- **Delta (sec)** displays the increment between time-tags in units of seconds
- **System** displays the time-tags in hours, minutes, and seconds based on the host computer clock settings

## *Monitor View Tool Bar*

Customize the properties of the Monitor View window through context menu commands, buttons, and the dialogs launched by certain buttons.




### Display Settings: Modes, Radix, and Time

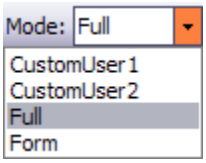
There are multiple settings that change how monitored data is displayed. These settings include options for modifying the display mode, radix settings, and time format.



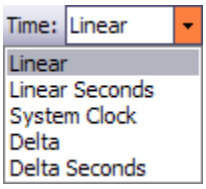



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Changing the mode setting changes how records are displayed. The display modes include CustomUser1, CustomUser2, Full, Form, and Statistics. The display settings of a Monitor View are controlled from the monitor view toolbar or via the monitor view context menu.




The radix format changes the display between hex, octal, binary, and decimal. The time format changes the time-tag format to any of the following: linear, linear (seconds), delta, delta (seconds), and system clock.




### Display Filters

The display of records can be limited by certain criteria. Use the buttons in the Monitor Filter toolbar to set criteria and toggle filtering.



### Navigation Controls

Monitor files can quickly become large with heavily loaded bus traffic. Use the Search, Step, and Go To Message controls to navigate through the Monitor View display.




### Pause Controls

The Pause button directly controls the sequential monitor in the Hardware Explorer tree. The pause button of a view connected to hardware directly controls the pause state of the sequential monitor.

Warning: pausing the sequential monitor can pause other channels (on the same core).



### Playback Controls

The controls in the Playback frame are unavailable unless the Monitor View window is linked to a board or channel that is in hardware or software playback mode. Please see the Hardware Playback and Software Playback sections for more information on these controls.

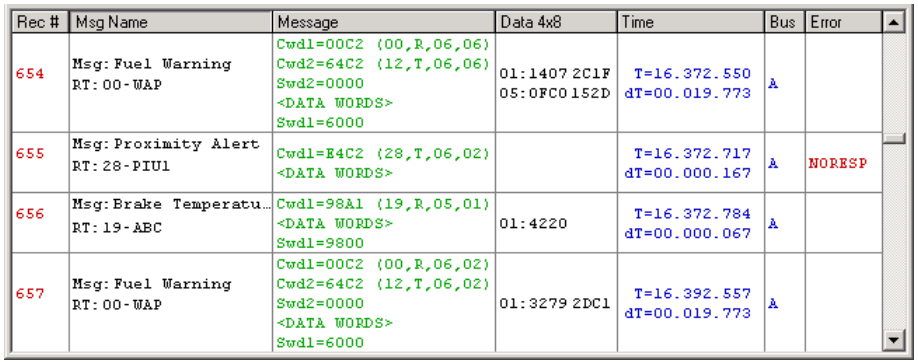




Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Monitor View Full Display

In Full view, message information is presented in a multi-line display, including all data words. The Message Name column displays the default or user-defined message and RT names. The Message column displays all the command, data, and status words in the message in order of transmission. If a message error is present, it is reported in the Error column.



| Rec # | Msg Name | Message | Data 4x8 | Time | Bus | Error |
|---|---|---|---|---|---|---|
| 654 | Msg: Fuel Warning<br>RT: 00-WAP | Cwd1=00C2 (00,R,06,06)<br>Cwd2=64C2 (12,T,06,06)<br>Swd2=0000<br><DATA WORDS><br>Swd1=6000 | 01:1407 2C1F<br>05:0FC0 152D | T=16.372.550<br>dT=00.019.773 | A | |
| 655 | Msg: Proximity Alert<br>RT: 28-PIU1 | Cwd1=E4C2 (28,T,06,02)<br><DATA WORDS> | | T=16.372.717<br>dT=00.000.167 | A | NORESP |
| 656 | Msg: Brake Temperatu…<br>RT: 19-ABC | Cwd1=98A1 (19,R,05,01)<br><DATA WORDS><br>Swd1=9800 | 01:4220 | T=16.372.784<br>dT=00.000.067 | A | |
| 657 | Msg: Fuel Warning<br>RT: 00-WAP | Cwd1=00C2 (00,R,06,02)<br>Cwd2=64C2 (12,T,06,02)<br>Swd2=0000<br><DATA WORDS><br>Swd1=6000 | 01:3279 2DC1 | T=16.392.557<br>dT=00.019.773 | A | |

### Data Words

Data words are presented in 4 columns and (up to) 8 rows. The first four words are in the first row; the second row starts with word 5, the third row with word 9, and so on. Data may be presented in hexadecimal, decimal, octal, or binary format. Click the middle button in the Display frame at the top of the Monitor View window to toggle the radix.

### Time-Tag

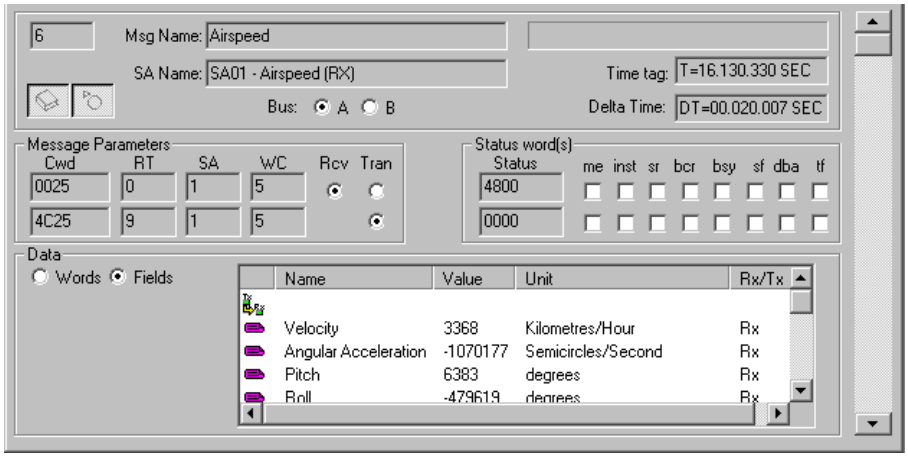
The linear and delta time-tag values may also be displayed in units of seconds by toggling the time-tag button in the Display pane at the top of the Monitor View window. The System setting displays the time-tags in clock time based on the host computer settings.

## Monitor View Form Display

In Form view mode, messages are displayed in fixed format so that changes in command word, error conditions, or data are easily detected. The record number is displayed in the upper left of the form, followed by the message name. Use the buttons below the record number to add a bookmark or breakpoint. Move between records with the navigation buttons or use the scroll bar. Drag the scroll box to view a tool tip display of record numbers. Release the scroll box when you reach the desired record.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




### Time-Tag

Two time-tag values are displayed: linear and delta. Both are displayed in units of seconds. The time-tag button in the Display pane at the top of the Monitor View window does not affect the time-tag display in Form view.

### Message Parameters

The command word, RT address, subaddress, word count, and receive or transmit mode of the selected message are displayed in the Message Parameters frame. If the message is an RT-to-RT message, the second RT is described below the first.

### Status Words

The status words are displayed across from the RT to which they apply. If status bits are set, they are noted in the corresponding check boxes.

### Data Words

When the Words option button is chosen in the Data frame, the data words for the selected message are displayed. The radix can be toggled between hexadecimal, decimal, and octal (but not binary) by pressing the middle button in the Display frame at the top of the Monitor View window.

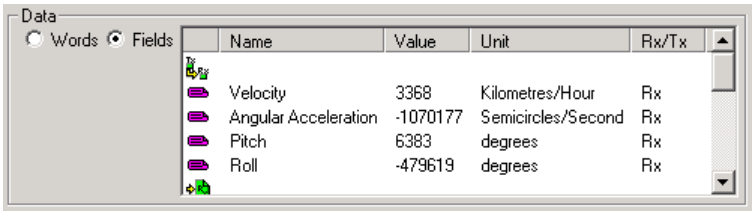



### Data Fields

Choose Fields in the Data frame to display the message and RT subaddress fields. The icon identifies the message or RT to which the following fields belong.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




The field name, value, engineering units, and receive or transmit mode are displayed.

## *Monitor View User-Defined Displays*

CustomUser1 and CustomUser2 may be customized by the user. In these views, each monitor message is presented in a single line, with types of information organized by columns. Limiting the display of certain columns is a way to filter and customize the display of monitor records. Once you have defined and named a custom view, you may save as the default.

| Rec # | Time | Msg Name | Data 2x16 | Word Count | TA # | SA # | Rcv | Cwd1 | Cwd2 | Swd1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16.067330 sec | Airspeed | 01: 006450 125637 03: 014357 127175 05: 000010 | 5 | 0 | 1 | Rx | 0025 | 4C25 | 4800 |
| 1 | 16.067557 sec | Fuel Warning | 01: 000727 000000 | 2 | 0 | 6 | Rx | 00C2 | 64C2 | 6000 |
| 2 | 16.067724 sec | Proximity Alert | 01: 000524 000000 | 2 | 28 | 6 | Tx | E4C2 | | E000 |
| 3 | 16.067841 sec | Fuel Status | 01: 113053 | 1 | 12 | 2 | Rx | 6041 | 0441 | 0000 |
| 4 | 16.067988 sec | Brake Temperature | 01: 173021 | 1 | 19 | 5 | Rx | 98A1 | | 9800 |
| 5 | 16.087338 sec | Fuel Status | 01: 113053 | 1 | 12 | 2 | Rx | 6041 | 0441 | 0000 |

### View Setup Dialog

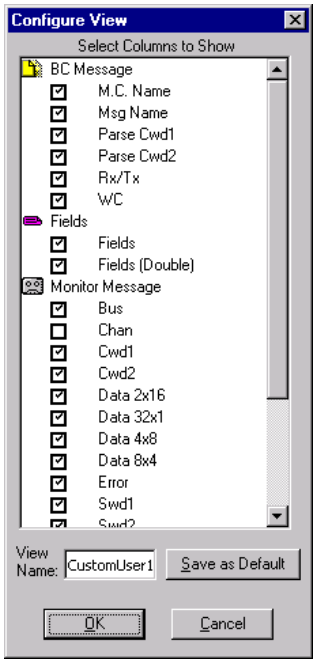
To define a custom display mode, toggle the display mode until CustomUser1 or CustomUser2 is selected. The options in the Configure View dialog will apply to the currently selected view.

To configure CustomUser1 or CustomUser2:

1. Be sure that CustomUser1 or CustomUser2 is selected as the display style
2. Right click in Monitor View, point to Display Settings to expand the submenu, and choose Configure View to open the Configure View dialog (see figure)
3. Configure the view by selecting columns to display and, if you wish, rename the view
4. (Optional) Click the **Save as Default** button to save the configuration as the default (for this one of the two user-configurable views)
5. Click **OK** to apply the configuration settings to the specified view and close the dialog



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*Sequential Monitor View column selection*

### View Name

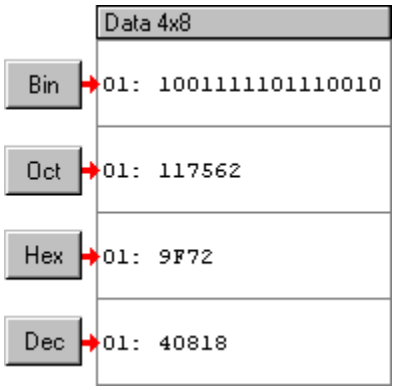
The text box at the bottom of the Configure View dialog allows you to customize the name of CustomUser1 or CustomUser2. The new name may be up to 8 characters long. This new name will appear in the context menu and on the display mode button (with Full and Form).

## *Monitor View Radix*

Click the radix drop list from the toolbar (see figure) of the Monitor View window to alter the radix of displayed data.




In CustomUser1, CustomUser2, or Full mode, the choices are hexadecimal, decimal, binary, and octal (see figure). In Form display, this button toggles between hexadecimal, decimal, and octal.




*Examples of the various display radix options*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

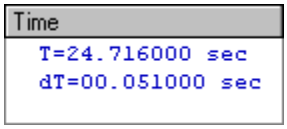
## Monitor View Time-Tag

Click the time drop list from the toolbar (see figure) of the Monitor View window to toggle between Linear, Linear (seconds), System Time, Delta, and Delta (seconds) to define the time-tag for the current display.




*Standard MIL-STD-1553 Monitor View toolbar*

Linear (seconds) and Delta (seconds) are displayed in units of seconds for easy comparison of time-tag values and for exporting data values for analysis and graphing.




*Linear (seconds) and delta (seconds) time-tags in Full Mode view*

### Linear and Delta Time Modes

The linear and delta time display modes display the time-tag in the following format:

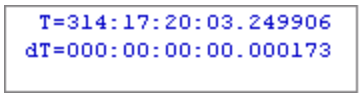
**000:00:00.000000 ddd:hh:mm:ss.ssssss**

ddd – days

hh – hours

mm – minutes

ss.ssssss – seconds.(3 digits of milliseconds)(3 digits of microseconds)




*Linear and delta time-tage in Full Mode view*

Note: The IRIG timer can be initialized with a seed value or can use the system time.

## Monitor View Display Filters

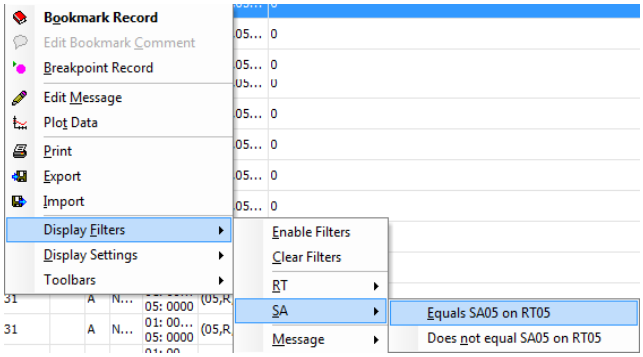
Display filters may be used to limit the display of captured data. Once defined, the display filter can be enabled or disabled through the On/Off button (see figure).



Display filters for individual RT and SA items can be modified to filter for or filter out the selected item from the context menu as shown in the image below.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*Sequential monitor view display filter configuration from the context menu*

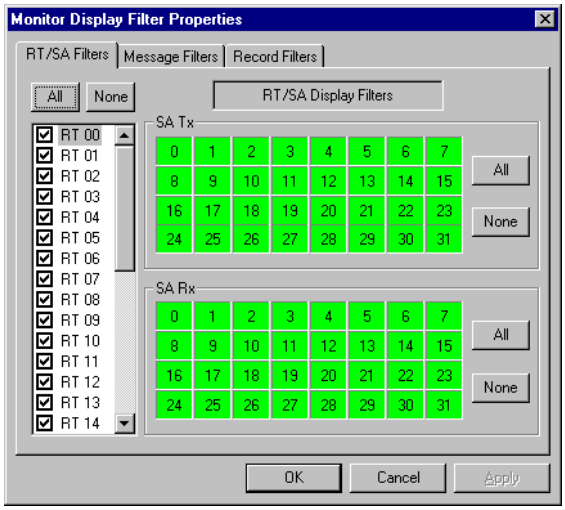
To define filter criteria:

- Click the **Edit** button in the Display Filter frame in the Monitor View window to open the Monitor Display Filter Properties window (see figure)

Note: All of the options in the three tabs described below are combined to define the display criteria. Therefore, if you deselect an RT on the first tab, any settings that apply to that RT on the next two tabs are irrelevant: nothing will be displayed for that RT.

### RT/SA Display Filters

The display of Monitor View records can be limited to selected remote terminals and subaddresses. All RTs and SAs are selected by default.




*Display filter property pages*

To filter for RTs:

- In the RTs listbox on the left side of the RT/SA Filters tab, specify the RTs you wish to display using the All or None buttons or select RTs individually by their check box



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To filter for SAs:

- In the SA area on the right side of the RT/SA Filters tab, specify the SAs you wish to display using the All or None buttons or select SAs individually by clicking their button (depressed green = selected; raised gray = deselected)

To filter for mode codes (or SA0 and SA31):

- If both SA0 and SA31 mode codes are legalized for the 1553 databus, then click on one mode code key to deselect both (if mode codes are illegalized SA0 and 31 revert to normal SAs and may be selected or deselected individually)

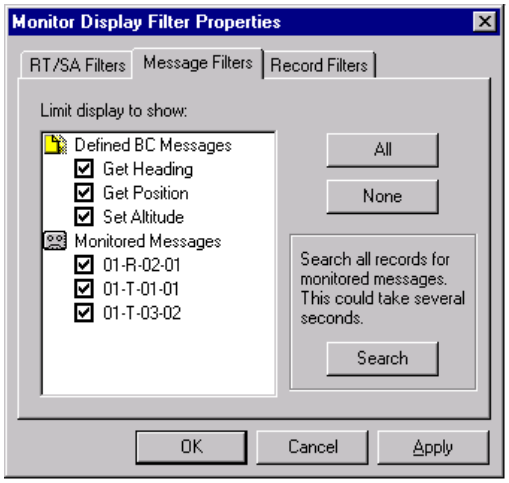
### Message Filters

Only those messages tagged with a check mark will be displayed (all messages are selected by default).

To filter for a message(s):

1. Click on the Message Filters tab
2. Click the Search button to load the monitored messages (see figure)
3. Click the All or None buttons or individually select or deselect messages with the check boxes

If a message is defined in the BC schedule and was also recorded by the monitor, deselecting it in one category will automatically deselect it in the other category.

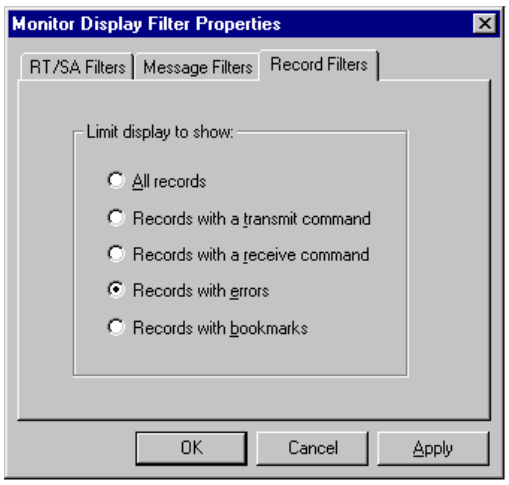



### Record Filters

The Record Filters tab allows you to choose from a list a criteria. Remember, the settings for these three tabs are combined to define the filter criteria. For example, if you wish to search for the errors in a particular RT, choose the "Records with errors" option on this tab, then deselect all but that RT in the RT/SA tab, and make sure you have not filtered out any messages to that RT in the Message tab.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




#### Display Filters versus Capture Filters

Capture filters and display filters are defined in exactly the same way; the difference is in the application. Display filters restrict the display of messages in the Monitor View window. They do not affect the content of the monitor file. Capture filters are a function of the Bus Monitor, and are useful for limiting the size and scope of monitor files.

## *Monitor View Bookmarks and Breakpoints*

CoPilot provides two forms of marking monitor records for future reference.

- **Bookmarks** are used within the Monitor View window. If you define bookmarks as the search criteria, you can move between bookmarked records with the direction buttons in the Search frame.
- **Breakpoints** are used in hardware and software playback to pause playback at specified records. You can use the Next and Previous breakpoint buttons to navigate between records marked with a breakpoint.

Since the marks are used for different purposes, a record can be tagged with both a bookmark and a breakpoint.

To set a bookmark or breakpoint:

1. Right click the monitor record of interest to access the context menu
2. Click the **Bookmark** or **Breakpoint** option

To remove a mark, highlight the record and repeat the two steps shown above. In the Form display mode, these marks can be set (and cleared) with pushbuttons.

#### Bookmark Comments

If a record has been marked with a bookmark, a comment can be added to explain why the record was bookmarked or to record notes about that record.

To add a comment to a bookmarked record:

1. Right click on a bookmarked record and choose **Edit Bookmark Comment** from the context menu

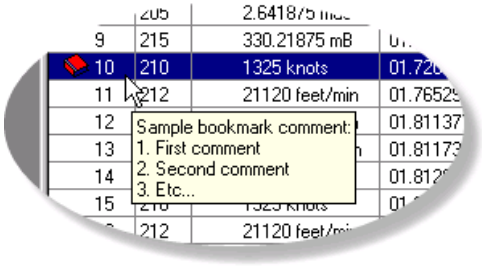


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. Enter your notes and click **OK** to close the dialog

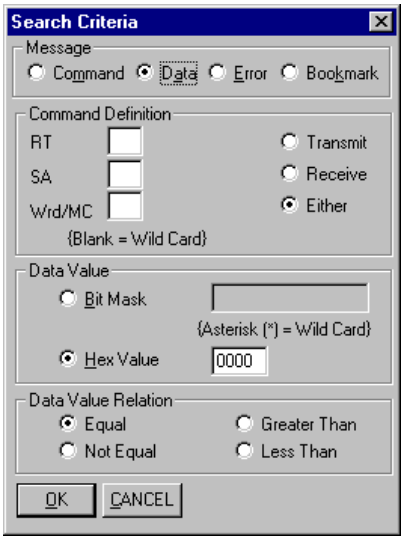
Note: If the Always Comment Monitor Bookmarks option has been selected in the CoPilot Options dialog, the Comment dialog will automatically open whenever a new bookmark is added.

To view the comment, hover the mouse pointer over the bookmark icon and the comment will appear in a pop-up display (see figure).




## Monitor View Search Options

Use the Search function to locate a specific record. First, click the Define button to open the Search Criteria dialog (see figure). Select the message category, define the criteria, and click OK.




Then, use the Forward and Back buttons to jump to the first record that meets the defined criteria.



## Monitor View Step and Go To Commands

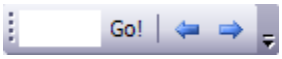
Because monitor files can become large, it can be difficult to navigate through so many records and find a particular record of interest. The Monitor View window contains powerful navigation controls to step through the records or jump to a specific message.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### Step

The Step function moves through the Monitor records incrementally from the currently selected message. Messages are highlighted as the Step buttons are engaged, visually identifying the record in focus.




Click on the buttons to move forward or backward one record.

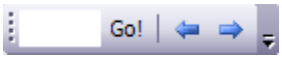
---

Note: The function of the Step buttons change for hardware and software playback modes.

---

#### Go To Message

The Go To button will automatically advance to the (index) number shown in the box. Jump to a specific monitor record with the Go To Message command. Type the message number in the text box and click the Go button. The display will advance to the specified record.




---

Note: Index numbers are based on display sequence, not the number of records in the monitor file. For example, if display filters are engaged on a file of 1000 records and only 200 messages are displayed, the "Go To" range will be 1 through 200.

---

### *Monitor View Recording Controls*

The recording controls, options, and properties in the Monitor View window do not affect the Bus Monitor's record of bus activity with one exception: the recording controls. The Run, Stop, and Pause commands directly control the Bus Monitor and the monitor record (if the 1553 Monitor View window is actively linked to the monitor in the Hardware Explorer tree).

- The **Pause**  button halts the recording of data with an opportunity to resume. Click the Pause button a second time to restart the monitor. The additional data will be appended to the same file.

By pausing and resuming the Bus Monitor, a segment of bus activity is not captured to file.

---

Note: These buttons perform a different function when CoPilot is in hardware or software playback mode.

---

### *Monitor View Message Editing*

After recording is complete, individual messages in the Monitor View window may be modified. Monitor record editing could be used to prepare a monitor file to be a hardware playback source with targeted error injection.

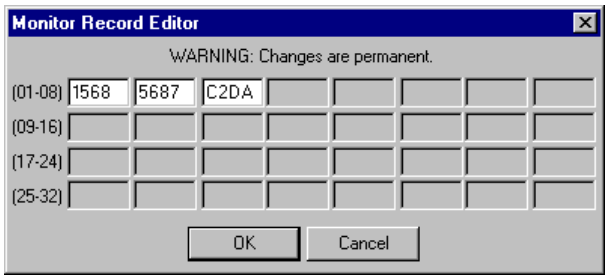
To modify a record in Monitor View:

1. Right click on the record you wish to edit and choose **Edit Message** from the context menu (not available in Form view)

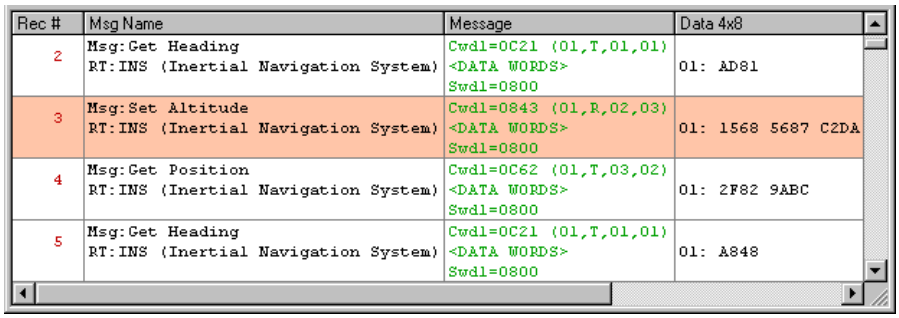


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. Enter the new data value(s) in hex in the available data cells
3. Click OK to apply the changes and close the editor (a warning dialog will ask for confirmation of this action)




Modified data becomes a permanent part of the monitor record and is saved with the Monitor View component. Modified records in the Monitor View window are marked with a special background color, in all views except Form (see figure).



| Rec # | Msg Name | Message | Data 4x8 |
|---|---|---|---|
| 2 | Msg:Get Heading<br>RT:INS (Inertial Navigation System) | Cwd1=0C21 (01,T,01,01)<br><DATA WORDS><br>Swd1=0800 | 01: AD81 |
| 3 | Msg:Set Altitude<br>RT:INS (Inertial Navigation System) | Cwd1=0843 (01,R,02,03)<br><DATA WORDS><br>Swd1=0800 | 01: 1568 5687 C2DA |
| 4 | Msg:Get Position<br>RT:INS (Inertial Navigation System) | Cwd1=0C62 (01,T,03,02)<br><DATA WORDS><br>Swd1=0800 | 01: 2F82 9ABC |
| 5 | Msg:Get Heading<br>RT:INS (Inertial Navigation System) | Cwd1=0C21 (01,T,01,01)<br><DATA WORDS><br>Swd1=0800 | 01: A848 |

## *Monitor View Data Plotting*

Users can create post-analysis strip charts that plot each and every point of recorded data (unlike Strip View, which is based on sampled data). Data Plot windows can always be accessed from a saved Monitor View window (although they cannot be saved separately).

Note: Data Plot is a CoPilot 1553 Professional feature. If monitor data was recorded with a Ballard card keyed for CoPilot Professional, then Data Plot charts can be created from that record (even if Professional mode is toggled off).

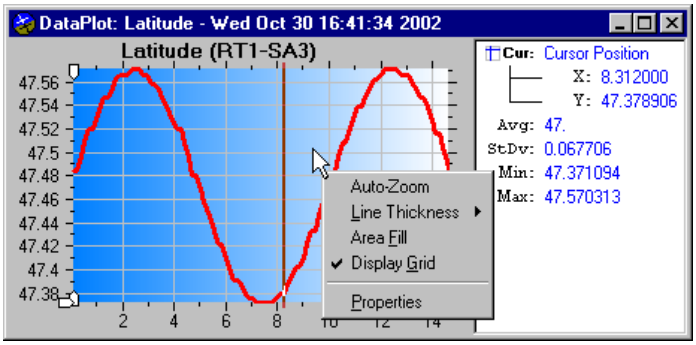
To plot data, there must be an open Monitor View window, with recorded data fields, and CoPilot must not be running (must be in Edit mode).

To plot data in Monitor View:

1. Right click in a Monitor View window and choose **Plot Data** from the context menu to open the Select Data Plot Items dialog
2. Check the box for each data field you wish to plot and click OK
3. A separate Data Plot window will open for each selected field (see figure).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




The right-hand pane displays the values at the current cursor position and a statistical analysis of the dataset.

Note: Because Data Plot computes all data, rather than data samples, these numbers can vary slightly from their equivalent in Strip View.

Data Plot configuration options are similar to those of Strip View: you can zoom in on the visible range of values and configure display properties such as line thickness or background color.

## *Monitor View Export Options*

Monitor data can be exported to Excel or other applications in several formats using the Export option.

### Choose an Export Type

There are three main types of 1553 Monitor View exports: record export, engineering units, and raw data.

- **Record Export**—exports records much as they are displayed in the 1553 Monitor View, using the same columns, data radix, time-tag format, etc. This option is useful for charting and comparing fields such as status word, errors, etc.
- **Engineering Unit Export**—exports only data fields and their time-tags in two titled columns. This format makes it easy to chart and compare data field values in Excel (or other application).
- **Raw Data Export**—exports all of the monitor content in an unformatted file. This type of file can be used in a third-party application or imported into a 1553 Monitor View window to append or overwrite existing records. See Monitor View import for additional information.

### Export the Data to File

The table below charts the steps and options for creating each of the three types of export files. The first three lines (display filters, display formatting, and record selection) are discussed in detail in the sections below.

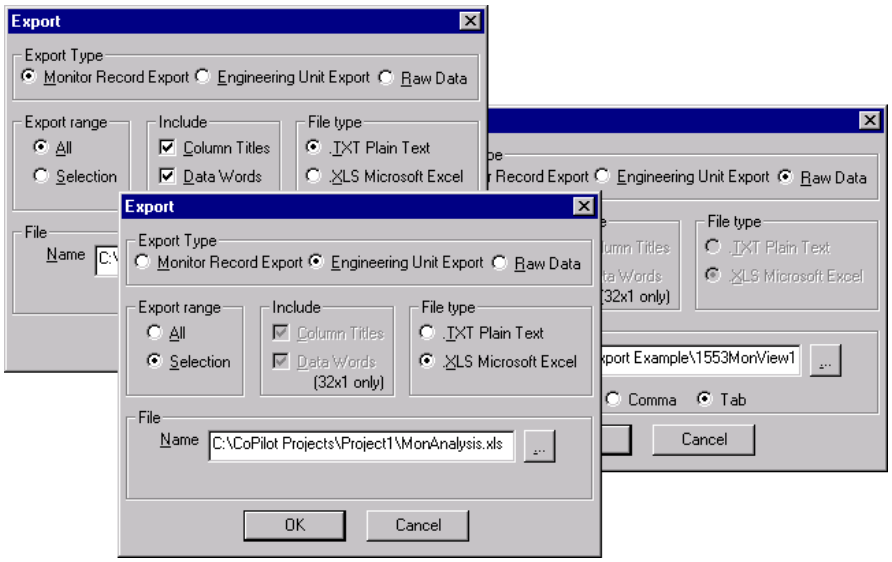


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| 1553 Monitor View Export Options | | | |
|---|---|---|---|
| | Record Export | Engineering Units | Raw Data |
| (Optional) Limit export with display filters | ✔ | ✔ | |
| Configure display format (radix, etc.) | ✔ | | |
| Specify range (all or selected) | ✔ | ✔ | |
| Include: columns? data words? | ✔ | | |
| Pick a file extension (.txt or .xls) | ✔ | ✔ | |
| Specify file name and path | ✔ | ✔ | ✔ |
| Choose delimiter (tab or comma) | | | ✔ |

To export 1553 Monitor View data to a file:

1. (Optional) Limit the exported records through display filters
2. (Record and Engineering Units export only) Determine the format of the export file by configuring the Monitor View format (see next section below)
3. (Optional) Limit the exported records to those selected through highlighting (see second section below)
4. Right click on the 1553 Monitor View window and choose Export from the context menu to open the Monitor Export dialog (see figure)
5. Choose options and specify a name and path for the export file
6. Click OK to create the file and close the Export dialog




The selected Export type determines which options are available in the Export dialogs pictured above (compare to the options charted in the table above).

### Configure the Export with Display Formatting

User choices through the three buttons in the Display frame on the Monitor View window that affect the display of data also affect the export of data (except for the Raw Data export type).

- Data is exported in Full, Form, or user-defined (CustomUser1/ CustomUser2) format
- Values may be transferred in binary, octal, decimal, or hexadecimal radix
- The time-tag may be transferred as linear, delta, or system clock values



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: When exporting for use with Excel, use the Linear (sec) or Delta (sec) settings, which do not append alphanumeric time or unit descriptions.

#### Export the Highlighted Selection

When one or more rows in the Monitor View display are clicked , the background is highlighted. Highlighting can be used to identify and limit the export (and print) of monitor data to specific rows. Shift or Ctrl (control) keys are used in conjunction with the mouse to highlight sections or selected lines.

Note: Multi-item selection (multi-select), as the following image shows, is not possible when the **Time Correlation Source** option is enabled for the Monitor View.

Highlighting in the image below was created through a sequence of three mouse clicks:

1. Click on row 1
2. Hold the (**Shift**) key while clicking on row 2, then releasing the (**Shift**) key and
3. Click on row 4 while holding down the control (**Ctrl**) key

| Rec # | Msg Name | Message | Data 4x8 | Time |
|---|---|---|---|---|
| 0 | Msg:<br>A BC->RT 00-R-01-02RT:<br>RT00 | Cwd1=0022 (00,R,01,02)<br><DATA WORDS><br>Swd1=0000 | 01: 2C1F 4220 | T=12.131.881<br>dT=12.131.881 |
| 1 | Msg:<br>A RT->BC 00-T-01-02RT:<br>RT00 | Cwd1=0422 (00,T,01,02)<br><DATA WORDS><br>Swd1=0000 | 01: 152D 3279 T.01 | T=12.131.999<br>dT=00.000.118 |
| 2 | Msg:<br>A RT->BC 01-T-02-03RT:<br>RT01 | Cwd1=0C43 (01,T,02,03)<br><DATA WORDS><br>Swd1=0800 | 01: 0FC0 6380 2DC1 | T=12.132.116<br>dT=00.000.117 |
| 3 | Msg:<br>A MC 31-T 8 Xmt Re... | Cwd1=FC08 (31,T,00,08)<br><DATA WORDS> | | T=12.132.253<br>dT=00.000.137 |
| 4 | Msg:<br>A BC->RT 00-R-01-02RT:<br>RT00 | Cwd1=0022 (00,R,01,02)<br><DATA WORDS><br>Swd1=0000 | 01: 2C1F 4220 | T=12.151.889<br>dT=00.019.636 |
| 5 | Msg:<br>A RT->BC 00-T-01-02RT:<br>RT00 | Cwd1=0422 (00,T,01,02)<br><DATA WORDS><br>Swd1=0000 | 01: 152D 3279 | T=12.152.006<br>dT=00.000.117 |

In the example above, the Monitor | Export command would transfer the three highlighted lines in Full display with data in hexadecimal format and a linear time-tag display.

### *Monitor View Import Options*

When exporting data from a Monitor View window, you can customize the content of the export file with display filters and highlighting. Then, by importing monitor files and appending them, you can create a highly customized monitor record, which could be useful for such applications as software playback.

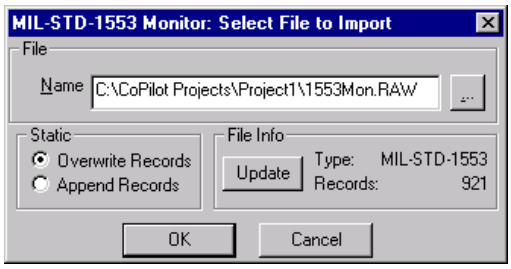
Monitor Views can import data for analysis and playing back data. Often, data is first exported to seed data values, then modified and re-imported into CoPilot. The import feature requires a .RAW file format, which are created using the Monitor View Export feature. The .RAW file format includes a header block followed by a data header row, then the data records. The file is intended to be self-documenting



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To import data into Monitor View:

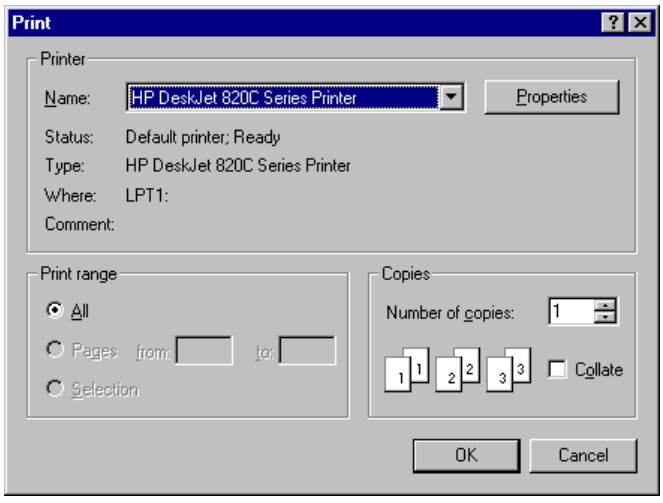
1. Right click in the Monitor View window and choose **Import** from the context menu
2. Click the **Browse** button, browse to the .RAW file you wish to import, select the file, and click OK to close the Open Hardware or View dialog (the path and file name will now be displayed in the file name field and the record type and count will be displayed in the File Info frame)
3. If you change the import file, click the Update button in the File Info frame to update the display
4. Choose the Overwrite or Append option
5. Click **OK** to insert the selected monitor file




If you chose to overwrite the current contents of the Monitor View window, the import file will start at record number 0. If you chose the append option, the import file will start after the last record and be number consecutively from that point. All other fields (such as time-tag) remain the same.

## *Monitor View Print Options*

Once the Sequential Monitor View is prepared through display format selections, display filters and highlighting, the user can print the target information through the Monitor | Print command. The content of this print file is governed by user choices in the open Monitor View window. Print, export and creation of playback files follow same rules. (For more information about the use of display filters and highlighting to control print selection, see the *Export Sequential Monitor Data* section on page *78*.)






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Users can redirect printing and set printer properties (e.g., paper size, portrait or landscape orientation, etc.) through the Print dialog box.

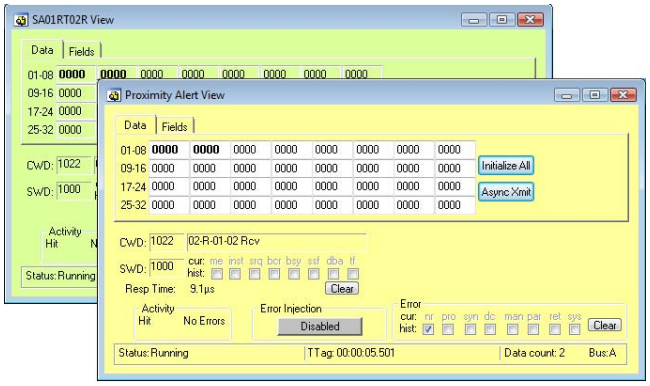
# Viewing MIL-STD-1553 Data

CoPilot has a variety of displays for viewing MIL-STD-1553 data. As the Hardware Explorer tree section described, the current values of fields and basic state information is displayed in the status column. Message and Subaddress Views are used for displaying detailed information for the message. The Engineering View provides a tabular display for viewing the current value of multiple messages and fields. The protocol browser (version 5) or the activity view (version 6) shows statistics and state information with the ability to drill in and out of each level. Finally, graphical displays provide a graphical representation for editing and viewing data. These different display types for MIL-STD-1553 are described below.

## Message/Subaddress View

### Using Message Views

Message View is a display container for information associated with a single MIL-STD-1553 message or subaddress. The Message View is typically used to focus on a message or subaddress when the bus is running. Any BC message or RT subaddress in the Hardware Explorer tree can be viewed and many Message View windows may be open at the same time. Information in the Message View is updated continuously when the project is running. Additionally, BC Messages may be asynchronously transmitted using the "Async Xmit" button.




*CoPilot 5 Message Views for a BC message and a subaddress*

The Message/Subaddress View window is divided into tabs where the Field tab contains information about the defined engineering unit fields.

The command word(s) and status word(s) are displayed in hex and by bit field. The response time (for messages) is displayed. If an error is present, a brief description appears in the Errors Detected frame. The description remains until displaced by another detected error. A record of all accumulated errors in displayed in the 'Error' group box.

Status line across the bottom displays the status, the time-tag ('TTag') and the data count (number of specified data words in the message).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Opening a Message View Window

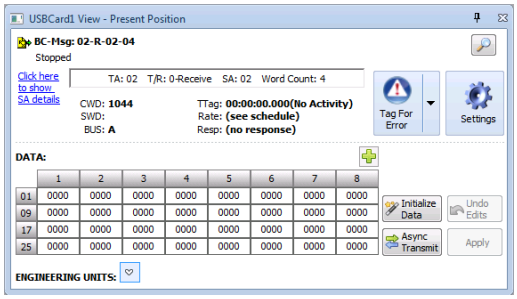
Message and Subaddress Views are accessed from the CoPilot Standard Views toolbar (version 5), the **Create** group of the **Home** ribbon tab (version 6), and from context menu of individual messages and subaddresses in the Hardware Explorer. The image below shows the Message View toolbar command.




Once the view is created from the toolbar or ribbon, simply drag and drop the message or subaddress onto the view. To open a message or subaddress view from the context menu, select the View message/View subaddress command. If the View command is bolded, the message can be double click to automatically open this view.

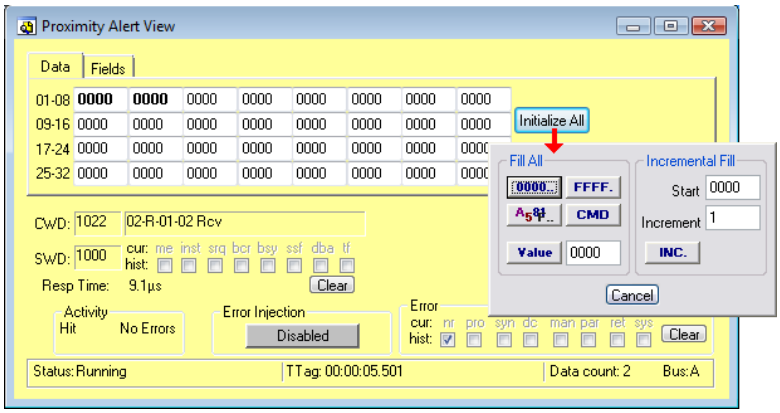
## *Message View Data*

The following is an image of the 1553 Message View in CoPilot 6.




*CoPilot 6 Message View*

In CoPilot 5, the background color of the view indicates whether a message (yellow) or a subaddress (green) is being viewed. The Data tab shows the message data. If a data word is included in the message, then the font is **bold** (otherwise normal). When a message or subaddress becomes stale (a timeout from inactivity) the text will become colored **red**. The Initialize All button is used for data initialization.



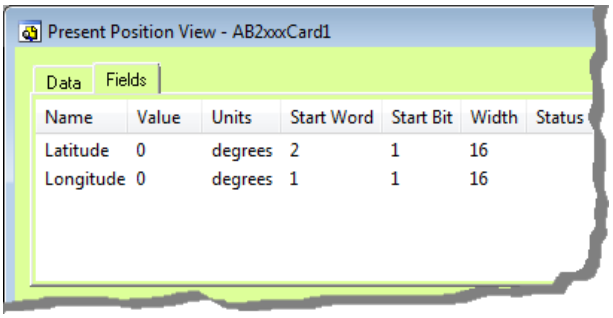

The message or subaddress can be tagged for error by clicking the Error Injection check box. The Accumulated Errors Detected frame displays the detected errors associated with this message or subaddress during simulation. Use the Clear button to clear the record of errors before another simulation. The lower-right hand corner displays the time-tag.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### *Message View Fields*

The Fields tab of the Message/Subaddress View window displays any data fields. If engineering units and meaningful names have been assigned to the fields, they display in this view. The data in the Value column is updated continuously while the bus is running.



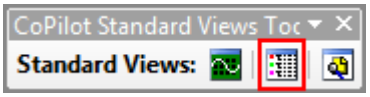

## Engineering View for MIL-STD-1553

#### Engineering View Overview

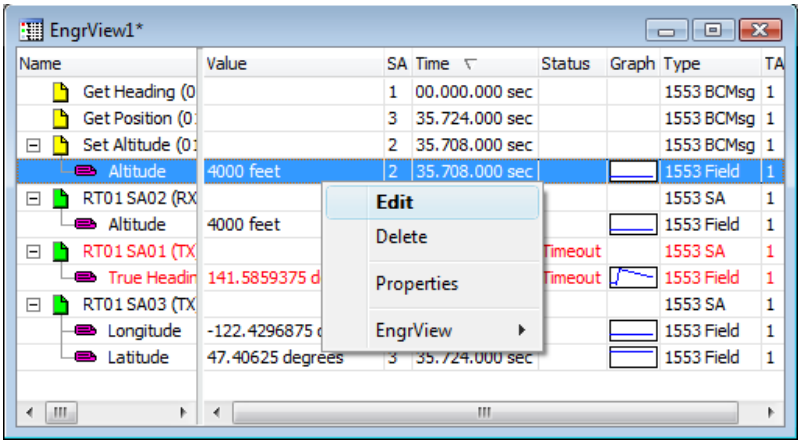
The Engineering View now replaces the MIL-STD-1553 View. Projects saved with previous versions of CoPilot with MIL-STD-1553 views in the project will automatically be converted to an Engineering View. The Engineering View is used for displaying MIL-STD-1553 BC messages, RT subaddresses, and their related data fields. Engineering Views can also host objects from other protocols such as ARINC 429 and AFDX.

The Engineering View displays the latest value of messages and fields in a list with the ability to configure whether or not data columns and graphs are shown. Multiple Engineering Views can be used to group specific information together using different column configurations.

An Engineering View can be created from the Engineering View icon in the CoPilot Standard Views Toolbar (version 5) or the **Create** group of the **Home** ribbon tab (version 6).




*Creating an Engineering View from the Standard Views Toolbar (version 5)*






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Display Customization

Messages and subaddresses often contain several data words whereas fields usually define a single fact. Consequently, it may be useful to use several Engineering View displays in a project. Each one may be customized through the selection of data properties, format, column width, and position to emphasize important information. Multiple views can also be used to group related information in common views.

### Engineering View Characteristics

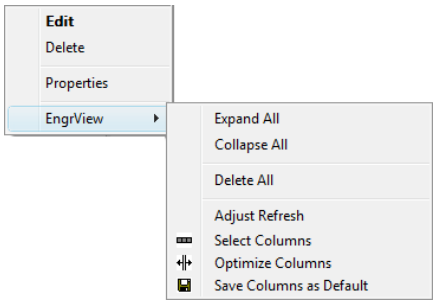
- MIL-STD-1553 Messages, subaddresses, and fields (and other protocols) are assigned to an Engineering View window by dragging a field from the Hardware Explorer to the Engineering View.
- Each item is entered in a row separated into columns. A context menu for each item provides access to properties.
- Users specify which attributes will be displayed and any combination of engineering unit, binary, octal, or hexadecimal data display.
- Drag either edge of column titles to change column width and drag entire columns to new positions.
- Users may save attribute selections, column width, and position as the Engineering View default for additional Engineering Views or for new projects.
- Color coding and special symbols simplify identification of items in Engineering View and the tracking of transmission intervals that exceed the timeout specification and out-of-range (operating limit) data conditions.

Messages are typically transmitted between five to ten times each second. Data in the Engineering View window is sampled to facilitate readability.

## *Engineering View Display Menu*

### Context Menu Commands

Right clicking on the background of an Engineering View or from the 'EngrView' menu item accesses the View context menu. This context menu provides access to configure the selected View. An Engineering View window can also be customized using the mouse to manipulate the display.




*Engineering View context menu commands*

- **Delete** removes the selected item from the view
- **Edit** opens an editor to edit the data



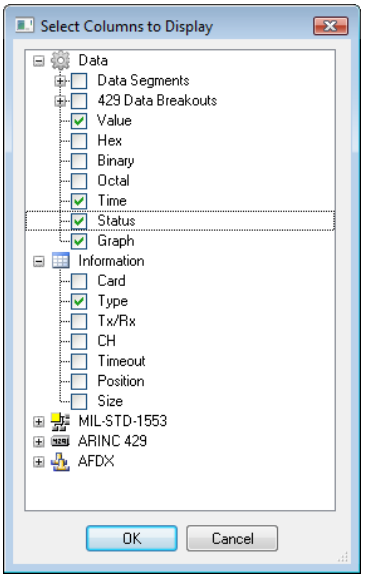
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Properties** opens the object property window
- **EngrView** shows the view context menu items when an item is clicked
- **Expand All** expands all collapsed nodes for items inserted in the view
- **Collapse All** collapses all child nodes for items inserted in the view
- **Delete All** deletes all items in the view
- **Adjust Refresh** allows the user to set a refresh rate for the view contents when the simulation is running. However, when not running, data in view updates at a slower rate.
- **Select Columns** **o**pens the Select Columns to Display Dialog window. Use this window to specify display columns.
- **Optimize Columns** **e**xpands or contracts all displayed columns to fit its contents
- **Save As Default** **s**aves the current display settings to the registry to be used as the default setup when new views are created

## *Available display columns*

### Available Column Groups

The Select Columns to Display dialog allows the user to define which types of information are displayed. (The Image column may not be hidden.)



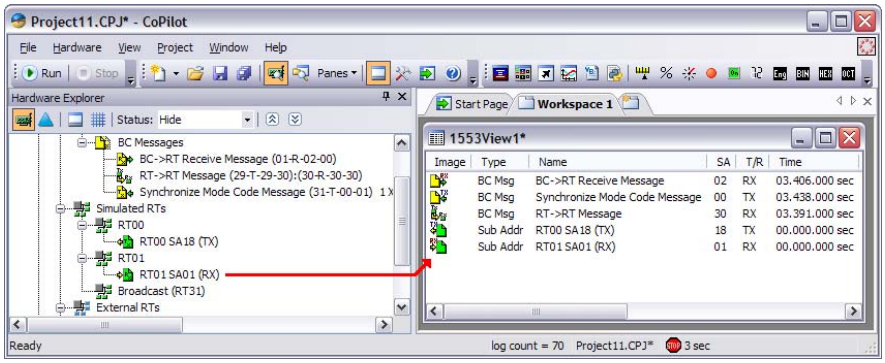

- **Data**: This category allows data to be displayed in several radices and provides additional data. The Value column displays the interpreted value as a real number (floating point); conversely, the Hex column displays the interpreted value as a Hex string. Other columns such as individual data segments, status, and graphs are in this category.
- **Information**: The category includes background information for the item such as type, card, position, and size.
- **MIL-STD-1553**: Contains MIL-STD-1553 specific columns.
- **ARINC 429**: Contains ARINC 429 specific columns.
- **AFDX**: Contains ARINC 664/AFDX specific columns.

## *Drag and Drop to Engineering View*

Items from the Hardware Explorer are assigned to the Engineering View through a drag and drop operation. Consequently, objects displayed in an Engineering View must exist in the Hardware Explorer. Items from the tree can be assigned to the Engineering View window individually or as a group. The columns in a Engineering View window may be customized before or after objects are added.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*Adding a MIL-STD-1553 SA to an Engineering View*

#### Assigning Single Items

Individual messages, subaddresses, and fields can be assigned to an Engineering View through drag and drop methods. Click the object in the tree (it will become highlighted) and hold down the left mouse button while dragging the cursor to the target Engineering View window.

#### Assigning Multiple Items

Objects in the Hardware Explorer often have other objects that branch out from them. To assign multiple items to the Engineering View window, drag an icon that has other items that expand below it. For example, you can drag and drop a single message and all of its associated fields will be added. Alternatively, you could drag the icon labeled “BC Messages” and all messages and fields beneath it will be automatically added to the Engineering View.

## *Engineering View Column Options*

#### Move Columns

To move a column in an Engineering View:

- Left click and hold the column that will be moved
- Drag it to the intended location then release the mouse button.

#### Resize Columns

To change column width:

- Left click on the right edge of the column title and
- Hold while dragging left or right, then release.

To auto-size columns (fit to width):

- Double click on the right edge of the column title

#### Sort Data

To sort the contents in ascending or descending order:

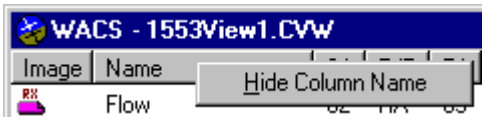
- Position the mouse pointer in a column title and click the left mouse button once. To reverse the order, click the column title again.

#### Show/Hide Columns

Columns can be removed through the column context box. Right clicking a column title header brings up a column context window.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




If a column is hidden, it can be restored through the Organize Columns window. The Image column (which displays a descriptive icon for each type of object) cannot be hidden.

#### Column Sequence

CoPilot users can rearrange columns according to preference. This is achieved by dragging the column header titles to the desired position.

#### Save Settings

If you wish to save your column configurations as the new default for additional Engineering Views, choose the Save As Default command from the Engineering View context menu. If you wish to preserve the columns without changing the defaults, you can save the Engineering View component (File | Save as).

### *Engineering View Display Status Indication*

CoPilot uses icons and coloring in the Engineering View window to indicate status information.

For example, if an object is removed from the Hardware Explorer after being assigned to an Engineering View window, references to that object are marked with a red X icon. Otherwise, the icon indicates the type of object and direction of data:

- receive message
- transmit message
- RT to RT message
- receive subaddress
- transmit subaddress
- receive field
- transmit field

### Reporting Out of Range and Timeout Conditions

Users can define a lower and upper operating range for scalar messages and fields (e.g., BNR, BCD, etc.) through the scalar tab of the message or field property page. When values are outside that range, the item icon is outlined in red to indicate an error/warning. Since the BNR and BCD interpreter property can be accessed prior or during a simulation, it is easy to modify operating limits to highlight conditions of interest.

When the interval between receipt of a specific message exceeds the timeout value set in message properties, the font color of the name, value, and time-tag will be changed to red.

## Protocol Browser

**Note**: *CoPilot 6 introduced the Ribbon Interface and simplified the Protocol Browser with the new Activity View. See New in CoPilot 6.*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The Protocol Browser provides a fast overview of what is happening on the MIL-STD-1553 (and other) databuses. The Protocol Browsers allow users to browse into various levels of the protocol to see the activity and statistics. The current and historical states of objects are displayed using color-coded icons. Drag a 1553 object into a Protocol Browser to view the details of that object and the summary of the objects below it. For example, displaying details for a channel will also show a summary of all the RTs on that channel. Then, double-click a particular RT for details about that RT with a subaddress summary.

## Graphical Displays

Graphical displays for displaying MIL-STD-1553 data include Quick Controls (also known as Quick Views), Strip Charts, and ActiveX controls in Control Views. CoPilot Professional provides these data control and visualization tools. Part 3 of this document beginning on page *367* describes the features of CoPilot Professional in detail.

# Error Injection

## Error Injection Overview

Protocol error injection is a method of modifying BC and RT messages so they do not comply with the MIL-STD-1553 specification. Non-compliant messages are typically used to test the response of a hardware system to bus faults. Parametric error injection (including transmit amplitude and zero crossing distortion) is the manipulation of the analog signal to deviate from the proscribed or expected waveform. Finally, the response time of RTs can be configured for 4G and 5G products for injecting errors within the range of 7.7 to 25.5 µs; the default RT response time is 9 µs.

Note: Some RT error injection features are not seen in the sequential monitor when the RT is tagged to inject RT errors and the monitor is on the same channel.

#### Device Dependent

Error injection is an advanced CoPilot feature of Ballard Technology's 1553-3 C and D model MIL-STD-1553 boards (LP1553-3C, LC1553-3C, PC1553-3C, CM1553-3C, BB14xx, all OmniBus boards with level C or D 1553 databuses), and all level M (Multi) hardware. Protocol error injection (word/gap/message errors) is available on all of these boards. Zero crossing distortion is available with level D hardware. Consult your hardware manual for more information about the features supported by your hardware.

#### Before or During Simulation

Error injection properties may be defined prior to or during a simulation. Therefore, a variety of errors could be intermittently injected without stopping the BC schedule. Error conditions specified prior to the run may be saved as part of the project, but error conditions specified during the run are transmitted but not preserved.

#### Using Error Injection

Error injection is a four-step process that includes the following:



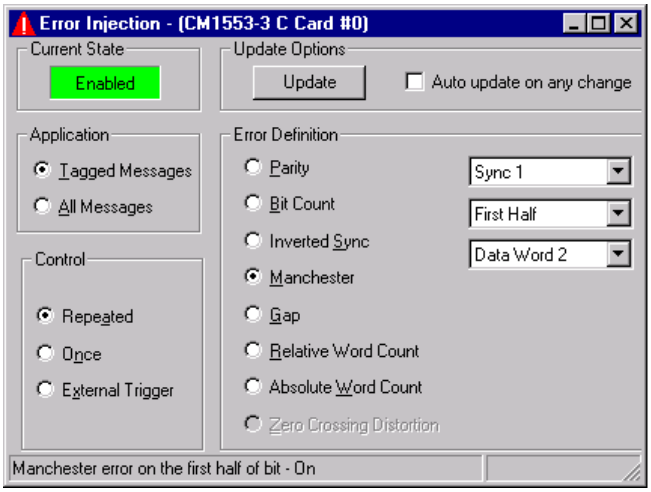
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Open the Error Injection window
- Define an error
- (Optional) Tag selected message(s) for errors
- Apply and control error injection

# Error Injection View

### Error Injection Definition Window

Error injection is defined and controlled through the Error Injection window. One error may be defined for a 1553 bus at any given time. That definition governs the injection of errors into the messages and subaddresses linked to error criteria until the link is terminated or until the error condition is changed.




To open the Error Injection window:

1. Right click on a 1553 channel icon (or the card icon for 3G hardware devices) in the Hardware Explorer tree
2. Select Error Injection from the context menu

The Error Injection View window will open in the display pane and the Project Explorer.

### Opening and Closing the Window

The Error Injection View window will disappear from the Project Explorer if it is closed in the display pane because it is tied to a board (or channel) in the Hardware Explorer tree and cannot be saved as a separate file. However, error injection will continue as it was defined when the window was closed (the error icon indicates whether injection is active). Repeat the steps listed above to reopen the view window from the Hardware Explorer tree.

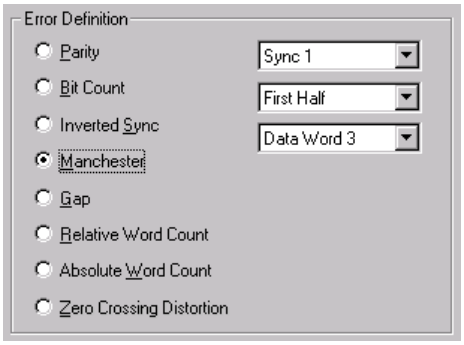
# Defining the Error

### Defining an Error

Errors are defined in the Error Definition frame of the Error Injection window. As an error type is selected with an option button (on the left), the required fields appear (on the right).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




- **Parity** inverts the parity bit of the selected word
- **Bit count** transmits a word with fewer or greater than 20 bits
- **Inverted sync** inverts the sync bits of the target word
- **Manchester** inverts either the first or second half-bit in a given bit position
- **Gap** inserts the specified time value (in units of 0.1 µs) before the selected word, thus causing the words in that message to not be contiguous
- **Relative Word Count** causes the actual number of data words to vary by the selected amount from that specified in the command word
- **Absolute Word Count** causes the word count in the command word to be ignored and the number of data words is established by the chosen value
- **Zero Crossing Distortion** (level D channels only) distorts the normal zero crossing point (of the specified bit in a specified word) on the leading zero or mid-bit zero

Once an error is defined, rules of application can be specified through the Application and Control frames of the Error Injection window.

### Redefining an Error

The application of a new error definition is controlled in the Update Options frame. To apply a new error definition, click the Update button.




To modify the definition dynamically, select the Auto update on any change option. If the simulation is running, the injection state is enabled, and this option is selected, changes are applied as they are made in this window (and injected when the correct conditions for application occur).

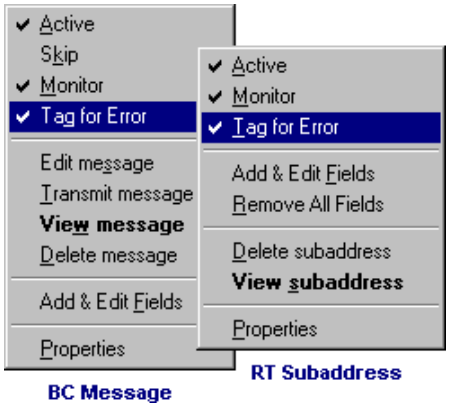
## Tagging Messages for Error Injection

### Selective Injection

Error injection is most useful when the error is associated with specific messages or remote terminals by tagging one or more BC messages or RT subaddresses for error injection. Messages and subaddresses may be tagged for error before or during a simulation.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




To tag a message or subaddress:

1. Right click on the message or subaddress in the Hardware Explorer tree
2. Choose **Tag for Error** from the context menu (see figure above)

Messages and subaddresses may also be tagged from their respective properties pages. If a message contains data words, it can be tagged from the Data Editor Window or Message/Subaddress View window.

When the Tagged Messages option (in the Error Injection window) is selected, error injection is limited to tagged messages. Otherwise, message tagging is ignored and error injection criteria are applied to all transmitted messages.

### Intelligent Application

Several factors could prevent a tagged message from containing an error:

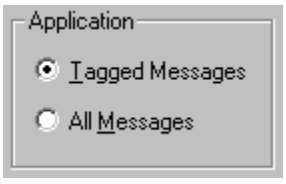
- The error definition does not make sense in the given message. For example, the error definition specifies the sixth data word in a message, but the tagged message only transmits three data words.
- Several messages have been tagged, but error injection is configured to only transmit one error after being engaged
- Error injection has been defined but not enabled

## Error Application and Control

The application and frequency of injected errors is specified in the Application and Control frames of the Error Injection window. The button in the Current State frame enables error injection.

### Application Rules

In normal practice, an error condition is assigned selectively to RT subaddresses or BC messages through a message context menu or properties page.




- **Tagged Messages** links the error criteria to those subaddresses and messages that have been selectively tagged.

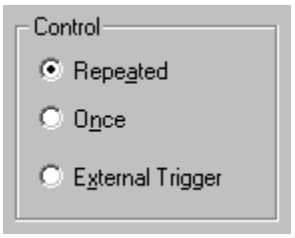


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- If the **All Messages** option is selected, assigned message tags are ignored and the error condition is applied to all messages.

#### Error Frequency

Errors may be injected singly at controlled times or triggered automatically and repeatedly. The frequency of error injection is established through the Control frame of the Error Injection window.



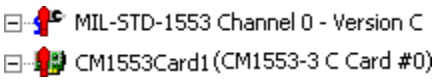

- The **Repeated** command applies the error condition continuously. Error injection is repeated without further user intervention until halted.
- When the **Once** option is selected, the error condition is applied to the next tagged message. Error injection is not applied to tagged messages that follow. The process is repeated each time the Update button is clicked.
- The same result can be achieved through an external hardware signal. If **External Trigger** is selected, the error injection criteria are applied to the next tagged message each time an external hardware signal is received.

#### Engaging or Suspending Error Injection

The button in the Current State frame is a master switch that controls the use of error injection specifications. If disabled, the information is preserved but not transmitted to the 1553 databus. The Enable/Disable button can be toggled prior to or during a CoPilot run.




When error injection is enabled, the error injection icon is visible on the board or channel in the Hardware Explorer.




# MIL-STD-1553 Database

## Overview

The MIL-STD-1553 database is used to define configurations and engineering unit interpretations. Definitions can be loaded from the database or saved to the database from CoPilot configurations. An XML format is used to import and export database information.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Using the database

### Assigning RTs, Subaddresses and Fields

The MIL-STD-1553 database holds definitions for RTs, subaddresses, and fields. For details on loading and saving these objects to or from the database, refer to the RT or field configuration section.

To extend a definition first load the definition from the database, alter the definition, then save the new definition into the database.

### Document Changes with Comments

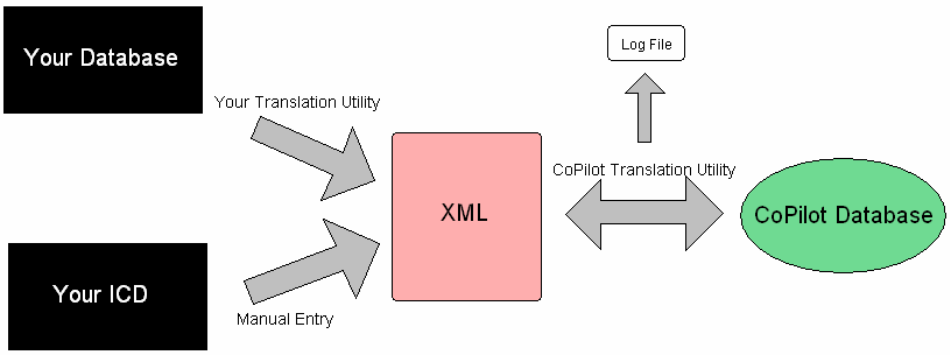
When adding or changing item definitions in the database, it is a good idea to add a comment to all new information added to the database. Comments could include initials of the author, data source and date.

## Sharing a database

The power of the database is in its reusability. You organization may have its own set of Equipment and/or Message definitions. By sharing common CoPilot database files with multiple users, all users will benefit from modifications made to the database. Once a definition is added or updated the next time any user loads that definition it will reflect the changes.

## XML database import/export utility

A utility application (BTIMDB.EXE) is provided with the CoPilot software to facilitate the definition the MIL-STD-1553 protocol configuration associated with engineering unit definitions. This application can be found on the CoPilot CD. Utilizing a defined XML interface, users may translate their own data definitions located ICDs (Interface Control Documents) or databases. The XML file describing the data definitions can be imported using the utility into a CoPilot MIL-STD-1553 database.

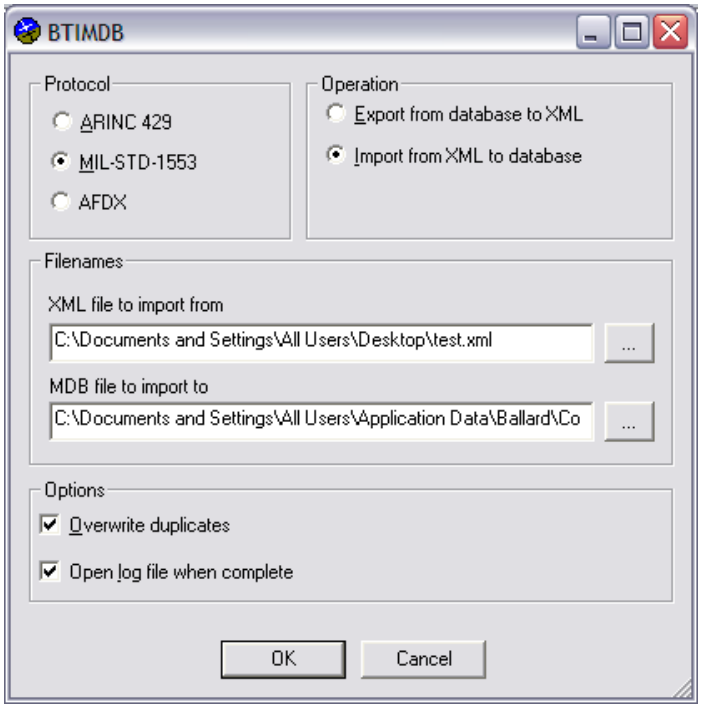



Through the use of this utility, the ease of providing data definitions for use by CoPilot goes up dramatically. Please see the “CoPilot Database Import-Export Utility-XML Format.PDF” located on the CoPilot CD.

The following image shows a screenshot of the utility, which allows you to specify various options such as which protocol database to import/export, the location of the XML and database files, and other options.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




# MIL-STD-1553 Scripting Events

### Overview

This section list and describes the events fired from various MIL-STD-1553 objects within CoPilot. For more detailed description of CoPilot scripting see *Automated Test Environment (ATE)* on page *399*. Refer to the *CoPilot Object Model* documentation in the Common Help Topics section of the Start Page (version 5) or the **Help and Learning** tab group on the **Backstage** (version 6) for a full description of all supported properties and methods.

### MIL-STD-1553 Channel Events

- **`OnEnableChange(`**`bEnable`**`)`** — Occurs when the enable state changes.
    - bEnabled— The current enable state

### MIL-STD-1553 Error Injection Events

- **`OnEnableChange(`**`bEnable`**`)`** — Occurs when the error injection enable state changes.
    - bEnabled— The current error injection enable state
- **`OnErrorDefine(`**type, ctrlflags, wordpos, wordcount, errmanchbitpos, bitcount, errbitcountval, errgaptime**`)`** — Occurs when an error is defined or the definition is changed. See details about this event in the CoPilot Object Model documentation.
- **`OnErrorSetAllMsgsChange(`**`bAllSet`**`)`** — Occurs when the error set all messages state changes.
    - bAllSet— The current error set all state



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### MIL-STD-1553 BC Events

- **OnEnableChange(**`bEnable`**)** — Occurs when the BC enable state changes.
    - bEnabled— The current BC enable state
- **OnPauseChange(**`bPaused`**)** — Occurs when the BC's pause state changes.
    - bPaused— The current pause state of the BC

#### MIL-STD-1553 BC Message Events

- **OnEnableChange(**`bEnable`**)** — Occurs when the BC Message's enable state changes.
    - bEnabled— The current BC message enable state
- **OnTimeTagStale(**`bStale`**)** — Occurs when the BC Message's Time Tag Stale state changes by the message either resuming transmission or by not transmitting for longer than the timeout.
    - bStale— The current time-tag stale state of the BC Message
- **OnMsgUpdate(**datawords, cwd1, cwd2, swd1, swd2, errors, activity, timetag, timetagh, MsgRespTime1, MsgRespTime2**)** — Occurs when the BC Message is updated.
- **OnControlChange(**`ctrlstate`**)** — Occurs when the BC Message's control state changes.
    - ctrlstate— The current BC message control state:
        - Continuous = 1
        - Skip = 2
        - Single Shot = 3

#### MIL-STD-1553 Field Events

- **OnValueChange(**`varValue`**)** — Occurs when the Field value changes.
    - varValue— The current Field's value
- **OnValueStringChange(**`bstrValue`**)** — Occurs when the Field's Value String changes.
    - bstrValue— The current Fields string value
- **OnTimeTagStale(**`bStale`**)** — Occurs when the Fields's Time Tag Stale state changes by the message either resuming transmission or by not transmitting for longer than the timeout.
    - bStale— The current time-tag stale state of the Field
- **OnLimitError(**`bHigh`**)** — Occurs when the Field's value is outside of the operating limits.
    - bHigh— The current bHigh value is set when the Field value is above the operating limits and bHigh is clear (zero) when below the operating limits



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

**MIL-STD-1553 Monitor Events**

- **`OnEnableChange(bEnable)`** — Occurs when the Monitor enable state changes.
    - bEnabled— The current Monitor enable state
- **`OnPauseChange(bPaused)`** — Occurs when the Monitor's pause state changes.
    - bPaused— The current pause state of the Monitor
- **`OnCountChange(count)`** — Occurs when the Monitor count changes.
    - count— The current count of number of records
- **`OnRunningChange(bRunning)`** — Occurs when the Monitor running state changes.
    - bRunning— The current Monitor running state

**MIL-STD-1553 RT Events**

- **`OnEnableChange(bEnable)`** — Occurs when the RT enable state changes.
    - bEnabled— The current RT enable state
- **`OnPauseChange(bPaused)`** — Occurs when the RT's pause state changes.
    - bPaused— The current pause state of the RT
- **`OnTimeTagStale(bStale)`** — Occurs when the RT's Time Tag Stale state changes by the message either resuming transmission or by not transmitting for longer than the timeout.
    - bStale— The current time-tag stale state of the RT
- **`OnStatusWordChange(status)`** — Occurs when the RT's status word changes.
    - status— The current value of the status word

**MIL-STD-1553 SA Events**

- **`OnEnableChange(bEnable)`** — Occurs when the SA enable state changes.
    - bEnabled— The current RT enable state
- **`OnTimeTagStale(bStale)`** — Occurs when the SA's Time Tag Stale state changes by the message either resuming transmission or by not transmitting for longer than the timeout.
    - bStale— The current time-tag stale state of the SA
- **`OnMsgUpdate(`**datawords, cwd1, cwd2, swd1, swd2, errors, activity, timetag, timetagh, MsgRespTime1, MsgRespTime2**`)`** — Occurs when the SA is updated.

Note: In addition to the events listed above, the CoPilot Object Model documentation describes all available properties and methods.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# ARINC 429 Protocol

CoPilot is Ballard Technology's software application that allows users to simulate and test ARINC 429 systems efficiently and easily, the perfect companion to Ballard's powerful avionics databus interface boards. CoPilot users can set up hardware channels to transmit or receive labels based on the latest label and equipment definitions released from ARINC. Consequently, information can be created or displayed in various radices (including binary, octal and hexadecimal) or translated into an easily readable engineering unit format. Data can also be monitored and stored in a Sequential Monitor file for later analysis.

## ARINC 429 Overview

### ARINC 429 Protocol Summary

#### Point-to-Point Communication

ARINC 429 is a specification that defines a local area network used on commercial aircraft and is the industry's standard for transfer of digital data between avionics system elements. The specification describes how an avionics system transmits information over a single twisted and shielded pair of wires (the databus) to as many as 20 receivers connected to that databus. Bi-directional data flow on a given databus (pair of wires) is not permitted.

#### Equipment Identifier

Each transmitting device ("source") is given a unique Equipment ID number. The Equipment ID number corresponds to a particular device on an aircraft. This number, along with the message label, is used to determine the meaning of the data. CoPilot 429 allows you to choose the equipment you wish to analyze and automatically assigns to it the correct ID number as specified by ARINC 429.

#### Data Types

There are five different data types in the ARINC 429 specification: BCD, BNR, Discrete data, Maintenance/Acknowledgement data, and ISO Alphabet No. 5. The majority of ARINC 429 labels are encoded in BCD or BNR format. In addition, CoPilot 429 extends the list of data types along with providing a user customizable data type (Custom Script).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **BCD** (Binary Coded Decimal): The BCD data field is defined within bits 11–29, where bit 29 is the Most Significant Bit (MSB). Bits 30–31 form the SSM for BCD data types.
- **BNR** (2's complement binary): The BNR data field is defined within bits 11–28. Bits 29–31 form the SSM for BNR data types.
- **Discrete**: Discrete words are used when non-standard labels are required.
- **Custom Script**: When the provided data types don't match your specification, CoPilot allows you to create your own.

### Transmit Rate

There are two transmission rates called for in the ARINC 429 specification, high speed (100 Kbps) and low speed (12.5 Kbps). CoPilot 429 can automatically detect the speed of receive channels attached through Ballard Technology's Ethernet, USB, PCI, and cPCI ARINC 429 boards. The bus speed must be defined for transmit channels because they are initiating the transmission.

CoPilot 429 and the Ballard 429 board will combine all labels for a particular channel to achieve a transmit schedule.

### ARINC 429 Messages

The basic element of ARINC 429 is a 32-bit message, which consists of five fields as shown in the diagram below:

| 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P** | **SSM** | | **DATA FIELD** | | | | | | | | | | | | | | | | | | | **SDI** | | **LABEL** | | | | | | | |

### LABEL

The 8-bit label at the beginning of an ARINC 429 word identifies what type of information is included in the word. Usually expressed in octal form, the label is transmitted MSB (most significant bit) first.

### SDI

The Source Destination Identifier (SDI) is optional and when used assumes bits 9 and 10. The SDI function is used when specific words need to be directed to a specific subset of a system or when a specific source of a system needs to be recognizable from the word content. When the SDI is not used, the bits should be padded with zeros.

The ARINC 429 CoPilot database can store up to five different message definitions for the same label/Equipment ID. This is done by providing the ability to save a label definition for each of the 4 possible SDI values plus one additional SDI-definition. Each of the database definitions can be unique and include the name, interpreter, and scheduling information.

### SSM

The Sign/Status Matrix (SSM) serves two purposes. The primary purpose is to express the sign (plus/minus, north/south, etc.) of the data field. The SSM may also be used to report hardware equipment condition (fault or normal) and operational mode or validity of data word content (verified or no computed data). The SSM uses bits 30 and 31. BNR uses bits 29 through 31.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### DATA FIELD

The Data field carries the actual message or information that is being transmitted. The data field consists of bits 11 through 28 or 29, and is transmitted LSB first. Unused bit positions are padded with zeros.

#### PARITY

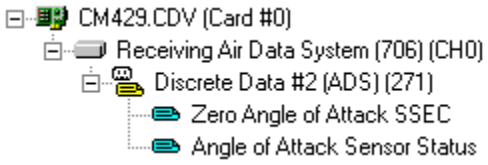
The Parity bit is set to odd parity by default. The parity bit is the MSB and is always bit 32 in an ARINC 429 message. The channel properties allow the parity to be configured for even parity, odd parity, and parity as data (parity as data not supported by CM429-1 devices).

## Hardware Explorer Object Hierarchy

**Note**: *CoPilot 6 introduced the Ribbon Interface and simplified the Protocol Browser with the new Activity View. The Hardware Explorer tree references remain for CoPilot 5 compatibility. See New in CoPilot 6.*

The Hardware Explorer pane holds the object representation of physical hardware devices along with display settings. These definitions of hardware devices are simplified by breaking them down into a hierarchical tree. Each node of the tree holds configuration settings (transmit and sample rates, schedules, settings, etc…) and interpretation definitions (engineering units) for the levels beneath them.

The figure below shows the Hardware Explorer tree displaying multiple object levels in a hierarchical structure.




These levels can include the:

- Ballard 429 hardware to be used. File | New | New Hardware and Views menu provides access to new hardware selection. Hardware can also be imported from an XML hardware definition file.
- (Optional) Cores. Cores represent groups of channels that share a Sequential Monitor, discrete banks and some configuration settings.
- Channel Configuration. Each channel represents one side of the ARINC 429 point to point communications (transmit or receive). Channels may be associated with Equipment IDs through auto detection or user specification.
- Messages of interest. Users can assign specific receive and transmit messages by adding labels to the channel from the context menu, or drag and drop. Additional receive messages are automatically added to the tree unless the Auto-detect Labels feature is disabled.
- Data fields defined within Messages.

Users can define each level of the tree explicitly or allow CoPilot to define them through auto-detection features built into the hardware and software (auto-detection is for receive channels only). The details of each level of the tree are known as branch properties.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# ARINC 429 Channels

## Overview

Hardware devices that transmit or receive on an ARINC 429 databus are referenced by an Equipment Identifier (ID). The companion ARINC 429 database that comes with CoPilot lists all Equipment IDs and the unique set of labels and fields associated with each. Therefore, if the equipment is identified for a channel, CoPilot translates incoming labels from their 32-bit structure into a recognizable engineering unit format. Channels are associated with Equipment IDs through the Hardware Explorer pane.

#### Equipment ID Information

An 8-bit value identifies ARINC 429 messages. The $2^8$or 256 label identifiers are not sufficient to document all ARINC 429 messages. Avionics systems often use the same label identifier on multiple systems, but the definition of the message may be different. Consequently labels in the ARINC 429 database are cross-referenced to an 8-bit label identifier and the transmitting Equipment ID.

## Setting the Equipment ID

The Equipment ID must be known before a label can be translated from the 32-bit string into an understandable engineering unit display. The ARINC 429 specification defines label 371 and label 377 (octal) as an equipment identifier label for general aviation and large aircraft. Although it is not mandatory, any transmitting system may encode its own Equipment ID into a 3-digit BCD field in the equipment identifier label. CoPilot includes a database of Equipment and associated Label IDs. See *The CoPilot 429 Database* on page *256*.

ARINC label definitions may vary significantly from one piece of equipment to another. For example, there are at least six different uses of Label 203. For that reason, the ARINC 429 database is indexed by Equipment ID and Label number.

### *Auto-Detect Equipment*

When the Auto-detect equipment option is enabled on receive channels CoPilot will watch for message label ID (377). CoPilot uses the data value of this message to consult the ARINC database (see the ARINC 429 database) for the Equipment definition. Auto-Detect labels can be used to complete the auto-detection process by associating database definitions with labels received on a channel and is described in a following section.

### *Assigning Equipment IDs*

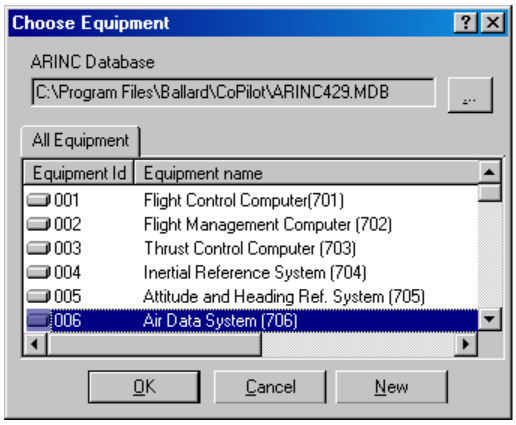
To set the Equipment ID:

1. Double click on a channel to access the Equipment ID window or right click on a transmit or receive channel in the Hardware Explorer tree to access the channel context menu and select **Set Equipment ID**.
2. (Optional) Click the **Equipment ID** or **Equipment Name** title bar to order the list in ascending or descending order.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3. Select the desired Equipment ID and press **OK** or double click on the desired Equipment ID to select and advance to the Label Selection window.




Once Equipment is selected, the channel icon in the Hardware Explorer tree changes from a connector to a system box.

## *Identifying an Unknown Equipment ID*

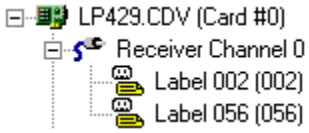
If one or more messages have been added to the channel, then the **Candidate Equipment** tab can help determine an unknown Equipment ID.

### Narrow Equipment Possibilities with Several Labels

The Candidate Equipments tab of the Choose Equipment window can be used to narrow the selection if more than one label is received. All Equipment IDs that transmit labels shown in the Hardware Explorer tree are listed based on information in the ARINC 429 database.

### An Example

Through Lookup Label, we can identify three Equipment IDs that transmit label 002 and seven that transmit label 056.

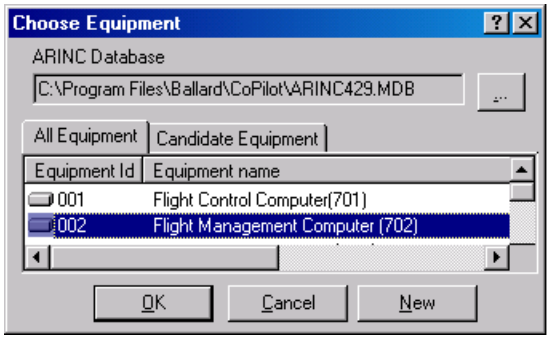



To isolate and specify the Equipment ID of a channel:

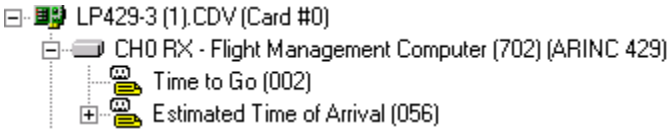
1. Right click the channel (e.g., Receiver Channel 0) in the Hardware Explorer tree to access the **Channel Context Menu** and select **Set Equipment ID**.
2. Click the **Candidate Equipment** tab to view a list of all possible Equipment IDs.
3. Double click the **Equipment ID** to select and assign equipment and label names and interpreters.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




Once selected, equipment and label titles for that channel are changed.




**Note**: SDI specific label information is not used for determining the Candidate Equipment.

# Channel Options

## *Channel Speed Settings*

### Overview

Speed options specify whether data transmitted on the receive channels will be at a high speed, low speed or whether it will be determined through auto detection.

### Auto Speed

Use the Auto Speed setting on receive channels (not available on PCMCIA devices) to auto-discover the transmitters data rate. When using this setting it not necessary to know the data transmission rate.

### High Speed

Standard high speed is defined as data transmitted at 100Kb/s.

### Low Speed

Standard low speed is defined as data transmitted at 12.5 Kb/s.

### Using Non-Standard Custom Speeds

Custom channel speed settings can be assigned on the **Speed and Parametrics** tab of a channel property page. A single custom speed setting controls both high and low speeds for all channels on a core (or card if no cores used). (Optional: Parametric channels can have independent custom speed settings.) Receive channel context menus show the current speed setting along with the frequency range used for signal decoding.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### *Monitor Options (Sequential Logging)*

For a detailed description of monitor options and how they affect the sequential monitor see *Sequential Monitoring for ARINC 429* on page *244*.

#### Monitor All

Monitor all ignores individual message settings and records all enabled messages on the channel.

#### Selective

Selective monitoring of a channel looks to individual messages settings and limits the recording to only include messages individually set with the monitor option.

#### Off/None

Monitor off (or none) ignores individual message settings and does not record messages on the channel.

### *Channel Options*

#### Enable

Enable turns a channel on and off (on by default). Channel icons of disabled channels are grayed out.

#### Pause

Individual channel operations can be halted and restarted during a simulation through the Pause option. When selected, the transmission or reception of labels through the ARINC 429 paused channel is halted. Although activity through the hardware is paused, CoPilot services are continued. For example, if transmit labels are attached to a data generator that process will continue. The data will not be transferred to the 429 databus through the card however. Channel activity can be resumed by deselecting the pause option.

#### Sample Rate

When enabled, the label sample rate is limited to the specified rate (in Hz). This setting is useful when the project contains a large amount of Messages. By limiting the Hardware Explorer object update rate you are freeing the system up to do other work. Sequential Logging functions are not affected by this setting.

#### Clear Channel

Clear Channel purges all channel information including the Equipment ID and label information from the channel definition. The channel properties are returned to the default values.

#### Save EquipID to Database

Save EquipID to Database places the equipment and label definitions in the ARINC database. For more details see *The CoPilot 429 Database* on page *256*.

## Receive Channel Specific Options

#### Auto-detect labels

Auto-detect labels direct the Ballard board to add incoming labels on receive channels to the tree as they are detected. This property is ON by default.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: Labels with parity errors (a parity value opposite to how the channel is configured) are not auto detected by Ballard receive channels but can still be monitored.

#### Auto-detect equipment

Auto-detect equipment directs the Ballard board to search for the Eq. ID label (377). This property is ON by default.

#### Fire event when undefined messages are received

The FireReceivedDefaultFilterMsg property configures receive channels to fire an event when previously undefined messages are received. The event contains the full 32-bit value of the message. When this property and Auto-detect labels are both enabled, events will fire until messages are auto-detected and configured in the Hardware Explorer tree. This property is OFF by default and can be modified at runtime. See *Using Object Events* on page *415* for more imformation about triggering and handeling events.

One example of using this feature allows users to receive an event to alarm the user when an unexpected ARINC 429 receive messages is received. Another example application for this feature is the generic handling of unconfigured messages for processing and retransmission on another channel.

Note: These events will only fire for undefined messages (i.e. not in the hardware explorer tree). When using label auto detection, as labels are detected and configured, the FireReceivedDefaultFilterMsg event is no longer fired for detected labels.

## Transmit Channel Specific Options

Transmit channel specific options are properties that only apply to transmit channels. Scheduling properties also apply to transmit channels, but are described immediately following this section.

#### Wrap-around

Wrap-around, sometimes called channel self-test, is a special feature of Ballard ARINC 429 hardware that creates an internal link between a single transmit channel and all active receive channels on the board. The internal board linkage is activated through a **Run** command. External connections are ignored when the wrap-around mode is in operation. Note, wrap-around is a hardware level option on the CM429-1 PCMCIA board.

#### Transmit Equipment ID

Transmit Equipment ID options automatically transmits the equipment identifier label (377) asynchronously approximately every 1 second. Each transmit channel can individually configure this option. The equipment ID value of the channel sets the value in the equipment identifier label.

Note: the equipment ID of a channel must be set to a non-zero value for the equipment identifier label to be transmitted.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### External Sync

The External Sync option outputs an external output synchronization pulse around message (label) activity. The external sync pulse will be active throughout the duration of each transmitted message. Alternatively, the external sync can be applied selectively to individual messages through message properties if the channel sync option is left unchecked. This property is located on the 'Channel' tab of the channel properties. The Sync Output Line Selection must be set on the External Sync tab for devices with more than one possible sync line (all devices with the exception of the CM429-1, Lx429-3, and the BUSBox) if sync is enabled.

Note: The CM429-1 (PCMCIA) card provides external sync functionality at the channel level only and is not configurable for individual messages.

#### External Trigger

The External Trigger options halts the transmission of messages until an external trigger input is received. Setting the external trigger option requires a trigger to step through each scheduled message; receiving the external resumes the schedule. Alternatively, the external trigger can be applied selectively to individual messages through message properties if the channel trigger option is left unchecked. This property is located on the 'Channel' tab of the channel properties. The Trigger Input Line Selection must be set on the External Trigger tab for devices with more than one possible trigger line (all devices with the exception of the CM429-1, Lx429-3, and the BUSBox) if triggering is enabled.

Note: The CM429-1 (PCMCIA) card provides external trigger functionality at the channel level only and is not configurable for individual messages.

#### Asynchronous Messages Transmission

If keyed ARINC 429 hardware device is properly defined and installed, messages may be transmitted asynchronously on the bus. Messages are sent when there is a gap in the schedule. Asynchronously transmitting a message is done from the command line using ATE (Automated Test Environment) with the "TransmitMessage()" methods to transmit messages via script events. ARINC 429 transmit channels also expose the "TransmitRawValue()" and "TransmitRawValueArray()" methods to asynchronously transmit either a single message or list of messages. These ARINC 429 methods transmit raw 32-bit message values. Asynchronous transmission directly from a transmit channel does not require the definition of messages or a schedule. In order to log the asynchronous message to the Sequential Monitor the ARINC 429 transmit channel must be configured to "Monitor All".

Note: Asynchronous message transmission for demonstration hardware (demo card) has no effect.

## Scheduling

Schedules are created to transmit messages using a sequence of messages and gaps. The length of the gap depends on transmit speed, required intervals, and the mix of labels. ARINC 429 labels must be separated by at least a 4-bit time gap. CoPilot automatically generates a default schedule from the list of active messages defined for a transmit channel. When creating large schedules, using



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

host timing to transmit very slow messages improves the speed of schedule building.

The user can also choose to transmit each label, one after the other, with the minimum inter-message gap by selecting the “Free-Running” schedule, which effectively loads the bus to capacity when there are numerous labels defined for the transmit channel.

In addition, with Ballard 429 hardware keyed for CoPilot Professional, the user has the choice of creating and running a custom schedule (not available on the CM429-1). Multiple schedules can be created to allow changing between schedules (while the simulation is not running). Custom schedules can be exported and imported to/from file.

## *Default Schedule*

By default, CoPilot automatically creates a schedule for transmit channels configured to use default scheduling. All active messages are scheduled at the transmit rates set by each message when the project is started. This default (rate-based) schedule is built to transmit each label at an interval within the minimum and maximum transmit intervals. The schedule building algorithm targets the midpoint between the minimum and maximum interval defined for each label (as specified in the ARINC 429 database or modified by the user). When the ARINC 429 Options are set to configure messages using the target transmit rate editing, the minimum and maximum interval are internally calculated using the global ’Maximum allowable transmit range’ setting to adjust the tolerance (on the ARINC 429 Properties tab of the CoPilot Options). The default schedule is illustrated below through an example using two ARINC 429 labels.

Note: CoPilot automatically adds an additional 40 bit times to last message in default schedules to allow asynchronous transmissions and equipment ID transmissions.

### Default Schedule Example

Two labels and their associated transmit criteria are shown in the table below assuming the channel is configured to high-speed (100 kHz). When constructing a schedule, CoPilot 429 attempts to transmit each label at an interval approximating the midpoint between the minimum and maximum interval. The midpoints in this example are 75 ms and 150 ms. Consequently, the resulting schedule will include two instances of Label 1 and one instance of Label 2 during a 150 ms cycle. That schedule will be repeated continuously until the board is halted through the Stop button.

| Label | Interval in Milliseconds | |
|---|---|---|
| | Min | Max |
| Label 1 | 50 | 100 |
| Label 2 | 100 | 200 |

Three 150 ms cycles of the default schedule (based on the two labels charted above) are shown in the table below. Although the schedule is based on label and gap times, the Start Time column is included to show the effects of the gaps. Notice that the start bit time for Label 1 is 0, 7500, 15000, 22500, 30000, etc. and the start bit time for Label 2 is 36, 15036, 30036, etc.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

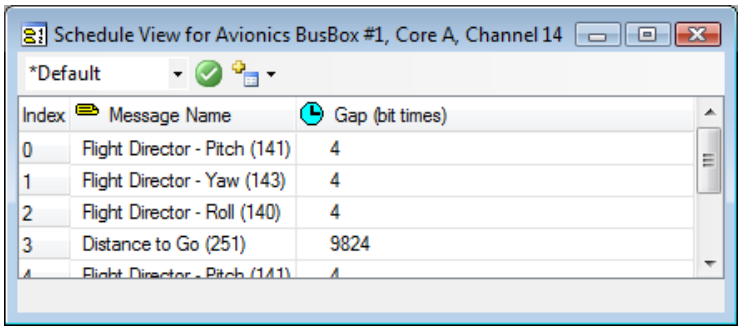
| | Label | Gap (bits) | Start Time (bit count) | Δ Label 1 (ms) | Δ Label 2 (ms) |
|---|---|---|---|---|---|
| Cycle 1 | Label 1 | 4 | 0 | 99.64 | |
| | Label 2 | 9896 | 36 | | 149.64 |
| | Label 1 | 4968 | 9964 | 50 | |
| Cycle 2 | Label 1 | 4 | 14964 | 99.64 | |
| | Label 2 | 9896 | 15000 | | 149.64 |
| | Label 1 | 4968 | 24928 | 50 | |
| Cycle 3 | Label 1 | 4 | 29928 | 99.64 | |
| | Label 2 | 9896 | 29964 | | 149.64 |
| | Label 1 | 4968 | 39892 | 50 | |

Each 150 ms cycle of the schedule consumes 108 bits (3 x 36 bits for each 32-bit label and minimum 4-bit gaps) out of a 15,000-bit interval, or less than one percent of the channel bandwidth. In this example, 20 labels are transmitted each second.

Note: The chart above is an example only. CoPilot may calculate the numbers slightly differently as it balances resources and processing tasks. The additional 40 bit times at the end of each cycle for asynchronous transmissions is also not included.

#### Viewing the Default Schedule

Choose **View Schedule** from the **Schedule** submenu found in a transmit channel context menu to open the Schedule View window. The figure below illustrates how our example schedule looks in the Schedule View window.




## *Software Timer Option with Default Scheduling*

Each ARINC 429 transmit message has an option to 'Use Software Timer Scheduling' on the Interpreter tab of the message property page to use the host software timing to transmit slow messages with large transmit intervals. Messages, such as status, heart-beat, or keep-alive messages, with a transmit interval greater than 500 ms may be enabled for this mode. If the average transmit interval is less than 500 ms, then the software scheduling setting is ignored. Additionaly, this setting is not used when the transmit channel is configured to use a free-running schedule since messages are transmitted back-to-back and timing rates are ignored. This setting is also ignored when the transmit channel is set to use a custom schedule.

Note: List Buffering of data cannot be used with the software timer scheduling option.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Free-Running Schedule

Users may choose to override the default schedule and run a compressed schedule by selecting the Free Running Schedule option. This schedule ignores the label timing specifications and transmits each enabled in rapid succession, separated by a 4-bit gap, which effectively loads the bus to capacity.

Note: CoPilot automatically adds an additional 40 bit times following the last message in the free-running schedule to allow asynchronous transmissions and equipment ID transmissions.

### Free-Running Schedule Example

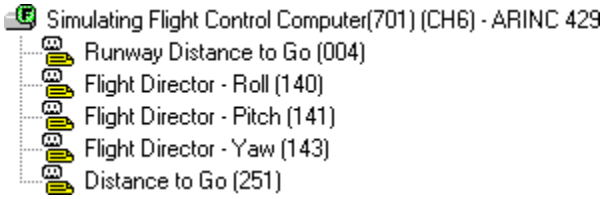
Using the three sample labels from the previous example a free-running schedule would look like the table below:

| | Label | Gap (bits) | Start Time (bit count) |
|---|---|---|---|
| Cycle 1 | Label 1 | 4 | 0 |
| | Label 2 | 4 | 36 |
| | Label 3 | 4 | 72 |
| Cycle 2 | Label 1 | 4 | 108 |
| | Label 2 | 4 | 144 |
| | Label 3 | 4 | 180 |
| Cycle 3 | Label 1 | 4 | 216 |
| | Label 2 | 4 | 252 |
| | Label 3 | 4 | 288 |

As a result, the channel is loaded to near 100% capacity and 2,777 ARINC 429 labels can be transmitted per second (on a high-speed channel). The schedule is executed continuously until the simulation is halted through the Stop button.

### Channel Icon

A free-running schedule is marked by an “F” icon on the transmit channel in the Hardware Explorer tree (see figure).




Note: The free-running schedule is not displayed in the Schedule View window.

## Custom User Schedules

### Custom Schedules

The CoPilot Professional Custom Schedule feature allows the user to create and fine-tune multiple transmit channel schedules to meet requirements. For example, the user could create one custom schedule one label is transmitted before another and another where that label follows another without affecting the transmit rates of either. Other possibilities include distributing gaps between labels to ensure specific spacing, or creating gap errors in the schedule. The Schedule View is used to view and edit schedules.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: The 429 Custom Schedule option requires a CoPilot Professional key. This feature is not available with CM429-1 hardware.

### Creating a Custom Schedule

Custom schedules are defined in the Schedule View window.

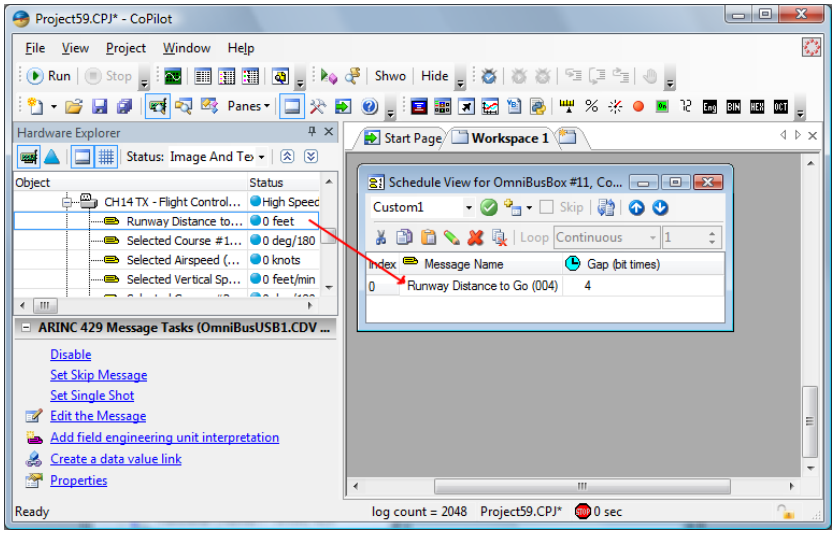
To create a new Custom Schedule:

1. Right click on the transmit channel, point to **Schedule** in the context menu, and then click **View Schedule** in the submenu to open the Schedule View window
2. Press the **New Custom Schedule** button in the view toolbar
3. Provide a name and optional comment then press the **OK** button

A custom schedule is defined by adding and rearranging labels and specifying gap times.

To add labels to the custom user schedule:

1. Click the Load Default Schedule button to load the current default schedule (will overwrite any labels already present), which can then be modified as needed
2. Alternately, drag and drop individual labels from the transmit channel in the Hardware Explorer tree (see figure below)




To modify gap times in the custom schedule:

1. Click on a row in the Schedule View window to select that label
2. Click the **Edit Gap** button
3. Enter a new gap time value

Other buttons (which can be identified by tool tip displays) are available to move selected labels up or down, clear the schedule, and cut, copy, paste, and delete labels.

Note: A gap time of less than the minimum 4-bit times may be defined. This is indicated by a red gap error icon 3 and the gap time is displayed red as well.

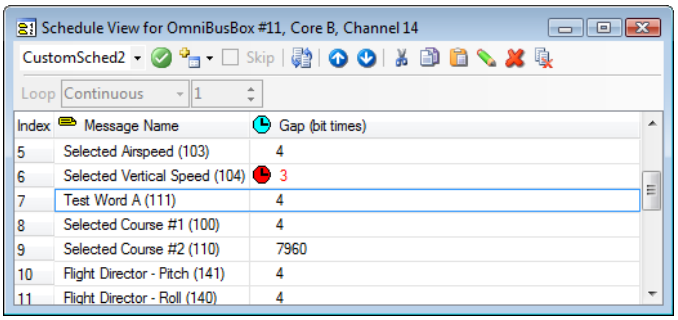


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: CoPilot automatically adds an additional 40 bit times to last message in the custom schedule to allow asynchronous transmissions and equipment ID transmissions.

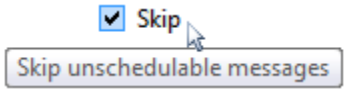
### Unschedulable Messages

If a label in the transmit channel is deleted or set to inactive, it can no longer be sent as part of the schedule. The Schedule View window updates dynamically as changes are made in the Hardware Explorer tree. The figure below illustrates the results in the window after a scheduled label was deleted:



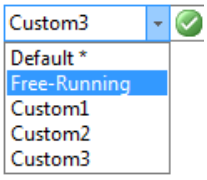

Note: Complicated schedules or schedules with tight timing tolerances take longer to build. You may want to close the schedule view when making multiple changes.

The absence of a scheduled label can have a ripple effect on the timing of the whole schedule. To compensate for this, CoPilot automatically preserves the timing sequence by inserting a gap for the length of the deleted message. If, however, you wish the schedule time reflect the missing label, set the **Skip (Skip Unschedulable Messages)** option in the toolbar of the Schedule View window (see figure below).




### Specifying the active custom schedule

Multiple custom schedules can be defined however only one can be active at any given time. The current schedule mode is indicated in the Mode drop down control by a * next the name.




*Schedule editor Mode drop down control*

To Set the Active Custom User Schedule:

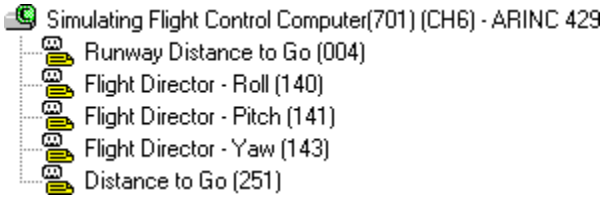
1. Select the custom schedule in the toolbar Mode drop down control
2. Press the **Select Schedule Mode** button in the toolbar



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

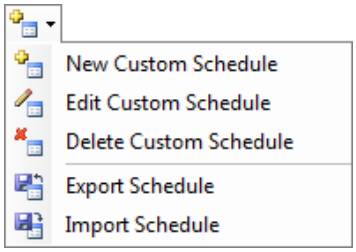
#### Channel Icon

A custom schedule is indicated by a "C" icon on the transmit channel in the Hardware Explorer tree (see figure below).




### *Schedule Import and Export*

Importing and exporting custom user schedules can be used to share/reuse schedules between projects, load programmatically created schedules (created outside the CoPilot environment) and for creating reports. The schedules are saved in XML file format.




#### Import Schedule (Load)

.To import a schedule select **Import Schedule** from the schedule view toolbar (see image above).

#### Export Schedule (Save)

To export a schedule, select **Export Schedule** from the schedule view toolbar (see image above). The exported file is formatted to provide a printable report of the schedule contents.

## Channel Summary

The channel summary tab of a channel property page contains a details list of all defined messages for the channel. Summary data may be exported to a text file using the **Export** button on the tab window. The columns are:

- **Index**–A zero based index number.
- **Label**–The octal label value of the message.
- **SDI**–The SDI value of the message or "NONE" if the message is not configured for SDI.
- **Min Time**–The minimum delta time measured from the previous occurrence of the message. If SDI specific labels are not defined, this value is measured between ANY SDI values of the label.
- **Max Time**– The maximum delta time measured from the previous occurrence of the message. If SDI specific labels are not defined, this value is measured between ANY SDI values of the label.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Copying Channel Settings

Use the “drag and drop” procedure to duplicate channel configurations, options, and settings onto another channel in the Hardware Explorer.

# ARINC 429 Messages and Fields

## ARINC 429 Messages

### Overview

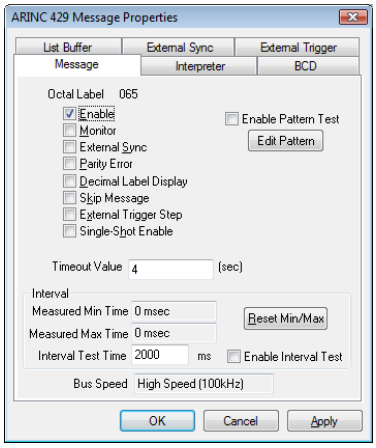
The basic element of ARINC 429 is a 32-bit message (or data word). The message consists of five parts that include the message label, SDI, SSM, data and parity. For more details on these fields see *ARINC 429 Protocol Summary* on page *217*.

### Optional Receive Message Definition

When emulating a transmit channel, messages must be defined before a schedule can be constructed. Manual entry of receive messages may not be required since they can be automatically added to the Hardware Explorer tree through the **Auto detect label** option on a transmit channel. To limit visibility to a subset of incoming message, select the message manually and turn off the **Auto detect** option.

## *Message Properties and Options*

ARINC 429 messages for both transmit and receive channels are configured from the message property page. The message properties have multiple tabs for organizing the various properties. The Message tab and Interpreter tabs are shown later in this section. The List Buffer tab is only valid for transmit messages on CoPilot Professional keyed hardware. The External Sync tab is used to configure digital outputs for output synchronization pulses. The External Trigger tab is used to configure digital inputs for input triggering. In addition to changing properties from the property page, properties can be configured through the context menu and through scripting.




*Message Tab of Message Properties*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Enable

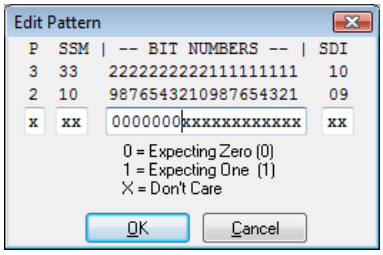
Enable turns a label on and off. Disabled messages are ignored when the ARINC 429 hardware is configured and activated. This property is located on the 'Message' tab of the message properties.

### Monitor

Monitor tags this message for selective monitoring. For a detailed description of monitor options and how they affect the sequential monitor see *Sequential Monitoring for ARINC 429* on page *244*. This property is located on the 'Message' tab of the message properties.

### Pattern Test

The pattern test option allows the user to test messages at the bit level. A pattern test is used to compare values of one or more bits in a message against expected values. Expected values are specified through the Edit Pattern dialog (see figure) and are engaged through the Enable Pattern Test checkbox. In the example below, bits 23 through 29 are expected to be zero.




*Editor for setting the Pattern Test value of a message*

If the expected values are not received, the Pattern Test error bit is set. This error state displays in the Activity column of Engineering Views and can be used to trigger events through scripting. The pattern test settings are accessed from the 'Message' tab of the message properties.

### Decimal Label Display

When set the message label identifier in the Hardware Explorer tree is expressed in both octal and decimal radices. This property is located on the 'Message' tab of the message properties.

### Skip Message

For transmit messages transmission of this message is halted/paused when the skip option is selected. For receive message the message is ignored (data and time-tag values are not updated). This option may be set during run time. Entire channels can be skipped (paused) through the Channel Pause option. This property is located on the 'Message' tab of the message properties.

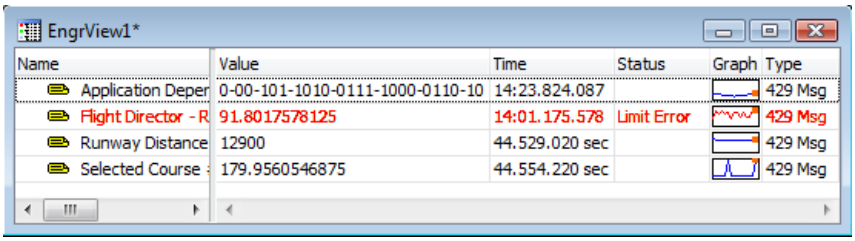
### Timeout Value

Timeout Value tests for message inactivity. If an expected message was never received/transmitted or is no longer being received/transmitted, the Timeout alarm will trigger when the timeout period has elapsed. For this reason, the value should not be set lower than the expected maximum interval rate for this message. The Timeout alarm is reported in the Activity column of Engineering Views and can be used to trigger events through scripting (operating limits and timeout items are highlighted in red). A timeout is cleared with new message



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

activity. This property can be configured from the ‘Message’ tab of the message properties.




*Engineering View with ARINC 429 data showing a message with a Limit Error*

### Parity Error

The parity of transmit messages are transmitted based on the channel rules (normally odd parity, but configurable through channel properties). When Parity Error is selected, a parity error is generated for that specific message by inverting parity. Other messages on the channel are transmitted according to the channel rules. Parity error cannot be set at the message level for the CM429-1 hardware. This property is located on the ‘Message’ tab of the message properties.

Note: Messages with parity errors (a parity value opposite to how the channel is configured) can still be monitored but are not auto detected by Ballard receive channels.

### Single-Shot Enable

When single-shot is enabled, the message will transmit once, then pause (the skip option is set and the single-shot enable is cleared). This property is located on the ‘Message’ tab of the message properties.

### External Sync

The External Sync option generates an external output synchronization pulse for the duration of the message. External sync can be specified for individual messages, or set for all messages on a channel. If the External Sync option is selected at the channel level (in the channel properties dialog), the message level option (in the message properties dialog) is ignored. CM429-1 (PCMCIA) cards provide external sync at the channel level only. This property is located on the ‘Message’ tab of the message properties. The Sync Output Line Selection must be set on the External Sync tab for devices with more than one possible sync line if sync is enabled.

### External Trigger Step

When the External Trigger Step option is selected, the message will wait for an external trigger input before transmitting. The schedule for this channel will wait until this label is triggered before proceeding. This option is not available for CM429-1 hardware. This property is located on the ‘Message’ tab of the message properties. The Trigger Input Line Selection must be set on the External Trigger tab for devices with more than one possible trigger line if triggering is enabled.

Note: Once external trigger step has been enabled and the schedule is halted, an external trigger pulse is required to resume the schedule, even if this option is cleared.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Interval Test

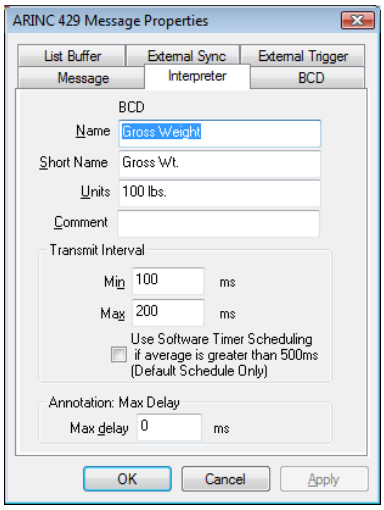
The Interval Test checks the actual interval (displayed in the Min and Max fields) between successive transmissions of this message against the value set in the Interval Test Time textbox. To enable the Interval Test, enter a value in the Test Time textbox and check the Enable Interval Test box. If the Measured Max Time exceeds the defined limit, the Interval Test error state is set. The Interval Test error state is displayed in the status column of Engineering View (see figure above) and can be used to trigger events through scripting. This property is located on the 'Message' tab of the message properties.

### Measured Min/Max Times

Measured Min/Max times are the minimum and maximum delta time measured from the previous occurrence of the message. If SDI specific messages are not defined, this value is measured between ANY SDI values of the message. Use r**eset Min/Max Time** to resets measured time to zero. This display of the measured minimum and maximum transmit interval times are shown on the 'Message' tab of the message properties.

### Interpereter Property Page

The Inperpreter property page allows additional configuration of properties as shown in the image below.




*Interpreter Tab of Message Properties (optionally display Transmit Rate)*

### Transmit Interval

The default channel schedule attempts to transmit each message at an interval approximating the midpoint between the minimum and maximum interval defined for each message (established in the ARINC 429 database or modified by the user). The transmit interval is specified on the Interpreter tab of a message property page. In addition to the transmit interval setting, the 'Use Software Timer Scheduling' option is used to make transmit channel schedules to build easier by using the host computer to asynchronously transmit the message for slow messages. The ARINC 429 settings in the CoPilot Options determine if the transmit interval is configured using a single value with a tolerance or using minimum and maximum transmit intervals as shown in the *ARINC 429 Properties* on page *52*. For more information on scheduling messages see *Scheduling* on page *225*.



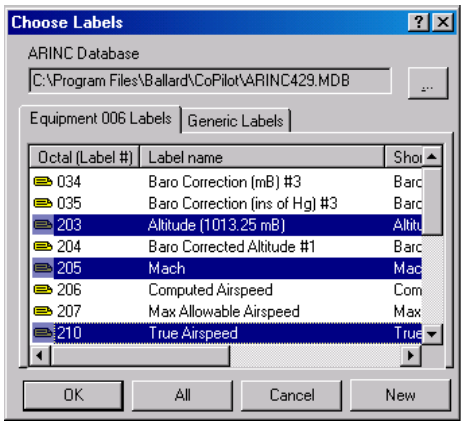
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Max Delay

The Max Delay annotation is information purposes only. This value is not used by CoPilot. Instead, when scheduling, the rates of transmission are kept between the min and max intervals. The max delay is shown on the 'Interpreter' tab of the message properties.

## *Adding Messages to a Channel*

Messages must be specified in the Hardware Explorer tree to create a transmit schedule or view a message using a Control View, Engineering View, or other CoPilot view window. Transmit and receive messages can be specified manually through the Choose Labels window. (Receive message are added to the channel list automatically if the **Auto detect** feature is selected.)




### Assigning Messages to a Channel

Immediately after the equipment ID is assigned to a channel the Choose Label dialog will open. All message associated with that equipment ID are listed. Select one or more items (or press the All button to select all items) and click **OK** to add them to the Hardware Explorer tree.

Warning: With all labels selected for, transmit channels for some equipment IDs may not be able to be scheduled.

### Adding Additional Messages

The example above describes an initial message selection sequence that began with equipment ID selection. To supplement an existing message list reopen the Choose Labels dialog by selecting Add Labels from the channel context menu. Select one or more items (or press the All button to select all items) and click **OK** to add them to the Hardware Explorer tree.

Warning: Adding labels to a channel when the schedule editor is open and using default scheduling causes the schedule to be recalculated. This can take a long time with large number of messages.

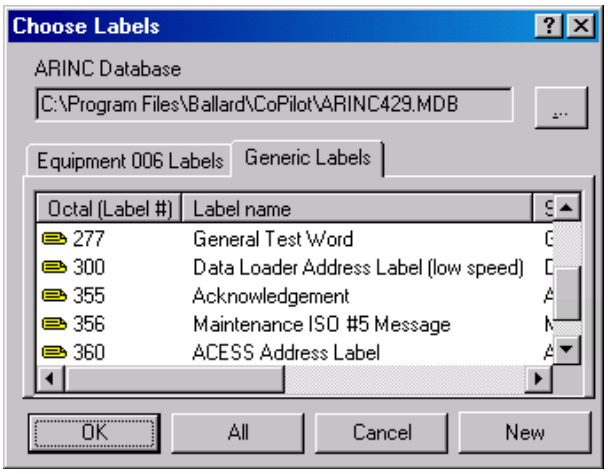
### Adding Generic Label Definitions

While most message entries in the ARINC 429 specification are linked to specific equipment, there are a few messages that may be transmitted by any



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

equipment (noted in the ARINC 429 specification without a defined Equipment ID). Labels 277 (General Test Word) and label 355 (Acknowledgement) labels are examples.




In a few cases, those generically defined labels may also be uniquely defined (e.g., The Flight Management Computer reassigns label 360 for Flight Information). The unique and generically defined labels are listed separately on the Choose Labels dialog window.

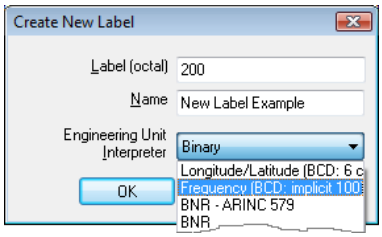
### Creating New Label Messages

New messages can be added if the required label is not already defined in the database or associated with an equipment ID. After new labels are created, they can then later be saved into the database.

To add a new message:

1. Click the 

button on the Choose Labels window
2. Type a message name and click on one of the items in the interpreter listing
3. Click **OK** to close the window and access the Interpreter

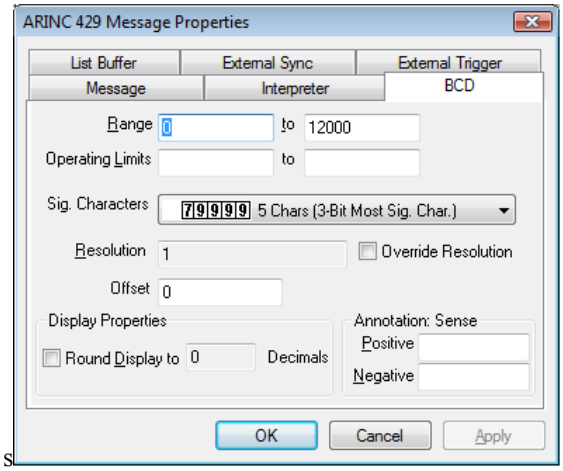



*Create New Label dialog*

4. Enter the min/max interval on the Interpreter Tab
5. Click on the interpreter tab (e.g., BCD) and type the necessary parameters
6. Click **OK** to close the window and add the message to the Hardware Explorer tree.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*Sample ARINC 429 BCD interpreter property page*

## *Message Interpreters*

### Overview

Once the equipment identifier is specified on a channel, the octal labels can be associated with the ARINC definition and displayed in engineering unit format. Engineering unit interpreters are associated with message data bits in order to decode the intended meaning. To learn more about interpreters and engineering units see the *Engineering Units* section on page *83*.

### Identifying an Unknown Label with Label Lookup

If the equipment ID of a receive channel has not been specified, incoming messages will be identified in the Hardware Explorer tree by its 3-character octal label value (e.g., 056). Display view windows will be in binary form since the message name and interpreter cannot be identified from the ARINC 429 database. The label lookup feature, available through the message context menu, lists the equipment ID and message definitions found in the ARINC database for the label value.

### Lookup Label Example

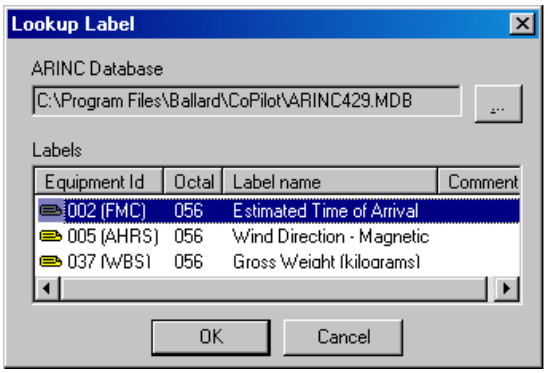
If the equipment ID of a receive channel has not been specified, incoming labels will be identified in the Hardware Explorer tree by its 3-character octal value (e.g., 056). Using autodetection, the database is used to automatically identify labels. If however, the message name and interpreter cannot automatically identify the label from the ARINC 429 database then value is displayed in binary.

To associate unidentified transmit or receive labels with a specific control system:

1. Select Lookup Label from the message context menu
2. Double click or highlight one of the descriptions and press OK



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




This method assigns names and interpreters one message at a time. Alternatively, you may modify all messages definitions on the channel by changing the Equipment ID.

## *Edit Data*

### Edit Interpreted Data (Engineering Units)

To edit interpreted engineering unit data values, select **Edit Data** from either a Message or Field context menu. A data editor window will open containing controls to change the value graphically (usually a slider control) or type in a value in the input box. To close the data editor window click the mouse anywhere outside the editor. The data editor window can also we opened from messages and fields in some view windows (e.g., Engineering View). For more information on editing data see *Generate and Edit Data* on page 91.

### Edit Raw

Messages can also be edited in Binary, Octal or Hexadecimal formats using the Raw Editor. To open the raw value editor select Edit Raw from the message context menu. To change the binary value double click on a binary bit (1 or 0) to toggle that bit value. Alternatively, use the alternate radix edit controls to view and edit the value in octal or hexadecimal.

### Edit data from Scripting

For information on editing data from scripting see the *Generate and Edit Data* section on page *91*.

## *List Buffering*

ARINC 429 data list buffers, or data buffers, create a list of data on the hardware associated with a transmit message structure. Data is loaded from an ARINC 429 playback file. Items added to a message's list buffer are sequenced in a circularly each time the message is transmitted (i.e. occurs in the schedule).

List buffers are configured from the List Buffer tab of the message property page. Once a file is loaded, the labels with the count of the number of times each label occurs is shown. Data from one or more labels may be selected, but the order of messages in the list buffer will match the order of the entries in the file. 3G and 4G devices support up to 16,375 list entries while 5G devices support up to 32,763 list entries.

Note: List buffering requires CoPilot Professional and is not supported by CM429-1 devices.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Using SDI Information

### Overview

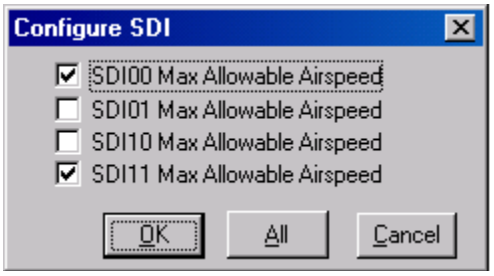
Just as with receive messages, it may be necessary to transmit messages in combination with unique SDI values. This is accomplished through the Configure SDI option and works in the same way for both transmit and receive messages.

### Utilizing SDI Information

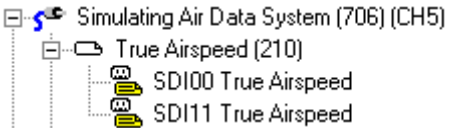
To actively display messages in an Engineering View or other view window, the message must be listed in the Hardware Explorer tree. When a message is assigned to a channel through the Add Label option, the SDI value is usually ignored. To view message based on their unique SDI value, the message must be configured to include SDI detail.

To configure a label to include SDI detail:

1. Right click the message
2. Select **Configure SDI** option and
3. Click the appropriate checkboxes in the SDI dialog box




The tree will be expanded as shown below. Additional onboard memory will be allocated at run-time and the SDI specific data may be displayed.




The SDI Option works the same for transmit and receive messages.

Note: When a message definition is changed to include unique SDI information, the changes can be saved to the database and recalled later with the SDI detail.

## Fields

### Overview

Although many messages are fully defined through the ARINC 429 database, there may be occasions when it is more convenient to view a portion or a single bit rather than the whole message. This can be done using fields. Fields, like Messages, use interpreters to represent data bits in engineering units. For more information about engineering units see *Engineering Units* section on page *83*.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Adding Fields to a Message

Fields are defined by the beginning and ending bit positions along with an Engineering Unit interpreter. Fields appear in the Hardware Explorer tree as an indentured branch under a message.

To add a field:

- Select **Insert Field** from a message context menu or task list
- Select the bit positions and assign a name
- Select the Engineering Unit interpreter and press **OK**

### Field Interpreter

Engineering unit interpreters are associated with data bits in order to decode the intended meaning. To learn more about interpreters and engineering units see the *Engineering Units* section on page *83*.

### Edit Interpreted Data (Engineering Units)

To edit interpreted data select Edit Data from either a Message or Field context menu. A data editor window will open containing controls to change the value graphically (usually a slider control) or type in a value directly. To close the data editor window click the mouse anywhere outside the editor. The data editor window can also be opened from linked items in some view windows (e.g., Engineering View). For more information on editing data see *Generate and Edit Data* on page *91*.

### Edit Raw Value

Field values can be edited in Binary, Octal or Hexadecimal formats using the raw value editor. To open the raw value editor select Raw Value Editor from a field context menu.

### Edit data from Scripting

For information on editing data from scripting see *Engineering Units* on page *83*.

## Engineering Units

### Overview

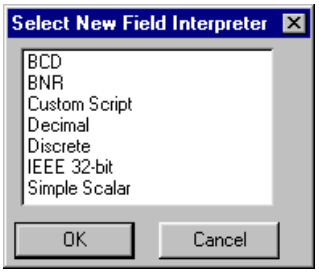
Engineering unit interpreters are associated with ARINC 429 message and field data bits in order to decode the intended meaning. An engineering unit definition consists of a full name, short name, units string, user comment and the data interpreter used to decode the data bits. For a full description of engineering units and the available interpreters see the *Engineering Units* section on page *83* of this manual.

### Change Interpreter

The interpreter is chosen and its properties defined when adding a new field to the Hardware Explorer tree. The interpreter for a data field may be changed at any time. Right click on a field in the Hardware Explorer tree and choose **Change Interpreter…** from the field context menu to open the Select New Field Interpreter dialog.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




Select a new interpreter from the list and click **OK**. The Interpreter Properties dialog will open so you can configure and define the new interpreter.

## Helpful Hints

### Limit Unnecessary Receive Processing

CoPilot creates a message in the hardware for each active message (label) defined in the Hardware Explorer. By default, CoPilot will continually service all active messages while CoPilot is running to allow event triggering and current value displays. This behavior can be altered by enabling "Message Sample Rate" (1553) or "Label Sample Rate" (429) on the channel property page to limit the frequency that these objects are serviced to a much slower rate.

Unnecessary processing of messages can affect performance, but can be skipped by either disabling the message or by deleting the defined label from the receive channel. When messages are disabled or not present in the Hardware Explorer, the message can still be recorded by the sequential monitor. Interpreters of inactive messages are still used for engineering unit interpretations in the sequential monitor data.

---

Note: Disable the auto-detect feature of receive channels to prevent the automatic insertion and processing of detected labels. Also, the interpretation of labels and fields captured in a sequential monitor file can be reapplied after the data has been collected; even if no interpreters were originally defined. See *Changing Engineering Unit Interpretations* section on page *79* for additional information.

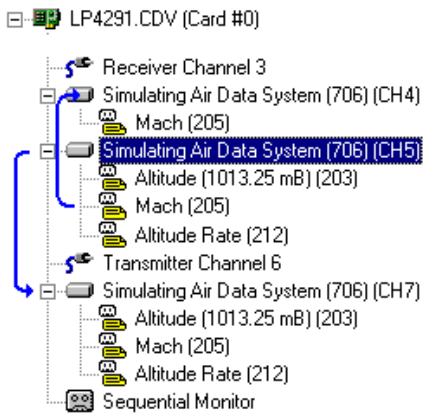
---

### Drag and Drop Assignments

Channel and message specification can be copied from one channel to another through drag and drop assignment. All information associated with the message is copied including title, interpreter, and current data values. An equipment designation will also be copied if an Equipment ID has not been defined for the receiving channel.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




**Examples:**

Single Label

Label 205 specification is copied from channel 5 to channel 4 by dragging a single label.

All Labels

All labels and Equipment ID are copied by dragging the channel 5 equipment title to channel 6.

Uses of the message (Engineering View, Control View or Data Generator assignments) are not copied to the new location.

### Updating Message Definitions

Although data values and properties are transferred to the new location when a drag operation is executed, an active data link is not established. If message names, values or interpreter specifications are subsequently changed and that information is needed in the drop location, that information should be transferred through a second drag operation.

### Creating Message Variations

Drag and drop is also a useful way to populate a new equipment ID with existing message definitions or to create message variations. Once dragged to the target channel, modified messages or whole equipment definitions can be saved to the database for reuse.

### Overlapping Fields

Fields may overlap one another provided the start bit of one is not the same as the end bit of an existing field.

# Defining Parity

### Overview

ARINC 429 is transmitted with a parity bit. Normally, ARINC 429 data is transmitted with odd parity but it is also possible to transmit with even parity or no parity. The transmitting equipment defines parity rules and all messages sent by that equipment should follow that parity rule. Consequently, the receiving device can detect transmission errors in individual messages.

Parity rules are defined for each channel. Transmit channels can be configured to the opposite parity as defined by the system to inject parity errors. Transmit messages can individually invert the parity while running for dynamic parity error injection. The display of the current parity value and parity errors of incoming data are then commonly viewed in an Engineering View or Sequential Monitor View.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### Parity Errors

Incoming messages are tagged as having parity error when bit 32 of that message does not match the intended rule of parity (see Rules of Parity below).

#### Parity as data

Although ARINC 429 data is normally transmitted with odd parity, users may select to use parity as data. When selected, parity checking is suspended and bit 32 may be used for other purposes.

#### Rules of Parity

The Parity frame in transmit and receive channel property windows define the intended parity rule. A message following the parity rule is said to have correct parity. Messages that do not follow the rules have incorrect parity or a parity error. Although ARINC 429 data is normally transmitted with odd parity, users may select even parity or data. When the latter is selected, parity checking is suspended and bit 32 may be used for other purposes.




The three options shown above appear on transmit and receive property windows. In both cases however, they define the rule that the transmitting equipment is expected to follow.

# Sequential Monitoring for ARINC 429

## Overview

CoPilot makes it easy to record databus. The quickest way to collect data is by selecting the Monitor All option in the Sequential Monitor. This option records all data on all channels. Users may also limit the recording by enabling capture filters at various levels of the protocol. Data recorded in the monitor data files can then be displayed using an ARINC 429 Sequential Monitor View. The interpreters are saved with Sequential Monitor Views, but the interpreters can be changed at a later date; see *Changing Engineering Unit Interpretations* section on page *79* for additional information.

## Configuration

### *Options*

#### Enable

Enable turns a Sequential Monitor on and off. Sequential Monitor icons are grayed when disabled.

#### Pause

Data collection is suspended when the Pause option is set.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### Clear pause on run

The pause option is automatically cleared when a project is run if the clear pause on run option is set.

#### Monitor All

Monitor all ignores individual message and channel settings and records all enabled messages on all enabled channels.

## *Capture Filters*

#### Overview

The Sequential monitor records data based on monitor options on the sequential monitor, channels and messages. Setting capture filters limits which records are logged to the sequential monitor file.

#### Selective Monitoring

The Sequential Monitor may be initialized to capture all bus traffic or collect a limited set of data based on message and channel monitor options. When the Sequential Monitor is not set to **Monitor All**, then data collection is based on settings at the channel and Message levels. When a channel is not set to **Monitor All**, then data collection is based on settings at the Message level. When selected for monitoring, a channel or message in the Hardware Explorer tree is marked with a monitor icon.

#### Overriding Capture Filters

If the monitor options for a channel is set to **Monitor Off** or **Monitor All**, or if the Sequential Monitor option is set to **Monitor All**, then the state of the monitor at the message level has no effect.

## *Monitor Data Collection Modes*

#### Capturing Monitor Records to File

When CoPilot collects data for a Sequential Monitor, data is collected using MonData files as described above. As records are collected, the count displayed in the status column of Hardware Explorer is updated to reflect the count of items recorded to each currently connected MonData file and the sequential monitor item displays the total count for the current run. The records in MonData files can then be viewed and analyzed using a 429 Monitor View. The 429 Monitor Views can be saved, reopened, modified, and shared with other projects.

#### Data Collection Modes

That default response can be changed by the user by selecting a different Data Collection mode:

- **Append to Current**—New data will be added to the monitor buffer each time the CoPilot simulation is run, thus accumulating until the monitor site is saved or overwritten
- **Create New Recording**—A 429 monitor data (MonData) file will be created each time CoPilot is run to capture the data until the simulation is stopped
- **Overwrite with Prompt**— A message will appear (with the number of records currently in the monitor buffer) asking for confirmation to run



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

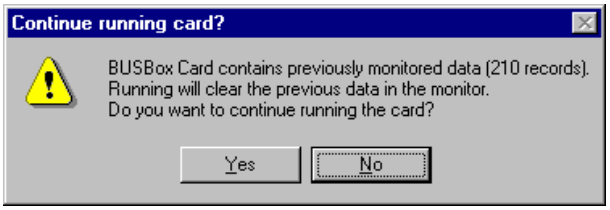
the card and overwrite those records (see figure) each time the project is run (simulation started). If accepted, previously recorded data in the monitor capture are erased. The new data overwrites the records in the 'connected' monitor data log (MonData1) each time CoPilot is run.

| Comparison of Data Collection Modes | | | |
|---|---|---|---|
| | **Overwrite with Prompt** | **Create New Recording** | **Append to Current** |
| *Description* | Overwrite existing records (on input from user) | A new MonData file is created | Add new records to existing file |
| *User-input dialog* | Yes | No | No |
| *Monitor buffer in hardware tree* | Records are erased and counter is reset to 0 | Creates new (active) MonData file and counter set to 0 | Records are pre-served and counter increments |
| *Monitor View window* | Uses active Monitor View window (if present) | Monitor Views are unaffected | Uses active Monitor View window (if present) |

*Table of Sequential Monitor data collection modes*

**Warning**: Overwritten data cannot be recovered.

In the default mode, a new view is created each time to prevent the possibility of data loss. If overwriting the data, then the following dialog will appear when the CoPilot Run button is pressed (if the monitor buffer contains records from a previous run). The dialog is intended to minimize the accidental loss of data.




*Prompt to overwrite previously recorded data when in overwrite mode*

#### Sequential Monitor Considerations

The specified Data Collection mode affects all channels being recorded by the Sequential Monitor. For example, if a Sequential Monitor is connected to a hardware device monitoring both ARINC 429 and MIL-STD-1553 channels, the record count is the total number of records for both.

## *Monitor recording modes*

The monitor can be configured to run continuously or for a specified length of time then is automatically set into the pause state. (If you have a Sequential Monitor, the recording mode affects all channels and protocols being recorded by the Sequential Monitor.) Un-pausing (resuming) the Sequential Monitor while running will again collect data for the specified time then pause again.

- **Default**—The monitor will run continuously when CoPilot is in simulation mode



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Timed Run**—The monitor will run until the specified time period is reached. This option can be useful when using scripting technology to only collect data around a triggered event such as a value out of range.

### *Data collection settings*

The data collection settings of the sequential monitor control the use of delta and interval modes. Important information is easily lost when large amounts of data are captured, and sometimes the volume and pace of data can exceed the capacity of host memory. Recording in Delta or Interval mode can improve capture efficiency because these can filter the monitor records via the Ballard hardware. This preserves the resources of the host computer to service the interactive requests associated with the CoPilot session.

#### Delta Mode

Using Delta Mode, ARINC 429 data is only added to the Sequential Monitor when data values change. Therefore, unless the purpose of the 429 session is to measure the frequency of received data, there is no point in cluttering the disk with redundant data.

#### Interval Mode

Using Interval Mode, ARINC 429 data is only added to the Sequential Monitor once per interval time for each message.

## Saving sequential monitor data

#### Overview

Sequential Monitor data is saved in a Monitor View. The Monitor View file preserves the data along with any associated engineering unit interpreters. For more information see *Saving and Loading Sequential Monitor Data* on page *75*.

## Analyzing recorded sequential monitor data

### *Monitor View*

#### Overview

The Monitor View window displays data and time-tag information in a columnar format. Each protocol in CoPilot has its own Monitor View window customized to the recorded protocol. For more information see *Analyzing Recorded Data* on page *76*.

#### Display Settings




- **Mode**—Mode is used to select between different display configurations. The Brief mode limits the displayed columns to channel and label numbers, label values, time-tags, error status and sequential record numbers. The Full mode adds several other columns including Label and Equipment titles. The Custom user modes can be configured by the user to show the columns they feel are most important.
- **Radix**—Radix controls the numeric representation of data



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Time**—Time controls how data time-tags are displayed

### Monitor View Time-Tags

Click the time drop list from the toolbar (see figure) of the Monitor View window to toggle between Linear, Linear (seconds), System Time, Delta, and Delta (seconds) to define the time-tag for the current display.




*Standard ARINC 429 Monitor View toolbar*

Linear (seconds) and Delta (seconds) are displayed in units of seconds for easy comparison of time-tag values and for exporting data values for analysis and graphing.

**00.000000** **ss.ssssss**

The Linear and Delta time display modes display the time-tag in the following format:

**000:00:00.000000** **ddd:hh:mm:ss.ssssss**

ddd – days

hh – hours

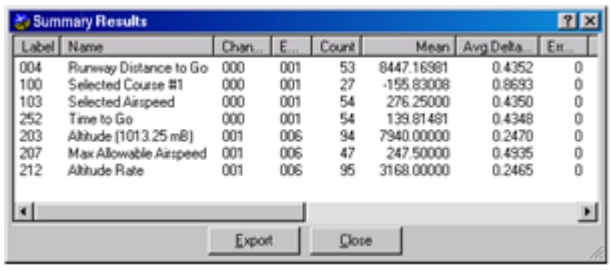
mm – minutes

ss.ssssss – seconds.(3 digits of milliseconds)(3 digits of microseconds)

Note: The IRIG timer can be initialized with a seed value or can use the system time.

### Monitor Summary

After sequential monitor data is collected, the ARINC 429 Monitor View can display summary information by analyzing the captured data. Selecting **Summary** from the view context menu shows the summary for all messages recorded in that file. Messages with unassigned or string-based interpreters display 'NONE' for the parameters that are calculated from engineering units (such as minimum, maximum, mean, standard deviation, variance, etc.). The results can be exported for further analysis with tools such as Microsoft Excel.




### Display filtering

Display filters limit the displayed data to only the data records that match a set of user defined criteria. Generally, the purpose of a display filter is to focus on one or a small number of messages in the sequential file. Messages defined in

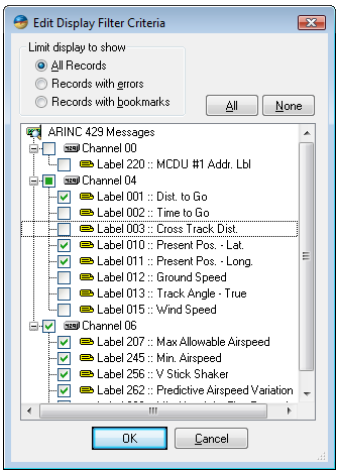


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

the Hardware Explorer and messages in the recorded data are shown in the list of available messages for each channel. Filtering can be configured by enabling/disabling the entire channel or by selecting individual messages. The **All** and **None** buttons will set/clear all messages. Messaged in the sequential monitor recording can be further filtered to only messages with errors or that have been bookmarked.

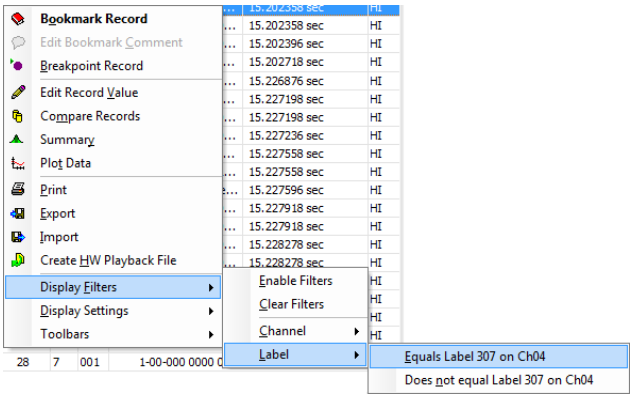


*Sequential Monitor View Display Filter toolbar*




*ARINC 429 sequential monitor view display filter dialog*

Display filters for individual channel and label items can be modified to filter for or filter out the selected item from the context menu as shown in the image below.




*Sequential monitor view display filter configuration from the context menu*

### Searching

Monitor records may be searched by label only, or a combination of label and value or error condition. Edit the search criteria in the Edit Search Criteria dialog. All checked labels will be a part of the search. Alternatively, you can search for bookmarked records. A search can be combined with data values by choosing the Engineering Value option button and defining the data relation.



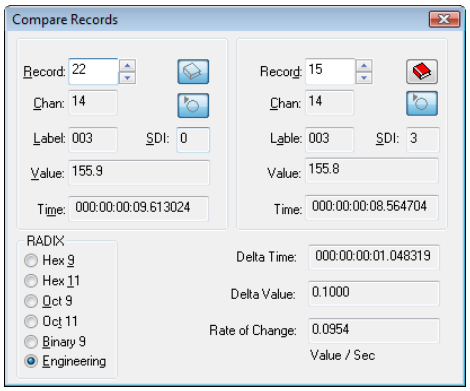
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The data criteria will apply to all checked labels. When the bookmark option is selected, all other criteria is ignored.



### Comparing records

The Compare Records window provides a way to compare the values and time-tags of two monitored records. As the image below shows, Compare Records allows side-by-side comparison of records. The display shows the record number, channel, bookmarks, breakpoints, label, SDI, time (time-tag), delta time, delta value, rate of change, and the value in a variety of different radices.




*Compare Records of a ARINC 429 Monitor View*

### Display Columns

- **Fields**—List of defined data fields (displayed using engineering units)
- **Fields (Double)**—Save as Fields except all data is represented using a double value
- **Eq#**—The Channel equipment number (if detected/specified)
- **Equipment**—The text string associated with assigned equipment number for the channel
- **Lbl #**—The label number of the message
- **Name**—The text string associated with the label number
- **Octal Data**—The message data represented in Octal format
- **SDI**—The SDI number of the message
- **SSM**—The SSM number of the message
- **Value**—The message data represented using the radix format selected in the Display Settings
- **Activity**— Channel speed and error conditions in messages and fields are shown as a text string in the activity column (the level of error detection varies by hardware). Activity data is available in all Monitor View display modes.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| Activities Reported in 429 Monitor View | | | | |
|---|---|---|---|---|
| Symbol | Description | BUSBox OmniBus 4G & 5G | LC429-3 LP429-3 | CM429-1 |
| HI | Message monitored at high speed | ✔ | ✔ | ✔ |
| LO | Message monitor at low speed | ✔ | ✔ | ✔ |
| ERR | Error detected | ✔ | ✔ | ✔ |
| PAR | Parity error detected in message | ✔ | ✔ | ✔ |
| GAP | Gap preceding message was less than 4 bit times | ✔ | ✔ | |
| LONG | Message was longer than 32 bits | ✔ | ✔ | |
| BIT | Long or short bit was detected in message | ✔ | ✔ | |
| TO | Decoder timed out while receiving the message | ✔ | ✔ | |

- **Ch#**—The channel number of the message
- **Parity**—The message parity
- **Time**—The message record time-tag represented using the Time format selected in the Display Settings

### *Protocol Browser*

**Note**: *CoPilot 6 introduced the Ribbon Interface and simplified the Protocol Browser with the new Activity View. See New in CoPilot 6.*

The Protocol Browser can be used to view the current activity on ARINC 429 channels. See *Engineering View Column Options* on page *255*.

# Viewing ARINC 429 Data

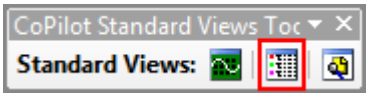
## Engineering View with ARINC 429 Data

#### Engineering View Overview

The Engineering View has replaced the ARINC 429 View from previous CoPilot versions with the capability to display ARINC 429 data simultaneously with data from other protocols. Previously saved ARINC 429 Views are automatically converted to Engineering Views.

The Engineering View displays the latest value of messages and fields in a list with the ability to configure whether or not data columns and graphs are shown. Multiple Engineering Views can be used to group specific information together using different column configurations.

An Engineering View can be created from the Engineering View icon in the CoPilot Standard Views Toolbar (version 5) or the **Create** group of the **Home** ribbon tab (version 6).




*Creating an Engineering View from the Standard Views Toolbar (version 5)*

#### Display Customization

Messages and subaddresses often contain several data words whereas fields usually define a single fact. Consequently, it may be useful to use several Engineering View displays in a project. Each one may be customized through the selection of data properties, format, column width, and position to emphasize



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

important information. Multiple views can also be used to group related information in common views.

#### Engineering View Characteristics

- Labels and fields (and other protocols) are assigned to an Engineering View window by dragging a field from the Hardware Explorer to the Engineering View.
- Each item is entered in a row separated into columns. A context menu for each item provides access to properties.
- Users specify which attributes will be displayed and any combination of engineering unit, binary, octal, or hexadecimal data display.
- Drag either edge of column titles to change column width and drag entire columns to new positions.
- Users may save attribute selections, column width, and position as the Engineering View default for additional Engineering Views or for new projects.
- Color coding and special symbols simplify identification of items in Engineering View and the tracking of transmission intervals that exceed the timeout specification and out-of-range (operating limit) data conditions.

Messages are typically transmitted between five to ten times each second. Data in the Engineering View window is sampled to facilitate readability.

## *Assigning the Data to Watch*

#### Adding items

ARINC 429 Messages and fields are assigned to an Engineering View window by dragging a message or field from the Hardware Explorer tree to the Engineering View window of choice. When a field is added and the parent message is present in the view the field is added as an indented child of the message.

## *Engineering View Display Status Indication*

CoPilot uses icons and coloring in the Engineering View window to indicate status information.

#### Reporting Broken Links

If a label or field is removed from the Hardware Explorer tree after being assigned to an Engineering View window, references to that label are marked with a red . Otherwise, the status icon indicates the object type: (ARINC 429 label) or (ARINC 429 field).

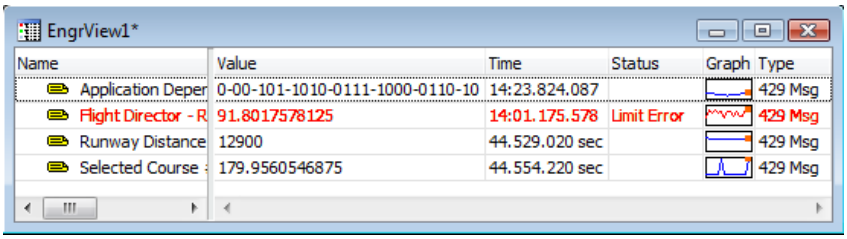
#### Reporting Out of Range and Timeout Conditions

Users can define a lower and upper operating range for scalar messages and fields (e.g., BNR, BCD, etc.) through the scalar tab of the message or field property page. When values are outside that range, the item icon is outlined in red to indicate an error/warning. Since the BNR and BCD interpreter property can be accessed prior or during a simulation, it is easy to modify operating limits to highlight conditions of interest.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When the interval between receipt of a specific message exceeds the timeout value set in message properties, the font color of the name, value, and time-tag will be changed to red.

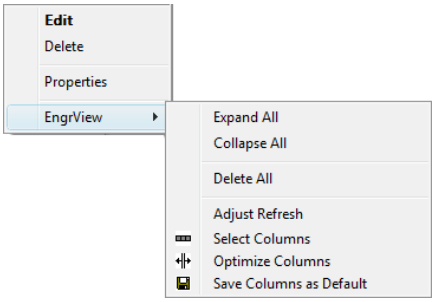



*Engineering View with ARINC 429 data showing a message with a Limit Error*

## *Engineering View Display Menu*

### Context Menu Commands

Right clicking on the background of an Engineering View or from the 'EngrView' menu item accesses the View context menu. This context menu provides access to configure the selected View. An Engineering View window can also be customized using the mouse to manipulate the display.




*Engineering View context menu commands*

- **Delete** removes the selected item from the view
- **Edit** opens an editor to edit the data
- **Properties** opens the object property window
- **EngrView** shows the view context menu items when an item is clicked
- **Expand All** expands all collapsed nodes for items inserted in the view
- **Collapse All** collapses all child nodes for items inserted in the view
- **Delete All** deletes all items in the view
- **Adjust Refresh** allows the user to set a refresh rate for the view contents when the simulation is running. However, when not running, data in view updates at a much slower rate.
- **Select Columns** **o**pens the Select Columns to Display Dialog window. Use this window to specify display columns.
- **Optimize Columns** **e**xpands or contracts all displayed columns to fit its contents
- **Save As Default** **s**aves the current display settings to the registry to be used as the default setup when new views are created

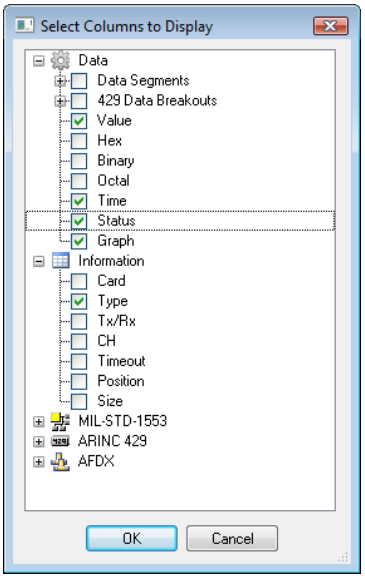


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Available display columns

### Available Column Groups

The Select Columns to Display dialog allows the user to define which types of information are displayed. (The Image column may not be hidden.)




- **Data**: This category allows data to be displayed in several radices and provides additional data. The Value column displays the interpreted value as a real number (floating point); conversely, the Hex column displays the interpreted value as a Hex string. Other columns such as individual data segments, status, and graphs are in this category.
- **Information**: The category includes background information for the item such as type, card, position, and size.
- **MIL-STD-1553**: Contains MIL-STD-1553 specific columns.
- **ARINC 429**: Contains ARINC 429 specific columns.
- **AFDX**: Contains ARINC 664/AFDX specific columns.

## Customizing the display

### Column Selection

Use the Select Columns to Display dialog to specify the columns of interest. Open the dialog by selecting **Select Columns** from the view context menu. Click a checkbox to select or deselect an attribute column.

### Hiding Columns

View context menus opened from a column title will contain a command to hide the current column. Use this command to quickly hide individual columns.

### Column Order

You can reorder columns by holding while dragging a column title left or right.

### Column Width

Use the optimize columns context menu command to auto fit the column contents into a column. Alternatively you can drag the right edge of a column title to resize the column.

### Sorting

Items in an Engineering View can be sorted by the contents of any displayed column. The current sort column has an arrow indicating the sort direction appended to the column title. Click on a column title to set that column as the sort column. Re-click on the sort column title to reverse the sort direction.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### Saving current settings as default

Select **Save As Default** from an Engineering View context menu to save the current display settings to the registry to be used as the default setup when new views are created.

### *Engineering View Column Options*

#### Move Columns

To move a column in an Engineering View:

- Left click and hold the column that will be moved
- Drag it to the intended location then release the mouse button.

#### Resize Columns

To change column width:

- Left click on the right edge of the column title and
- Hold while dragging left or right, then release.

To auto-size columns (fit to width):

- Double click on the right edge of the column title

#### Sort Data

To sort the contents in ascending or descending order:

- Position the mouse pointer in a column title and click the left mouse button once. To reverse the order, click the column title again.

#### Show/Hide Columns

Columns can be removed through the column context box. Right clicking a column title header brings up a column context window.

If a column is hidden, it can be restored through the Organize Columns window. The Image column (which displays a descriptive icon for each type of object) cannot be hidden.

#### Column Order

CoPilot users can rearrange columns according to preference. This is achieved by dragging the column header titles to the desired position.

#### Save Settings

If you wish to save your column configurations as the new default for additional Engineering Views, choose the Save As Default command from the Engineering View context menu. If you wish to preserve the columns without changing the defaults, you can save the Engineering View component (File | Save as).

## Protocol Browser



**Note**: *CoPilot 6 introduced the Ribbon Interface and simplified the Protocol Browser with the new Activity View. See New in CoPilot 6.*

The Protocol Browser provides a fast overview of what is happening on ARINC 429 (and other) databuses. The Protocol Browsers allow users to browse into various levels of the protocol to see the activity and statistics. The current and



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

historical state of objects are displayed using color-coded icons. Drag an ARINC 429 object into a Protocol Browser to view the details of that object and the summary of the objects below it. For example, displaying details for a channel will also show a summary of all the messages (labels) on that channel. Double-clicking a particular item in the Protocol Browser drills into that object.

## Graphical Displays

Graphical displays for displaying ARINC 429 data include Quick Controls, Strip Charts, and ActiveX controls in Control Views. CoPilot Professional provides these data control and visualization tools. *Part 3* of this document beginning on page *367* describes the features of CoPilot Professional in detail.

# The CoPilot 429 Database

## Overview

The ARINC.MDB database shipped with CoPilot is the key to creating and translating ARINC 429 data. That information is used to define channel and message assignments in the Hardware Explorer tree, to create or present data in a readable form and to associate data with graphical ActiveX controls. After data is imported from the ARINC database into a CoPilot project, it can be modified or supplemented as required. Those changes become a part of the project file but not a part of the ARINC database.

### User Modification

The ARINC.MDB database can be modified to include changes in the ARINC 429 specification or special requirements within your organization. This may include new Equipment IDs and the assignment of new or additional fields or messages.

### The ARINC Database Simplifies Message Selection

One of the many advantages of the ARINC protocol is that most labels are identified with specific equipment and the message structure of labels is already defined. All of that information is recorded in the ARINC 429 database incorporated into the CoPilot system. Consequently, the creation of the Hardware Explorer tree is a simple process. Equipment is selected from a list of predefined systems. Once selected, labels are also selected from the short list of labels identified with that Equipment ID. This is done through the channel context menu.

## Loading from the database

### Assigning Equipment ID

See the *Setting the Equipment ID* section on page *220*.

### Adding Messages to Channels

See the *Adding Messages to a Channel* section on page *236*.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### Adding Graphical ActiveX Controls to a Control Form

Graphical ActiveX controls are automatically loaded from the database when ARINC 429 messages or fields are dropped on a Control Form from the Hardware Explorer. This is described in more details in the *Professional Displays* section on page *369*.

## Saving to the database

CoPilot ships with a pre-populated ARINC 429 database and allows the included definition to be extended or replaced for custom implementations or changes.

To extend a definition first load the definition from the database, alter the definition, then save the new definition into the database.

#### Document Changes with Comments

Information added to the database is typically an exception to the published ARINC 429 specification. Since the database can be shared with others, it is a good idea to add a comment to all new information added to the database. Comments could include initials of the author, data source and date.

#### Saving Equipment ID Definitions to the Database

When saving Equipment IDs to the database all Messages and fields associated with the Equipment in the Hardware Explorer, at the time of the save, are saved with the equipment. All database loading operations (see Loading from the database above) will now include this new Equipment definition.

To save an Equipment ID to the database:

Select the **Save EquipID to Database** option in a channel context menu from the Hardware Explorer tree

#### Saving Message Definitions to the Database

When saving Messages to the database all fields associated with the message in the Hardware Explorer, at the time of the save, are saved with the message. All database loading operations (see Loading from the database above) will now include this new message definition. If SDI-specific messages are defined, then each individual SDI message can also be saved to the database.

To save a Message to the database:

Select the **Save to Database** option in a Message context menu from the Hardware Explorer tree

#### Saving Field Definitions to the Database

After saving a field to the database all database loading operations (see Loading from the database above) will now include this new field definition.

To save a Field to the database:

Select the **Save to Database** option in a Field context menu from the Hardware Explorer tree

#### Saving ActiveX Control Links to the Database

The Control View, part of CoPilot Professional, allows the creation of graphical controls to show engineering values in a visual manner. These control



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

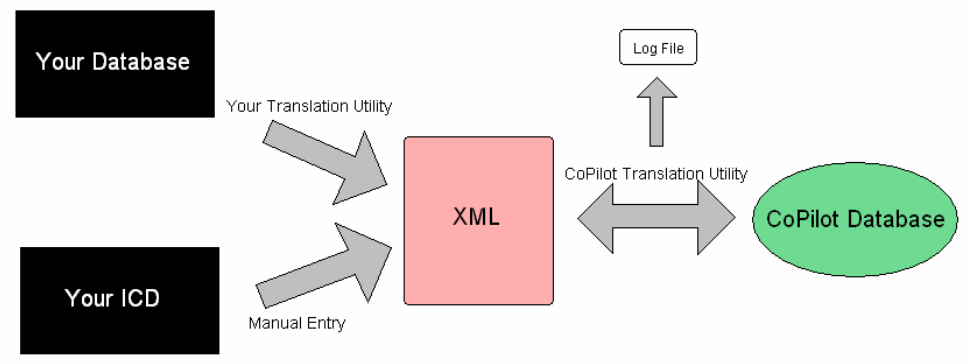
configurations can be saved to the database associated with a particular field or label. This is described in more details in the *Professional Displays* section.

## Sharing a database

The power of the database is in its reusability. You organization may have its own set of Equipment and/or Message definitions. By sharing common CoPilot database files with multiple users, all users will benefit from modifications made to the database. Once a definition is added or updated the next time any user loads that definition it will reflect the changes.

## XML database import/export utility

A utility application (BTIMDB.EXE) is provided with the CoPilot software to facilitate the definition of ARINC 429 labels and fields. This application can be found on the CoPilot CD. Utilizing a defined XML interface, users may translate their own ARINC 429 data definitions located in their own ICDs (Interface Control Documents) or databases. The XML file describing the data definitions can be imported using the utility into the CoPilot ARINC 429 database.

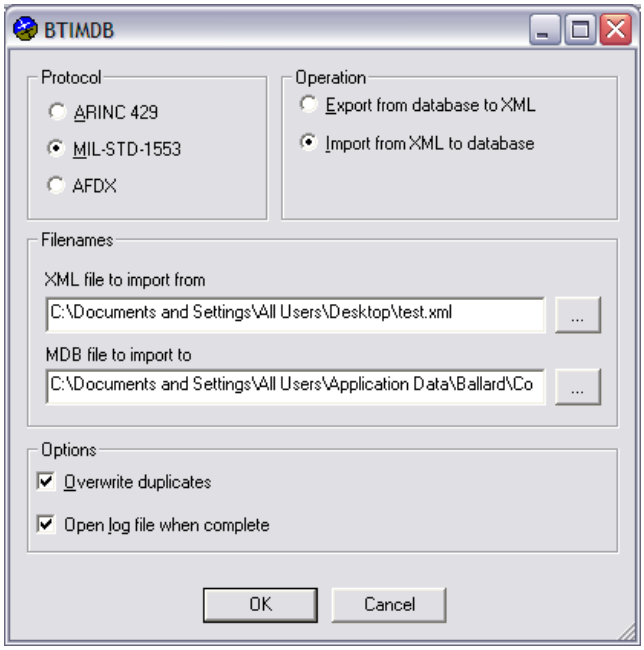



Through the use of this utility, the ease of providing data definitions for use by CoPilot goes up dramatically. Please see the "CoPilot Database Import-Export Utility-XML Format.PDF" located on the CoPilot CD.

The following image shows a screenshot of the utility, which allows you to specify various options such as which protocol database to import/export, the location of the XML and database files, and other options.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




## Helpful Hints

### Avoiding Unintended Effects of Automatic Detection

If Auto-detect equipment ID is enabled on a receive channel, the equipment ID is updated when label 377 (the equipment identifier label) is received. If the detected equipment ID is different from the previous value, then message descriptions and the interpretation of the message in the Hardware Explorer tree are updated to match the definitions found in the ARINC 429 database. The auto-detect equipment ID feature should not be checked if you want to ensure your message names and interpretations will not change. Additional message or updated message definition can be saved in the ARINC database through the **Save to Database** option of the message context menu.

# Error Injection

### Inverting Parity

When the **Parity Error** option is selected on a 429 (transmit) Message, the parity bit is reversed. This can be used to inject errors on ARINC 429 channels. This option is not available on the CM429-1 board.

### Inter-Message Gap Time

The ARINC 429 Specification states that ARINC 429 message must be separated by at least a 4-bit time gap. The CoPilot Professional Custom Schedule feature allows the user to specify less than 4-bit time gaps. To learn more about Custom Schedules see the *Custom User Schedules* section on page *228*.

### Parametrics

- **Custom speed**—Parametric custom speed can be set on individual hardware channels (device dependent)



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Amplitude**—Parametric signal amplitude can be set on individual (transmit) channels (device dependent)

# ARINC 429 Scripting Events

### Overview

This section list and describes the events fired from various ARINC 429 objects within CoPilot. For more detailed description of CoPilot scripting see *Automated Test Environment (ATE)* on page *399*. Refer to the *CoPilot Object Model* documentation in the Common Help Topics section of the Start Page (version 5) or the **Help and Learning** tab group on the **Backstage** (version 6) for a full description of all supported properties and methods.

### ARINC 429 Channel Events

- **`OnAutoDetectLabel(`**`nLabelNum`**`)`** — Occurs when the channel auto-detects a new label.
    - nLabelNum— The label value of the auto-detected message
- **`OnPauseChange(`**`bPause`**`)`** — Occurs when the channel's pause state changes.
    - bPause—The new pause state of the channel (TRUE or FALSE)

### ARINC 429 Message (Label) Events

- **`OnIntervalError()`** — Occurs when the message detects an interval error.
- **`OnLimitError(`**`bHigh`**`)`** — Occurs when the message's value is outside the engineering unit interpreter's specified operating limits.
    - bHigh—Indicates whether the High or Low limit was exceeded (TRUE or FALSE)
- **`OnLimitValid()`** — Occurs when the message's value returns to a valid range from a range that was outside of the interpreter's specified operating limits.
- **`OnMsgUpdate(`**`ValueRaw, activity, timetag, timetagh, mintime, maxtime`**`)`** — Occurs when a message object data structure is updated. For more information on the parameter values see the `MsgBlockRd(…)` function in your hardware manual.
    - ValueRaw—The 32-bit message
    - activity—Message activity
    - timetag—The low 32-bit value of the message time-tag
    - timetagh—The high 32-bit value of the message time-tag
    - mintime—Minimum repetition rate
    - maxtime—Maximum repetition rate
- **`OnPatternError(`**`bPatternError`**`)`** — Occurs when the message detects a pattern error change. The bPatternError value equals zero with no error and non-zero when there is an error.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **`OnTimeTagStale(bStale)`** — Occurs when the message's time-tag becomes stale. A message time-tag becomes stale when the delta time since the last transmission on the databus exceeds the timeout value.
    - bStale — The stale value of the time-tag (TRUE or FALSE)
- **`OnValueChange(varValue)`** — Occurs when the message's engineering unit interpreted value changes.
    - varValue —The new engineering unit interpreted value of the message
- **`OnValueStringChange(bstrValue)`** — Occurs when the message's text string interpreted value changes.
    - bstrValue —The new text string value of the message

**ARINC 429 Field Events**

- **`OnEnableChange(bEnabled)`**— Occurs when the enable state changes.
    - bEnable — The new enable state (TRUE or FALSE)
- **`OnLimitError(bHigh)`** — Occurs when the field's value is outside the engineering unit interpreter's specified operating limits.
    - bHigh—Indicates whether the High or Low limit was exceeded (TRUE or FALSE)
- **`OnLimitValid()`** — Occurs when the message's value returns to a valid range from a range that was outside of the interpreter's specified operating limits.
- **`OnTimeTagStale(bStale)`** — Occurs when the field's (message) time-tag becomes stale. A message time-tag becomes stale when the delta time since the last transmission on the databus exceeds the timeout value.
    - bStale — The stale value of the time-tag (TRUE or FALSE)
- **`OnValueChange(varValue)`** — Occurs when the field's engineering unit interpreted value changes.
    - varValue —The new engineering unit interpreted value of the message
- **`OnValueStringChange(bstrValue)`** — Occurs when the field's text string value changes.
    - bstrValue —The new text string value of the field

Note: In addition to the events listed above, the CoPilot Object Model documentation describes the available properties and methods. Open the CoPilot Object Model documentation from the Start Page (version 5),the **Help and Learning** tab group on the **Backstage** (version 6), or the Help menu for the latest information.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# ARINC 708 Protocol

## ARINC 708 Overview

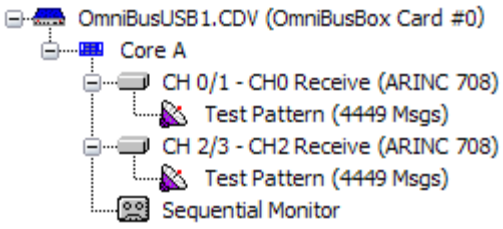
### ARINC 708 Protocol Summary

ARINC 708/453 describes the characteristics of an airborne pulse Doppler weather radar system intended for installation in commercial transport type aircraft. Display data and system mode/status information is transmitted over the display databus from the Transmitter-Receiver unit (T-R) to the Control/Display Unit (CDU). The signal characteristics on the display databus are very similar to MIL-STD-1553, but differ in length, following sync, and absence of parity. Data on the display databus consists of 1,600-bit messages (words) that are preceded by and followed by a sync.

### Hardware Explorer Object Hierarchy

The Hardware Explorer pane holds the object representation of physical hardware devices along with display settings. These definition of hardware devices are simplified by breaking them down into a hierarchical tree. Each node of the tree holds configuration settings (transmit and sample rates, schedules, settings, etc…) for the levels beneath them.

The figure below shows the Hardware Explorer tree displaying multiple object levels in a hierarchical structure.




These levels can include the:

- Ballard 429 hardware to be used. **File** | **New** | **New Hardware and Views** menu provides access to new hardware selection.
- (Optional) Cores. Cores represent groups of channels that share a Sequential Monitor, discrete banks and some configuration settings.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Channel Pair Configuration. A channel pair consists of one transmit and one receive ARINC 708 channel.
- Datasets. Datasets are sequential data recording files with display radar settings.

# CoPilot for 708 Overview

## Simulating a Weather Radar System in CoPilot

ARINC 708 weather radar systems include a T-R (Transmitter-Receiver) unit, a display unit, and a control panel. Often the display and control panel are combined into a CDU (Control-Display Unit). CoPilot can transmit, receive, monitor, and display ARINC 708/453 display data. In addition, CoPilot provides a variety of tools and features to enhance the analysis of weather radar data.

**CoPilot for 708 Overview**

The following sequence of topics summarize the capabilities of CoPilot for 708. New users can quickly see the "big picture" and discover the benefits and features of CoPilot.

### *Transmitter-Receiver Unit*

In a typical weather radar system, the T-R unit performs two functions. It relays the results of radar pulses to the display unit via ARINC 708/453 and accepts and responds to control data from the control panel via ARINC 429.

CoPilot simulates the T-R unit by resending previously recorded data onto the 708 databus. You can modify this recorded data in special editing windows before transmission. Transmit options allow the group of 708 messages being transmitted to loop continuously or a specified number of times. You can also view this data as a radar image while it is being transmitted.

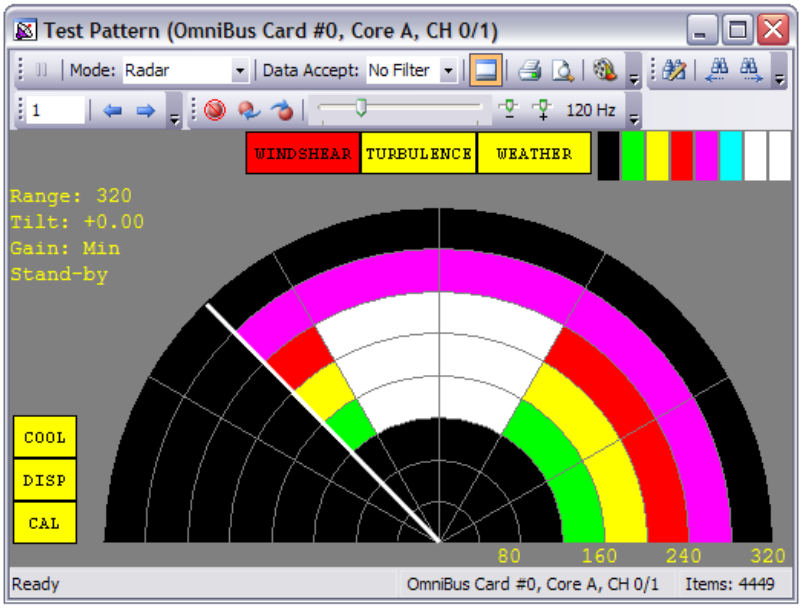
### *Display Unit*

In a typical Doppler radar system, the display unit receives and processes ARINC 708/453 data from the T-R unit and creates a continuous display of detected weather, windshear, turbulence, terrain mapping, etc.

In CoPilot, the weather radar display is simulated in the display window. You can view data as a radar image as you monitor the 708/453 databus. The radar display is fully configurable, including weather colors, status, faults, warnings, and sector scan display area.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




CoPilot can also simulate a system that contains two display units (identified as Indicator 1 and Indicator 2, as described in the ARINC 708 protocol). When monitored data is addressed to two display units using the Data Accept field, CoPilot can display two radar screens and filter data to the correct screen. In addition, a single radar screen can be filtered by the Data Accept field.

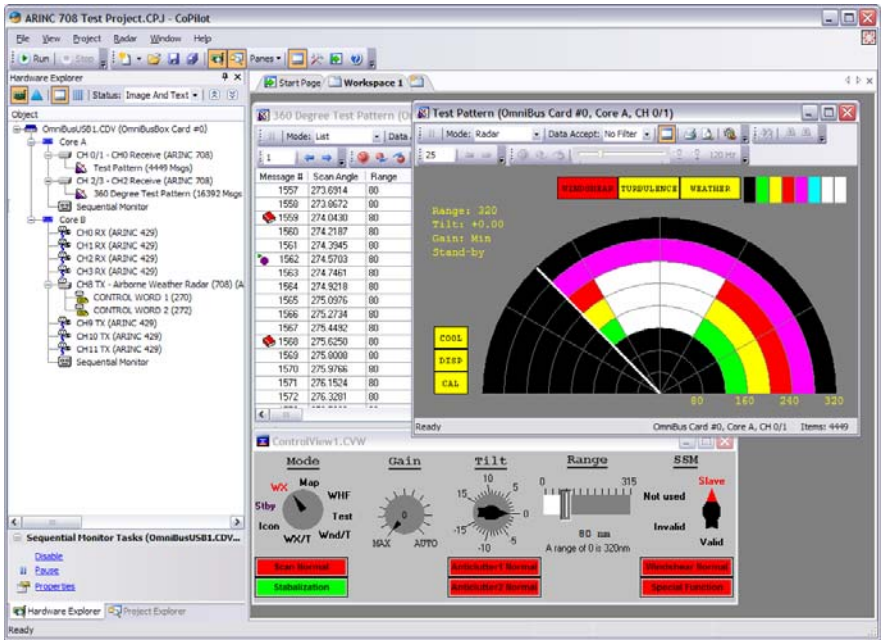
## *Control Panel*

Standard weather radar system controls are range, tilt, gain, mode, etc. These controls may appear on the display unit (CDU), in a separate control panel, or a combination. Control data is sent to the T-R unit via the ARINC 429 databus.

Because CoPilot is a unified environment that supports multiple avionics protocols, CoPilot users can combine ARINC 429 with ARINC 708 in a single CoPilot project to control a T-R unit (see figure below). The T-R unit can be controlled through the panel of virtual instruments (see the "ControlView1.CVW" window in the figure below).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




# CoPilot for 708 Simulation and Analysis Tools

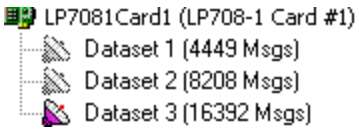
CoPilot contains many software tools and features to enhance the analysis of ARINC 708/453 display data.

- CoPilot can record monitored display data for later use
- Saved data can be transmitted, copied, edited, exported, and imported into a CoPilot project
- Monitored data can be replayed and analyzed in CoPilot using software playback

## *Data Recording*

As you monitor the display databus, CoPilot automatically collects and stores received messages, along with time-tags and error conditions detected by the hardware, in a "dataset." (A dataset is a sequential collection of ARINC 708/453 1,600-bit messages.) CoPilot adds messages to this dataset as they are received until reception is paused or halted. You can pause and resume the recording at any time to omit a segment of bus activity.

Each dataset and message count is displayed with the associated board in the Hardware Explorer tree.




Once a dataset has been created, you can save and reuse it or overwrite it.

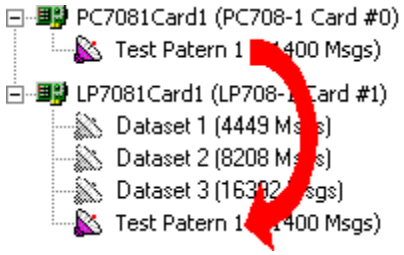
## *Data Propagation*

As CoPilot monitors the bus, it collects 708/453 messages into a reusable dataset. All datasets listed under the 708 board in the Hardware Explorer tree are saved with this board when the CoPilot project is saved unless explicitly deleted



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

by the user. You can copy a dataset from one 708 board to another through a drag and drop procedure.

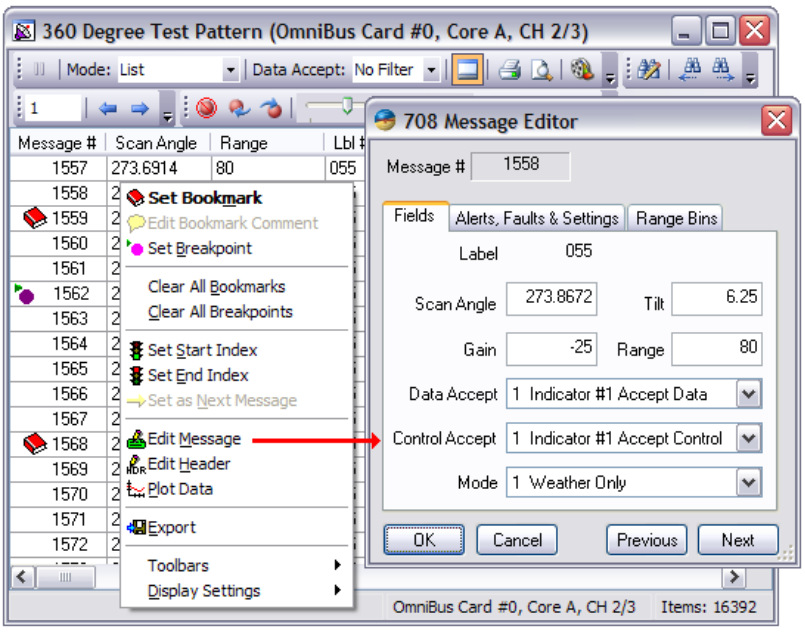



The raw data (1,600-bit messages only) from a dataset may also be saved to a file (in a .dat data file format). This allows collected data to be used in another CoPilot project, with Ballard's 708 Utility program, or other application. When a 708/453 data file (in .dat format) is imported into CoPilot, it appears as a dataset in the Hardware Explorer tree.

Any dataset in the Hardware Explorer tree may be examined in software playback mode or transmitted on the display databus.

## *Displaying and Modifying Messages*

The RadarView window can be selected to display the 708/453 messages in a list format for analysis and editing. Each message in the list is interpreted and displayed in engineering units based on the 055 label defined in the ARINC 708 protocol (see figure below). Users can navigate through the message list using the Search and Step functions, bookmark selected records for future reference, and view range bin data.




Individual values for each 708 message field and the 512 range bin values can be viewed or modified in engineering units through the 708 Message editor.

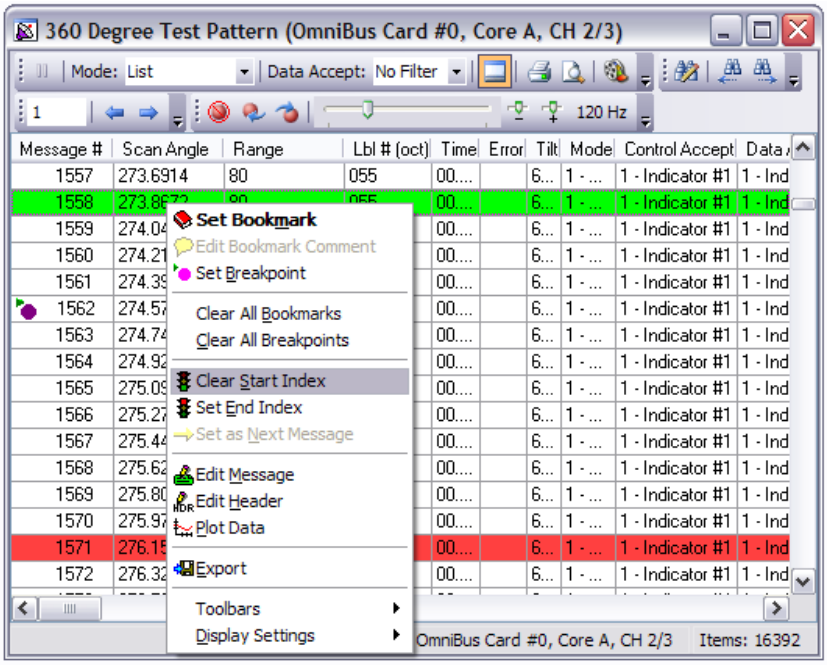
## *Analyzing Data with Software Playback*

CoPilot provides a special software playback mode that allows recorded data to be analyzed in detail. In software playback mode, interaction with the Ballard



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

708 card and ARINC 708 databus is suspended. A previously recorded or imported dataset is “played” as if it were being received or transmitted. The replayed data may be viewed in the radar display, just like transmitted or received data. In the List view display, you can configure a dataset for software playback.




Software playback tools and features provide the means of controlling the playback for optimum analysis. Playback features include:

- **Pause**—Suspend playback at any time with the pause button
- **Step**—When paused, you can advance the playback in steps (choose a step size of 1 to 100 messages)
- **Speed**—Increase or decrease the speed of playback using the speed controls in the playback tool bar
- **Breakpoints**—Use breakpoints to pause the dataset at specified messages
- **Skip**—When playback is paused, you may specify a new resume point, thereby skipping a segment of the dataset
- **Index**—Select a subset of messages for playback be setting the start and end index
- **Loop**—Choose to replay a dataset a set number of times or to loop indefinitely until halted



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Monitor and Display Data

## How to Monitor and Display Data

### Monitoring the Display Databus

When monitoring an active display databus, CoPilot collects monitored messages (with time-tags and detected errors) in a "dataset," and lists that dataset and message count in the Hardware Explorer below the 708 board or channel.

### What is a Dataset?

A dataset  is a CoPilot object (located in the Hardware Explorer) that holds a sequential collection of ARINC 708/453 1600-bit messages. Each dataset object holds radar display configuration settings (default or user-defined). Datasets can also contain time-tag and error information (for recorded data) and user-assigned formatting such as bookmarks and breakpoints.

### Multiple Display Formats

In CoPilot, weather radar data can be viewed in a radar screen display, a label 055 interpreted message list and in a raw hex list format. All of these displays are hosted in a single display window that is associated with and launched from a dataset in the Hardware Explorer tree. This display can be modified through menus options and tool bar buttons to enhance the analysis and display of ARINC 708/453 data.

- **Radar Screen**—Data from the display databus can be viewed as a radar image. The Radar view display is fully configurable, including weather colors, status, faults, warnings, and sector scan area. Data addressed to an individual display unit by the Data Accept field can be filtered out of a single radar screen or separated into two radar screens (left/right or pilot/copilot).
- **Message List**—The List view display interprets and displays each message in engineering units based on the 055 label (defined in the ARINC 708 protocol). The List view display can be customized to show or hide range bin data and header field columns. You can bookmark records for future reference and navigate through the message list using the Step and Search functions.
- **Raw Hex**—The Raw Hex view displays un-interpreted message data in 10 x 10 hex grid layout. Like the List view display, the Raw Hex mode can be customized to show or hide header field columns. You can bookmark records for future reference and navigate through the message list using the Step and Search functions.

## Monitoring Data

CoPilot makes it easy to monitor display data from the 708/453 display databus. You can view this data graphically while you are monitoring.

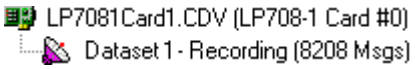


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To monitor data:

1. Right click on the 708 card (or 708 channel pair ) icon in the Hardware Explorer tree and choose Receive from the context menu to select Receive mode
2. Click the CoPilot Run button to begin simulation
3. Click the CoPilot Stop button to return to Edit mode

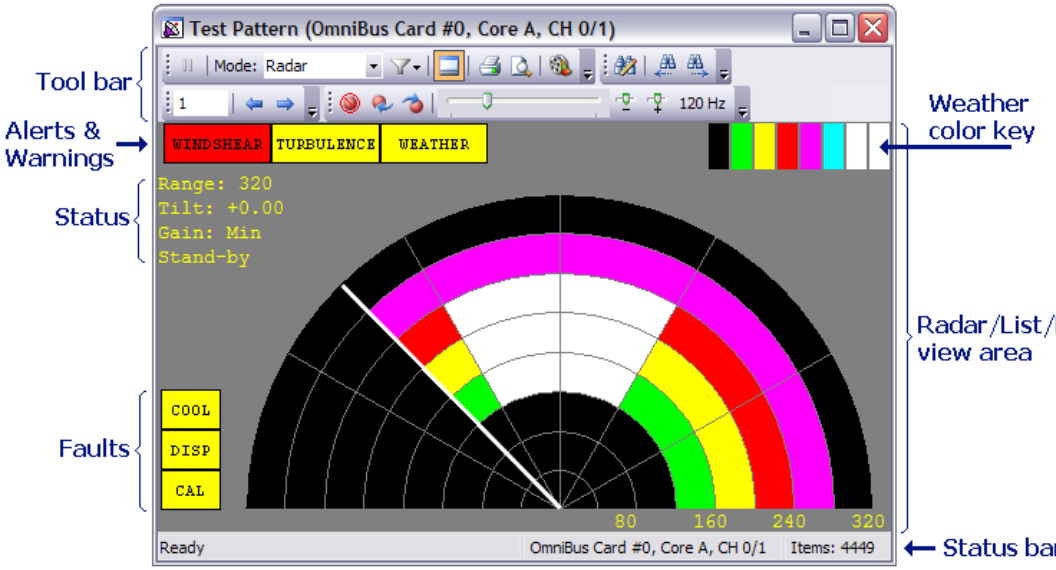
As data is received, CoPilot stores it in a dataset. The name and message count of each dataset is listed under the 708 card object in the tree.




## Displaying Data

### Dataset Display Window

In CoPilot, the weather radar display is simulated in the display window. You can view display data as a radar image, interpreted message or raw hex list.




To open the dataset display window:

- Right click on a dataset icon in the Hardware Explorer and select **View** from the context menu to open the view window for that dataset

The dataset display window opens in the display pane area of the CoPilot desktop. You can have several display windows open at one time. Use the Window menu to manage multiple windows.

---

**Tip:** Double clicking on a dataset (in both Edit mode and Simulation mode) will automatically open the dataset display window.

---

### Radar, List and Raw Display Modes

Users can switch between Radar mode, List mode and Raw mode through the view toolbar (see figure above). In Radar mode, raw data from the dataset is



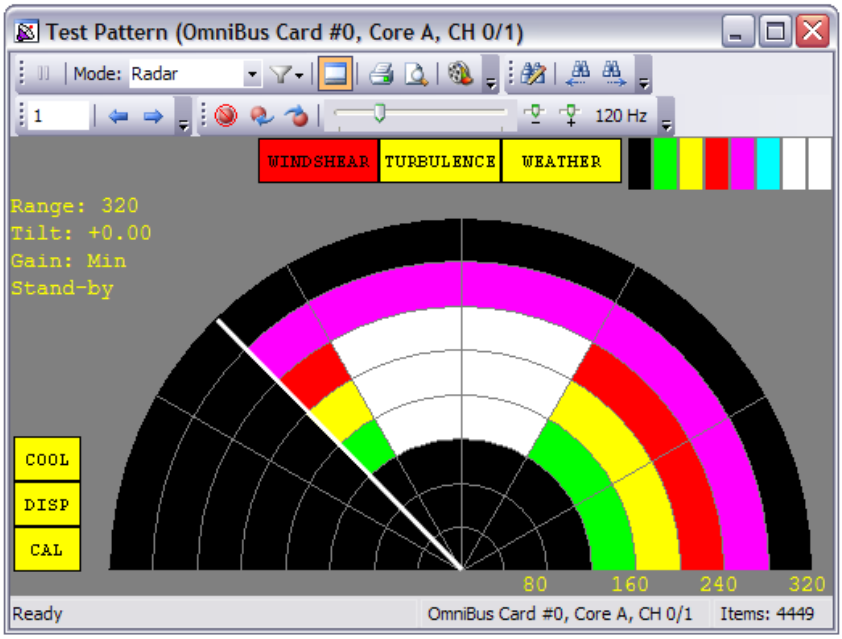
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

translated into a user-configurable graphical radar display. In List and Raw modes, messages can be viewed in engineering units, searched by various criteria, modified, and saved. Those changes will then be visible in Radar mode.

## Viewing the Radar Image

### Drawing the Radar

When using Radar mode, the dataset display window is set to Radar mode, data is visually represented on a radar screen, with a moving wand drawing the image.




To view data as a radar image:

- Double click a dataset in the Hardware Explorer tree to open its display window (defaults to Radar mode) or right click on an open window and choose **Display Settings | Radar** from the context menu
- Click the CoPilot **Run** button to begin the simulation (whether in receive, transmit, or software playback mode, the Run button animates the radar display by starting Simulation mode)

### Configuring the Radar

The radar display is fully configurable by the user. Radar configuration options include the following:

- Configure the radar grid, wand, background color/picture, etc.
- Weather colors may be redefined by the user
- Status, Faults, & Warnings indicators and text may be displayed and configured
- Screen area may be specified by the user as wider or narrower than the actual sector scan area received from the bus



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

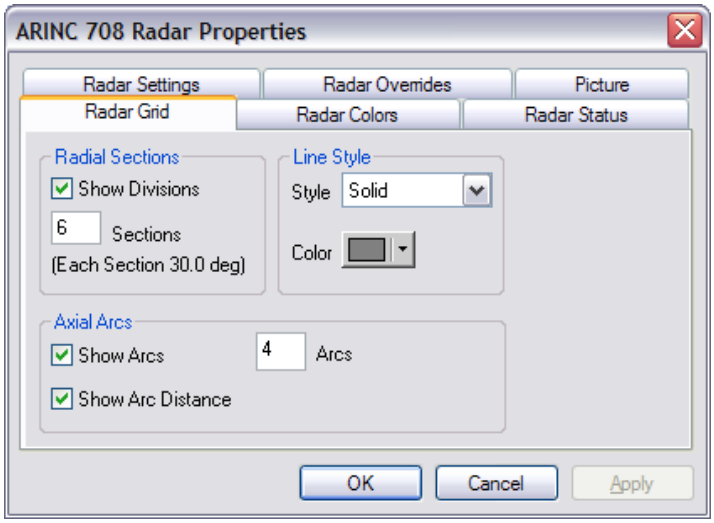
- Default configuration for the radar display can be applied or redefined by the user
- Filter out data by Data Accept field when data is addressed to two display units
- Split the radar into two screens to simulate both displays when data is addressed to two display units

## *Configuring the Radar Screen*

The user can customize the radar screen display through the radar Properties dialog. Users can define unique configurations for each dataset in the tree, and these will be preserved when the project is saved and reopened. In addition, a custom configuration can be saved as the default for new datasets.

To access radar properties:

- Double-click the radar display or right click in the radar display window or dataset icon in the Hardware Explorer and choose **Properties**




**Tip:** Click the Apply button as you go along to view and adjust your current choices without exiting the dialog.

### Radar Grid Tab

To configure radial sections:

1. Click the **Show Divisions** checkbox in the Radial Sections frame (see figure above) to enable or disable radial divisions on the radar display
2. If Show Divisions is enabled, you may specify the number of sections in the textbox

To configure axial arcs:

1. Click the **Show Arcs** checkbox in the Axial Arcs frame (see figure above) to enable or disable arc lines on the radar display
2. If Show is enabled, you may specify the number of arcs in the textbox



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3. If Show is enabled, you may choose to show arc distances on the display

To configure line style:

1. Click the **Style** listbox in the Line Style frame (see figure above) to expand the list of choices and click a line type
2. Click the **Color** button to choose a line color

The line style options affect how the radial sections and axial arcs are drawn on the radar image.

**Radar Settings Tab**

To configure the wand:

- Click the **Visible** checkbox in the Wand Options frame in the Radar Settings tab to enable or disable the wand

If the wand is enabled, the choices below become available:

  - Click the **Thick** checkbox to toggle the wand thickness
  - Choose a wand color by clicking the **Color** button

---

**Note:** To change the sector scan area, see *Changing the Scan Area of the Radar Screen*.

---

To change the background:

- Select from Frame, Image or None in the Background Options frame to set the background.
  - If the None option is selected, the background area will display the color selected for weather target level 0.
  - If Frame is selected, you may specify the color by clicking the **Color** button
  - If Image is selected, you may specify an image on **Picture Tab**

To adjust the radar refresh rate:

- In the **Refresh Rate** frame, slide the bar to the left to reduce the refresh frequency, and right to increase it

The highest refresh rate setting provides the most accurate real-time view of incoming data. If your computer has limited processing power, a lower refresh rate may work better.

To configure the Airplane:

- Click the **Visible** checkbox in the Airplane Options frame in the Radar Settings tab to enable or disable the airplane.
- Airplane draws a small airplane below the Radar image at 90 degrees.

If the airplane is enabled, the choices below become available:

  - Choose a airplane color by clicking the **Color** button

To configure the Max Delta Scan:

- Select the Max Delta Scan (maximum difference of scan angle between two consecutive messages) from the drop down list.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The area between consecutive messages is filled with the range bin data. When the scan angle difference between the consecutive messages exceeds the *Max Delta Scan*, the control will not fill the area between those consecutive angles. Instead, the range bin data is used to draw a sliver at the angle of the preceding value. Then the wand is moved to the location of the following scan angle value.

**Tip:** Many of the commands listed above may be executed directly from the Radar view context menu (right click in radar area to access).

## Specifying Weather Colors

A key to the weather colors is displayed in a band (see figure below) in the upper right-hand of the Radar view pane. The eight colors from left to right correspond to the eight range bin values (0 to 7).

To change weather target colors:

1. Right click in the Radar view display or right click the dataset icon in the Hardware Explorer and choose **Properties** to open the Properties window
2. Select the **Radar Colors** tab
    - Choose a new weather target color (levels 0 through 7) by clicking the appropriate color button
3. Click **OK** to close the properties window




**Tip:** Click the Apply button as you go along to view and adjust your current choices without exiting the dialog.

## Displaying Status, Faults, & Warnings

Text and indicator lights for alerts, faults, and status information, and warnings can be displayed in Radar view.

To display text and indicators:

1. Right click in the dataset view window (or on the dataset icon in the Hardware Explorer tree) and choose **Properties** to open the Properties window
2. Select the **Radar Status** tab
3. Click the **Show Text, Show Faults and Show Warnings** checkboxes to enable or disable

When these features are enabled, the following options become available:

- Click the listbox in the Fault Indicators frame to select the length of time an indicator light will remain on after a fault is detected
- Click the **Set Font** button to specify the size, font, and color of all display text in the Radar image
- Click the **Auto Calculate Text Size** checkbox to allow text to resize with the window



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Click the **Show Timetag** checkbox to show the time-tag value in the status area of the Radar image
- Click **Show User String** to show a user supplied string in the status area of the Radar image
- Enter a user supplied sting in the user string edit box to be shown in the status area of the Radar image

4. Click **OK** to close the properties window

---

**Tip:** Click the Apply button as you go along to view and adjust your current choices without exiting the dialog.

---

## *Changing the Scan Area of the Radar Screen*

The sector scan area display can be set anywhere from 10 to 360 degrees and can be set to a narrower or wider arc than the supplied data.

To specify the visible scan range:

1. Right click in the dataset view window (or on the dataset icon in the Hardware Explorer) and choose **Properties** to open the Properties window
2. Select the Radar Settings tab
3. In the Scan Range frame, enter a value (in degrees) in the textbox




The numbers in parentheses to the right of the textbox update to reflect the range of the radar display image.

---

**Tip:** Click the Apply button as you go along to view and adjust your current choices without exiting the dialog.

---

## *Saving and Loading a Radar View Configuration*

The Radar display properties may be individually defined for each dataset in the tree, and these configurations are saved and reloaded with the project, A default Radar view configuration is automatically applied to created (recorded) or newly imported datasets, but can also be applied to an existing dataset display at any time. Users may define a new default, which is saved to the host system registry.

The dataset display window must be open and set to Radar mode for this command to appear in the context menu.

To load the default settings:

1. In the Radar view display area, right click to access the context menu
2. Choose **Load Default Radar Properties** to overwrite the current settings with the saved default

To save the current settings as the new default:

1. In the Radar view display area, right click to access the context menu

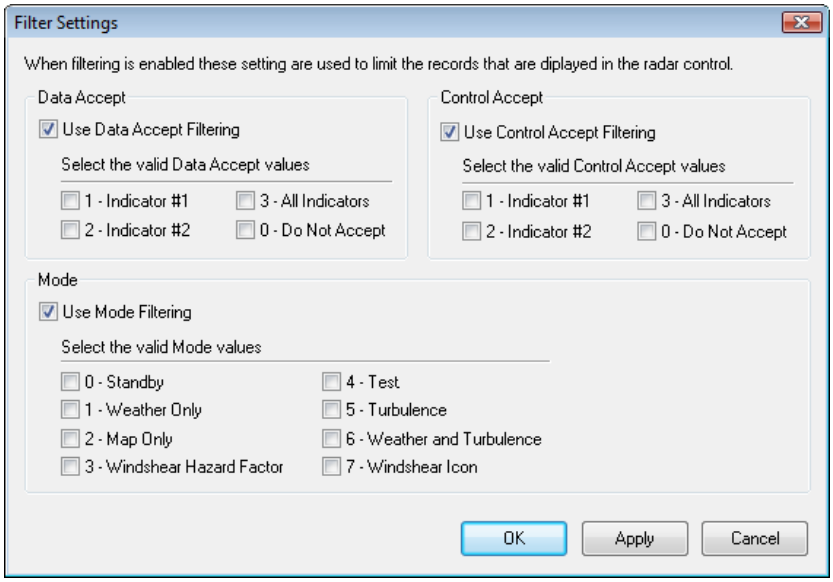


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. Choose **Save Radar Properties As Default** to overwrite the old default with the current settings

## *Filtering the Radar Display*

Weather radar data on the ARINC 708/453 databus can be addressed to one or more display units using a combination of Data Accept, Control Accept and Mode field values. Filtering is used to limit the radar display to only show the data with a certain combination of these values. Filtering affects the radar display only; the collection of messages (in the dataset) is not affected.




*Radar View Filter Settings*

**Tip:** You can also display data in two different radar screens with independent filters.

## *Displaying Data in Two Radar Screens*

CoPilot can simulate a weather radar system that contains two display units. Each radar display has its own filter. For instance, you could set up the two radar displays for Pilot/Copilot (left/right) radar displays using Data Accept Filtering.

A dataset display window must be open and toggled to Radar mode to access and view this feature.

To split the radar into two screens:

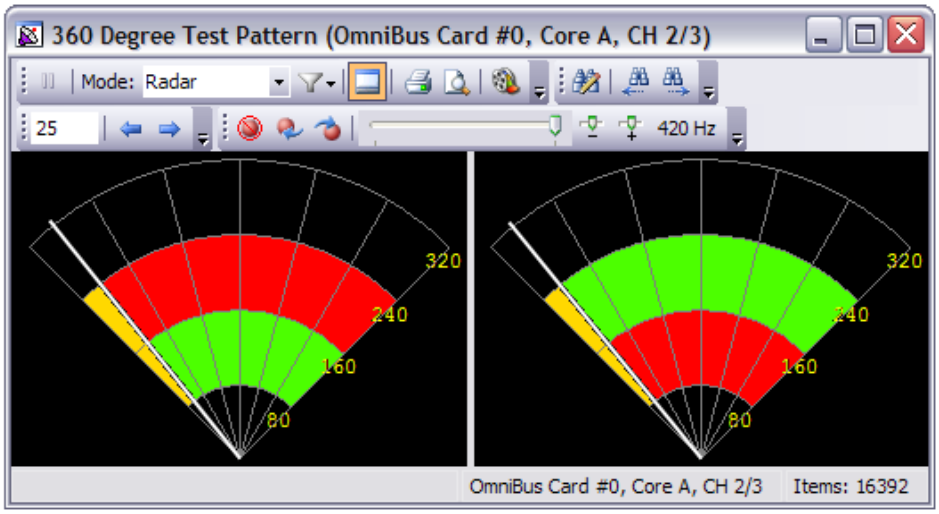
1. Right click on the Radar view pane in the dataset display window and choose **Split Radar** from the context menu
2. (Optional) Drag the bar between the screens or the window edges to resize them

When the Split Radar feature is enabled, the following option become available:

- Right click on the split Radar view pane and click **Equal Split** on the context menu to resize the screens equally



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

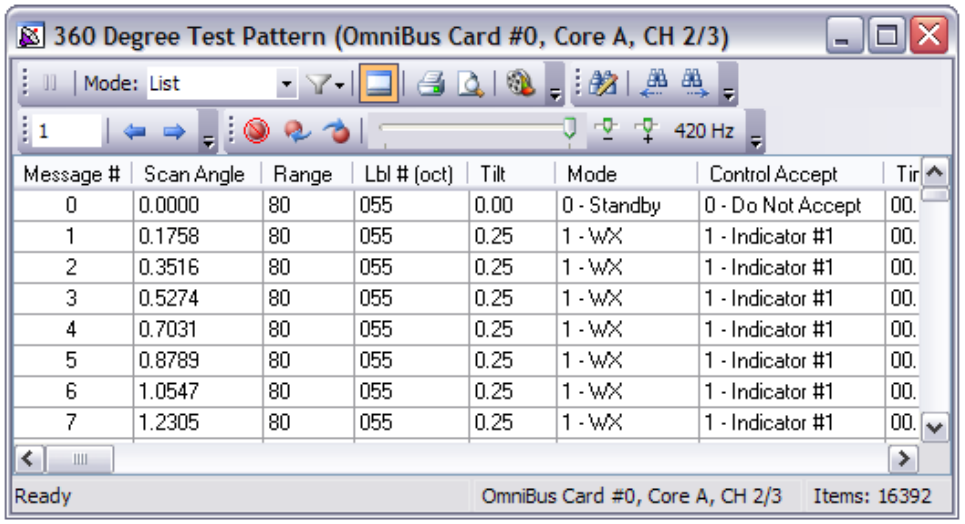



**Tip:** It is also possible to filter a particular Data Accept field out of a single radar screen display

## Viewing the Message and Raw Hex Lists

### Displaying the Message List

Each row in List mode displays the 64-bit header of a message in engineering units, organized into columns (by fields). These fields and engineering units are based on the 055 label defined in the ARINC 708 specification. Range bin data for each message can also be displayed in a separate pane.




The first column displays the message number. If the dataset was recorded (not imported), then monitor data such as time-tag and error information are displayed. Use the table below to determine the type of error:



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| Error | Description |
|---|---|
| TO | Timed out |
| WORD | Word error |
| SYNC | No following sync |
| MAN | Manchester error |
| LONG | More than 1,600 bits |
| SHORT | Less than 1,600 bits |

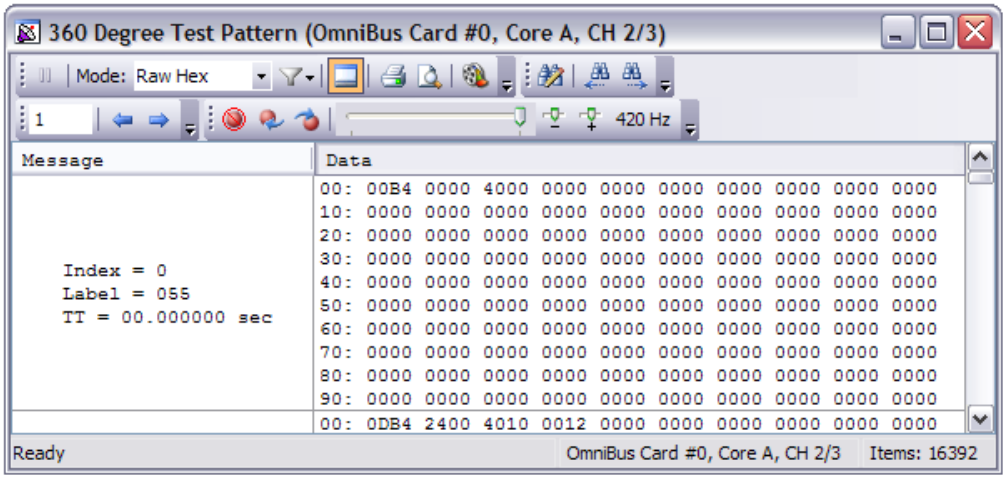
**Note:** Imported datasets do not have time-tags or errors associated with them, so these columns contain zeros in the List view display.

To view data as a message list:

1. Right click on the dataset view window and choose **Display Settings | List** from the context menu (dataset display window must be open)
2. Navigate through the messages in the dataset using the scroll bars, arrow keys, or Step control
3. (Optional) Right click in the message list and choose **Display Settings | Show Range Bin Data** to display range bin data for the highlighted message

### Displaying the Raw Hex List

Each row in Raw Hex mode displays 1,600-bits of the message in a 10x10 grid. Range bin data for each message can also be displayed in a separate pane.




The first column displays the message number, label number and time-tag. If the dataset was recorded (not imported), then monitor data such as time-tag.

### List & Raw Hex Mode Features

Special features and functions are available:

- **Step** through the list in user-defined steps of 1 to 100 messages
- **Search** the dataset by range bin value, fault conditions, or other criteria
- **Bookmark** records of interest for future reference
- **Configure** the columns to optimize the display
- **Range Bin Data** may be displayed in a separate pane



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

In addition, users can select and modify messages in through a hexadecimal editor or engineering unit editor.

## Displaying the 055 Label

### 055 Label Messages

The Weather Radar Display Databus format defined in the ARINC 708 specification is known as "Label 055." This format assigns the first 64 bits (of the 1,600-bit 708 word) as the "header" and allocates the remaining 1,536 bits into 512 three-bit range bins. The first eight bits of the 64-bit header (bits 0–7) contains the 055 label identifier. The remaining 55 bits (8–63) are allocated to fields holding alerts, faults, warnings, flags, and system mode and status information such as range, tilt, mode, control accept, data accept, gain, scan angle, etc. (as displayed in the List view window). Field values could be decimal numbers (40 nm range), a true or false condition (windshear alert true), a discrete condition (weather only mode), etc. The three-bit range bins hold reflectivity and turbulence data (color-coded by the intensity of the return). CoPilot 708 interprets the display databus weather data (i.e., 055 label) so that users may view and modify data in engineering units.

### Custom Messages

Unlike ARINC 429, there is only one label defined for the ARINC 708 weather radar databus. Since only one label is used, naturally non-055 messages are not displayed in the 055 format and engineering units in CoPilot. However, CoPilot can receive and record custom messages that meet the electrical characteristics of the 708/453 bus. To view or edit the bits in a custom message (non-055 label), use the Header Data window. This editing window allows you to isolate and modify any field you specify within the first 64 bits of the (non-055 label) message.

## Stepping through the Message List

Users can quickly navigate through the list of messages using the Step control. The dataset display window must be open and toggled to List view to access this feature.

To step through the message list:

1. Enter a number in the textbox in the Step frame to establish a step size (1 to 100)
2. Select the message you wish to step from by clicking on its row (row will become highlighted)
3. Click the forward and back buttons in the Step frame to move forward and backward by the number of messages (step size) you have entered




Using the Step buttons moves the message highlight and scrolls the display window if needed.

---

**Tip:** In software playback mode, when playback is paused, you can use the Step control in Radar view to advance playback in steps.

---



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

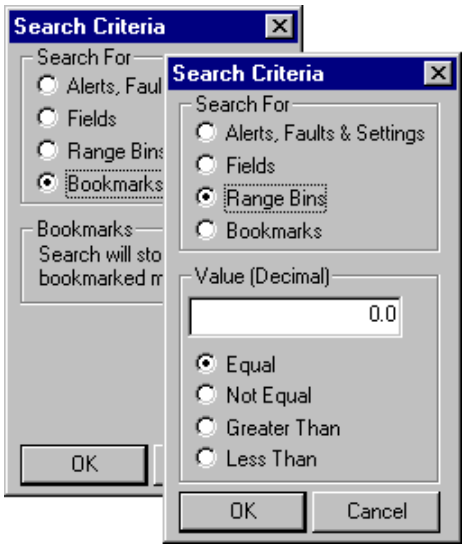
## Searching the Message List

You can search the messages in a dataset by various criteria. The search criteria may be set as a specific value of a range bin or header field. You can also search for bookmarked messages. The dataset display window must be open and toggled to List or Raw Hex mode to access this feature.

To search by criteria:

1. Click the **Define** button in the Search toolbar of the dataset display window to open the Search Criteria window
2. Select one of the four options in the Search For frame

The lower part of the Search Criteria window changes to display the choices linked to the selected option in the Search For frame (see figure below).




3. Define the search criteria
4. Click **OK** to close the Search Criteria window
5. Use the forward and back buttons in the Search frame (see figure below) to find and highlight the next message that meets one or more of your specified criteria



## Setting and Clearing Bookmarks

Bookmarks can be used to tag messages of interest in the List and Raw Hex mode display. Users can navigate through the dataset to bookmarked messages with the Search command. A dataset display window must be open and toggled to List or Raw Hex mode to access and view this feature.

To set a bookmark:

1. Click on a row in the view window to select it
2. Right click on the row and choose **Set Bookmark** from the context menu

A bookmark icon will appear in the Message # column.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To clear bookmarks:

- Right click on a bookmarked row and choose **Clear Bookmark** from the context menu to clear that bookmark
- Right click anywhere in the view pane and select **Clear All Bookmarks** from the context menu to clear all bookmarks in that dataset

## *Configuring the List View Columns*

There are several ways to customize the display of columnar data in a List view display to fit the information you wish to view.

To move a column:

1. Left click and hold the column title you wish to move
2. Drag it until the intended location becomes highlighted, then release the mouse button

To change column width:

1. Left click and hold on the right edge of the column title
2. Drag right or left to resize, then release

To auto-size (fit to width) a *single* column:

- Double click on the right edge of the column title

To auto-size *all* columns:

1. Right click in the list area to access the context menu
2. Choose the **Optimize Columns** command

To show or hide columns:

1. Right click in the list area to access the context menu
2. Choose the **Organize Columns** command to open the Organize Columns dialog
3. Show or hide individual columns by checking or clearing the appropriate checkboxes
4. Click **OK** to close the Organize Columns dialog and view the results

## *Displaying Range Bin Data*

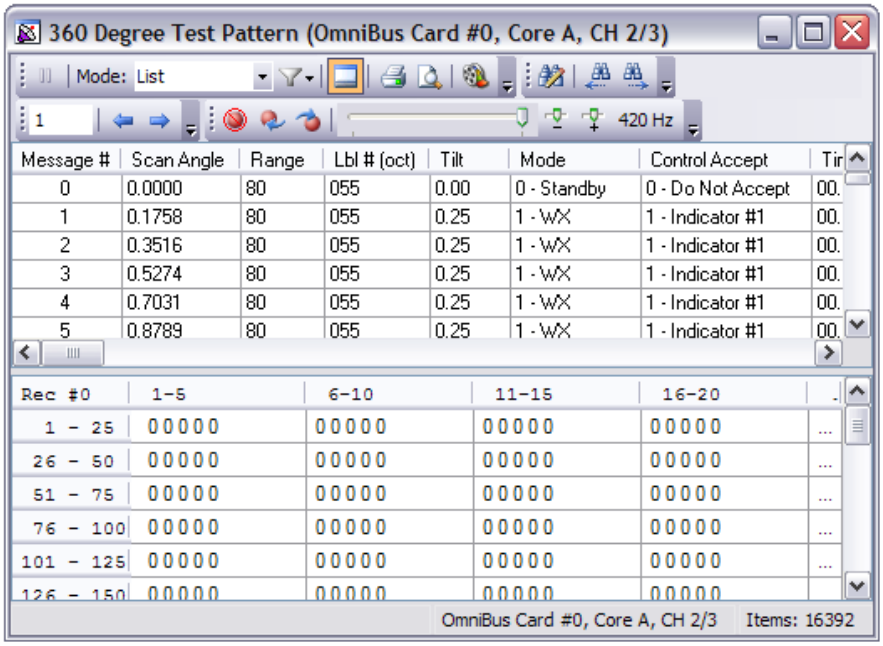
Range bin data can be displayed in a pane at the bottom of the display. Each three-bit field is displayed as a 0–7 value (corresponding to the eight weather colors) with 25 range bins in each row (see figure below).

To view range bin data:

- Right click in the display area and choose the **Display Settings |Show Range Bin Data** command from the context menu
- Repeat the above step to hide the range bin display pane



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




Resize the window or use the scroll bar(s) to view data outside the display area.

To auto-size (fit to width) the columns:

1. Right click in the range bin pane to access the context menu
2. Choose the **Display Settings |Optimize Columns** command

Range bin data can be modified in engineering units (i.e., weather colors) through the 708 Message Editor.

# Record and Reuse Data

## How to Record and Reuse Data

### Weather Data in CoPilot

CoPilot not only simulates a display unit by monitoring the display databus, but it can also record and reuse 708/453 data. CoPilot gives you the ability to save, modify, copy, import, and export ARINC 708/453 display data.

### Recording Data

CoPilot automatically collects data into a "dataset" object as it monitors the bus. (A dataset is a sequential collection of ARINC 708/453 1,600-bit messages.) Once captured, you can test, analyze, and even modify the recorded data.

### Reusing Data

Monitored data can be saved, edited, and copied within the CoPilot project as a dataset. In addition, raw data from a dataset can be extracted and saved to a file (in a .dat data file format) for use with CoPilot or other applications. These data files can be imported into any CoPilot project.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

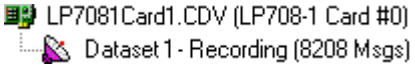
## Recording Data

As you monitor the display databus, CoPilot automatically collects and stores received messages, along with time-tags and error conditions detected by the board, in a dataset. CoPilot adds messages to this dataset as they are received until reception is paused or halted. You can pause and resume recording at any time to omit a segment of bus activity.

To record data:

1. Right click the 708 card (or channel pair) icon in the Hardware Explorer tree and select **Receive** from the context menu
2. (Optional) Choose to reuse the current dataset or to create a new dataset for each run
3. Click the CoPilot **Run** button to begin Simulation mode

When the Run button is pressed, recorded messages begin accumulating in a dataset in the Hardware Explorer. The cumulative message count is displayed to the right of the dataset icon (see figure below).




4. (Optional) Pause and resume recording by right-clicking on the dataset icon in the Hardware Explorer and choosing **Pause** from the context menu
5. Click the CoPilot **Stop** button when you are done recording to return to Edit mode

---

**Tip:** Pause and resume recording by toggling the Pause button on the dataset view window.

---

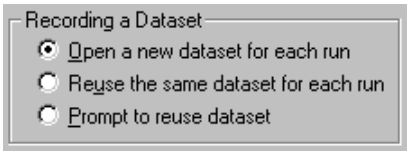
Once data has been recorded, you can save and copy the dataset within CoPilot or export the raw data as a data file. All datasets in the Hardware Explorer tree are saved in their current state when the CoPilot project is saved.

### *Setting Recording Options*

Users can control whether recorded data is placed into a new dataset each time the Run button is pressed or if data in the active dataset is overwritten. The CoPilot default is to create a new dataset for each run. Recording options may be specified uniquely for each 708 board (or 708 channel pair) through 708 board properties.

To specify recording options:

1. Access the Properties dialog by right clicking on the card (or channel) icon in the tree and choosing **Properties** from the context menu
2. Select the **Receive** tab
3. Choose a dataset recording option






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### *Working with Datasets*

Because all data within CoPilot is handled in the dataset object, it is important to understand what datasets are and how to use and configure them. A dataset is a CoPilot object (located in the Hardware Explorer tree) that holds a sequential collection of ARINC 708/453 1,600-bit messages. Radar display configuration settings (default or user-defined) are associated with each dataset. Datasets can also contain time-tag and error information (for recorded data) and user-assigned bookmarks, index points, etc.

#### Renaming a Dataset

A default name is assigned when a new dataset is created (e.g., Dataset1, Dataset2). When datasets are imported, the filename is used as the dataset name.

To rename a dataset:

1. Right click on the dataset icon in the Hardware Explorer and select **Properties** from the context menu
2. Enter a name in the Name textbox in the 708 Dataset tab of the Dataset Properties window
3. Click **OK** to close the dialog

The new dataset name appears in the both the Hardware Explorer tree and title bar of the dataset view window.

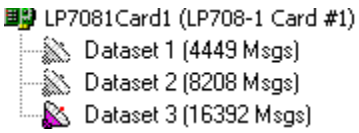
#### Enabling a Dataset

Each 708 board (or 708 channel pair) can host many datasets, but only one dataset can be active.

To specify the active dataset:

- Right click on a dataset icon in the Hardware Explorer tree and choose **Enable** from the context menu

A colored icon marks the enabled dataset and a dimmed ("grayed out") icon represents an inactive dataset.




Recording, transmission, and software playback are linked to the enabled dataset.

## Reusing Data

Recorded data can be modified through message editors, replayed through software playback, or used as a source for transmitting on the bus. A recorded dataset can be used within the same CoPilot project, exported as a data file, or imported and used in another CoPilot project.

- Copying Data—Datasets can be copied from one 708 board to another
- Exporting Data—Data can be extracted from a CoPilot dataset and saved to an external file (in a .dat data file format)



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

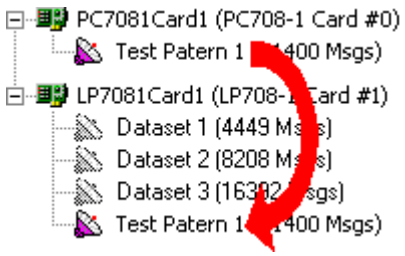
- Importing Data—Datasets created within CoPilot or created with the Ballard 708 Utility program can be imported and listed as a dataset object in the Hardware Explorer

## *Copying Data*

A dataset can be copied from one Ballard 708 board (or channel pair) in the Hardware Explorer tree to another through a drag and drop procedure. The dataset will be copied in its entirety, including name, data, and display configuration.

To copy a dataset to another 708 board:

1. Right click and hold on the dataset icon you wish to copy
2. Drag the dataset to the target 708 board or channel icon (icon will become highlighted) and release the mouse button




## *Exporting Data*

Raw data from a dataset can be extracted and saved to a file (in a .dat data file format) and used with Ballard's 708Utility program, in another CoPilot project, or other application.

To export data:

1. Right click on the dataset icon in the Hardware Explorer and select **Export** from the context menu to open the Save Dataset dialog
2. Choose a name and location and click **Save**

---

**Note:** Exporting data creates a .dat data file that contains 708 data only. Dataset configurations such as breakpoints, bookmarks, radar image settings, etc. are not exported.

---

## *Importing Data*

When a 708/453 data file (in .dat format) is imported into CoPilot, it appears as a dataset object in the Hardware Explorer. This dataset can then be examined in software playback mode or transmitted on the display databus.

The .dat data files can be obtained from an exported dataset within CoPilot or created by Ballard Technology's 708 Utility program.

---

**Tip:** Sample datasets were copied to your computer (in the Dataset folder in the CoPilot Projects folder) as part of the CoPilot installation.

---

To import data into CoPilot:

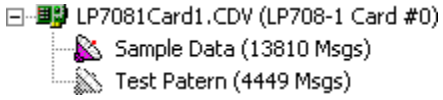
1. Right click the 708 board (or channel pair) icon in the Hardware Explorer to access the board context menu



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. Choose the **Import Dataset** command to open the Import Dataset dialog
3. Browse to the .dat data file you wish to import and click **OK**

You can import as many datasets as you wish, but only one dataset can be active at a time.




**Note:** Since imported datasets do not have time-tag or error information, these columns in the List view display contain zeros.

# Modify and Transmit Data

## How to Modify and Transmit Data

**Modifying Data in CoPilot**

CoPilot provides the tools to edit recorded 708/453 messages through hexadecimal or engineering unit editors. For example, selected messages could be modified to trigger certain fault conditions to test how a display unit detects and responds to those conditions.

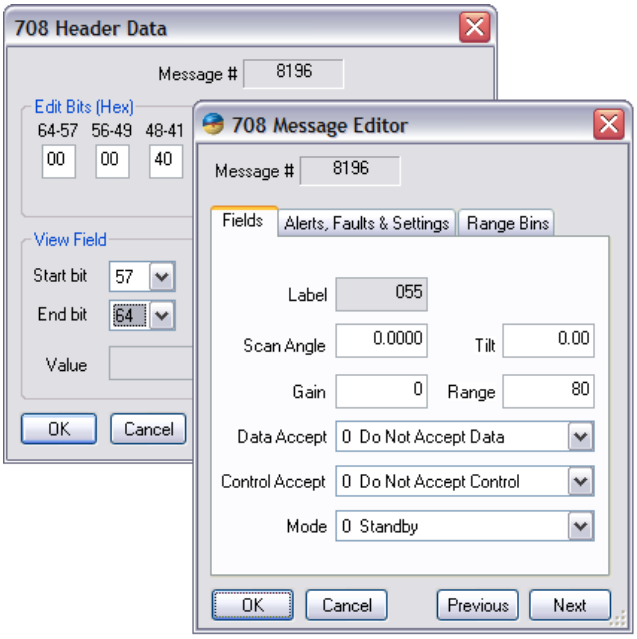
The data used in transmit mode may come from a previously recorded radar transmission or data specially prepared for testing.

CoPilot provides two data editing windows:

- **708 Message Editor**—Easily edit message fields and range bins in engineering units.
- **708 Header Data**—Quickly edit the 64-bit header of a message by entering just 14 digits in a hexadecimal radix (excluding the 8-bit label). In addition, you can select and view segments of the header in octal, binary, decimal, and hexadecimal radices.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




**Transmitting Data in CoPilot**

CoPilot can simulate a T-R unit transmitter by resending previously recorded data onto the display databus. The data source for transmission is an active dataset in the Hardware Explorer. You can specify transmit options that determine the transmit speed and set the dataset to loop continuously or a specified number of times.

## Modifying a Message in Engineering Units

The 708 Message Editor allows you to edit the 64-bit header (of a standard 055 label 708 message) in engineering units, by fields (based on the 055 label). You can also edit range bins individually or initialize them all to a single weather color or to random colors. Alternatively, you can modify the 64-bit header through the 708 hexadecimal editor.

A dataset display window must be open and toggled to List view to modify messages.

To modify message data:

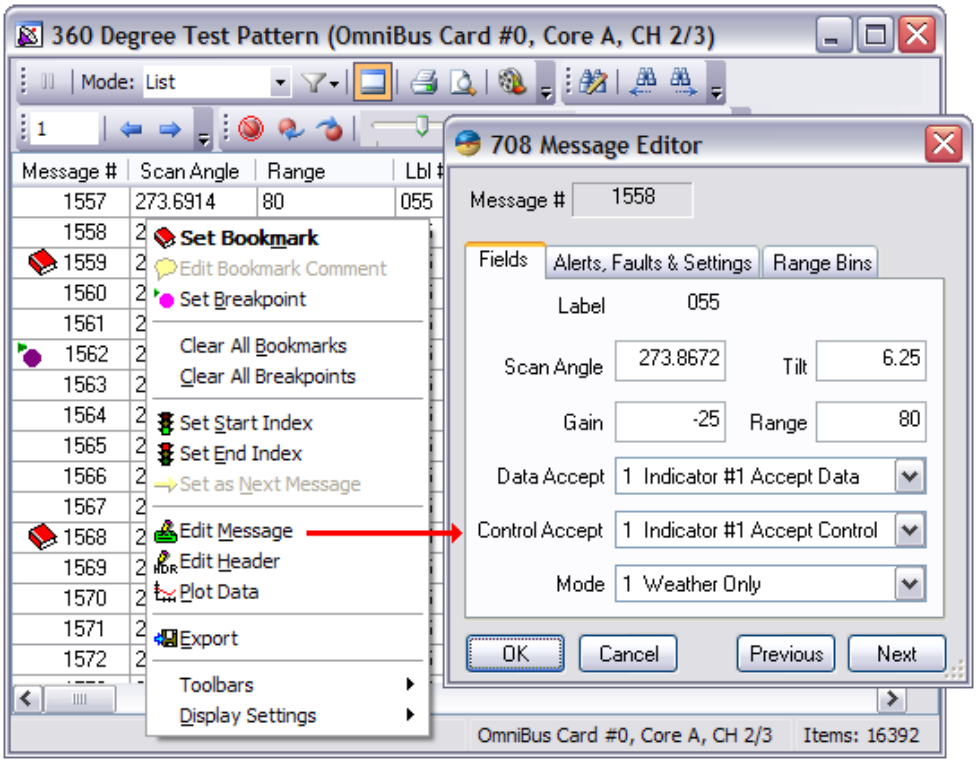
1. Right click on the row you wish to edit and choose **Edit Message** from the context menu to open the 708 Message Editor

The message number is displayed at the top of the three-tab 708 Message Editor (see figure below).

2. Select and change header information (in the Field tab and Alerts, Faults & Settings tab) and range bin values (in the Range Bins tab)
3. Click the **Previous** or **Next** button to continue editing messages, or click **OK** to save your changes and close the dialog



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




## Modifying a Header in Hexadecimal

Header information can be defined through hexadecimal values in the 708 Header Data window. The dataset display window must be open and toggled to List view to access this editing window.

To edit a message header:

1. Right click the message you wish to edit and choose **Edit Header** from the context menu to open the 708 Header Data dialog
2. Modify the header by entering data in hex in the textboxes in the Edit Bits frame
3. Click the **Previous** or **Next** button to continue editing message headers, or click **OK** to save your changes and close the dialog

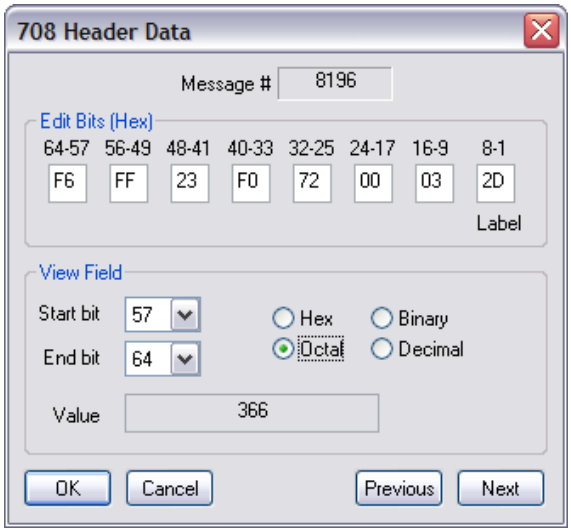
To view a portion of the header in a different radix:

1. Set the start and end bits in the View Field frame using the listboxes to define a section of the header
2. Select a display radix through the Hex, Octal, Binary, or Decimal option buttons
   *(Example: a value of F6 hex for bits 64–57 is a value of 366 octal, 246 decimal, etc.)*

The translated number is displayed in the Value textbox (see figure below).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




To modify the header by engineering unit fields or to modify the range bin data of a message, see Modifying a Message in Engineering Units.

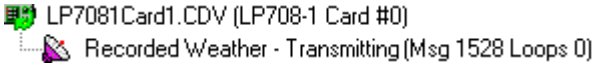
## Transmitting Data

CoPilot 708 can simulate the transmitter in a T-R unit by resending previously recorded data onto the display databus. The data source for transmit is a dataset in the Hardware Explorer tree. To obtain a dataset, you can record or import one. If there is more than one dataset, only the active dataset for a board will be transmitted. The active dataset can be changed during the transmission to chain different datasets together.

To transmit data:

1. Right click the 708 board (or channel pair ) icon in the Hardware Explorer tree and choose **Transmit** from the context menu
2. (Optional) Right click the dataset source you wish to transmit and select **Enable** from the context menu (if your source dataset is not already enables)
3. (Optional) Specify the transmit speed and loop condition
4. Click the CoPilot **Run** button to begin transmission
5. (Optional) Change the active dataset to switch the data source during transmission
6. Click the CoPilot **Stop** button when you are done transmitting to return to Edit mode

A green transmit arrow on the card or channel icon in the Hardware Explorer indicates transmit mode. During run time, a display appears to the right of the dataset icon in the tree containing status ("Transmitting") message number (being currently transmitted) and number of loops.




You can view data as a radar image while it is being transmitted.



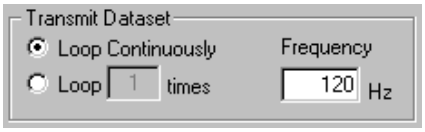
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Setting Transmit Options

In normal operation, datasets are transmitted in a continuous loop at 120 Hz. You can change the transmit speed and duration through dataset properties.

To specify transmit options:

1. Right click on the dataset to be transmitted and select **Properties** from the context menu
2. Click the **Transmit** tab in the Dataset Properties dialog
3. Select a Loop option
    - Specify the number of loops (if you selected the second option)
4. Enter the transmit speed (in hertz) in the Frequency textbox (between 1 and 500 Hz)
5. Click the **OK** button to close the dialog




Transmit options may be uniquely defined for each dataset in the Hardware Explorer.

# Software Playback

## How to Analyze Data with Software Playback

### A Tool for Analysis

CoPilot's software playback mode makes it easy to review, analyze, and process a dataset. Fault indicators that light too briefly to react to in real time can be frozen and reexamined using software playback. Slowing the playback rate to "human speed" facilitates detailed analysis.

### How Software Playback Works

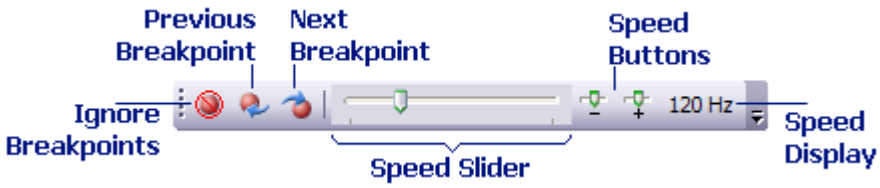
Interaction with the Ballard 708 card and ARINC 708 databus is suspended in software playback mode. The previously recorded or imported dataset is "played" as if it were being received or transmitted. The special tools and features available in software playback allow you to examine and manipulate a dataset in ways not possible in other modes.

### Software Playback Controls

Software playback is viewed and controlled through the dataset display window. Playback is controlled through user-specified boundaries and breakpoints and special controls hosted in the SW Playback frame of the dataset display window (see figure below).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*The software playback toolbar in the dataset display window*

**Prepare the Dataset**

You can focus playback analysis by using breakpoints and a start and end index. These are set through the List view context menu. Breakpoints are used to pause playback at specified points. The start and end index allow you to limit the playback to a portion of the dataset.

**Run Time Features**

Additional playback tools are available during run time. For example, you can adjust the playback speed while running. You can also pause software playback, then set a new resume point or advance the playback in steps.

## Running Software Playback

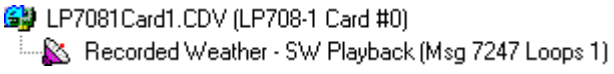
CoPilot's software playback mode allows recorded data to be replayed and analyzed in detail. The data source for software playback is a dataset in the Hardware Explorer. To obtain a dataset, you can record activity from the 708 databus or import a previously recorded dataset. If multiple datasets are listed in the tree, the targeted playback dataset must be set to active.

To reexamine data in software playback mode:

1. Right click the 708 board (or 708 channel pair) icon in the Hardware Explorer tree and choose **Software Playback** from the context menu
2. (Optional) Specify the initial playback speed and loop condition
3. Click the CoPilot **Run** button to start the software playback simulation
4. (Optional) Pause and resume playback by toggling the **Pause** button in the dataset display window
5. Click the CoPilot **Stop** button to return to Edit mode

**Tip:** It is possible to pause the dataset before pressing Run, thus starting software playback in a paused state.

A blue playback arrow on the board or channel icon in the Hardware Explorer indicates software playback mode. During run time, a display appears to the right of the dataset containing status, message count, and number of loops (see figure below).






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### *Setting Software Playback Options*

You can specify the (initial) speed of software playback and number of loops from the dataset properties window. (The default setting is continuous playback at 120 Hz.)

To specify playback options:

1. Right click the dataset and select **Properties** from the context menu
2. Click the **SW Playback** tab
3. Select a Loop option
    - Specify the number of loops (if you select the second option)
4. (Optional) Enter the playback speed (in hertz) in the Frequency textbox to initialize playback speed
5. Click the **OK** button to close the dialog




During software playback, the speed controls in the dataset display tool bar can override the initial frequency value entered in this properties dialog.

### *Moving the Playback Resume Point*

When software playback is paused, the next message to be replayed is highlighted (usually yellow; see figure below) in the List or Raw Hex mode pane. If you wish playback to resume from a different message, you can move the resume location.

A software playback simulation must be running, but in a paused state, and the dataset display window must be in List view for the Set Next Message command to be available.

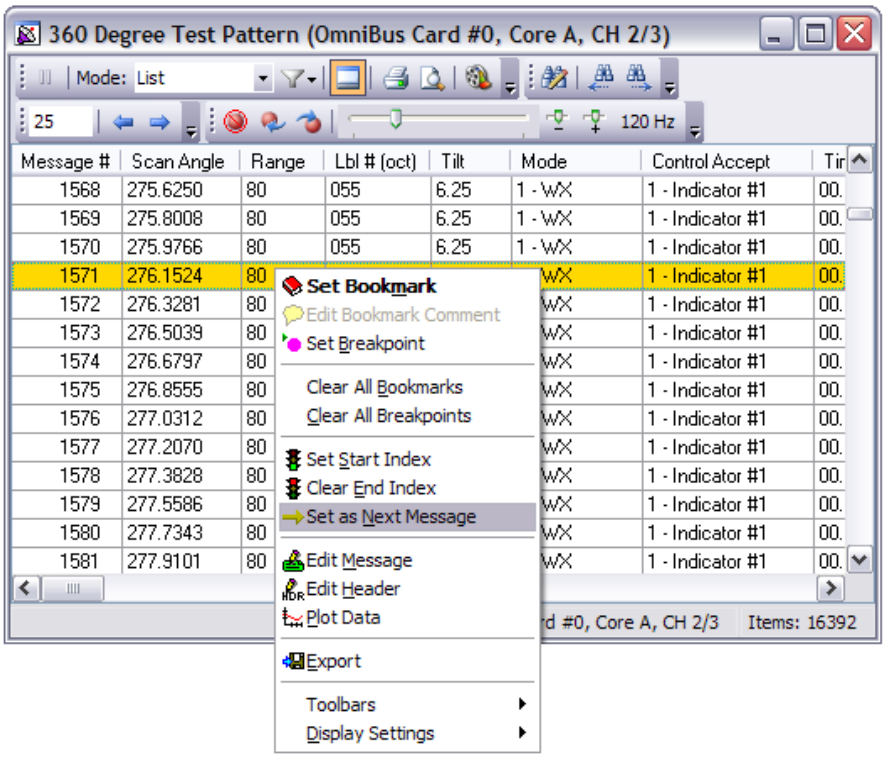
To move the playback resume point:

1. Navigate through the List view display to the message at which you wish to resume playback
2. Right click the message and choose **Set Next Message** from the context menu

The message you have selected will be marked with a (usually yellow) background color. You can now resume playback from this message by using the Step control or toggling the pause button.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




## *Advancing Software Playback in Steps*

In Edit mode or Simulation mode, the Step control allows you to navigate the messages in List view. However, during software playback, the Step feature allows you to replay data (in Radar view) in increments of 1 to 100 messages. Software playback must be engaged through the Run button and the dataset display window must be toggled to Radar view to access this feature.

To advance playback in steps:

1. Press the **Pause** button in the dataset display window to suspend software playback
2. Enter a step size in the textbox in the Step frame
3. Use the forward and back buttons to advance or rewind the software playback




4. Press the **Pause** button again to resume automatic playback

Clicking the forward button will cause the wand to advance the specified number of messages, thus drawing the radar image in "chunks." Clicking the back button erases that portion of the radar image and repositions the wand.

## *Controlling Software Playback Speed*

Software playback speed can be set at any time through the slide control in the SW Playback frame of the dataset display window tool bar. You may adjust the speed during run time or prior to the software playback simulation.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To adjust the playback speed:

- Slide the speed control to the right or left to increase or decrease software playback speed
- Alternately, use the control buttons to the right of the slider bar for smaller increments

As you slide the speed bar, the speed displays to the right in hertz.

## Breakpoints

Breakpoints can enhance analysis by pausing software playback at preselected messages. Although breakpoints are defined in List and Raw Hex mode, the effects of breakpoints during run time are visible in all modes.

Software Playback mode must be selected and the dataset display window must be set to List or Raw Hex mode for the breakpoint commands to appear in the context menu.

To set a breakpoint:

- Right click the target message and choose **Set Breakpoint** from the context menu to tag the message with a breakpoint

A breakpoint icon  will appear next to the message number.

To clear breakpoints:

- Right click a row in List view marked by a breakpoint and choose **Clear Breakpoint** from the context menu to clear the breakpoint
- Right click anywhere in List view and select **Clear All Breakpoints** from the context menu to clear all breakpoints in that dataset

To find the breakpoint you wish to clear, use the Next/Previous Breakpoint buttons in the SW Playback frame.

Software playback enters a pause state when a breakpoint is encountered. Toggle the Pause button to resume playback. To run playback without stopping at breakpoints, press the Ignore Breakpoints button in the SW Playback frame.

### *Navigating between Breakpoints*

Breakpoint navigation buttons are available through the SW Playback frame in the dataset display window tool bar when the 708 board (or 708 channel) is in software playback mode and the display window is toggled to List view.

To move between breakpoints:

- Click the **Next Breakpoint** button in the Playback toolbar to move to the next message tagged with a breakpoint
- Click the **Previous Breakpoint** button to move to the previous breakpoint

Using these buttons moves the List display to the tagged message, and highlights that row. If the row is outside of the current view area, the display automatically scrolls to that location. These buttons will navigate to breakpoint even if the Ignore Breakpoints command is activated.

---

**Note:** Navigating between breakpoints when playback is paused does not move the playback resume point.

---



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### *Ignoring Breakpoints*

Software playback enters a pause state when a breakpoint is encountered. The effect of breakpoints can be suspended or enabled from any view mode.

To toggle pausing on breakpoints:

- Click the **Ignore Breakpoints** button in the Software Playback frame in the dataset view window to prevent the software playback from pausing on any of the messages tagged with a breakpoint
- Click the **Ignore Breakpoints** button again to reenable breakpoints

The Ignore Breakpoints button can be toggled on or off before or during a software playback simulation.

## Setting the Start and End Index

Software playback is controlled through a start and end index. If the user does not explicitly define index points, the start and end index default to the first and last message of the dataset. You can define one index point, or both, or none. When an index point is cleared, the default is restored.

The start and end index can be set through the List view context menu. The 708 board or channel must be in software playback mode for the index point commands to appear. Index points cannot be set during run time.

To set an index point:

- Right click the target message and choose **Set Start Index** or S**et End Index** from the context menu to define the index

Background colors are used to identify the start (usually green) and end index (usually salmon).

To clear an index point:

- Repeat the above steps to clear the Start or End Index (the menu commands change to Clear Start/End Index)

To find the index point you wish to clear, use the index navigation buttons in the SW Playback frame of the dataset display window tool bar.

#### Navigating between Index Points

The start and end index can be located easily through special keys in the SW Playback frame of the dataset display window tool bar.

To move between index points:

- Click the **Start Index** button in the SW Playback frame to move to the message marked as the start index (or message 0 if the start index was not explicitly defined)
- Click the **End Index** button to move to the end index (or the last message if the end index was not explicitly defined)

The List display will scroll to the appropriate message, and the row will be highlighted.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Controlling a T-R Unit

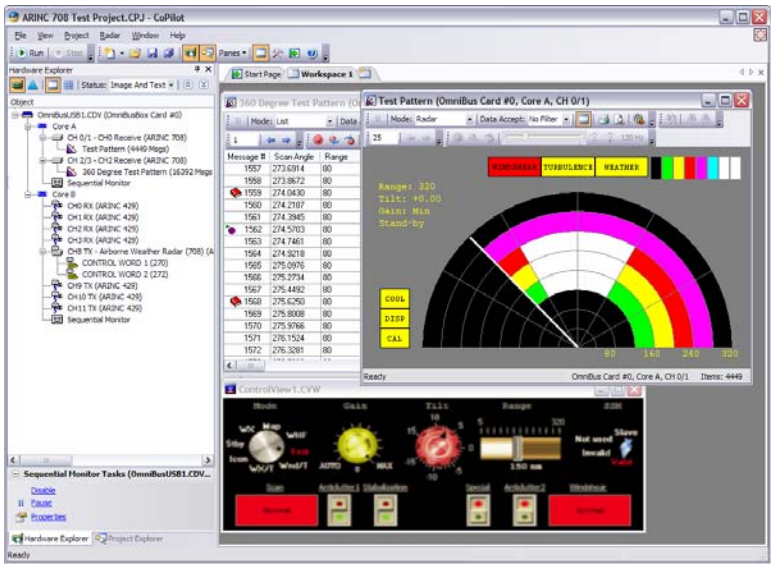
## How to Control a T-R Unit with CoPilot

### The CoPilot System

Because CoPilot can host multiple Ballard boards and multiple avionics protocols in the same project, CoPilot users can add ARINC 429 channels to control a T-R unit. With a CoPilot Professional key, the T-R unit can be controlled with virtual instruments.

### 429 and 708 in One Project

ARINC 429 and ARINC 708 could be accessed, controlled, and monitored through separate Ballard 429 and 708 cards or a single multi-protocol hardware device. As an example, in the figure below, an OmniBusBox card appears in the Hardware Explorer tree that supports both 708 and 429. Core A supports ARINC 708 receive and transmit and Core B transmit channel 6 is used to simulate an Airborne Weather Radar system (equipment ID number 708) over ARINC 429. When the Run button is engaged, CoPilot interacts with both databuses simultaneously.




### Control and Monitor a T-R Unit

The T-R unit receives control data over the ARINC 429 databus and transmits display data over the ARINC 708 databus. The values of the control word fields (pictured above) can be modified and transmitted to the T-R unit over ARINC 429 through engineering unit editors (or, using CoPilot Professional, through virtual instruments) before or during transmission. (The ARINC 429 labels 270 and 271 pictured above and their fields are defined in the ARINC 708 specification.) The T-R unit can be monitored over the 708 databus and incoming data could be observed in the dataset display window.

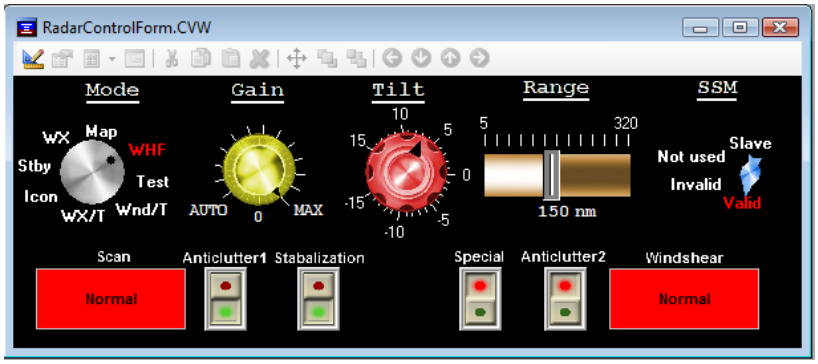
For more information on using 429, please refer to the *ARINC 429 Protocol* section.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Using CoPilot 429 Virtual Instruments

The components of CoPilot include many advanced features, including virtual controls and simulated aircraft instruments. These are created and saved in Control View windows. Virtual instruments may display (sink) or modify (source) data. The figure below illustrates several controls. These can each be linked to ARINC 429 labels and fields in the Hardware Explorer so that when a knob is "turned" (with the mouse) the field is modified. Thus, by combining ARINC 429 and ARINC 708 protocols, you can control and monitor a T-R unit.




Note: You must have a Ballard 429 board with a CoPilot Professional key to communicate over an ARINC 429 databus with virtual instruments in CoPilot.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# ARINC 664 / AFDX Protocol

## ARINC 664/AFDX Overview

CoPilot provides the necessary tools to record databus activity with the Sequential Monitor, analyze received data, and create simulations for ARINC664/AFDX (Avionics Full Duplex Switched Ethernet) Networks. The Monitor View provides multiple modes for displaying and filtering recorded data. Views for networks, VLs, ports, and fields allow users to display current status, change configuration, and control error injection. CoPilot Professional components further extend the feature set by adding graphical control displays and scripting capabilities.

### ARINC 664/AFDX Protocol Summary

ARINC 664 is a multipart specification that defines an Ethernet data network for the new generation of aircraft installations. Part 7 of ARINC 664 defines a deterministic network known as AFDX, which stands for Avionics Full Duplex Switched Ethernet. IEEE Standard 802.3 (Ethernet) is an integral part of the AFDX specification. The AFDX extensions to IEEE 802.3 address the special requirements (quality of service, reliability, etc.) of an aircraft environment. AFDX is a new standard providing much higher data rates than existing avionics databus protocols. Both Airbus and Boeing are developing aircraft that use AFDX.

### Hardware Explorer Object Hierarchy

The Hardware Explorer pane holds the object representation of physical hardware devices along with display settings. These definition of hardware devices are simplified by breaking them down into a hierarchical tree. Each node of the tree holds configuration settings (transmit and sample rates, schedules, settings, etc…) for the levels beneath them.

The levels in the Hardware Explorer can include the:

- **Card**. The Ballard AFDX hardware to be used. File | New | New Hardware and Views menu provides access to new hardware selection.
- **Networks**. Networks include the redundant (AB) network and two independent networks (A and B). The networks can either be operated redundantly or independently.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **VL**. Virtual Links (VLs) are configured below a network. A VL is identified by the VL number (or the destination MAC address). The Bandwidth Allocation Gap (BAG) of a VL specifies how often that VL can transmit on the network.
- **Port**. Ports are messages on the AFDX bus and are configured below a VL. Ports define AFDX frame and payload definitions.
- (Optional) **FDS** and **DS**: Functional Data Sets (FDS) and Data Sets (DS) provide logical groupings of data in a port.
- **Fields**. Fields are used to define sections of data in the payload of a port and can define engineering unit interpretations to the data. Fields are defined below a port or a DS if DS's are used.
- **Sequential Monitor**. Sequential data recording (logging) to files are configured using the sequential monitor.

## Monitoring AFDX Networks

The Sequential Monitor of the AFDX interface boards records specified databus activity. Records are displayed using a multi-pane Sequential Monitor View where capture filters limit what is recorded and display filters limit the displayed records. See the Sequential Monitor chapter for additional information.

## Analyzing Receive Traffic

In addition to recording databus traffic for AFDX networks, CoPilot maintains detailed information for VLs, ports, and fields. This information is organized according to the hierarchy defined by the AFDX specification. Auto-detection of receive VLs and auto-detection of ports further simplifies analysis of traffic. Statistics such as MIB counts, jitter, analysis, BAG comparison, and others are calculated and maintained by CoPilot.

## Simulating Transmit Traffic

CoPilot automatically generates schedules for transmit VLs and ports based on timing parameters. VLs can be configured to transmit at the BAG or at user-defined transmission rates. VLs may use up to four sub-VLs to share bandwidth between logical groupings of ports. The use of sub-VL schedules is optional. When using sub-VL scheduling, the sub-VL schedules use timing factors to control the relative transmission rates or VL schedules can be configured to process ports in a round-robin mode. When transmitting, data generators, graphical control, links, error injection and scripting are used to manipulate data frames.

## AFDX Views

### *Network View*

The Network View displays network health and statistics, and controls parametric frame frequency error injection. Error injection is only available with Model B hardware. A Network View can be displayed for each network.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### *VL View*

The VL View displays VL health and statistics, and controls RSN error injection. Error injection is only available with Model B hardware. A VL View can be displayed for each VL.

### *Port View*

The Port View displays the port's health, statistics, and configuration. The Port View also controls the port error injection. Error injection is only available with Model B hardware. Transmit ports allow engineering unit field and entire frame editing. A Port View can be displayed for each port. See *Displaying Port Information with Port View* section for additional information.

### *Engineering View with ARINC 664/AFDX*

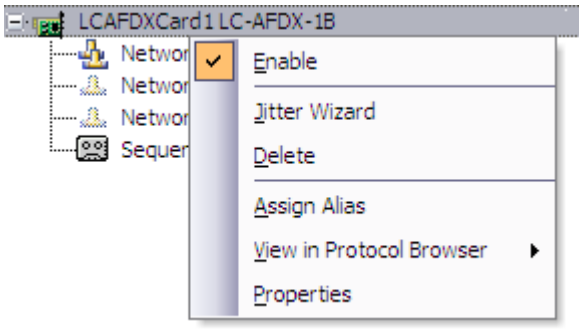
Engineering Views display field and port information along with their corresponding data values and AFDX configuration. If a port with defined fields exists in the view, the fields are grouped together. They can be expanded and collapsed. The display columns and the sort order are configurable. Sort order is changed by clicking the column headers. The Engineering View allows fields from multiple ports from multiple AFDX boards and also data from other protocols. Commands such as editing transmit fields are accessed by right-clicking an entry in the view.

# ARINC 664/AFDX Hardware Features

## Card Configuration

### *Card Context Menu*

To configure a card after it is inserted into a CoPilot Project, right-click on it to access its context menu.




- **Active** toggles the active/inactive status of the card. A card must be active to interact with the databus during run time
- **Jitter Wizard** walks you through the steps to a successful jitter measurement of a known source
- **Delete** removes the card from the Hardware Explorer pane



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Assign Alias** sets the ATE script reference name
- **View in Protocol Browser** in version 5 displays a protocol browser the hardware information shown (the Protocol Browser is not available for all objects and protocols)
- **Properties** opens the Properties dialog box (described below)

### *Card Properties*

The Card Properties window is accessed from the context menu (shown above) or by double clicking on the card icon in the Hardware Explorer pane.

### *Card DMA Mode*

AFDX hardware is capable of running in DMA (Direct Memory Access) mode. DMA mode enables host DMA processing of receive data and the sequential monitor. When DMA mode is enabled, receive port structures are allocated in host memory for direct host access (receive host enable). Sequential monitor data is also stored directly in host memory to optimize data transfer from the card to the host. This setting is configured from the Card Properties window.

# Sequential Monitoring for ARINC 664/AFDX

## How to Record and View Sequential Data

### *Recording Databus Traffic in the Monitor*

CoPilot makes it easy to record AFDX databus traffic. The quickest way to collect data is by selecting the Monitor All option in the Sequential Monitor. This option records all data on the databus. Users may also limit the recording by enabling capture filters on networks, VLs, or ports.

Model B hardware and the demo card allow concurrent recording of sequential monitor data while simulating VLs and ports.

To monitor data:

Right click on the Sequential Monitor icon in the Hardware Explorer and check Active in the context menu.

1. Then check the Monitor All option also from the Sequential Monitor the context menu
2. Click the CoPilot Run button to begin monitoring
3. Click the CoPilot Stop button to stop monitoring and return to Edit mode

CoPilot AFDX collects the monitored data with time-tags, activity information, and detected errors. A record count is displayed next to the Sequential Monitorin the Hardware Explorer.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### *Viewing Monitor Records*

The recorded databus traffic is viewed in an AFDX Monitor View window. The Monitor View window displays data and time-tag information in several formats, and decodes the protocol layers of the AFDX frame.

To open the AFDX Monitor View:

1. Right click the Sequential Monitor in the Hardware Explorer.
2. Select Open Monitor View in the context menu

## Sequential Monitor View Capabilities

### *Using the Monitor Controls*

A control button toolbar is located at the top of the Monitor View. These controls facilitate analysis and display of sequential record data. Display options can be defined anytime, even while data is being collected. Display filters can be engaged before the run is initiated or after a recording is complete.

Controlling the Recording Session:

The Monitor Pause  button halts the recording of data with an opportunity to resume. Click the Pause/Continue button a second time to restart the monitor. The additional data will be appended to the same file.

### *Saving the Monitor View*

**Monitor Views Components**

CoPilot automatically names each Monitor View component as they are created in the project. The names follow the pattern MonViewN.CVW (where N is the first unused number in the project folder). The user can change the name however, through the File menu. The Project Explorer lists each of these components.

Each Monitor View is saved to the project folder in a separate file. A user can have as many Monitor Views in the current project as they wish, but only one Monitor View is updated when the project is run. Saved Monitor Views can be added to the project using the File | Open Hardware or View command.

**File Menu Options**

Sequential Monitor views are loaded and saved through the File menu or CoPilot toolbar (version 5) or the **Home** tab of the ribbon (version 6). All Active Monitor files are saved simultaneously with the Save Project command. Monitor files may be saved selectively with the Save button  or File | Save command (version 5).

### *Exporting Data*

Recorded data can be exported and saved to a file for use by other applications such as Excel or Ethereal. CoPilot exports data in one of three formats (Export Type):

- **libpcap** – The libpcap file format is a raw packet capture format used with applications like Ethereal
- **Monitor Record** – Exports all columns presently displayed in the Monitor View Window into a delimited text file



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Engineering Unit** – Exports each field into two titled columns (time-tag and value) into a delimited text file

To export data:

1. Right click on the Frame List Pane of the Monitor View and select Export from the context menu
2. In the Export Dialog, select the file format (Export Type), a filename, and the export record range
3. Click the OK button to create the new export file

# Filtering Monitor Recordings

## *What is Monitor Capture Filtering*

Capture filters limit the amount of data that is collected into the sequential record. Capture filters can be set for Cards, Networks, VLs, and Ports. Enabling the recording of all traffic for a particular level overrides the settings at the lower levels in the Hardware Explorer. If capture filters are not set, then the filters of the lower levels are used. See *Analyzing Recorded Data in a Sequential Monitor View* for information about the use of display filters after the data has been recorded.

## *Sequential Monitor Capture Filters*

When the Monitor All option is enabled on the Sequential Monitor, CoPilot records all databus traffic, and ignores filter settings for all networks, VLs, and ports. If the Monitor All option is disabled, capture filters for the entire card can be configured on the Capture Filter tab of the Sequential Monitor property page.

## *Network Capture Filters*

When the Monitor All option for a Network is enabled, CoPilot records all Network-specific databus traffic, and ignores capture filters for child VLs and ports. If the Sequential Monitor is set to Monitor All, this setting is ignored. All capture filters for a card can be set from the Capture Filter tab of the Sequential Monitor property page.

## *VL Capture Filters*

When the Monitor All option for a VL is enabled, CoPilot records all VL-specific databus traffic, and ignores capture filters for child ports. If a higher level parent, such as a Network, is set to Monitor All, this setting is ignored.

## *Port Capture Filters*

The Monitor option for a Port determines if the bus traffic for the Port will be recorded. If a higher level parent, such as a Network, is set to Monitor All, this setting is ignored.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Sequential Monitor Modes

## *How to limit the size of the recorded data file*

The Sequential Monitor has options to limit the amount of data to be recorded. This can be done through timed runs, by discarding the frame payload and keeping the frame header, or by setting Capture Filters. See *Sequential Monitor Capture Filters* for more information.

To set data collection parameters:

1. Right click the Sequential Monitor in the Hardware Explorer and select Properties from the context menu
2. Change the desired parameters of the Sequential Monitor.

## *Timed Run Mode*

The Sequential Monitor is automatically paused after the specified number of seconds. Unpausing the Sequential Monitor will again run for the specified number of seconds and repause.

## *Header-Only Setting*

The Sequential Monitor records only the AFDX frame header and RSN values. The AFDX payload data is not recorded.

## *Monitor Data Collection Modes*

### Where the Data Goes

When CoPilot collects data for a Sequential Monitor, data is collected using Monitor View (and MonData) files as described above. As records are collected, the count displayed in the status column of Hardware Explorer is updated to reflect the count of items recorded to each currently connected MonData file and the sequential monitor item displays the total count for the current run. The captured data records can be displayed and analyzed using a AFDX Monitor View. The AFDX Monitor Views can be saved, reopened, modified, and shared with other projects.

### Monitor Collection Modes

That default response can be changed by the user by selecting a different Data Collection mode:

- **Append to Current**—New data will be added to the monitor buffer each time the CoPilot simulation is run, thus accumulating until the monitor site is saved or overwritten
- **Create New Recording**—An AFDX Monitor View data file will be created each time CoPilot is run to capture the data until the simulation is stopped
- **Overwrite with Prompt**— A message will appear (with the number of records currently in the monitor buffer) asking for confirmation to run the card and overwrite those records (see figure) each time the project is run (simulation started). If accepted, previously recorded data in the monitor capture are erased. The new data overwrites the records in the 'connected' monitor data file each time CoPilot is run.



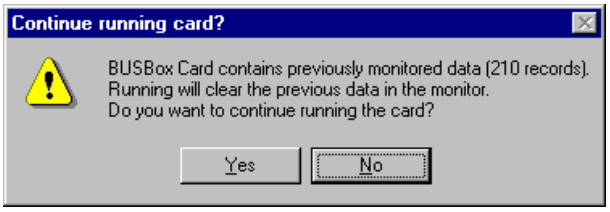
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| Comparison of Data Collection Modes | | | |
|---|---|---|---|
| | **Prompt** | **Create New View** | **Append** |
| *Description* | Overwrite existing records (on input from user) | New Monitor View window for new records | Add new records to existing file |
| *User-input dialog* | Yes | No | No |
| *Monitor buffer in hardware tree* | Records are erased and counter is reset to 0 | Records are extracted to view and counter is reset to 0 | Records are pre-served and counter increments |
| *Monitor View window* | Uses active Monitor View window (if present) | Creates new (active) Monitor View window | Uses active Monitor View window (if present) |

*Table of Sequential Monitor data collection modes*

**Warning**: Overwritten data cannot be recovered.

In the default mode, a new view is created each time to prevent the possibility of data loss. If overwriting the data, then the following dialog will appear when the CoPilot Run button is pressed (if the monitor buffer contains records from a previous run). The dialog is intended to minimize the accidental loss of data.




*Prompt to overwrite previously recorded data when in overwrite mode*

#### Sequential Monitor Considerations

The specified Data Collection mode affects all channels being recorded by the Sequential Monitor. For example, if a Sequential Monitor is connected to a hardware device monitoring both ARINC 429 and ARINC 664/AFDX, the record count is the total number of records for both.

## *Monitor recording modes*

The monitor can be configured to run continuously or for a specified length of time then is automatically set into the pause state. (If you have a Sequential Monitor, the recording mode affects all channels and protocols being recorded by the Sequential Monitor.) Un-pausing (resuming) the Sequential Monitor while running will again collect data for the specified time then pause again.

- **Default**—The monitor will run continuously when CoPilot is in simulation mode
- **Timed Run**—The monitor will run until the specified time period is reached. This option can be useful when using scripting technology to only collect data around a triggered event such as a value out of range.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Analysis of the Sequential Monitor Record

## *How to analyze a single record*

Data in the Sequential Monitor is displayed in a Sequential Monitor View. It displays the contents of each frame plus additional information such as error and activity. The data is divided into three panes - the Frame List Pane, the Detail Pane, and the Hex Pane.

### The Frame List Pane (Record list)

The Frame List pane is at the top of the Sequential Monitor View. This is a list of the recorded frames ordered by time-tag. The list can be filtered by setting Display Filters (see *Time Correlation*). Delta time calculations are based on the time difference between adjacent items in the list.

To customize the Frame List Pane:

1. Add or remove columns by selecting **Display Settings | Select Columns** from the Frame List Pane context menu.
2. Select the time-tag format by selecting **Display Settings | Time Display | <time option>** from the Frame List Pane context menu. Cycle through the time formats by pressing the right button in the Display group on the Monitor View toolbar.
3. Save the current Frame List settings to the host system registry. Subsequent Monitor Views will be created using the same default settings.

### The Detail Pane

The Detail Pane in the middle of the Sequential Monitor View breaks down a single Frame by protocol layer. Each layer is represented as a node in a list. Users can expand the nodes to get even more details about the frame. By setting a time reference  in the Frame List, you can see the delta time in reference to the currently selected frame under the Sequential Record node.

### The Hex Pane (Frame bytes)

The Hex Pane on the bottom of the Sequential Monitor View displays a hexadecimal dump of each frame. It shows the same frame as the Detail Pane, only with different representation. As each item in the Detail Pane is selected, the associated bytes in the Hex Pane are highlighted.

## *Display Filter Settings*

Display Filters limit the information visible in the Monitor View for easier analysis. Display Filter criteria include networks, VLs, ports, transmit/receive, error conditions, bookmarks, and more. Display Filter settings are accessed through the Display Filter frame on the Sequential Monitor window toolbar. Display Filters can be defined and engaged before data capture begins or after recording is complete.

To Specify a Display Filter:

1. Click the Edit button in the Display Filter Frame.
2. Add specific VLs and ports to the filter lists, or search for items using the Auto Fill buttons.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3. Select items, at any level, to be filtered out by clicking on the items state icon.
4. Select a limit option to further limit the viewable items. (e.g., Selecting Transmit Records will only show records transmitted)

### *Search Options*

Use the Search function to locate a specific record.

To search by criteria:

Click the Define button in the Search frame on the Sequential Monitor window toolbar

1. Select the search criteria to use, and click OK.
2. Use the Forward and Back buttons to jump to the next record that meets the defined criteria.

### *Exporting Sequential Monitor Data*

A Sequential Monitor View can export data after it is recorded by a Sequential Monitor. See *Common Sequential Monitor View Features* for additional information regarding exporting.

# CoPilot AFDX Networks

## Why Are There Three Networks?

AFDX is a redundant protocol. Ballard AFDX boards allow the ability to operate the two busses as a single redundant network or to operate two independent non-redundant networks. CoPilot displays the three networks for configuration:

- NetworkAB – The redundant network consisting of bus A and bus B.
- NetworkA – Independent network on bus A.
- NetworkB – Independent network on bus B.

Either the redundant network or both independent networks can be operated at a time. Activating the redundant network while either of the independent networks are enabled will disable both independent networks. Alternately, activating either independent network while the redundant network is enabled will disable the redundant network. Only active objects are simulated.

## Differences between Redundant and Independent Networks

The redundant definition applies to both bus A and bus B. Defined VLs and ports on the redundant network will receive data packets from both databuses and transmit data packets on both databuses. Independent networks use different configurations and operate as distinct networks. Independent networks receive and transmit data packets on their respective buses only. Drag-drop functionality



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

has been added to the Hardware Explorer to assist with the configuration of multiple networks.

## Displaying Statistics with the Network View

Network health and statistics are displayed in the Network View.

- To open a Network View, select **View Network** from the Hardware Explorer network context menu.
- To reset the counts used to create the statistics, select **Reset All Counts** from the Network View context menu.

## Network Error Injection

Error injection is a Model B hardware feature. Network-level error injection consists of Parametric frame-time frequency scaling of the transmission rate for all VLs on the network.

Parametric frame-time frequency is configured using the Network View. See the error injection chapter for additional information.

# Simulating VLs and Ports

## How to Simulate VLs and Ports

CoPilot AFDX is designed to monitor, simulate and test avionics networks. The first step to simulate an AFDX network or end system is to define the configuration. The configuration includes:

(1) Defining the Virtual Links (VLs)

(2) Creating Ports to hold the data

(3) Setting the timing parameters (such as BAG or transmission rate) for each Port.

CoPilot AFDX provides a simple and powerful method of configuring the Ballard AFDX hardware. The CoPilot Hardware Explorer provides the framework for defining which AFDX boards to use, which VLs and Ports to transmit or receive on the databus, and which AFDX data packets to record in the Sequential Monitor. The hardware configuration is loaded onto the Ballard AFDX board when the CoPilot project is run.

## Configuring VLs and Ports

### *Virtual Links*

Virtual Links (VLs) are created on AFDX Network objects in the Hardware Explorer. VLs can be created on any of the three Networks (See *CoPilot AFDX Networks*). Once VLs are created, they can be reconfigured using the VL property page.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To create a VL:

Right click on a Network in the Hardware Explorer and select **Add VL**

1. Enter the VL Number or VL ID
2. Select Receive or Transmit
3. Click the **OK** button

When the VL is created, the VL properties page is opened. Use the VL property page to set timing and other options specific to the VL being simulated.

## *Ports*

Ports on a VL hold the frame definitions and payload definitions. Fields are inserted on ports to give users engineering unit interpretations of payload data. Optionally, Functional Data Sets (FDS) and Data Sets (DS) logically group bytes of the payload together and may contain fields to interpret their data. See Defining Field Engineering Units for more information. Once ports are created, they can be reconfigured using the Port property page.

To create a Port:

Right click on a VL in the Hardware Explorer and select **Add Port**

1. Enter the Port Number (the UDP destination port)
2. Select Sampling or Queuing service
3. Click the **OK** button

When the Port is created, the Port properties page is opened. Use the Port property page to set options specific to the Port being simulated.

## *Functional Data Sets & Data Sets*

The use of Functional Data Sets (FDS) and Data Sets (DS) in CoPilot is optional. The Port property page allows users to set this option to use FDS. Once an FDS or DS is created, they can be reconfigured by opening the property page from the Port View context menu or from the Hardware Explorer context menu.

To create an FDS or DS from the Port View:

1. Right click on a Port in the Hardware Explorer and select **View Port**
2. Click the **Configuration** button to select the Configuration mode
3. Click the **Insert** button to create a new item under the selected item in the Item tree
4. Enter the Start Byte and Length to set the location of the FDS or DS in the data packet
5. Optionally set a Name
6. Click the **OK** button

The Port Layout display of the Port View graphically shows where the FDS or DS is located in the packet.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To create an FDS or DS from the Hardware Explorer:

1. Right click on a Port or FDS in the Hardware Explorer and select **Insert FDS/DS**
2. Enter the Start Byte and Length to set the location of the FDS or DS in the data packet
3. Optionally set a Name
4. Click the **OK** button

## Displaying VL Statistics with VL View

During simulation the VL View gathers data from the Ballard AFDX hardware and displays vital statistics and network health information. The user can reset the counts used to calculate these values by opening the VL View context menu and selecting Reset All Counts. A VL View is created by selecting View VL from a VL context menu.

## Displaying Port Information with Port View

The Port View is a powerful tool for data packet organization and viewing. A Port View is created by selecting View Port from a Port context menu. This view has four modes: Summary, Packet Edit & Display, Configuration, and Error Injection.

### *Summary Mode*

The summary mode of the Port View displays Port Activity, Errors, Statistics (MIB), and Engineering Unit values for a Port.

### *Packet Edit & Display Mode*

The Packet Edit & Display mode displays detailed frame information and a hexadecimal dump of the data. Selecting detailed items or fields within a frame will highlight the corresponding bytes in the hex editor. Transmit packet data is edited using the detail view, field engineering units and the hex editor.

To edit a transmit packet:

1. Open the Port View context menu and select **Lock Display for Edit**
2. Edit the data by:
    a. Hex Editor: Place the mouse cursor in front of the first byte to edit. Enter hex characters to replace the bytes values.
    b. Detail Edit: Select the item to edit by double clicking the value. Type the new value and press **<Enter>** to replace the old value.
    c. Field Edit: Select the Fields item from the list on the left then double click the field to edit. Type the new value and press **<Enter>** to replace the old value.
3. To update the port with the edited data packet, press the Update Port button or



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4. Deselect **Lock Display for Edit** when editing is complete. This will update the port with the entire data frame.

### *Configuration Mode*

The configuration mode of the Port View organizes the AFDX data payload into smaller logical pieces. Users can insert, edit or delete Functional Data Sets, Data Set or Engineering Unit Fields.

### *Error Injection Mode*

The error injection mode of the Port View is used to inject errors into transmit packets. To learn more see the Error Injection chapter.

## Defining Field Engineering Units

Individual bits, bytes, or byte segments within a data packet can be defined as data fields. Once declared, fields can be named and associated with an interpreter, permitting data to be defined, translated, edited and viewed in an "engineering unit" format. The engineering unit interpreters assign meaning to fields and can be saved and loaded in the AFDX database as described on page *316*.

To create a field from the Hardware Explorer:

1. Select **Insert Field** from the Port or DS icon in the Hardware Explorer
2. Enter a Name
3. Enter the location in the data packet by setting the Start Byte and optional Length
4. Select a data interpreter. For more details see *Engineering Units* on page *83*
5. Click the **OK** button

Fields can also be created from the Configuration Mode of the Port View.

### *Engineering Unit Interpreters*

Field data formats are based on several different data interpreters. CoPilot uses interpreters associated with appropriate data types to display field values correctly. It is possible to change the interpreter for a field with CoPilot as long as the new interpreter can accept the number of bits/bytes in the field. Currently CoPilot AFDX supports the following interpreters:

**Integer 32-bit** – A 32-bit signed integer using the two's complement data representation format.

**Integer 64-bit** – A 64-bit signed integer using the two's complement data representation format.

**Float 32-bit** – A single precision floating point format with 3 fields. These fields are: (1) the sign bit, (2) an 8-bit exponent and (3) a 23-bit mantissa.

**Float 64-bit** – A double precision floating point format with 3 fields. These fields are: (1) the sign bit, (2) an 11-bit exponent and (3) a 52-bit mantissa.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

**Boolean 32-bit** – Booleans are packed into 32-bit structures. Each bit represents a single discrete. A single 32-bit Boolean can represent up to 32 discrete entities.

**String** – A sequence of ASCII characters with one ASCII character per byte. The first 2 bytes of the string structure is a 16-bit unsigned integer declaring the length of the string (number of characters in the string).

**Opaque (Fixed Length)** – An un-interpreted sequence of n bytes.

**Opaque (Variable Length)** – A 16-bit integer representing the length followed by an un-interpreted sequence of n bytes.

**BCD** – The BCD (Binary Coded Decimal) interpreter is not part of the AFDX specification, but is added to CoPilot for ARINC 429 compatibility. The BCD interpreter allows the user to adjust the range of the data, to establish the BCD character rules, and to define other optional parameters. The resolution is generated automatically from the upper and lower ranges unless Override Resolution is checked. The operating limits provide a visual notice in the Engineering View window if data is outside the specified operating range.

**BNR** – The BNR interpreter is not part of the AFDX specification, but is added to CoPilot for ARINC 429 compatibility. BNR is a widely used binary format that is compatible with many types of data. The BNR interpreter defaults to a 2's complement format (sign bit indicates positive/negative). To encode a positive number, select the Unsigned Values option button. CoPilot automatically sets the low range to 0 (zero) and recalculates the resolution.

### *Embedding ARINC 429 Engineering Units*

Many implementations of AFDX leverage data definitions from ARINC 664's predecessor - ARINC 429. In addition to the standard AFDX interpreters, CoPilot adds ARINC 429 BNR and BCD field interpretations to AFDX payload definitions.

## Viewing Field Engineering Units

CoPilot provides multiple ways to view data in engineering units. The Port View allows users to view and configure fields for the port they are defined on, while the Engineering View allows fields from multiple ports (even from different cards). Controls and graphs are also used to display data.

### *Displaying Fields with the Port View*

The Port View displays fields in two ways. The first is accessed by selecting Fields in the Item list when in the Summary or Packet Edit & Display Modes. The second is accessed by selecting a field parent object in the Item list when in the Configuration Mode. In this mode the Port View will graphically display a chart showing the field locations within the data packet.

### *Displaying Fields with the Engineering View*

The Engineering View displays interpreted values along with time-tags and other important data. Create an Engineering View by selecting the New Hardware or View command in the File Menu and dragging Fields from the Hardware Explorer to the new view. The display can be customized through the Engineering View context menu. Multiple Engineering Views can be created and saved within a CoPilot Project, allowing users to organize data into logical groups.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### *Additional Field Displays*

Refer to the *Professional Displays and Controls* on page *369* for more information.

## Editing Field Engineering Units

CoPilot provides multiple ways to edit engineering units. In addition to editing the fields using engineering units, they may also still be edited using hex, binary, etc. Only transmit fields (fields belonging to a transmit VL) can be edited.

### *Editing of Fields in-place*

Transmit fields in the Hardware Explorer, Port View, and Engineering View can all be edited in-place. Additionally, right-clicking the field shows additional edit options.

To in-place edit a transmit field:

1. Double-click the field to open the editor.
2. Type the new value and press **<Enter >** to replace the old value.
3. Click away from the editor or press <**Esc**> to close the editor

### *Editing Fields indirectly*

CoPilot enables multiple editing and display mechanisms in addition to in-place editing of field engineering units. Some examples of the possibilities include:

- Links – Creating links to source or sink data to/from other objects in the Hardware Explorer or controls or views. The link properties are accessible from the CoPilot **Project | Links** menu. An example could link an ARINC 429 field to an AFDX field.
- Quick Controls – A graphical display to view and/or edit engineering values. Quick controls are inserted by right-clicking a field and selecting the desired control. Error information is displayed in the status of each quick control.
- Control View Controls – ActiveX controls on a Control View used to display to view and/or edit engineering values. CoPilot provides a variety of controls in the Control View palette.
- Scripting – User implemented script (code) can run to execute tests or perform complex simulations where field values are modified.
- Data Generators – Automatically generate field data based on configurable functions.
- Port Payload editing –Performing operations to edit the payload of a port where a field is defined will modify the data value of the port.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## CoPilot Professional for Display and Edit

CoPilot Professional provides additional data control and visualization tools not available with CoPilot Standard. Part 3 of this document beginning on page *367* describes the features of CoPilot Professional in detail.

# Scheduling for Transmit

## How to Schedule VLs and Ports for Transmit

CoPilot AFDX automatically creates a schedule based on the transmit rate and optional sub-VL factor specified by the VL. Transmit schedules for a VL use either a round-robin or a sub-VL scheme. Sub-VLs are used for bandwidth sharing between logical groupings of ports. If using sub-VL scheduling, each port identifies the sub-VL it belongs to; each sub-VL uses round-robin processing of the defined ports. If not using sub-VL scheduling, all ports of the VL are scheduled using a round-robin scheme. When a port does not have data to transmit (such as with an empty queuing port), the next item in the round-robin is processed.

## Transmission Rate

The transmission rate is the time interval between subsequent transmissions of a VL. The default transmission rate for a VL is the BAG (Bandwidth Allocation Gap) value. BAG values are defined for each VL. The BAG specifies the minimum amount of time that must expire before another transmission of the identical VL. CoPilot allows the configuration rate of a VL to be slower (larger gap value) than the BAG to limit the amount of traffic on the network and CoPilot allows transmission rates faster (smaller gap value) than the BAG for injecting timing errors.

The effective transmission rate for a Port is not necessarily the transmission rate of the VL. A port's actual transmission rate depends on the number of ports defined for the VL, the usage of sub-VLs, and existence of queuing ports.

## Sub-VL Scheduling

Sub-VLs are used for bandwidth sharing between logical groupings of ports. By default, ports are assigned to Sub-VL number one. The default configuration for sub-VLs is to use a round-robin scheme where each defined sub-VL equally shares the VL's bandwidth. Normalization factors of sub-VLs specify the relative transmission rates for each sub-VL. When all normalization factors are equal, the sub-VLs are transmitted at the same rate. If a VL is using Sub-VL scheduling then a Normalization Factor is required for each Sub-VL. The port properties specify the sub-VL number of the port. The VL property pages configure the transmit rate and sub-VL usage.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# AFDX Database

## What is the AFDX database

The AFDX database that ships with CoPilot is the key to preserving data translations for AFDX data between multiple projects. CoPilot includes a standard AFDX database, but users are free to copy and rename database files. CoPilot AFDX users can save engineering unit, AFDX port, Virtual Link, or even entire AFDX Network definitions to the AFDX database. Saved definitions can then be loaded into a CoPilot project and modified or supplemented as required. Auto detected VLs can be loaded from the database automatically during simulation.

## Adding definitions to the AFDX database

All Networks, VLs, Ports, and Fields in CoPilot AFDX have a “Save to Database” option in their context menu. Saving an object will save that item and all the item’s children. For example, saving a VL to the database will also save the VL’s ports and fields.

When saving an object to the database, the user is prompted for a user description. This description is used to facilitate the selection process. After entering a meaningful description and pressing the OK button all of the associated definitions including all child definitions are saved to the database.

## Loading definitions from the AFDX database

Objects saved to the database can be used to configure objects in the Hardware Explorer. Network, VL, Port, and Field information can all be saved and loaded with the AFDX database.

Network definitions are extracted from the database by selecting the “Load Network” option on the network context menu. This will open a dialog listing all of the saved networks in the database. Selecting a saved network and pressing the OK button will load the entire definition including all VLs, Ports, Datasets, and Fields.

VL, Port, and Field definitions are extracted by selecting a saved item listed in their add item dialog. The database definitions are added to the dialog by selecting the “Select from database” option. Note that key parts of the definition can be changed when loading from the database. For example when loading a VL from the database you can specify a different VL number. The result is a new VL is added to the Hardware Explorer with the exact definition and child definitions as the item saved, except the VL number is different.

## Default VL definition for Auto-Detection

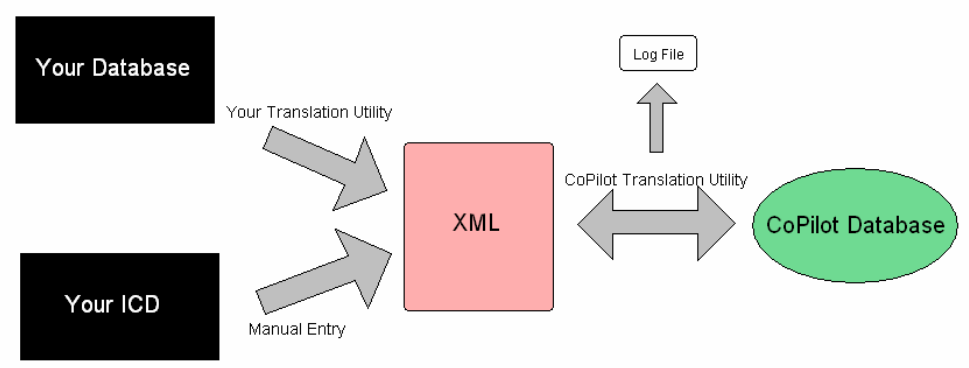
AFDX networks in CoPilot have an option for Auto-Detection of VLs. If a Network in the Hardware Explorer is set to auto detect VLs and lookup detected VLs is enabled for a network, then CoPilot searches the database for the default VL information when that VL is detected on the network. The entire definition for that VL number is loaded into the current running project. The VL may be marked as the default VL definition when saving the VL to the database.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

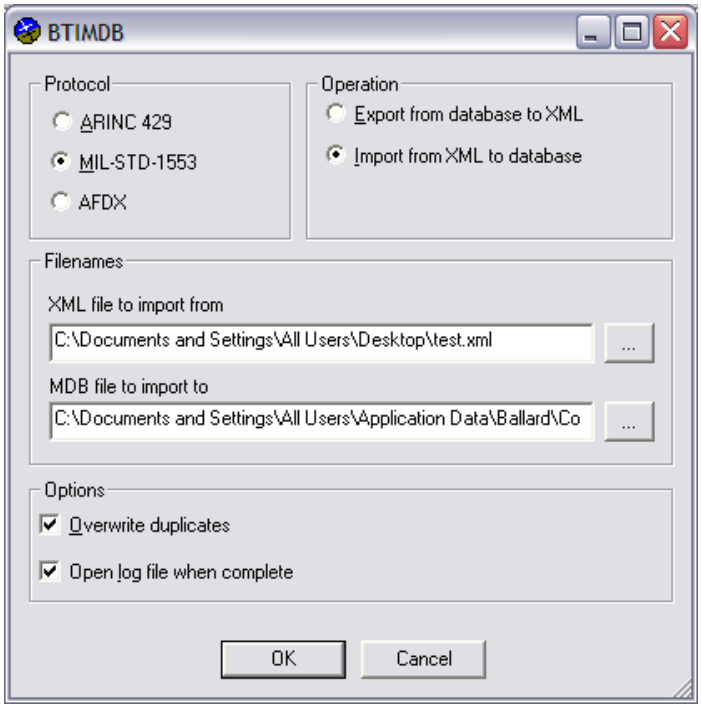
## XML import/export utility

A utility application (BTIMDB.EXE) is provided with the CoPilot software to facilitate the definition of AFDX Virtual Links and fields. This application can be found on the CoPilot CD. Utilizing a defined XML interface, users may translate their own AFDX data definitions located in their own ICDs (Interface Control Documents) or databases. The XML file describing the data definitions can be imported using the utility into the CoPilot AFDX database.




Through the use of this utility, the ease of providing data definitions for use by CoPilot goes up dramatically. Please see the "CoPilot Database Import-Export Utility-XML Format.PDF" located on the CoPilot CD.

The following image shows a screenshot of the utility, which allows you to specify various options such as which protocol database to import/export, the location of the XML and database files, and other options.






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Error Injection

## What is Error Injection

Error injection is a method of modifying the timing or contents of a frame to create timing errors, AFDX protocol errors, or logical errors using real AFDX hardware. These non-compliant frames are used to test system responses to bus faults. Error injection properties may be defined prior to or during a simulation. A variety of errors can be intermittently injected without stopping transmit schedules. Control of error injection is done manually, with scripting, or automatically using conditions in hardware like single-shot and triggers. Error injection settings are part of the project and as such, are saved with the project. Hardware supported Error Injection is available with Model B AFDX boards. The AFDX demonstration card does not support error injection. In addition to the hardware Error Injection features, all boards support additional errors created by deviating from standard transmission rates or modifying the frame with invalid data (such as wrong constant values).

The error injection enable at the network level operates as an override to other error injection settings. Error injection for the network must be enabled in order to inject errors for the VLs and ports defined below the network. The Enable Error Injection option is available from the network property page or the network context menu.

## Controlling Error Injection

Different types of Error injection exist at the Network, VL, and Port. Network error injection consists of the parametric frame-time frequency and is accessed on the Network View. VL error injection consists of the RSN errors and is accessed on the VL View. The Port level error injection consists of schedule errors (Transmission Control), frame errors (Packet Blending), payload data errors (Packet Blending), and Bus Control errors; port level error injection is accessed on the Port View. Each of these views has an error injection form with easy to use controls for adjusting and specifying the type of error desired. If error injection is disabled at the network level then all other error injection settings are ignored.

Each form has an enable/disable button. When the enable/disable button is enabled (pressed), it displays with a green background, and the settings on the form are active. Users can change the error settings at any time. However, if CoPilot is in the running state and the enable/disable button is pressed, then each change is automatically applied to the hardware. If users want to delay the transmission of errors while they are editing the settings they can simply disable error injection. When the enable/disable button is enabled again the current settings on the form are applied to the hardware for error injection.

The VL and Port Level Error Injection forms have a Single-Shot option. When the error condition is set to be in single-shot mode, the specified error is injected into a single frame instance then the error is disabled. Once CoPilot detects that the error was injected the enable/disable button will be switched to disabled. To activate the error condition again, simple re-enable the condition. When the Single-Shot option is turned off the specified error is applied over and over again.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Network Parametric Frame-Time Frequency

Each Network with defined transmit VLs creates a transmit schedule. When CoPilot and the hardware generate the schedule, schedule items (VLs or sub-schedules) are grouped into frames where each frame is processed within a frame time interval. The parametric frame time frequency modifies the frame times in the schedule by scaling the frame times up or down. The parametric frame frequency effectively scales the transmission rate for all VLs on the network. This type of error injection is used to create timeouts for expected ports and VLs or to create BAG errors by over-using the allocated bandwidth.

## VL RSN Error Injection

RSN Error Injection is configured from the VL View. Use this form to modify the sequence number stream of the VL transmission. Note, error injection for the RSN is ignored when Auto RSN generation is disabled.

There are two types of RSN errors available. The step type is used to permanently jump the sequence number to the value specified and continue from there. The impulse type is used to temporarily overwrite the sequence number with the specified value. With the impulse type, once the error is disabled the sequence number will resume from the normal sequence. The value can be altered by a relative value or set to an absolute value. When using relative values the hardware will advance the RSN to the next value in the normal sequence then add the specified value to create the result. The RSN rollover condition is automatically handled for the user by the hardware.

## Port Error Injection

### *Schedule Errors (Transmit Control)*

#### Stutter Mode

Stutter Mode of the Transmit Control error injection for port objects causes a transmit Port to immediately retransmit the exact same frame a second time. The result is the transmission of two identical frames (exact same message) back to back regardless of VL timing restraints. When a redundant network is configured, the transmission is stuttered on both Network A and Network B (if enabled).

#### Repeat Mode

Repeat Mode of the Transmission Control error injection for port objects causes a transmit Port to be processed twice for each entry in the schedule. The result is the transmission of two frames from the same Port (updating RSN, etc…) back to back regardless of VL timing restraints. When a redundant network is configured, the transmission is repeated on both Network A and Network B (if enabled).

### *Frame Errors (Packet Control)*

Error injection of frame data controls how the packet is created for transmission on the bus. The port contains data for a normal frame and another entire frame for error packet blending. The blend settings control the pieces of the error packet inserted into the transmitted frame. Interface ID and CRC values can also be configured for error injection.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### Auto-Interface ID

The Interface ID specifies the bus of the transmitted port. The default operation of the AFDX hardware is to automatically insert the interface identifier. Error injection can disable this automatic insertion. When the automatic Interface ID insertion is skipped, the transmitted value is either the value from the normal packet or the value from the error packet if Interface ID blending is enabled.

#### Frame Header Values

The Port View's Packet Control error injection controls the ability to inject errors in the frame's headers. The various Interface ID, Blending, and CRC settings affect which pieces of the Error Packet are injected into the frame. Frame header errors can be any combination of the following:

- MAC Layer – Interface ID or entire MAC address
- IP Layer – IP length, entire IP header or IP CRC value
- UDP Layer – UDP length, entire UDP header or UDP CRC value

#### Payload Data

The Port View's Packet Control error injection controls the ability to inject errors in the frame's payload. When the user wants to use payload data error injection the entire payload is replaced with a user specified payload.

### *Network Errors (Bus Control)*

Port Bus Control Error Injection changes the bus enable and the bus transmission order. This error injection is available for redundant networks only.

#### Invert Bus Enable

Invert Bus Enable changes the enable state of the selected bus. If the bus is currently inactive (disabled at the port or the VL level) then inverting the bus state enables transmission on that bus. Conversely, if the bus is enabled with the invert error injection setting on, then the bus does not transmit.

#### Invert Bus Transmission Order

Invert Bus A/B Order changes the transmission order between the two buses. For example, if the port is configured to transmit on bus B before bus A and the invert bus order is set, then the frame transmits on bus A followed by bus B.

## ARINC 664/AFDX Scripting Events

#### Overview

This section list and describes the events fired from various ARINC 664/AFDX objects within CoPilot. For more detailed description of CoPilot scripting see *Automated Test Environment (ATE)* on page *399*. Refer to the *CoPilot Object Model* documentation in the Common Help Topics section of the Start Page (version 5) or the **Help and Learning** tab group on the **Backstage** (version 6) for a full description of all supported properties and methods.

#### ARINC 664/AFDX Card Events

See the _IBT664CardEvents Class reference in the CoPilot Object Model documentation.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### ARINC 664/AFDX Network Events

See the _IBT664NewtorkEvents Class reference in the CoPilot Object Model documentation.

#### ARINC 664/AFDX Monitor Events

See the _IBT664MonEvents Class reference in the CoPilot Object Model documentation.

#### ARINC 664/AFDX VL Events

See the _IBT664VLEvents Class reference in the CoPilot Object Model documentation.

#### ARINC 664/AFDX Port Events

See the _IBT664PortEvents Class reference in the CoPilot Object Model documentation.

#### ARINC 664/AFDX Field Events

See the _IBT664FieldEvents Class reference in the CoPilot Object Model documentation.

---

Note: In addition to the event interfaces listed above, the CoPilot Object Model documentation describes the available properties and methods. Open the CoPilot Object Model documentation from the Start Page (version 5),the **Help and Learning** tab group on the **Backstage** (version 6), or the Help menu for the latest information.

---



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Part 2: CoPilot Standard-Advanced Topics

## Part 2 Overview

This section covers the advanced topics of CoPilot Standard. All of these advanced features and all other features of CoPilot Standard are also a part of CoPilot Professional. Included in this section are hardware playback, software playback, linking properties and data values, and the CoPilot automation model.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Hardware Playback

## Hardware Playback Overview

### What Is Hardware Playback?

Hardware Playback is a CoPilot feature that allows Ballard hardware to replay previously recorded bus activity. For example, you can capture bus traffic through the CoPilot monitor, and then replay all or part that data onto the databus.

MIL-STD-1553:

Playback allows you to simulate BC and RT traffic based on previously monitored data. If an RT is selected for playback, the RT response is also supplied from monitored data. If an RT is not selected for playback, BC commands referencing those RTs are still transmitted but the RT response is suppressed, allowing an external RT to respond. The bus activity during Hardware Playback, including external RT responses, can be recorded with concurrent monitoring of a level C or higher hardware.

ARINC 429:

Hardware Playback provides a way to simulate individual systems (LRUs) on an ARINC 429 databus by retransmitting previously recorded activity. For example, you could capture data from one or more 429 systems through the CoPilot Sequential Monitor, and then replay part or all that data onto the 429 databus bus. Typically, one transmit channel is used for each 429 system being simulated and the playback rate is controlled by the time-tags.

### Hardware Playback versus Software Playback

Two kinds of playback are available through CoPilot: Hardware and Software Playback . Both use information captured through the monitor, but they serve different purposes.

- In **Hardware Playback**, previously recorded monitor records are replayed onto the databus. The objective is to test or interact with real (external) 1553 terminals or ARINC 429 Equipment using realistic data.
- In **Software Playback**, a monitor record is replayed without any interaction with the databus or hardware. The objective of Software Playback is to examine previously recorded data at a readable speed using all the displays, graphics, and tools available in CoPilot.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The table below clarifies the differences between these two features and illustrates the usefulness of each approach.

| CoPilot Playback Modes | |
|---|---|
| **Hardware Playback** | **Software Playback** |
| Used for testing external equipment | Used for analyzing data in detail |
| Actively transmits on the databus | No interaction with databus or board |
| The monitor source can be created using a keyed Ballard hardware | The monitor source must be created using a Ballard board with CoPilot hardware key |
| Hardware playback requires a Ballard board with a CoPilot hardware key | Software playback does not require hardware or CoPilot software key, but must have been recorded with a CoPilot key |
| Playback speed is based on recorded time-tags<br>(ARINC 429) Playback speed can be scaled per-channel | Playback speed can be increased or reduced to facilitate analysis |
| (MIL-STD-1553) Replays the BC and only selected RTs from the monitor record data source | Replays all parts of all messages from the monitor record data source |
| (MIL-STD-1553) Can concurrently monitor external RTs to verify playback (requires C model hardware) | No concurrent monitoring capabilities (no actual transmission to monitor) |

### Applications for Hardware Playback

There are many useful applications for Hardware Playback. You can use Hardware Playback to:

- Capture operational information in the field and reproduce the results in the laboratory without the need for costly equipment
- Test external equipment on the databus in a stable environment with known data timing
- Create repeatable test sequences and easily share these tests with other CoPilot projects and users
- Determine how a system responds to special conditions by selectively modifying records in the monitor source file

### Hardware Requirements

The concurrent monitoring feature of level C or higher MIL-STD-1553 boards allows you monitor the databus during Hardware Playback (see *CoPilot-Compatible Hardware* on page *61*).

### How Does Hardware Playback Work?

Setting up and running Hardware Playback is a three-step process:

- Prepare the data source
- Configure the card for Hardware Playback
- Run the Hardware Playback simulation

The following sections explain each of these steps in detail.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Preparing the Data Source for Hardware Playback

### Collect Data for Playback

Hardware Playback uses information collected through the Sequential Monitor. Transmit 429 and 1553 channels placed in Hardware Playback mode will access recorded monitored data (MonData)from the Hardware Explorer tree or via a playback file exported from a Monitor View (file used for 429 only). Each has their advantages. CoPilot Playback files (*.CPB) are simple files that allow users to filter records or manually edit playback values and timing. This is particularly helpful when trying to recreate scenarios, like errors, that were previously monitored. Alternately, playing back data from monitored data is extremely simple.

### Prepare the Data Source

Before playing back on a databus, data must be collected through the Sequential Monitor into a MonData file or a playback file must be loaded. When viewing a MonData file in a Monitor View, the status bar of the Monitor View displays the status of the MonData file. To be able to connect to a MonData file as a playback source, the MonData file must disconnected from the Sequential Monitor. A disconnected MonData file is one that is not actively configured to record new monitor records. Connected MonData files cannot be used simultaneously to record new data and function as a Hardware Playback source. Playback can only be linked to an unbound MonData files.

To unbind a MonData file:

- Stop the simulation, then right-click the MonData file and 'Disconnect' to disconnect the MonData file from the Sequential Monitor object in the Hardware Explorer tree
- Alternatively, running the hardware again will automatically unbind the currently connected MonData file and create a new one as long as the sequential monitor is not configured to append records to the existing file

### Optional Customizations

This monitor record can be used as recorded for analysis, modified through the record editor, customized through display filters, or exported into special Hardware Playback files. For more information, see *Export* on page *78*. Playing back from an (unbound) Monitor View is the most flexible Hardware Playback source, since it is requires few steps to configure and can be easily viewed, filtered, and edited. For more information on customizing Monitor View content through display filters, see *Time Correlation* on page *78*.

Note: Display filters have no affect on which records are replayed during Hardware Playback. However, for MIL-STD-1553, RT responses can be filtered out for particular RTs.

### Modifying Monitor Records for Hardware Playback (ARINC 429)

In most cases, you will probably want to play back data exactly as it was received. When testing systems however, it is sometimes useful to see how the system responds to unexpected conditions. One way to accomplish this is to modify one or more records in the Monitor View file targeted as the Hardware Playback data source.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To change the value of a monitored record:

1. Right click on the record you wish to edit and choose **Edit Record Value** from the context menu to open the Record Value Editor
2. Select a radix from the listbox in which to view and edit the record data (see figure below)
3. Enter the new data value(s) in the New textbox
4. Click **OK** to apply the changes and close the editor (a warning dialog will prompt you to confirm this action)

Modified data becomes a permanent part of the monitor record and is saved with the Monitor View component. Changed monitor records are highlighted by a light peach background.

## Limiting 1553 Hardware Playback with Index Points

In normal operation, CoPilot will replay the entire contents of a MIL-STD-1553 monitor data file from beginning to end in a continuous loop. Alternately, a portion of the record can be selected by specifying the start and end index.

In order for the Index commands to be available, the Monitor View must be selected as the data source and Hardware Playback must be enabled (to accomplish this, see *Configuring Hardware Playback* on page *330*).

To define the index:

- Click on a row in the Monitor View window to select it
- Right click to access the context menu or select the Monitor menu from the menu bar
- Choose the **Set Start Index** or **Set End Index** commands

When a row is designated as the start or end index, it is marked with a background color.

If you have defined the index, you can navigate between those points with the Go To Playback Start/End buttons (see figure below). If the index is not explicitly defined, these buttons move between the first and last record in the Monitor View file.



To remove a start or end index:

- Right click the row marked with an index to access the context menu
- Choose the **Remove Start Index** or **Remove End Index** command

## Creating a Hardware Playback Export File

### Advantages of an Export File

Although any open Monitor View window can be used as the source for Hardware Playback, there are certain advantages to creating a special playback file. Since the export can be limited with message highlighting, you can replay export a segment of the monitored data rather than the entire file. That can be a great advantage if you want to focus on a few minutes of a very large monitor



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

file. A Hardware Playback export file (rather than a Monitor View window) may also be easier to share between projects and other CoPilot users.

Note: Hardware Playback files contain the information needed for playback but are not formatted for user interaction. Since they cannot be reread through the Monitor View window, any change in data values should be done before the export is created.

### Display Filters and Highlighting

Through display filters and highlighting, the user can control which records from a monitor file are extracted to playback, print, and export files. Display filters can be used to limit the monitor display to a specific filtered records before the playback file is written.

For example, to filter out all but a single ARINC 429 channel:

1. Click the Display Filter **Edit** button in the Monitor View tool bar to open the Filter Criteria dialog
2. Click the **None** button to clear all selections
3. Select a single channel by checking its box, and click **OK** to close the dialog and apply the filter

You can also filter channels out of the data source when you configure a channel for Hardware Playback. Preserving multiple channels in the monitor record allows multiple channels to configure Hardware Playback from the same source file and still play back unique channels.

The user may export the entire view to a playback file or limit the export through highlighting. See *Export* on page *78* for more information about the use of highlighting. See the following *Setting Up Hardware Playback* section below for specifics on configuring ARINC 429 and MIL-STD-1553 hardware playback.

### Export File Creation

Once the Monitor View window is prepared through display filters and highlighting, the user can export the target information to file through the monitor menu or monitor context menu.

To create a Hardware Playback file:

1. Right click the Monitor View window to access the context menu and select **Create HW Playback File**
2. Choose **All** to export all records as displayed, or choose **Selection** to export only those records selected through highlighting
3. Click the **Browse** button on the Export dialog to specify the path, enter a filename, then click **OK**

The file is typically saved in the project folder, but that is not required. Hardware Playback files are saved with a ".CPB" file extension.



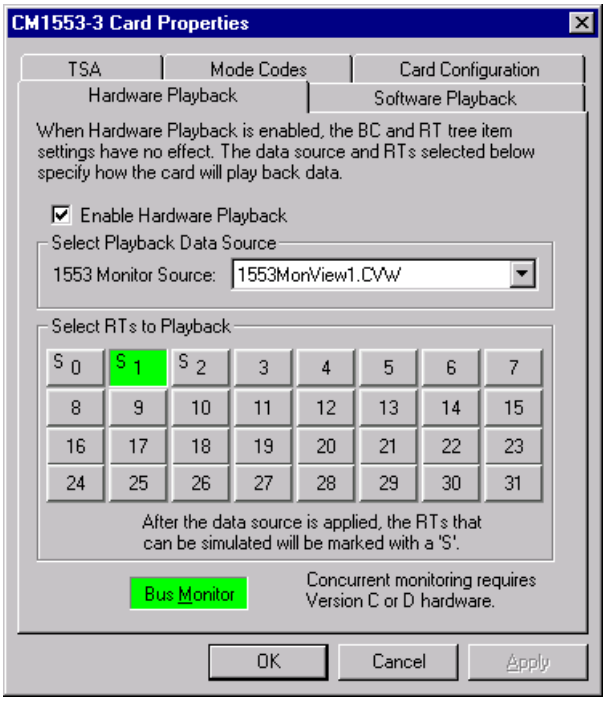
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Configuring Hardware Playback

## Configure a MIL-STD-1553 card for Hardware Playback

The following steps will configure MIL-STD-1553 for hardware playback:

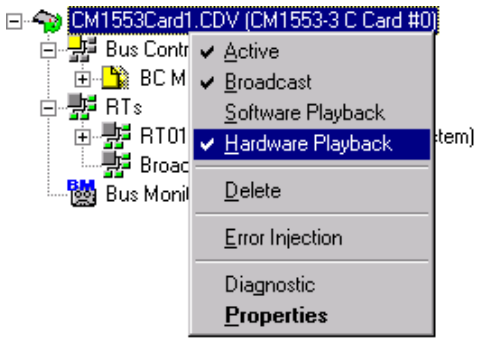
1. Right click on the board icon in the Hardware Explorer tree and choose **Properties** to open the Card Properties dialog
2. In the Hardware Playback tab, select the **Enable Hardware Playback** option (see figure below)
3. Select the data source from the listbox and click the **Apply** button to load the source file (RTs that were addressed by the BC in the source will be marked with an "S" symbol)
   - *The data source for Hardware Playback is a 1553 Monitor View file that is disconnected from the monitor.*
4. (Optional) Select any **RT** button with an S symbol that you wish to simulate using monitored data (key will become depressed and green)
   - *To allow an external RT to respond to playback, do not select it here. By selecting an RT addressed in the BC schedule, you choose replay its response (if any) from the data source and thus ignore any external activity for that RT.*
5. (Optional on model C boards) Engage the **Bus Monitor** button if you wish to concurrently monitor external RTs during playback



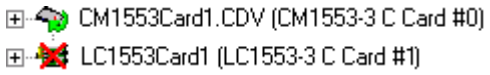

The card icon in the Hardware Explorer tree will be marked with a green arrow icon. You can toggle the card in and out of Hardware Playback mode from the context menu (see figure below).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




The board icon will display a red X (see figure below) if there is a problems with the Hardware Playback configuration.



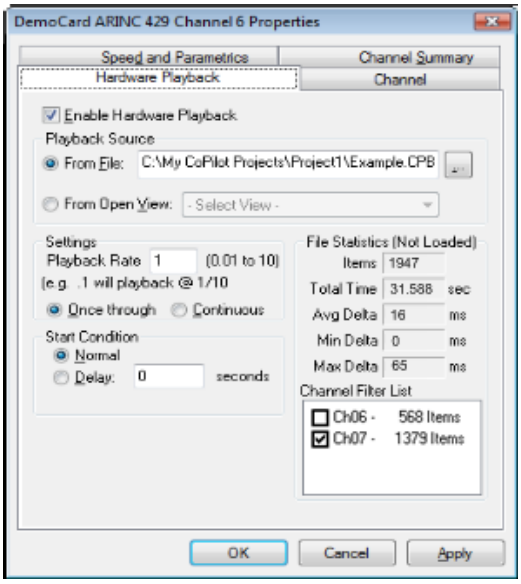

To configure a transmit channel for Hardware Playback, the transmit channel must be on a board with a CoPilot key and a data source must be ready for use (see *Preparing the Data Source for Hardware Playback* on page *327*).

## Configure a 429 transmit channel for Hardware Playback

1. Right click on a transmit channel in the Hardware Explorer tree and select Properties to open the Channel Properties dialog (see figure below)
2. Select the Hardware Playback tab and choose a type of data source:
   - If you chose a playback file, click the browse button to browse to a previously saved .CPB Hardware Playback file
   - If you chose MonData file, select a file from the list box
3. Click the Apply button (so that information on the selected source is loaded into the File Statistics frame)
4. Specify which channels from the data source to play back on this transmit channel in the Channel Filter List (in the File Statistics frame)
5. (Optional) Specify playback settings such as speed, loop condition, or start condition
6. (Optional) Click the Enable Hardware Playback checkbox (you can also toggle Hardware Playback mode through the channel context menu)
7. Click OK to apply the changes and close the dialog



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*ARINC 429 Hardware Playback property page*

The ability to specify the channel(s) to replay allows individual transmit channels can run ARINC 429 hardware playback from the same source file and choose the same or different channels from that common source.

Note: Playing back data from two or more channels on a single transmit channel is possible but unusual since the playback does not represent a real external system. If you select more than one channel from the Channel Filter List (see figure above), the labels for all channels will transmit in the same order as they appear in the Monitor View source file. For CM429-1 boards, engaging Hardware Playback mode for one transmit channel places all transmit channels on the card into Hardware Playback mode.

Now that the card is set up and configured, you are ready to run Hardware Playback. You can replay the entire file or selected potions.

# Running Hardware Playback

### Runtime Options

Hardware Playback is started with the CoPilot run button. Once in playback mode, playback controls in the Monitor View data source allow you to suspend and restart playback. Other runtime controls include:

- **Breakpoints**—Use breakpoints to pause playback at record of interest
- **Skip**—When playback is paused, you may specify a new resume point, thereby skipping (or repeating) a segment of the record
- **Step**—When paused, you can playback one record at a time, stepping forwards or backwards

### Runtime Controls

To activate a Hardware Playback simulation, the card must be configured and enabled for Hardware Playback.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To activate Hardware Playback:

1. Click the CoPilot Run button to start the Hardware Playback simulation initially and to animate the objects and displays in CoPilot
2. Click the CoPilot Stop button to end the Playback completely and return CoPilot to edit mode

### Monitor View Controls

While running, Hardware Playback can also be paused with the control buttons in the Monitor View data source.

- Click the Pause button to pause (or resume) the playback

---

Note: Because data is preloaded onto the card during playback, there may be a lengthy delay between when you press the pause button and when records actually cease transmitting.

---

### Runtime Behaviors

Various runtime behaviors unique to Hardware Playback are described below.

Messages, 1553 SAs, fields, etc. on the channel that is running Hardware Playback are ignored. However, Hardware Explorer tree specifications are still used by the Monitor View window to interpret the recorded data into engineering units. Note that all 429 receive channels and 429 transmit channels not configured for playback operate normally.

- Records in the Monitor View are highlighted as they are replayed
- Hardware Playback speed is controlled by the time-tags in the data source
- Because each transmit channel can be configured for Hardware Playback individually, they can start at different times (if a delay time is specified)
- A channel may be configured to replay its file once and then stop, or to loop continuously

## Pausing Hardware Playback with Breakpoints

Breakpoints are used to mark points of interest in a Monitor View window. They may be applied at any time. During 1553 hardware and both 1553 and 429 software playback, playback activity is paused when a breakpoint is encountered. Breakpoints are not used with 429 hardware playback.

### Adding Breakpoints

To tag a monitor record with a breakpoint:

1. Click on a row in the Monitor View window to select it
2. Right click to access the context menu or select the Monitor menu from the menu bar
3. Choose the **Breakpoint Record** command to tag the selected monitor record with a breakpoint icon
3. Repeat the above process to clear a breakpoint



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When a record has been tagged with a breakpoint, the breakpoint icon appears. The breakpoint icon is visible in all display modes except 1553 Form.

When Hardware Playback is started, it will enter a paused state when it encounters a breakpoint. To continue from that point, press the pause button to resume the playback.

#### Navigating between Breakpoints

Press the Go To Previous Breakpoint and Go To Next Breakpoint buttons to move the highlight to the record marked with a breakpoint (when playback is not running).



If the Go To Breakpoint button is pressed when Hardware Playback is paused, the resume point is moved to that breakpoint.

#### Ignoring Breakpoints

When the Ignore Breakpoints button is depressed, breakpoints are ignored during playback.

## Specifying the Hardware Playback Resume Point

When Hardware Playback is paused, the current message is highlighted. In normal operation, playback resumes at the next message. Alternatively, you can move the resume point to playback at a different location using the Go To Breakpoint buttons or the Set as Next Message command.

#### Set As Next Message

Set as Next Message command is available when Hardware Playback simulation is running, but in a paused state.

To move the playback resume point navigate through the Monitor View display to the message at which you wish to resume playback. Then right click that message and choose **Set as Next Message** from the context menu

#### Go To Next Breakpoint

When Hardware Playback is paused, use the Go To Previous Breakpoint and Go To Next Breakpoint buttons to move the playback resume point to that record.



After you have moved the Hardware Playback resume location, you can continue playback from this message by using the Step control or toggling the pause button.

## Advancing Hardware Playback in Steps

You can advance Hardware Playback in steps, forward and backwards one message at a time, using the Monitor View Step control.

The Hardware Playback simulation must be running but in a paused state to access this feature.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To advance playback in steps:

1. While playback is paused, click the forward and back buttons in the Step frame in the Monitor View window
2. Press the Monitor View Pause button to resume continuous playback



As you click the Step buttons, the Monitor View highlight will indicate which message is being replayed.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Software Playback

## Software Playback Overview

### Need for Replayed Data

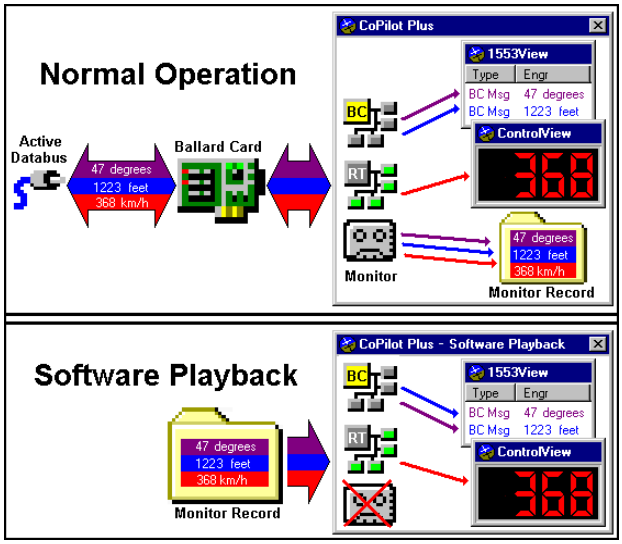
Software playback is used to view, analyze, and process previously recorded data sets multiple times. In addition, activity on a databus often occurs too fast to view to in real time, making it hard to locate or identify anomalies. CoPilot's software playback feature can slow down, speed up and step through the replay of data through the software.

### What Is Software Playback?

Software Playback replays previously recorded databus activity through all CoPilot views and displays as if it were being received from the databus.

### How Does Software Playback Work?

In software playback mode, interaction with the Ballard hardware and the databus is suspended (see figure below). A previously recorded monitor record becomes the source for all channel activity. Monitor recording is halted and the Data Generator is paused to prevent data from being driven by two sources.




Software Playback viewing data with displays and controls



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### What Are the Benefits of Software Playback?

Software playback can increase your productivity and save you time. Some of the advantages of software playback are:

- Allowing recorded activity to be viewed and processed repeatedly
- Slowing bus activity down to "human speed" to locate areas of interest
- Pausing playback at specified points to analyze data, transitions, and error conditions
- Limiting the playback to a user-defined subset of the monitor record

### Applications for Software Playback

There are many uses for software playback. The following are examples of possible applications:

- Record in the field, then recreate the activity in the lab for detailed post-analysis
- Share a record of bus activity with other CoPilot users for comment and analysis
- Process the same file repeatedly using scripts to automate analysis and discovery

### Using Software Playback

If you have data recorded by the monitor displayed in an open Monitor View window, using software playback can be as simple as engaging software playback mode and pressing the CoPilot Run button. However, additional features and tools are provided to give you greater control and further options.

# Software Playback Operation

Configuring a CoPilot board for a successful software playback simulation includes setting the board into software playback mode, linking to monitored data, and running the simulation.

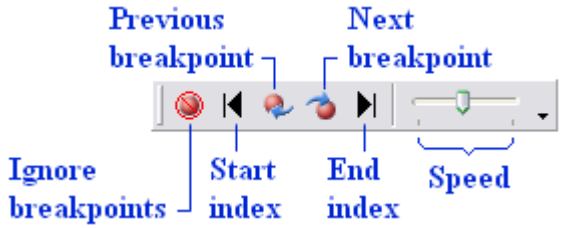
To run a software playback simulation:

1. Turn on software playback mode for a board in the Hardware Explorer tree
2. Select a monitor record to be the playback data source
3. Start software playback by pressing the **Run** button
4. (Optional) Control speed, insert breakpoints, and define the index from the Monitor View window

Once in Playback mode, configuration options and Playback controls are accessed from the Monitor View window. The Monitor View window hosts the Playback tool bar (see figure below).



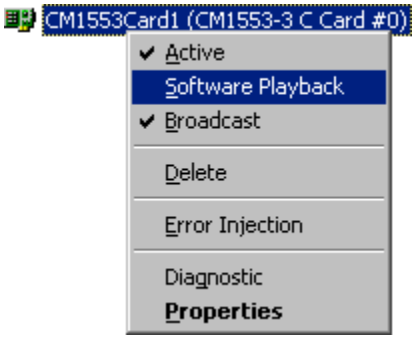
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




Note: Software playback is a run-time only mode. Configurations, settings, or links unique to software playback cannot be saved to file.

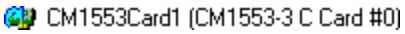
## Engage Software Playback

### Activate Software Playback Mode




- Right click on the board icon and choose Software Playback from the context menu
- Alternately, choose Properties from the context menu, select the Software Playback tab, and click the check box to enable software playback mode

When software playback is enabled, the playback icon is visible on the board in the Hardware Explorer tree.




In addition, the playback controls in the Monitor View window hosting the data source for software playback change from dimmed to available when playback mode is engaged.

### Software Playback Behavior

When CoPilot is put in software playback mode, interaction with the Ballard hardware and the databus is suspended and monitor recording is halted. Objects in the Hardware Explorer tree respond to the playback data just as they do to an external databus and all display windows and CoPilot methods are available.

Once software playback mode is engaged, the next step is selecting the data source for playback and hosting it in a Monitor View window so that the playback controls are accessible.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

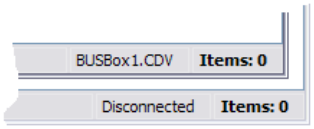
## Select Software Playback Source

Software playback data can be sourced from active or inactive data residing in the board's monitor or from a saved Monitor View file.

### Select the Current Monitor Record

If there is recorded data in the monitor, the record count will be displayed next to the monitor icon in the Hardware Explorer tree. Be sure that the data you wish to replay has been recorded before running software playback.

The Monitor View window that is linked to the monitor icon in the tree when the monitor is selected as the software playback source is identified in the status bar (see figure below).




### Select a Saved Monitor Record

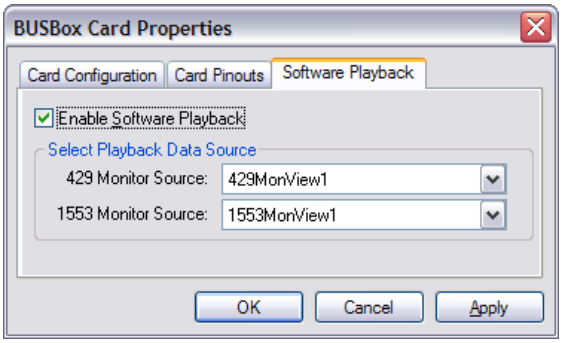
If you open and play back from a saved Monitor View, be aware that objects referenced in the playback source must be in the Hardware Explorer tree to work with software playback. The exception to this is auto-detected items, such as ARINC 429 receive labels and MIL-STD-1553 subaddresses, that are added while running.

Note: Selecting a Monitor View as the playback data source, rather than the board monitor, can result in a difference between the Monitor View hosting the playback source and the Monitor View holding the most recent data.

In order to use a saved monitor record as the source of data for software playback, it must be visible in the CoPilot Workspace display area.

To specify a save monitor file as the playback data source:

1. Right click on the board icon in the tree and select **Properties** from the context menu to open the Board Properties dialog
2. Click the **Software Playback** tab to access the software playback properties
3. Select the Monitor View file of your choice from the drop down list box
4. Click **OK** to close the Board Properties dialog






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### After the Source Has Been Selected

When a data source for software playback is selected, the playback controls in the Monitor View window hosting that source change from dimmed to available.

Now that the source for software playback is selected and hosted in a Monitor View window (so that the playback controls are accessible), the next step is running the playback simulation. An optional step is setting the index range or adding breakpoints.

## Run Software Playback

### CoPilot Tool Bar Controls

The Run and Stop buttons in the CoPilot tool bar normally toggle between edit mode and simulation mode. When software playback is engaged, they also control playback.

Before the software playback simulation can be run, software playback mode must be enabled.

- Click the CoPilot **Run** button to start the software playback simulation initially and to animate the objects and display windows in CoPilot
- Click the CoPilot **Stop** button to end the playback completely and return CoPilot to edit mode

When the software playback simulation is started, the Monitor View display steps through the monitor record as it is played back, tracking with the playback and scrolling as needed. The currently highlighted record is the one being played back.

### Monitor View Controls

Some of the Monitor View buttons and controls used in edit and simulation modes may also be used in software playback mode, including the Recording controls.

In order to control software playback with the Monitor View recording buttons, the Monitor View window must be open on the desktop and must contain the data selected as the source for software playback.

- Click the **Pause** button to pause (or resume) the playback from the currently selected record

Note: Playback may only be paused from the pause button in the linked Monitor View window, not from the monitor icon in the Hardware Explorer tree.

While software playback is running you can change the speed, navigate through the record using the Go To Breakpoints or Index points buttons, or even advance the playback one record at a time with the Step buttons.

## Control the Rate of Software Playback

You can run software playback continuously at varying speeds or one record at a time, forwards or backwards.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Speed Control

Software playback speed is controlled by the slider in the Playback toolbar of the Monitor View window. When software playback is slowed, changing data values are more easily detected than under normal run conditions. By increasing the rate of software playback, a variation in a constant data pattern over long segments of bus activity can be quickly located.

To change the speed of software playback:

- Drag the slider control in the Playback toolbar in the Monitor View window to the left to slow the playback rate and to the right to increase it

When the playback speed is changed, it can be tracked by the rate that the records scroll by in Monitor View. The refresh and sampling rate of other CoPilot controls and windows do not change.

### Single Step

It is also possible to advance the playback a single record at a time, both forward and backward. This function uses the Monitor View Step keys. In edit or simulation mode, the  and  keys simply advance the monitor display to the previous or next record. In software playback mode, however, they cause the next or previous record to be replayed through CoPilot.

To single-step the software playback simulation:

1. Put playback into pause by engaging the Pause  button in the Monitor View window that is hosting the playback data source
2. Use the forward and back Step keys (see figure above)
3. To resume normal playback, click the Pause button again to release the pause state

In order to single step, you can engage the pause button, but software playback can also be put into a paused state with breakpoints.

## Pause Software Playback with Breakpoints

### Mark a Record with a Breakpoint

Breakpoints can enhance analysis by pausing the playback at specified points of interest. Breakpoints may be applied at any time, even when software playback mode is turned off. However, breakpoints perform no function outside of software playback.

To tag a monitor record with a breakpoint:

1. Click on a row in the Monitor View window to select it
2. Right click on the record and choose the **Breakpoint Record** command from the context menu to tag the selected monitor record with a breakpoint  icon
3. Repeat the above process to clear a breakpoint

When a record has been tagged with a breakpoint, the breakpoint  icon appears. The breakpoint icon is visible in all Monitor View display modes except MIL-STD-1553 Form view. When the software playback simulation is started, it will enter a paused state when it encounters a breakpoint.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To resume playback from a paused state, you can toggle the pause button to restart continuous playback. Alternatively, you could advance playback in one record at a time using the Step control.

### Ignore Breakpoints

When the Ignore Breakpoints button is engaged, all breakpoints remain in place but are ignored during playback. The Ignore Breakpoints button can be toggled while software playback is running.

To ignore breakpoints:

1. Click the Ignore Breakpoints button in the Playback tool bar in the Monitor View window to prevent the software playback from entering a pause state when encountering a breakpoint
2. Click the Ignore Breakpoints button again to reenable pausing on breakpoints

### Move Between Breakpoints

You can move between breakpoints at any time, even when software playback mode is disabled.

To move between breakpoints:

1. Click the **Next Breakpoint** button in the Playback tool bar in the Monitor View window to move to the next record marked with a breakpoint
2. Click the **Previous Breakpoint** button to move to the previous breakpoint

Using the Go To Breakpoint buttons moves the highlight to the appropriate record. If the record is outside of the current view, the display will automatically scroll to the location.

---

Note: When software playback is paused, these buttons move the resume point to the next breakpointed record. The resume point can also be set with a menu command.

---

## Specify the Software Playback Resume Point

When software playback is paused, the current message is highlighted in the Monitor View window. In normal operation, playback will resume at the next message. However, you can move the resume point to a different location in the monitor file using the breakpoint navigation buttons or the Set as Next Message menu command.

### Set as Next Message

The Set as Next Message command is only available when software playback is running, but in a paused state.

To move the playback resume point:

- Navigate through the Monitor View display to the message at which you wish to resume playback, right click that message, and choose **Set as Next Message** from the context menu



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Go To Next Breakpoint

When software playback is running or stopped, the breakpoint navigation buttons move the display to the next breakpointed record and highlight it. However, when software playback is running but paused, the navigation buttons also move the resume point to the next breakpointed record.

### Resume Software Playback

After you have specified the software playback resume location, you can continue playback from this message by using the Step control or by toggling the pause button.

## Set Software Playback Start and End Index

### Define the Index

By default, CoPilot will replay the entire contents of the monitor record selected as the software playback source from beginning to end in a loop. Alternately, a portion of the playback record can be selected by specifying the start and end index.




To define the index:

1. Click on a row in the Monitor View window to select it
2. Right click on the record and choose the **Set Start Index** or **Set End Index** command from the context menu

When a row is designated as the start or end index, it is marked with a unique background color (in all display modes except MIL-STD-1553 Form).

### Move to the Start or End of the Index

In order to move to the start or end of the software playback index, CoPilot must be in software playback mode, and the Monitor View window hosting the index must be the source for playback data. If a start or end index has not been explicitly set, these buttons move between the first and last records.

To move between index points:

1. Click the Go To **End Index** button in the Playback tool bar in the Monitor View window to move to the record marked as the end index
2. Click on the Go To **Start Index** button to move to the beginning of the software playback index.

Using the Go To Index buttons moves the highlight to the appropriate record. If the record is outside of the current view, the display will automatically scroll to the location.

### Remove the Start or End Index

The start and end index can be moved to different records by selecting a new record and choosing the appropriate Set command from the context menu. The index points can also be removed completely.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To remove a start or end index:

1. Right click the row marked with an index to access the context menu
2. Choose the Remove Start Index or Remove End Index command

# Software Playback Examples

The examples in the sections that follow illustrate how CoPilot's software playback feature may be used to accomplish specific tasks. The following is a summary of each example:

- Analyze the same record several times, focusing on different data each time, by combining software playback with scripting.
- Record 100% of bus traffic in the field, then replay selected portions in the laboratory to focus on just the information you wish to analyze.
- Compare findings with others by sharing a saved monitor record with CoPilot colleagues so they can recreate and analyze the same databus you recorded.

## Combine Scripting with Software Playback

### Analyze a Stable Data Set

When analyzing a segment of bus activity, software playback provides a reproducible record of data values. This record can be processed several times, based on different criteria. Comparisons have meaning because the control sample is always the same. The example in this section illustrates how to use scripting to process the same monitor file several times for different data.

### Set Up the Software Playback Simulation

To set up software playback:

1. Turn on software playback mode
2. Select the data source (if necessary)
3. (Optional) Set start and end index points
4. Place a breakpoint at the end point

By default, software playback loops through the monitor record until stopped or halted. In order to compare the data field reports, we want to pass through the record just once for each data field. Placing a breakpoint at the last record pauses software playback at the end of the monitor record so we can then stop the simulation and return to edit mode.

### Set Up the Script

Let's assume you wish to analyze several different data fields. We will use a script to gather statistics about a data field while the software playback simulation is running, then print a report when the CoPilot simulation is stopped.

To set up the data field report script:

1. Open a Script View window
2. Paste in the "CreateReport.txt" example script
3. Add the first field to the Script View object pane



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4. Rename the field “Field1” (click twice on the name to edit it)

### Run the First Analysis

Press the Run button to start CoPilot in software playback mode. Wait for the playback to reach the last record (tagged with a breakpoint) and enter a pause state. When you press the CoPilot Stop button, you will see a report in the Script View output pane listing the minimum, maximum, and average values for the data field you selected. This information is also saved to a file.

### Analyze the Remaining Data Fields

To analyze the next field:

1. Delete the first field in the object pane
2. Drag and drop in the new field
3. Rename it “Field1”
4. Comment out the line “fso.CreateTextFile filename” (to add this second report to the first by appending the file)
5. Run the simulation again and exit on breakpoint

You will see the report for the second field listed below the first (see figure below).

```
        -- Script View Output --
        Started Mon Jun 03 11:50:00 AM
Report for *True Heading*
The maximum value was 345.921875
The minimum value was 256.1796875
The average value is  300.767299107143

Report for *Altitude*
The maximum value was 3000
The minimum value was 3000
The average value is  3000
```

Repeat this process for as many fields as you wish. If you wish, you can add code to the script to record, calculate, and display additional information.

## Use Software Playback for Post-Analysis

### Recording Bus Activity for Software Playback

With software playback, you can collect real-time data in the field for later analysis. There is no need to worry about when to start and stop recording. Take the entire record of bus activity back to the lab, and replay the sections you want using software playback mode.

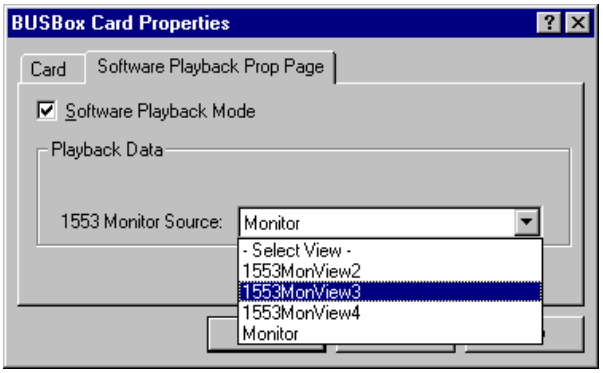
### Playing Back from a Saved Monitor Record

To select the monitor record for playback:

1. Open the Properties dialog from the card’s right click context menu
2. Select the Software Playback tab
3. Select the monitor file from the drop down list box



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




You can replay the entire record or limit the scope to a specific subset by setting the start index and end index. Observe the playback through CoPilot's graphical displays to aid in locating areas of interest. Pause the playback at specified records using breakpoints.

### Using Multiple Monitor Records

If you wish to compare data samples for the same bus, taken at different times perhaps, you can choose a series of monitor files as the playback source. When the first playback is run, record and observe the activity you wish to compare. You could record analysis information with a script. Then, following the steps above, select a new playback source. If you have several Monitor View files open on the CoPilot desktop, you could move between them as desired to analyze and compare.

## Share Data with Others Using Software Playback

### Recreate the Databus

With CoPilot, you can disseminate a recording of bus activity to other CoPilot users to get feedback, compare findings, or verify results. They cannot only examine a static list of data values or read a report; they can actually recreate the experience using software playback and CoPilot.

By using the monitor record (and perhaps a saved project) you have sent them, they can view the record dynamically. In addition, they can slow it down, or feed it through graphical displays. In short, the monitor record becomes a substitute for an active databus. Software playback is the difference between reading a transcript of a movie or watching a video of it. In addition, they can then run scripts, create displays, analyze data, and print or export data just as in normal simulation mode.

### Easily Distribute Data for Playback

To share playback data with others:

1. Record a monitor file(s) and save it to disk
2. Deliver it via e-mail, network, floppy, etc.
3. (Optional) Send the project file for that monitor record

They can create a CoPilot project that matches the bus topology and transmit schedule of the recorded bus (or open a saved project you have sent them). After they have opened the Monitor View file and selected it as the playback source, they can engage software playback mode and run the playback simulation.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Linking Objects

## Links Overview

Links are created in CoPilot to connect events (e.g. Value Change) to object properties. On one side an event serves as the "source" and is responsible for "sourcing" data to the other property, known as the "sink."

This simple concept allows the value changes from one object to be propagated to another automatically with no action required from the user. More complicated link patterns can be formed by creating multiple links from one object source (a one-to-many link pattern) or by chaining multiple links together from object to object (daisy chain). Some examples of different links that can be created are described in the following sections.

### Linking Hardware Explorer Objects

CoPilot links allows users to connect objects from the Hardware Explorer. Objects include channels, messages, fields, and other objects. A receive field on one terminal could, for example, become the data source for a specific transmit field or message (i.e. 429 label). Such a connection is created through the **Project | New Link** menu command.

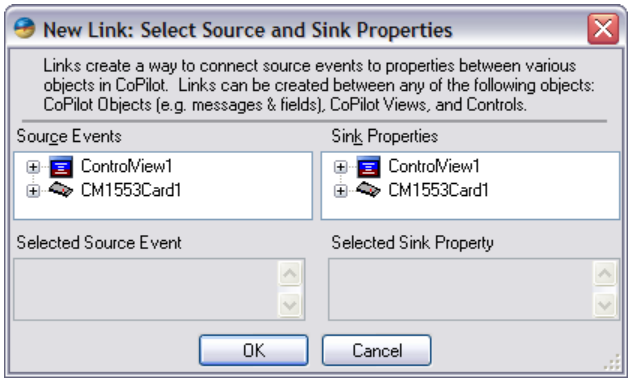
### Linking Controls and Graphical Objects

Links may also be created for connecting to/from Graphic Objects and Controls. These various controls can be hosted in the Control View. Most controls often serve as sinks to display data, such as a strip chart or aeronautical display. However, some controls lend themselves to being sources as well, such as in the case of a slider control. The linking of controls and fields is established through the **Project | New Link** menu command.



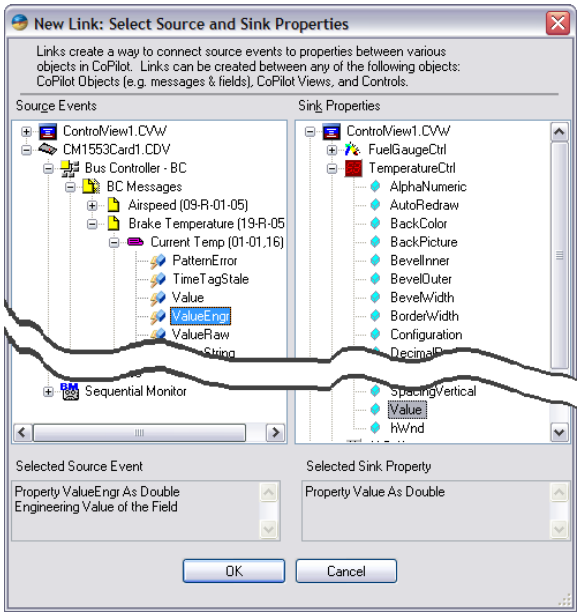
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Creating a Link




When **New Link** is selected from the Project Menu, the New Link dialog appears, prompting the user to input a selection for the source event and sink properties to be linked. Once the Link Dialog is opened:

1. Double-click an item in the Source pane (with a ⊞ icon) to expand it and view its properties. Note that sometimes objects are nested, so finding a field or label (429) for instance requires you to navigate first to the card, then channel, and so forth. See the next section for a more detailed description of link properties.
2. Select the desired source event. A description of the event will appear in a text box below the Source pane to help you choose the correct one.
3. Repeat the process for the Sink pane, selecting the desired object first, expanding its properties, and selecting a property.
4. Click **OK** to create the link.



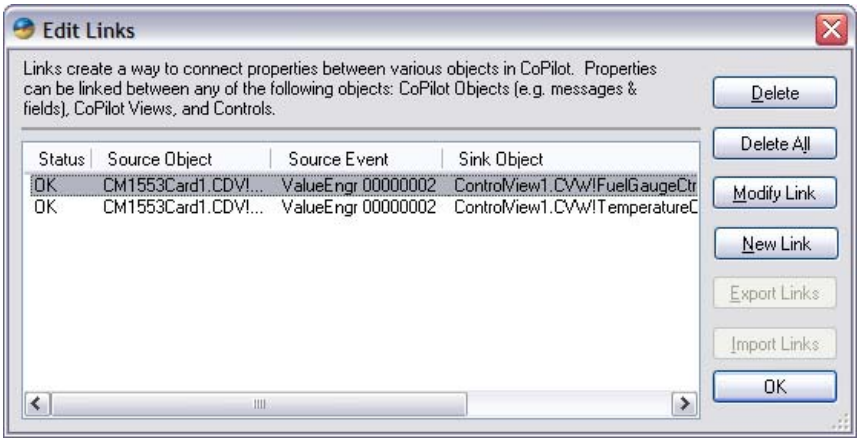

*Dialog for creating and editing links*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Viewing Link Status

Accessing the Edit Links window shows the status of all of the links available in the current project, as well as allowing the modification and deletion of links. This dialog can be accessed via **Project | Links Status** menu command.




*Link status dialog showing the existing links*

The Edit Links dialog shows each link on its own line with the status, source and sink information. The "Status" column will say "Broken Source" or "Broken Sink" if the source or sink object has been deleted since the creation of the link. These links can be repaired by pointing to a new property via the "Modify Link" button.

The "Modify Link" button will bring up a dialog similar to the New Link dialog, except the linked source and sink properties will be highlighted and shown in the dialog automatically.

The "Delete" button will remove the selected link from the project, whereas the "Delete All" button will remove all links from the project.

# Linkable Properties of Objects

## Description

As discussed above, linking objects requires a property to be selected for linking, as objects can be defined by one or more properties.

Fields are typically linked using the ValueEngr property, which is the human-readable engineering value. ActiveX controls, such as the various aeronautical controls available through the ControlView, are more complex. In this case, sometimes it makes sense to link to the Value property, although on some of the more specialized controls the best linkable property is similar to the control name itself. A table of commonly used control properties is available in *Appendix E: Control View Library* on page *445*.

The following sections describe the various linkable properties, and whether they are available exclusively as a source or sink. Properties that are sources are the values and state that can be 'read' from the object and then passed to the sink of another object. Sink properties are those that can set the new value of the property from the incoming source. Properties that are not labeled as source only or sink only, can operate as either a source or a sink.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: New properties and methods are frequently added to the CoPilot Object Model. Open the CoPilot Object Model documentation from the Start Page (version 5),the **Help and Learning** tab group on the **Backstage** (version 6), or the Help menu for the latest information.

# MIL-STD-1553 Linkable Properties

## *BC Properties*

- Pause is a property for the pause state of the BC to suspend the schedule.
- Enable (sink only) is not a runtime linkable property.
- Default (sink only) sets the BC to use a default schedule. This is not a runtime linkable property.
- RateBased (sink only) sets the BC to use rate-based scheduling. This is not a runtime linkable property.

## *BC Message Properties*

- BusBOpt (sink only) indicates whether the alternate bus is being used.
- Enable (sink only) is not a runtime linkable property.
- ErrorOpt (sink only) is the status of error injection for this message.
- MonitorOpt (sink only) indicates if the message is being monitored.
- MsgActivity (source only) is a field that contains a variety of activity flags.
- MsgError (source only) is a field that contains a variety of error flags.
- MsgSwd1 (source only) is the value of status word for command word 1.
- MsgSwd2 (source only) is the value of status word for command word 2.
- MsgTimeTag (source only) reports the time interval (in seconds) from the beginning of the run to the transfer or receipt of the message.
- Singleshot (sink only) indicates a message will be sent and then placed into the Skip state.
- Skip (sink only) is a toggle that causes a scheduled message to be paused and ignored. Skip has no effect on BC messages on the demonstration card.
- TimeTagStale is a Boolean value that indicates whether the time-tag has not been updated in *TimeoutStaleVal* seconds.
- TimeoutStaleVal (sink only) the number of seconds after the time-tag has been updated before message becomes stale.
- WordCountCwd1 is the number of words requested in command word 1.
- WordCountCwd2 is the number of words requested in command word 2.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### *Subaddress Properties*

- Enable (sink only) is not a runtime linkable property.
- ErrorOpt (sink only) is the status of error injection for this message.
- MsgActivity (source only) is a field that contains a variety of activity flags.
- MsgError (source only) is a field that contains a variety of error flags.
- MsgSwd1 (source only) is the value of status word for command word 1.
- MsgSwd2 (source only) is the value of status word for command word 2.
- MsgTimeTag (source only) reports the time interval (in seconds) from the beginning of the run to the transfer or receipt of the subaddress.
- MonitorOpt (sink only) indicates if the message is being monitored.
- SyncOpt (sink only) indicates a sync output pulse frames the response when set.
- TimeTagStale (source only) is a Boolean value that indicates whether the time-tag has not been updated in *TimeoutStaleVal* seconds.
- TimeoutStaleVal (sink only) the number of seconds after the time-tag has been updated before message becomes stale.

## ARINC 429 Linkable Properties

### *Channel Properties*

- Pause is a property for the pause state of the channel.
- Enable (sink only) is not a runtime linkable property.
- AutoEquip (sink only) automatically detect equipment ID for receive channels and automatically transmits the channels equipment ID for transmit channels.
- AutoFilter (sink only) automatically detects label activity and adds the label to the Hardware Explorer for receive channels.
- EquipmentNum (sink only) sets the equipment number for the channel.
- HighSpeed (sink only) is not a runtime linkable property.
- Label Rate (sink only) is the (maximum) update rate for servicing labels when LabelRateEnable is set in units of .1 Hz. For example, LabelRate=10 is sampled at 1 Hz.
- LabelRateEnable (sink only) sets/clears the label rate enable flag which samples the labels at the ‘specified label’ rate.
- MonitorType (sink only) sets the monitor type to: 0=Monitor Off 1=Monitor Selective 2=Monitor All.
- ParamAmplitude (sink only) sets the parametric amplitude (for hardware that supports this feature) with the range between 100 (full scale value) and 0.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- ParityType (sink only) is used to set the parity type: 0=even 1=odd 2=data.
- Wrap (sink only) configures the card to wrap transmission on this channel to receive channels. Note, only one transmit channel can have wrap set at a time.

### *Label Properties*

- Enable (sink only) is not a runtime linkable property.
- ExtTrigStep (sink only) the Channel stops at this message until an external trigger.
- IntervalTestEnable (sink only) turns on the interval test.
- IntervalTestError (source only) indicates an interval error.
- IntervalTestVal (sink only) value in milliseconds.
- Monitor (sink only) indicates if the label is being monitored.
- ParityErr (sink only) used to inject a parity error.
- PatternError indicates the pattern error state.
- PatternTestEnable (sink only) turns on the pattern test.
- SingleShotEnable (sink only) transmits message once, and sets skip after sending.
- SkipMsg (sink only) skips message when true.
- Sync (sink only) transmits an output sync pulse.
- TimeoutStaleVal (sink only) the number of seconds after the time-tag has been updated before message becomes stale.
- TimeTagStale is a Boolean value that indicates whether the time-tag has not been updated in *TimeoutStaleVal* seconds.
- Value is the engineering value.
- ValueDataRaw (sink only) is the raw upper 24-bits of the message.
- ValueEngr is the engineering value.
- ValueRaw is the raw 32-bit value (including label bits).
- ValueString (source only) is the string value of the message.

## ARINC 664/AFDX Linkable Properties

### *Network Properties*

- AutoDBVLs (sink only) Auto lookup detected VLs in the database.
- AutoDetectVLs (sink only) Auto detects receive VLs and add them to the project.
- DisableNetA (sink only) disables the “A” side of the redundant network (for redundant networks only).
- DisableNetB (sink only) disables the “B” side of the redundant network (for redundant networks only).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- EnableParamFreq Enable/disable the parametric frame frequency option.
- ErrInjEnable enables/disables error injection.
- MonAll (sink only) records all traffic on the network when enabled.
- OverrideMACConst (sink only) enables/disables the use of non-standard MAC constants. Cannot be modified while running.
- ParamFreqVal gets/sets the parametric frame frequency value (0 to 100 as a percentage).
- Pause pauses the network to stop processing and collection of data.
- PhyStatus (source only) gets the status of the card's physical interfaces.
- UserMACConstant (sink only) is the user specified, non-standard MAC constant value. Cannot be modified while running.

### *VL (Virtual Link) Properties*

- AutoDetectPorts (sink only) automatically detects and adds new ports to the project when set.
- DisableAutoRSN (sink only) disables automatic generation of RSN values.
- DisableIPCRC (sink only) disables automatic calculation and insertion of correct IP CRC value.
- DisableIntID (sink only) disables interface ID (bus number) insertion.
- DisableIntegrityA (sink only) disables intergrity checking on Network A.
- DisableIntegrityB (sink only) disables intergrity checking on Network B.
- DisableNetA (sink only) disables activity on Network A.
- DisableNetB (sink only) disables activity on Network B.
- DisableRedundancy (sink only) disables redundancy checking.
- DisableUDPCRC (sink only) disables automatic calculation and insertion of correct UDP CRC value.
- MonAll (sink only) records all traffic on the VL when enabled.
- RSNErrCtrl controls RSN error injection.
- RSNErrEnable enables RSN error injection.
- RSNErrType selects RSN error injection type.
- RSNErrValCtrl controls RSN error injection value.
- RSNErrValue selects RSN error injection value.

### *Port Properties*

- DisableNetA (sink only) disables activity on Network A.
- DisableNetB (sink only) disables activity on Network B.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- ErrPktAutoGen_IPCRCMode (sink only) controls mode of packet error injection of auto IP CRC.
- ErrPktAutoGen_InterfaceIDSkip (sink only) controls packet error injection of auto Interface ID generation.
- ErrPktAutoGen_UDPCRCMode (sink only) controls mode of packet error injection of auto UDP CRC.
- ErrPktBlend_IPHdr (sink only) uses IP header from error packet when set.
- ErrPktBlend_IPLen (sink only) uses IP length from error packet when set.
- ErrPktBlend_InterfaceID (sink only) uses interface ID from error packet when set.
- ErrPktBlend_MACHdr (sink only) uses MAC header from error packet when set.
- ErrPktBlend_Payload (sink only) uses payload from error packet when set.
- ErrPktBlend_UDPHdr (sink only) uses UDP header from error packet when set.
- ErrPktBlend_UDPLen (sink only) uses UDP length from error packet when set.
- ErrPktCtrl (sink only) controls packet error injection options.
- ErrPktEnable (sink only) enables packet error injection.
- ErrPortBusCtrl (sink only) controls packet bus error injection options.
- ErrPortBusEnable (sink only) enables packet bus error injection.
- ErrPortBus_InverseEnableA (sink only) enables bus error injection inverse on network A.
- ErrPortBus_InverseEnableB (sink only) enables bus error injection inverse on network B.
- ErrPortBus_InverseOrder (sink only) enables bus error injection reverse transmit order.
- ErrXmtCtrl (sink only) controls the number of times the transmit error is injected.
- ErrXmtEnable (sink only) enables transmit error injection.
- ErrXmtType (sink only) sets transmit error injection type (stutter or repeat).
- IPDestUnicast (sink only) uses unicast IP destination address format when set.
- MIB (source only) contains MIB (Management Information Base) statistics.
- Monitor (sink only) records all traffic on the port when enabled.
- PacketActivity (source only) contains packet (frame) activity flags.
- PacketBusNum indicates packet (frame) bus number (interface ID).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- PacketCount is the byte count of the packet (frame).
- PacketLength is the packet (frame) length.
- PacketSourcePort is the packet (frame) source port from the UDP header.
- PacketTimeTag (source only) time-tag low 32 bits.
- PacketTimeTagEx (source only) time-tag full 64 bits.
- PacketType is the packet (frame) type.
- Skip (sink only) skips the processing of port activity when skipped.
- TimeoutStaleVal the number of seconds after the time-tag has been updated before message becomes stale.
- TimeTagStale (source only) is a Boolean value that indicates whether the time-tag has not been updated in *TimeoutStaleVal* seconds.
- TransmitBBeforeA (sink only) transmits packets on network B before A when enabled.

### *FDS (Functional Data Set) Properties*

- FunctionalStatusSet (sink only) is the FSS (Functional Status Set) value.

### *DS (Data Set) Properties*

- Status (sink only) is the DS status property from the FSS (Functional Status Set).

## Generic Linkable Properties

### *Field Properties*

- OperatingLimitError (source only) is a non-zero value indicates an outside of operating limits error.
- TimeTagStale (source only) is a Boolean value that indicates whether the time-tag has not been updated in *TimeoutStaleVal* seconds.
- TimeoutStaleVal (sink only) is the number of seconds after the time-tag has been updated before message becomes stale.
- Value is the engineering value.
- ValueEngr is the engineering value.
- ValueRaw is the raw of the field.
- ValueString (source only) is the string value of the field.

### *Card Properties*

- Enable (sink only) is not a runtime linkable property.

### *Sequential Monitor Properties*

- All (sink only) sets the sequential monitor to capture all bus activity.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Enable (sink only) is not a runtime linkable property.
- Pause is a property for the pause state of the monitor to suspend data collection while running when set to a non-zero value (true).
- Run is the running state of the sequential monitor. This value can be modified to resume the sequential monitor while running.
- Run429 is the running state of the ARINC 429 portion of the sequential monitor. This value can be updated while running

# Helpful Hints

## Changes for Links in CoPilot 5 and 6

CoPilot version 5 changed the behavior of links in order to improve their performance in responsiveness within the CoPilot architecture. Now, a subset of link sources are available as compared with previous versions of CoPilot. This has brought about a marked improvement in link speed.

The drawback is that some link sources are now no longer available in CoPilot 5 and 6. The ones that were dropped were usually uncommonly used, so impact should be minimal. However, old CoPilot projects will need to be converted to be compatible with CoPilot 5 and 6. The conversion process occurs automatically, and it will display a report of how links were changed to be compatible with CoPilot 5 or 6, including a count of successful and failed conversions. Links that could not be upgraded will be indicated in the Link Status dialog, and may be changed to be compatible. Leaving broken links in place will not prevent the project from operating, however. Scripting can handle any links that cannot be repaired.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# CoPilot Automation

## CoPilot Automation Overview

### What Is OLE Automation?

OLE Automation is a protocol that defines an "object interface" as a means of exposing a program's functionality to development tools, macro languages, and other applications that support Automation. The object interface is a collection of objects that can be programmed. Each object has *properties*, which equate to variables that can be set, and *methods*, which equate to functions that can be called. Objects are nested (tree-style hierarchy) within a single application object.

The following sections present an introduction to the CoPilot Object Model, examples of CoPilot automation, and information on the mechanics of automating CoPilot.

### Controlling CoPilot Programmatically

CoPilot can be controlled programmatically with languages such as Visual Basic® or C++ using OLE Automation technology. CoPilot can also control OLE Automation–compliant applications using the Python Script Editor and the Command Prompt (see *Automated Test Environment (ATE)* on page *399*) or the Script View (see *Appendix H: Legacy CoPilot Support* on page *487*).

CoPilot's OLE Automation object interface exposes a subset of key functionality normally available through the user interface (mouse and keyboard input). For example, instead of right clicking a monitor in CoPilot and selecting "Pause" from the context menu, you could write a line of code to set the "Pause" property of the "monitor" object to "TRUE."

### Benefits of Automating CoPilot

So, why control CoPilot with lines of code instead of a mouse? There are many benefits to automating CoPilot. For example, you can:

- **Automate tasks**—Write code to automate and standardize frequently repeated test procedures in CoPilot.
- **Save time and money**—Use a few lines of code to run CoPilot and thereby run its data generators, scheduling tools, and other features instead of the hundreds of lines of code required to do the same tasks with custom logic and function calls from the Ballard API library.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Integrate systems**—Make CoPilot a seamless and invisible part of a larger system, application, or test procedure.
- **Achieve precise timing**—Modify data or perform other tasks with precise timing, based on events in a larger system.
- **Minimize human intervention**—Allow CoPilot to respond to other applications or even equipment, instead of only user input. Create autonomous test procedures that can run for long periods without requiring user input.
- **Use your custom interface**—Use code to bypass CoPilot's user interface and create your own interface for CoPilot functions that fit within your existing application.

### Applications for CoPilot Automation

There are many practical applications for controlling CoPilot remotely using custom software. The following are a few possible applications:

- Integrate CoPilot functionality into a test program embedded in an Excel® spreadsheet, powered by VBA (Visual Basic for Applications).
- Acquire avionics data from a Ballard board controlled by a CoPilot virtual instrument (VI) from within LabVIEW®.
- Incorporate the power and speed of CoPilot automation tools into your existing C/C++ application.

# The CoPilot Object Model

### CoPilot Object Model Reference

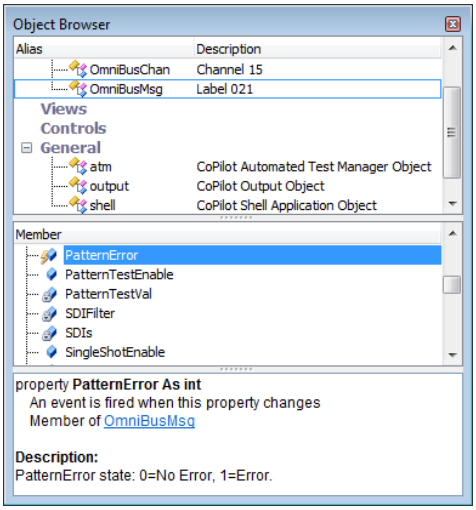
CoPilot exposes a subset of its functionality to OLE Automation, ATE, and scripting through an object model interface. The CoPilot Object Model documentation can be accessed from the Start Page (version 5),the **Help and Learning** tab group on the **Backstage** (version 6), or the Help menu.. Alternatively, you can use the object viewer of the programming environment you are using (such as the Object Browser for CoPilot ATE, the Object Browser window in Visual Basic, or in the object pane of the CoPilot Script View under the <globals> object).

The CoPilot Object Model provides useful information about:

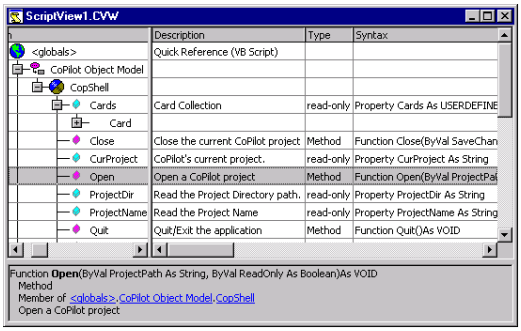
- The name and type of each CoPilot object
- The hierarchical path to each object
- The properties and methods of each object
- The description and calling syntax for each property and method



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*Sample screenshot of the ATE Object Browser*




*Sample of CoPilot Object Model displayed in a Script View window – for legacy applications*

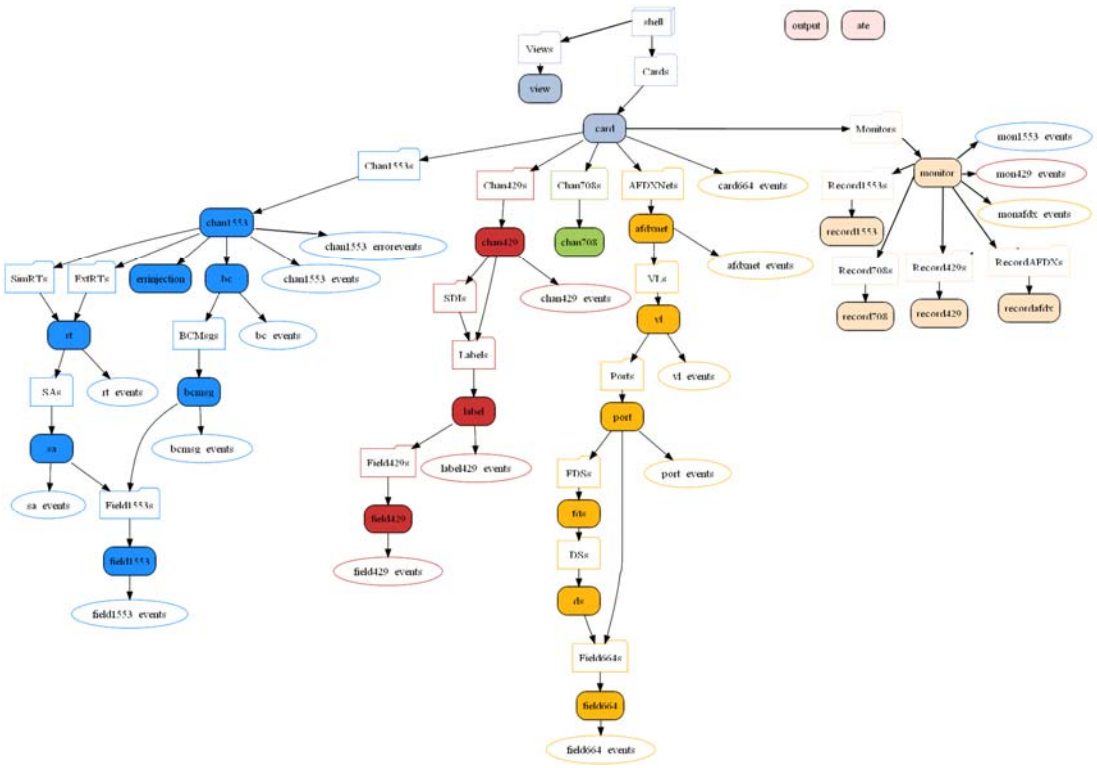
### CoPilot Object Model Overview

The figure below summarizes the available CoPilot objects and collections and is a visual overview of the hierarchical relationship between them. Objects are represented by a white oval and collections are represented by a gray box.

This figure shows a single instance of each *possible* object and collection. The objects and collections in actual projects will vary. For example, a CoPilot project with two 429 cards would have two 429 Card objects in the Cards collection and no objects in the Chan1553s or Chan708s collections.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*CoPilot Object Model diagram*

The object model chart illustrates the path to accessing the object you wish to call. For example, to pause the first monitor of the first card in a project **from an external application**, you would access the monitor object with the following syntax (based on the chart above) where 'Application' represents the CoPilot application:

```
Application.Cards.Item(0).Monitors.Item(0).Pause = TRUE
```

As you examine the CoPilot Object Model, you will notice that its structure resembles that of the CoPilot Hardware Explorer tree. If you are familiar with using CoPilot, the structure of the object model will be intuitive and easy to understand.

The ATE Command Prompt provides a mechanism to interact with the CoPilot Object Model from within CoPilot.

### CoPilot Type Library

In order to instantiate the CoPilot application object **from an external application** using code, you will need to reference or include the CoPilot type library (named "Ballard Technology CoPilot Shell"). The type library is part of the CoPilot executable (installed by default at C:\Program Files\Ballard\CoPilot\COPILOT.EXE).

After your program has access to the type library, you can then instantiate the CoPilot application object in your code as "BallardTechnology.CopilotShell.1."



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Below is a Visual Basic (not Visual Basic script) example:

```
Sub Button1_Click()
    'Create the CoPilot application object
    Dim copilotApp As Object
    Set copilotApp = CreateObject("BallardTechnology.CopilotShell.1")
    ...
```

Below is a C++ example:

```
#import "C:\...\CoPilot.exe" rename_namespace("COPILOT") raw_interfaces_only named_guids

void main(void)
{
    // Automation return value.
    HRESULT hr;

    // Initializes the COM library.
    hr = OleInitialize(NULL);

    // Create a Smart Pointer to the Application
    COPILOT::IBTPublicCopilotShellPtr pShell = NULL;

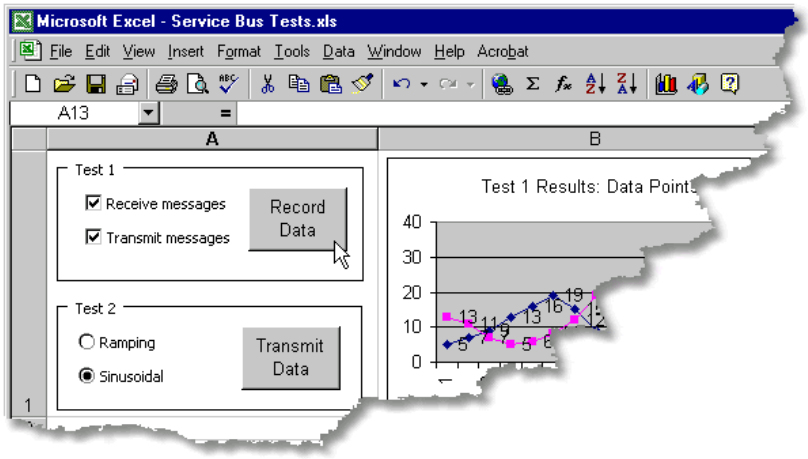
    // Creates a new running instance of the CoPilot Application Object.
    hr = pShell.CreateInstance(L"BallardTechnology.CopilotShell.1");

    ...
```

# CoPilot Automation Examples

### Embed CoPilot Functionality into Excel

VBA (Visual Basic for Applications) is a seamless part of Excel (as well as other Office applications). You could use VBA within an Excel spreadsheet to launch a series of tests using CoPilot or other applications. Each procedure could be executed in CoPilot with the push of a macro button, and the results could be saved and charted in Excel automatically.




In this way, a developer could set up a repeatable, controlled procedure with results that are automatically tracked in Excel. The tester or technician who runs the test does not need to understand CoPilot. CoPilot can integrate just as easily with other applications that support OLE Automation.

CoPilot ships with several OLE Automation examples in the directory "C:\UserName\My Documents\My CoPilot Projects\Samples\CoPilotAutomation." These samples show some simple applications written in C++, VBA for Excel, and VB6.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Acquire Avionics Data from CoPilot in LabVIEW

Through OLE Automation, LabVIEW users can access avionics data directly from Ballard cards by automating CoPilot. A CoPilot virtual instrument (VI) could be easily integrated into a larger test system and automatically provide the required data. For example, a CoPilot VI could be programmed to give a continuous reading of engine temperature and trigger an alarm if a defined threshold is exceeded.

### Incorporate CoPilot into a C/C++ Application

If you are working in C/C++, you don't have to use Ballard's API library to control a Ballard board, you can program CoPilot to do it. Because CoPilot has built-in tools to configure boards and channels, initialize data, set up schedules, etc., you can save countless lines of code and programming man-hours.

In addition, it is much quicker to change a configuration in CoPilot than in code. For example, if you want to add another message to your schedule, just open CoPilot and add the new message. Then, the next time you run your application, the function for running the schedule is the same, but the schedule now contains the new message.

### Created an Automated Test Framework in CoPilot

Through the use of CoPilot ATE (Automated Test Environment) and the Test Manager, customers can write entire script applications using the Python programming language to control a sequence of tests, utilizing user inputs and generating reports and output based on test parameters. A customized GUI can additionally be created for the test technician using one of the various Python GUI frameworks (e.g. WxWidgets, TKinter, or PyQT).

# Using Third-Party Automation Documents

### Enlarge CoPilot's Scope to Include Other Applications

Just as CoPilot can be controlled programmatically using OLE Automation, so CoPilot can control applications that are OLE Automation–compliant. Using CoPilot Professional with ATE (or the Script View's VBScript engine for legacy applications), CoPilot can seamlessly incorporate the resources of Microsoft Excel, Word, or other applications to enhance the analysis and simulation of avionics data. Some of the sample scripts included with CoPilot that illustrate this technique are described below.

### Excel Examples

Microsoft® Office® applications such as Word®, Access®, Excel®, etc. are OLE Automation–compliant. Therefore, you could use a script in CoPilot to access the power of these applications.

- **"Save Data to Excel" script**—This simple example illustrates how to open Excel invisibly in the background, add CoPilot data to a spreadsheet, and save and close Excel. You could add code to this script to also format and chart the data.
- **"Export 1553 Monitor to Excel" script**—This example uses Excel to simulate some of the functions of a CoPilot monitor. As messages are received in CoPilot, they are copied to labeled columns in Excel. Excel uses a formula to calculate the delta value between time-tags.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Internet Explorer Examples

Another application that can by automated by CoPilot scripting is Internet Explorer. You could automate Internet Explorer to send an e-mail or web page of avionics data as you receive it, giving remote colleagues instant access to your test results.

- **"Web Browse" script**—This simple example opens Internet Explorer and browses to a web page. You could easily modify this script to open other websites or documents.
- **"Web Page Output" script**—This example creates a web page with text and a hyperlink and saves it to the host computer hard drive. Then, Internet Explorer is launched and the new web page is loaded. You can expand this example to implement HTML code from within CoPilot.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Part 3: CoPilot Professional

## Part 3 Overview

CoPilot Professional provides all the features of the stardard CoPilot, plus the most comprehensive graphical display capability available anywhere. *Appendix A: CoPilot Standard and Professional Differences* on page *435* provides a description of the CoPilot Professional features and lists the differences between the CoPilot Professional and CoPilot Standard.

The features of CoPilot Professional are visible to CoPilot Standard users even if a CoPilot Professional license key is not installed. However, a CoPilot Professional license is required to run the Professional components with an active databus.

Part 3 of this document describes the CoPilot Professional Views and the ATE features. Other protocol related features, such as List Buffering and 429 Custom Schedules, are documented in the protocol sections. *Appendix E: Control View Library* on page *445* provides information about included ActiveX controls. *Appendix H: Legacy CoPilot Support* on page *487* provides information about the Legacy Visual Basic scripting.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Professional Displays and Controls

## Professional Display Windows Overview

### Display Window Introduction

CoPilot Professional displays can be created, opened, and saved via the **File** menu, the professional toolbar, or the New , Open , and Save  buttons in the CoPilot toolbar (version 5). In version 6, use the **Create** group of the **Home** ribbon tab. Once a display is created, it becomes a part of the project it was created in until it is physically deleted from the project. Deleting the underlying file will not remove it from the project, but will cause an error the next time the project is opened, indicating the file is missing.

Creating new displays is an important part of managing a useful project. Understanding the various purposes and uses for each of the available Display windows can be facilitated by reading their documentation in this manual. The table in the following subsection lists all available Display windows, as well as where their documentation can be accessed in this manual.

CoPilot 5 also hosts two toolbars specifically for creating new Display windows, known as the **CoPilot Professional Views Toolbar** and the **CoPilot Standard Views Toolbar**. These toolbars can be shown and hidden via the toolbar context menu (see the *Menus & Toolbars* section on page *47*).

### Summary of CoPilot Professional Display Windows

The table below summarizes the characteristics of the view windows in CoPilot. Each view is named and described. Additional information includes whether the view can be saved to file and the means of creating each view. (Each of the views summarized below is discussed in detail in its own section).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| Professional Displays | | | |
|---|---|---|---|
| **View Name** | **Description** | **File** | **Create From…** |
| Cockpit View | Displays and allows manipulation of aircraft controls and aircraft dynamics in a simulated cockpit format | Yes | New Hardware or View dialog or toolbar |
| Control View | Create virtual controls and simulated aircraft instruments | Yes | New Hardware or View dialog or toolbar |
| Map View | Simulate a moving map display (with TerraServer maps engine) | Yes | New Hardware or View dialog or toolbar |
| Strip View | View historical strip charts with statistical summaries | Yes | New Hardware or View or data field in tree or toolbar |
| Quick View | Launch a variety of virtual controls instantly from a data field | Yes | Data field in Hardware Explorer tree or toolbar |
| Script View | Control CoPilot programmati-cally with VBScript routines | Yes | New Hardware or View dialog or toolbar |
| Python Script Editor | Use Python scripts to control CoPilot automatically | Yes | New Hardware or View dialog or toolbar |
| Data Plot | View a chart of all data points of a monitored label or field | No | Monitor View window |

### Managing Display Windows

The display and placement of view windows can be controlled through buttons in the view's title bar, through the Project Explorer, and through the Window menu. Views that can be saved to file can be saved, opened, and named from the File menu.

- **Title bar buttons**—Use the button in the upper-right corner of the view to minimize, maximize, or close the window.

Note: Closing the window hides it from view but the display is still active and serviced during the simulation. To prevent a view from running, make in inactive or delete it through the Project Explorer. Also note that fileless views are deleted when they are closed.

- **Project Explorer commands**—Use the view's context menu in the Project Explorer to set the active/inactive state, show (or restore) the view in the display area, move the view to a different workspace, delete it from the project, or access summary information. Some views may have special context menu commands (such as the Python Script Editor).
- **Window menu**—Use the Window menu to select or arrange open views.
- **File menu**—(version 5) Use the File menu to save, open, close, and name views.
- **Ribbon**—(version 6) Ues the ribbon **Home** or **Backstage** tab to save, open, close, and name views. Use the **View** tab to arrange views and workspaces.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Workspace Tabs**—Views can be moved between various workspaces, which serve to logically organize views into groups that can be viewed separately.

### Other Display Window Concepts

- **Default Names**—All view components are given a default name by CoPilot when they are created.
- **Saving**—Most views can be saved to file and named by the user through the **File | Save As** command (version 5, see File column in table).
- **Fileless Views**—Some views do not save to file. These views can be recreated from their object in the Hardware Explorer.
- **Third Party Components**—In addition to the CoPilot views, you can open selected applications and third-party component windows within CoPilot. See the following section for more details.
- **Menu Merging**—(version 5) Some views have unique menus which are merged with the standard CoPilot menus when active. These menus offer special commands unique to that particular view. Please see the view's documentation for details.

### Diplay Windows Versus Display Panes

The CoPilot application hosts two types of windows, classified either as Display Windows (or views), or Display Panes (or panes). While they may look similar in appearance, they differ greatly in their functions.

Display Panes are not tied to a particular project, whereas Display Windows are. When a project is closed, all of its views are also closed with it, however the panes will remain. In fact, panes can exist even when a project isn't open. Finally, while views can strictly be hosted in the CoPilot Workspace area, panes can be dragged, docked, or floated anywhere on your computer's desktop, even on a secondary monitor.

# Professional Display Panes

CoPilot supports several "docking panes" (see *Docking Panes* section on page *43*) in Professional Mode. These panes are only available with professional-keyed hardware. Attempting to use these panes without professional keyed hardware will result in error messages.

Like mentioned in the CoPilot User Interface section of the manual, docking panes may be manipulated and moved around anywhere on your computer's desktop. They do not necessarily need to be "docked" into the CoPilot application window.

The following table lists the available Professional Display Panes and a brief description.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| Professional Display Panes | |
|---|---|
| **Pane Name** | **Description** |
| Output Pane | Displays textual output from CoPilot, Python, and user-generated custom output |
| Command Prompt | Provides a command line interface for performing automated tasks within CoPilot using its Automation model and Python syntax |
| Object Browser | Shows the various “aliases” available for manipulation via Python scripts and the command line. See ATE section for more details. |
| Watch Window | Display (and edit) variables and object properties for debugging and tracing. |

# Professional Displays

## Strip View

### Assign Fields through Drag and Drop

The Strip View is a display container (available with CoPilot Professional) for displaying fields in a strip chart format. Fields are assigned to this container through a drag and drop operation.

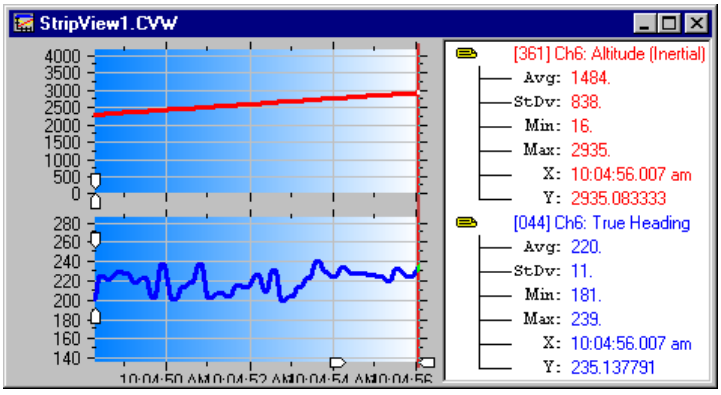
### Quick-Launch from Hardware Explorer Tree

A Strip View window can be automatically launched from a field context menu, thus bypassing the New Hardware or View dialog. A Strip View can be launched this way at any time, even during Simulation mode, and additional fields can be added to the display by dragging and dropping them from the Hardware Explorer.

To open a Strip View from a field in the Hardware Explorer:

1. Right click on the item and choose **View with Strip Chart** from the context menu

A Strip View window will open in the display area and if CoPilot is running, it will begin charting that field. You can then add other fields to the Strip View if you wish, through the drag and drop procedure.






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Strip View Options

Display attributes are changed through the Strip View context menu. Right click on the Strip View container to access the context menu below. Attribute selections can be made while the bus is active.




**Auto-Zoom** resizes the y-axis (vertical) to the min/max range of the data currently displayed in the x-axis (horizontal)

**Analysis** provides an average and standard deviation for each displayed field

**Multiple Tracks** assigns fields to separate charts (when multiple fields are present)

**Time Display** toggles the display and x-axis between elapsed seconds and time (hours, minutes, and seconds)

**Line Thickness** provides a selection of line thickness

**Area Fill** extends the line color to the x-axis

**Display Captions** turns field titles on or off

**Display Grid** turns grid lines on or off

**Display Color** toggles line display between black and white and color

**Display Operating Limits** adds shading to the display to demarcate the operating limits defined in the data interpreter for that field

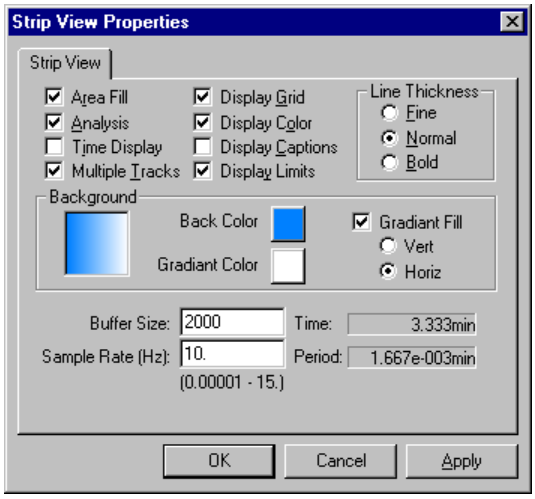
**Delete Message** allows user to delete specific fields

**Clear All Messages** purges all field assignments

**Properties** provide access to Strip View properties

## *Strip View Properties*

Strip View Properties are accessed through the Strip View context menu. Most of the options presented through the context menu are repeated in properties.




### Buffer Size

Data for all assigned fields is placed in the buffer as it received. The sizing of the time frame depends on the buffer size and sample rate. At 10 samples per



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

second and 2000 word buffer, the x-axis of the strip view window displays 200 seconds or 3.333 minutes worth of data. The maximum buffer size is 50,000.

#### Sample Rate

Data can be sampled at up to a maximum of 15 times per seconds, but with contention between tasks, it is unlikely that the maximum sample rate can be sustained. Alternatively, a lower sample rate could be used (e.g., .01 samples/second or one sample every 100 seconds) to display data over a long time period.

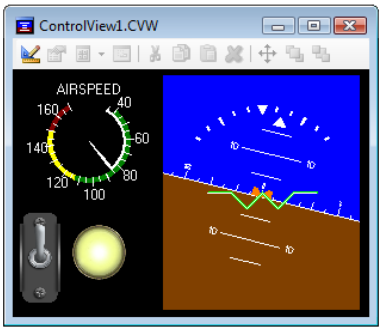
#### Background Color

The background can be a single color or a gradient. A gradient background can be vertically or horizontally oriented. To select a color, click on the Back Color or Gradient Color button to open the Color dialog.

## Control View

### *Overview*

The Control View window is a form/panel used for hosting virtual instruments, gauges, dials, widgets and other types of controls. These controls use the ActiveX architecture which also allows users to add their own controls. Each of the ActiveX controls can be adapted and customized through the control's properties.




#### Controls and Links

The controls added to a Control View are connected to other CoPilot objects via links. Controls can be used to modify data (control linked as a source) or display data (control linked as a sink). Values that can be edited, by default, assume the control will be used to source (modify) the data. When the control sources the data, the value changed on the control modifies the engineering value connected to it; thus the control sources the data to the engineering value. Conversely, values that cannot be edited (for example a receive ARINC 429 message) use controls to display data from the engineering value connected as a sink. See *Linking Controls* on page *380* for a more information.

### *Design and Operational Modes*

Control View windows support the Design mode for designing the view, adding, moving and modifying controls and the Operational mode when a CoPilot project is running. When in Design mode, links to the controls are not enabled. Running CoPilot automatically sets the Control view to the Operational mode and services the links to the controls.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Customizing the View

The Control View window is customizable to hold multiple controls/displays. The view is customized by adding various controls and by changing properties of the controls and the view in the Design mode.

When the view is set to Design Mode, properties are customized using the Control Property Browser. When no control is selected this property browser shows all of the available view window properties. Use the property browser to change the background, grid settings, and more as described in the following *Customizing a Control* section on page *380*.

A selected control shows grab handles for moving and resizing the control. The green grab handle in the upper left corner is used to move the control without changing the control size. The blue grab handles are used to stretch and resize the control.

Multiple controls can be selected by holding down the <CTRL> button while clicking multiple controls or by using the mouse to drag a box around the controls. Once multiple controls are selected, they can be moved, aligned and resized together using the grab handles.

## Adding Controls to the View

There are several ways to add controls to a Control View. A simple drag and drop of en engineering unit value to the control view adds a control. When dropping the engineering unit object from the Hardware Explorer onto a Control View, the view attempts to add and connect to the control defined in the database. Users may save their own definitions to the control database. In the absence of particular saved controls, the Control Selection Gallery provides a simple process of selecting a control and automatically creating a link. See *The Control Selection Gallery* on page *377* or *Control Database* on page *379* for more detailed discussions.

### Drag Drop

An easy way to add a control to the view is to drop an engineering unit object from the Hardware Explorer onto a Control View window. If a control has been defined in the controls database for the specific engineering unit, then that stored control will be added to the view. Otherwise, the 'Control Selection Gallery' will assist with picking an appropriate control. Once the control is added the view will attempt to connect the new control back to the engineering unit by creating a link.

The default behavior is for editable data values (i.e. ARINC 429 transmit, MIL-STD-1553 BC receive messages, and MIL-STD-1553 SA transmit messages) to create controls as sources and non-editable values create sink controls. (To change the source/sink assignment, hold the (**Shift**) key as you drag and drop.)

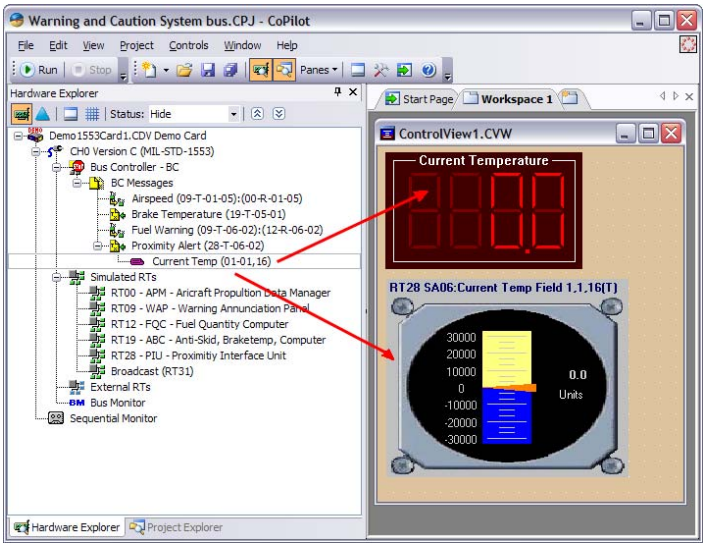
---

Note: Hold down the (**Shift**) key while dropping a control attempts to select the control stored in the database for the opposite (source vs. sink) direction of the link. If there is no control in the database, then the Control Selection Gallery is shown.

---



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




As illustrated in the figure above, the same Current Temp field has both a control saved for sourcing the data to the field, the slider control, and the other for sinking the data from the field to the LED control. A normal drag-drop operation would add one control and drag-drop of the field with the (**Shift**) key held down would give the other. Again, if the controls have not already been saved in the database for the object (field/message), then Control Selection Gallery is shown. In this case, the control acting as a data sink with the LED is the default. See *Linking Controls and Graphical Objects* for detailed information about manually creating and modifying links.

---

Tip: To remove a control from the Control View(Control View window must be in Design Mode), right click on the control and choose **Delete** from the context menu.

---

### Control Palette

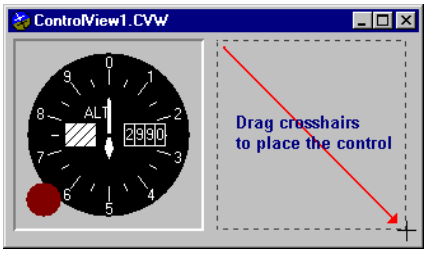
The control pallet is tool window used to add ActiveX controls to a control view. The palette contains a buttons liked to a library of controls. Adding a control from Control Palette adds the unmodified version of the control without any of the additional settings made in the Control Selection Gallery. The Control Selection Gallery is described in the following section.




To add a control from the palette, first click on the control button of your choice. Then click and drag a box in the Control View window where you want to insert the selected control as shown in the figure below.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

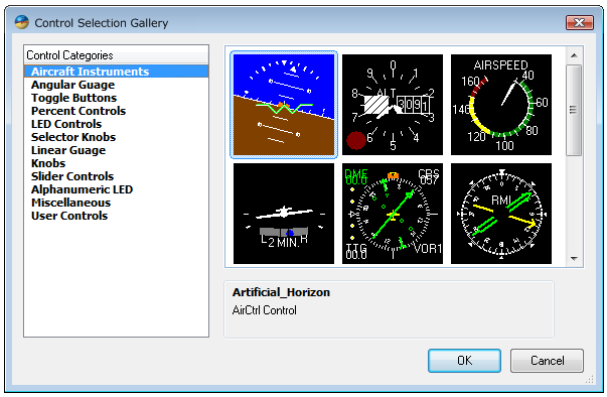



The properties of the control may be modified after the control is inserted. To add additional controls to the control palette, see the *Using third-party ActiveX controls* section on page *381*.

## The Control Selection Gallery

### Overview

CoPilot comes with a pre-populated control database containing categories of preconfigured controls. The control selection gallery is used to quickly add one of these controls from an organized control picker. The Control Selection Gallery is launched when the 'Add Control to Form' command is selected from the context menu.

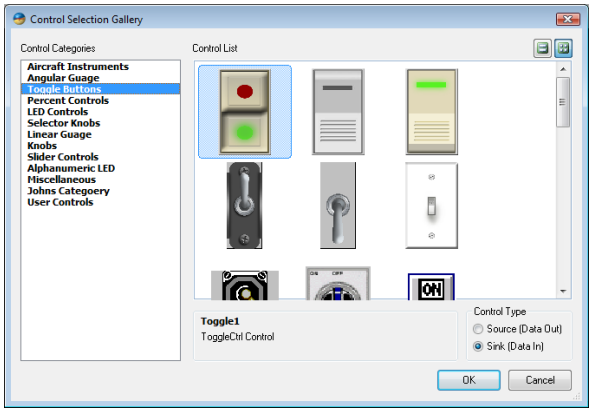



*The Control Selection Gallery from the 'Add Control to Form' command*

Adding a control the control view cannot automatically determine what object to connect to without providing any additional information. However, when a message or field object is dropped on a Control View and the Control Section Gallery is shown, a link will be created. The image below shows that the control type can be set to act as either a source (the control sources the data to allow changing values) or a sink (control used for displaying data values).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*The Control Selection Gallery adding a control using drag-drop*

### The Suggested Control Category

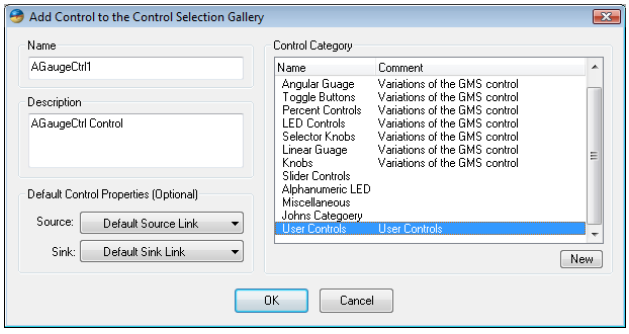
When dropping an engineering unit object onto the view and no prior saved control is found, then the Control Selection Gallery dialog is shown. If possible, CoPilot will offer a 'Suggested Control' category will one or more suggested controls. The 'Suggested Control' category is added to the top of the list. Press the **OK** button to accept the suggested control; otherwise, select a control from another category.

Note: The suggested control category uses the interpreter type to make the default selection; not all interpreter types suggest a default control.

### Saving Definitions to the Control Selection Gallery

Controls in a Controls View can be added to the Control Selection Gallery. These controls are saved in the control database and are available to future projects.

When saving a control to the gallery, a description, sorting category, and default source/sink properties can all be specified. The default source/sink properties automatically create a link between that property and the field or message object's value. The following image shows a control being added to the Control Selection Gallery.




*Adding a control to the Control Selection Gallery*

To save a control to the Control Selection Gallery:

1. Select **Add to Control Gallery** from the controls context menu



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. Set the name, description, default links (optional), and category (we suggest adding to the **User Controls** category or creating a new category)

3. Click **OK** to complete

## *Control Database*

### Overview

The control database stores information about controls in the Control Selection Gallery. The protocol databases, such as the ARINC 429 and MIL-STD-1553 databases, contain the definitions for the engineering unit interpreters as well as the specific controls saved with the engineering unit messages and fields.

When dropping an engineering unit object onto a control window the view will attempt to add and connect to a control from the list of controls in the protocol database. If a control is not found in the database for that object with the matching source/sink value, then the Control Selection Gallery is opened and attempts to find a default control based on the engineering unit interpreter type. If found, the control is added as the 'Suggested Control.' Users may save additional control definitions by saving the linked controls using the 'Save to Database' command or the controls may be added to the Control Selection Gallery as described in the previous section.

### Connecting Engineering Units to Controls

When a control is linked to a CoPilot engineering unit object (see *Linking Controls and Graphical Objects* on page *349*) that association can be saved to a database to create a default control. After a control has been saved to the database as part of the data definition, you can immediately drag and drop that field into any Control View to produce the saved control. Since it is part of the database, it is available to all CoPilot projects.

### Duplicating Control Database Definitions

Some field names appear in the database many times and are defined by the same interpreter properties. Airspeed, longitude, latitude, and various temperature settings are transmitted by many systems. Once a "custom" control has been established and linked to a field, it can be propagated to other fields in the database easily.

To assign and save a custom control to another field:

1. Start by adding a control from the Control Selection Gallery or add a control that you have saved in the database by dragging the field it was saved with into a Control View window

2. Click **Project | Links Status** and select the field/link combination of interest

3. Click **New Link** to create a link for a control added from the Control Selection Gallery or click **Modify Link** to reassign the control to the EngrValue property of another field.

4. Click **OK** to complete the linking

5. With the Control View window in Design mode, right click the control and select **Save to Database**

With a little practice, control, property, and field assignments can be performed quickly.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

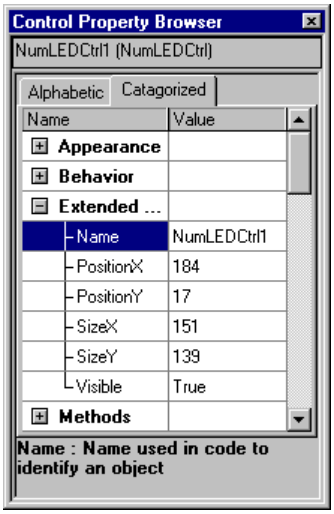
## Linking Controls

Links are used to connect one or more properties of a control to the properties of other controls or objects (such as messages, channels, and fields).See *Linking Controls and Graphical Objects* on page *349*.

## Customizing a Control

### Control Property Browser

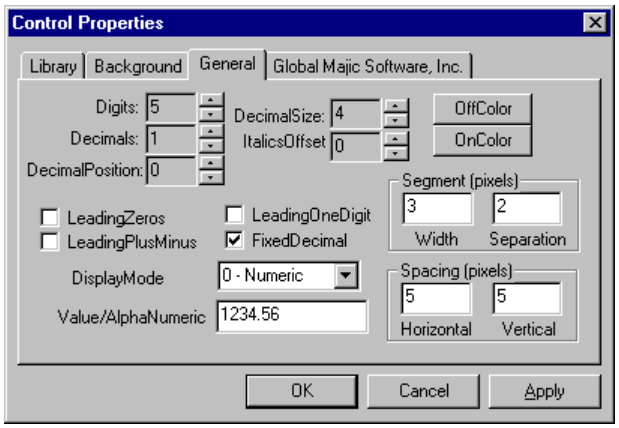
When a control is selected in Design Mode all of the available settings and properties are loaded into the Control Property Browser. The browser gives you access to view and modify all available settings. Certain properties like name and location are only available through the Control Property Browser.




*The GMS Alphanumeric LED control in the Control Property Browser*

### Control Properties

If the ActiveX control provides a properties editing window the Properties option will be available in a control context menu. Selecting this Properties menu item will open the control properties window. This window will contain one or more tabbed panels provided by the author of the control. In most circumstances this is the preferred way of customizing a control.




*The property dialog above is associated with the GMS Alphanumeric LED control*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

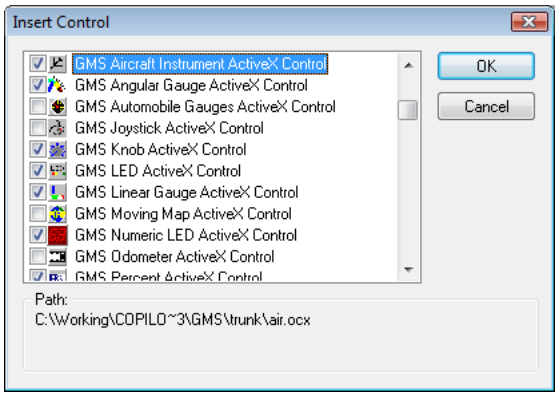
## *Using third-party ActiveX controls*

Third-party ActiveX controls can be added to the Control Palette and assigned to a Control View window (see Adding Controls to the View ).

Before importing a third-party control, the control must be properly installed on the host computer and the Control View window must be in Design Mode. The Control Palette does not need to be visible.

To add an ActiveX control to the Control Palette:

1. Right click on the Control View background (or the Control Palette background) and choose **Add Control to Palette** to open the Insert Control dialog
2. Scroll through the ActiveX controls available in your system and click on a control to select it
3. Click **OK** to close the Insert Control dialog and add the selected control to the palette




After a control has been successfully imported, a button will be added to the Control Palette for that control (see figure below). The imported control will appear in the Control Palette until removed by the user.




Tip: To remove an imported control from the palette, right click on the control button and choose the **Remove** command.

## *Mutually Exclusive Controls*

If several controls, each of which have a value of type OLE_OPTEXCLUSIVE (e.g., Radio Buttons), are placed within a view, only one member of the group will be capable of being set to True, all others will be set to False. Controls capable of inclusion in this group will have a True/False property named “Group



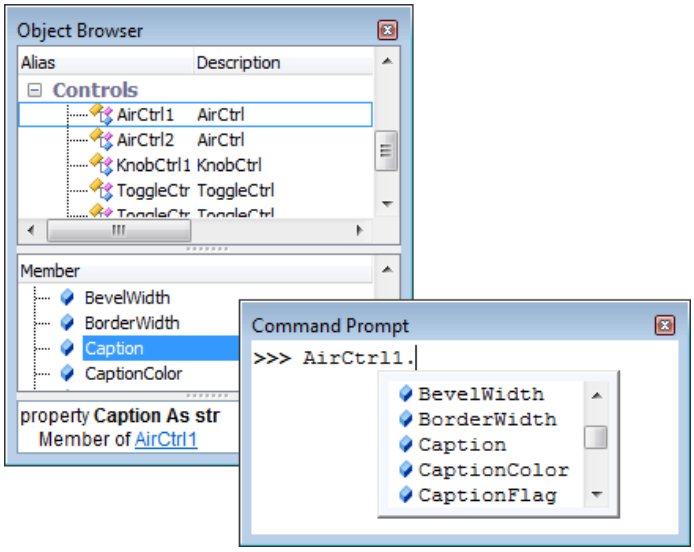
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Item" available in the property browser. Only controls with this property set to True will be members of the mutually exclusive group.

## *Controls and Scripting*

### ATE : Shell scripting

All controls are automatically aliased (see Automated Test Environment (ATE)) by the CoPilot application. Control properties and methods can be accessed by name via the Command Prompt, Watch Window, and in Python scripts.




### Controls and the VB Script View

The Visual Basic Script View has been deprecated. The use of the VB Script View is still supported for legacy compatibility, but it is suggested new applications use ATE which uses Python. Advanced functionality includes additional variables, a debugger, watch window, improved editor, runtime control, a command prompt, auto-completion, intellisense and more.

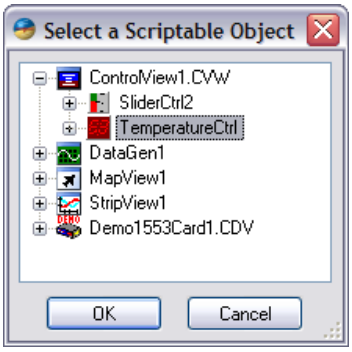
Like any other CoPilot object, to read or modify its properties in Script View, the ActiveX control must be added to the Object Browser and referenced in the script.

To include a control in a script:

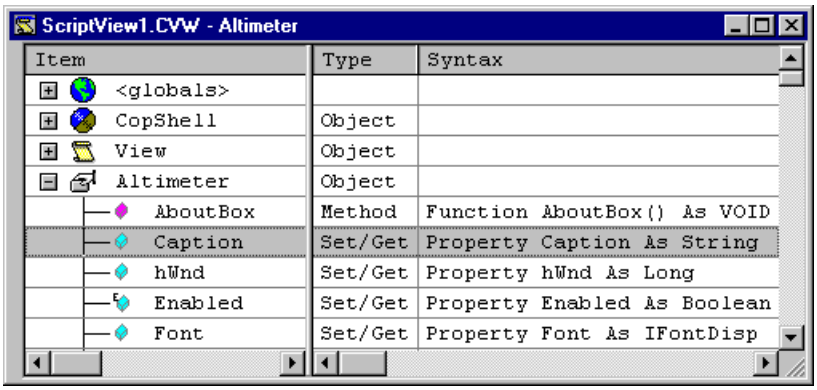
1. Right click in the Script View's Object Browser (right-hand pane of Script View window) and choose **Insert New Object** from the context menu
2. Browse to and select a control in the Pick a Scriptable Object dialog (see figure below) and click **OK**
3. Add or modify your code to handle the events of this object



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




Clicking the show/hide icon in the Object Browser will expand the display of the control properties and methods (see figure below).




### Scriptable Control Properties and Methods

***property**: A characteristic or attribute of an object (e.g., Visible = True).*

***method**: A procedure, function, or routine associated with an object (e.g., ShowPropertyPage).*

ActiveX controls are defined by properties and methods. The property browser in both Control View and Script View retrieves and displays the "exposed" or "public" properties and methods native to each control. All of these native attributes are visible and accessible in the Script View window, with the exception of the Extended properties that assigned to a control by the Control View window (such as position and size). The object browser in Script View also adds information such as type, calling syntax, and a brief description for each attribute. For more information on controlling CoPilot objects (such as controls) through scripting, see the *Appendix H: Legacy CoPilot Support* section on page *487*.

### Example Script

The script below is a slight variation of the MonitorControl.txt script described in *Appendix F: Script Examples* on page *467*. Three lines of code have been added to manipulate an LED control in Control View.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
'Globals
Const UPPERLIMIT = 5000

Sub ScriptStart
    'Make sure the monitor is not set to pause
    Mon1.Pause = False
    LEDCtrl1.Value = 1
End Sub

Sub Field1_ValueEngr
    If Field1.ValueEngr <= UPPERLIMIT Then
        Mon1.Pause = False
        LEDCtrl1.Value = 1
    Else
        Mon1.Pause = True
        LEDCtrl1.Value = 0
    End If
End Sub

Sub ScriptStop
End Sub
```

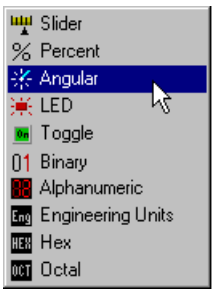
## Quick View

### Overview

Quick View is a display for a special library of controls that launch directly from a message or field in the Hardware Explorer tree. Users can select the control type while CoPilot is running and individual views can be used to set the value of transmit labels and fields and display the values of labels and fields being received on the databus. A field can be viewed in a variety of ways through different Quick View controls.

To launch a Quick control:

1. Right click on a label or field in the Hardware Explorer and point to Quick Controls to expand the submenu
2. Click on one of the Quick controls to create it (see figure)




The specified control will open in a Quick View window in the display pane. The status bar identifies the message or field and whether the control is read only. If the control is not read only, you can use your mouse to manipulate the control and modify the data.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




### Operating Limits

For messages and fields that have Operating Limits defined (see Engineering Units section), the Quick Controls will display a different color theme when the data is out of the operating range. The out-of-range color theme is usually red to indicate that an error has occurred with the data value, allowing an operator to quickly see range problems.

### Error Display

The status bar of Quick View controls display status information such as ‘timeout’. The image above shows a message with a timeout error condition. An operating limit error condition is described above.

## *Quick View Library of Controls*

The library of Quick controls (illustrated in the table below) provides a variety of ways to view data graphically. Most of these can be used for any field type but others, like Angular, have a more specialized use.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| Quick View Controls | | | |
|---|---|---|---|
| | **Description** | **Edit** | **To modify data** |
| **Slider** | Indicates the current value of the message/field within the defined range. | Rx = Read Only | Drag pointer with mouse |
| **Percent** | Expresses the current value of the message/field as a percentage of the full range. | Rx = Read Only | Drag ring with mouse |
| **Angular** | Indicates the current value of the message/field within a range of –180 to +180. | Rx = Read Only | Drag needle with mouse |
| **LED** | Displays OFF (unlit) when the value of the message/field is zero or below, and ON (lit) when the value is above zero. | Read Only | n/a |
| **Toggle** | Displays ON (green) when the value is not zero, and OFF (red) when the value is zero. | Rx = Read Only | Single click with mouse |
| **Binary** | Displays the raw data value of the message/field in a binary radix. | Rx = Read Only | Double click or 1/0 keys |
| **Alphanumeric** | Displays the interpreted value of the message/field. The number of displayed digits can auto-size or be explicitly set. | Read Only | n/a |
| **Engineering Units** | Displays the interpreted value of the message/field with the units of measurement. | Read Only | n/a |
| **Hexadecimal** | Displays the raw data value of the message/field in a hexadecimal radix. | Read Only | n/a |
| **Octal** | Displays the raw data value of the message/field in an octal radix. | Read Only | n/a |

#### Read Only

All of the Quick controls referenced above can be used to display current data values A few of the controls can be used to edit or display information. When a Quick View control for a transmit field or label is launched from the Hardware Explorer, CoPilot always activates the control without setting the read only flag if possible. If the selected control is “Read Only,” the control will display the value of the transmit field but not allow data to be set through the control. Receive fields are always represented by Quick View displays in read only mode.

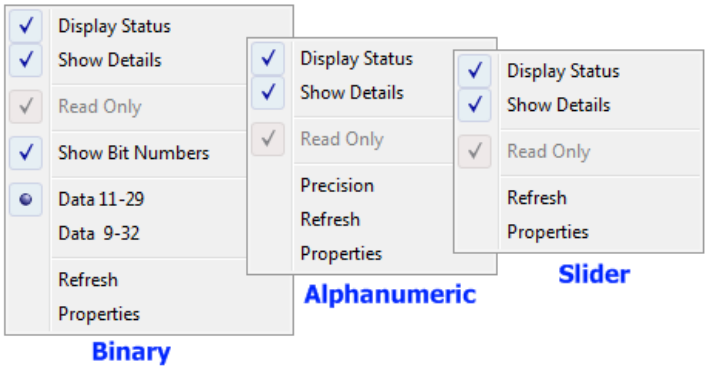
## *Configuring Quick View Controls*

#### Configure through Menu Options

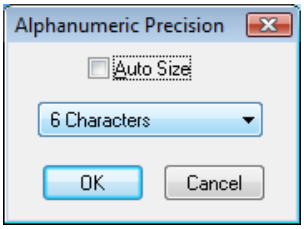
To customize a Quick View control, right click on the control in the display pane to access its context menu. The available menu options vary by control (see figure below). All configuration options can be modified while the simulation is running.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




- **Display Status** toggles the status bar, which displays the name of the linked label or field and the source/sink state of the control
- **Show Details** toggles the details display, which shows the engineering unit string and time-tag value on the view
- **Read Only** enables value editing by mouse control (only available when a Slider, Percent, Angular, Toggle, or Binary control is linked to a transmit message/field)
- **Show Bit Numbers** toggles the display of bit number above each bit character (Binary control only)
- **Data 11–29/9–32** limits the display to the range of bits selected (Binary, Hex, and Octal controls only)
- **Precision** opens the precision dialog where you can specify the number of characters or enable auto-sizing (Alphanumeric control only)
- **Refresh** forces the control to redraw
- **Properties** opens the properties dialog for this control (see figure below)



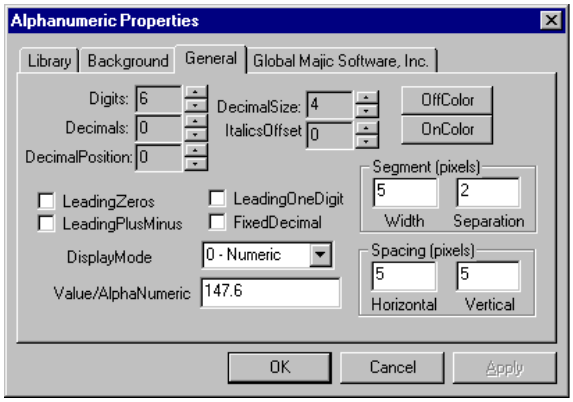

Tip: Drag and drop the linked field (identified in status bar) from the Hardware Explorer onto the Quick control to update the display.

### Customize through Properties

Control properties provide a number of configuration options. Many controls host a multi-tabbed properties page with numerous configuration options (right click on control and choose Properties to access). A listing of predefined controls is often available on the Library tab.



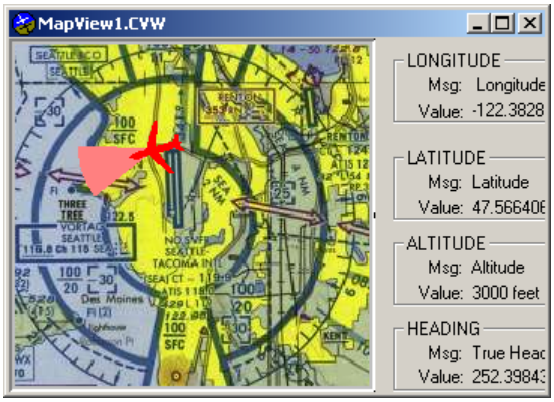
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




# Map View

### View Aircraft Position

Map View is a view for displaying aircraft position over a moving map image. Incoming data is linked to properties in the status pane and displayed using an airplane (or other image) over the map. Objects, waypoints, and grids may be added and customized. The Map View display is configured through the context menu and properties page.

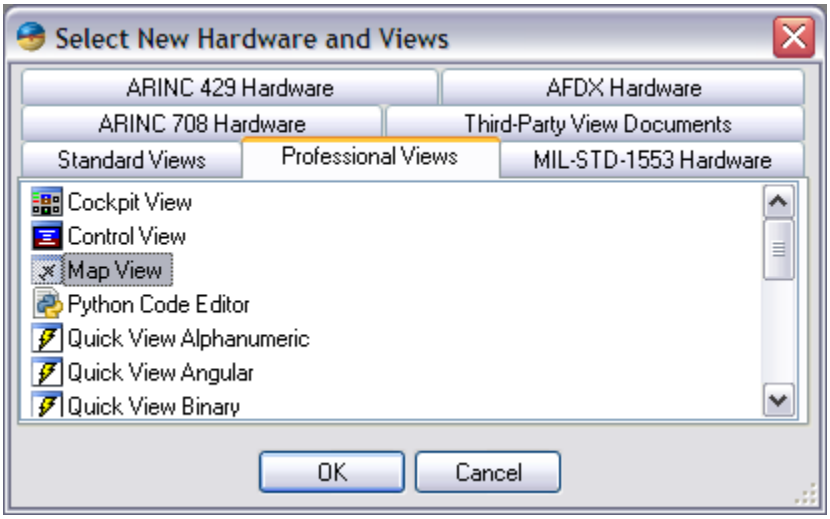



### Accessing a Map View Window

Map View windows are selected through the **File | New Hardware or View** command or the **New Hardware or View** button. Map displays are listed in the New Hardware or View dialog under the Professional Views tab.



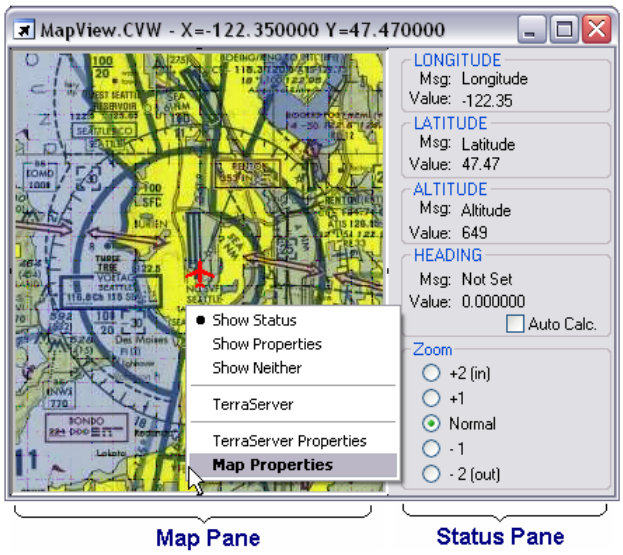
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




## *Map View Window*

### Map and Status Pane

The Map View window consists of a map pane and a status pane as shown in the following figure. The map display may be floating or stationary, and objects may be placed at any location. The status pane displays the current links and data values for longitude, latitude, altitude, and heading. The relative size of each pane may be adjusted by dragging the edge with the left mouse button.




Map resolution may be controlled through option buttons in the Zoom frame. Alternately, users may zoom in or out with the mouse. Mouse-controlled pan and zoom are enabled by default, but may be disabled through the Map Properties window.

To zoom or pan with the mouse:

- To zoom in or out, click and drag the right mouse button
- To pan, click and drag the left mouse button



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### Drag and Drop Assignment

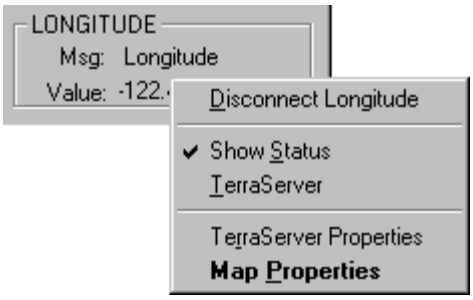
Positional data is obtained from data fields in the Hardware Explorer. Fields may be linked to a property in the status pane by dragging them from the Hardware Explorer. The status pane displays changing data values while the bus is running.

To link data to a Map View window:

1. Select the data field from the Hardware Explorer
2. Drag and drop it onto the property of your choice
3. Update the links at any time by re-dragging them from the tree

## *Map View Context Menu*

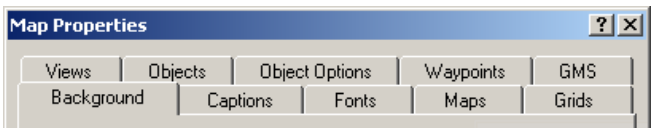
To access the Map View context menu, right click anywhere on the Map View window. If you click in the map pane, the context menu will have four choices: Show Status, TerraServer, TerraServer Properties, and Map Properties. If you click on one of the four data links in the status pane, an option to disconnect that link will appear at the top of the context menu.




The Show Status option is selected by default. To hide the status pane, choose Show Status to clear the check mark (toggles the status pane on and off). To enable TerraServer maps in the Map pane, select TerraServer. To configure the TerraServer display, choose TerraServer Properties to open a properties page. Choosing Map Properties opens the Map View properties dialog.

## *Map View Properties*

Map images, objects, grid patterns, center point, and zoom and drag options are established through the Map Properties window. To access the properties dialog, right click anywhere in the Map View window and choose Map Properties from the context menu. The Map View properties page has ten tabs. To view your current choices without closing the properties page window, click the Apply button.




#### Initializing and Changing Properties

At a minimum, users must specify the location and boundary of maps that form the background for a moving airplane or other objects. Specification of objects, grids and other refinements is optional. Map properties can be changed prior to or while a CoPilot simulation is running.

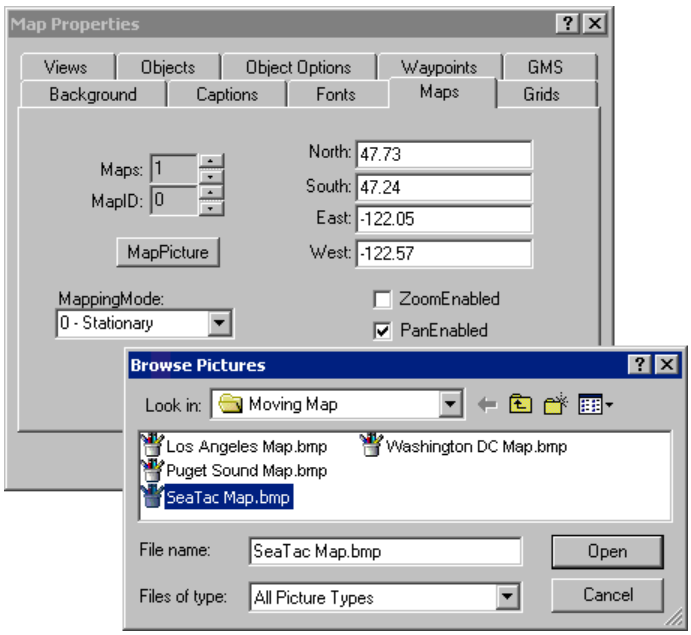


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## *Map View Maps*

### Single Map

Maps are defined through the Maps tab of the Map properties dialog (unless you are using TerraServer to generate your map display). One or more graphics files define the map field. In the figure below, the map display will be defined through a single bitmap file (SeaTac Map.bmp). The north, south, east, and west boundaries define the latitude and longitude of this bitmap.




### Multiple Maps

Graphics files can be quite large. That may lead to a tradeoff between putting all map content into one or several files. When more than one file is required, specify the number of map files (in the Maps text box), then identify and set the boundaries of each file.

To establish a unified map from multiple files:

1. Click the right mouse button over the Map View to access the context menu.
2. Select **Map Properties** and click the **Maps** tab
3. Enter the number of maps in the Maps textbox (1 is the default value)
4. With MapID set to 0, click the **MapPicture** button and select the file through the Browse Pictures window
5. Type the longitude and latitude boundaries of map 0 in the **North**, **South**, **East**, and **West** text boxes
6. Repeat steps 4 and 5, incrementing the MapID, until all maps have been configured

When complete, these multiple images are integrated into a single map based on their coordinates.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Overlapping Maps

Maps may overlap or may be separate, but the coordinate system must be consistent. Overlapping maps are layered by MapID number. If, for example, maps 0 and 1 overlap, map 1 will be drawn on top of map 0.

### Display Mode

If the Mapping Mode is set to stationary, the object (plane) moves relative to the map. If the mode is set to Floating, then the map moves relative to an object (determined through the ViewObjectID property in the View tab).

Note: Because Stationary mode takes less processing power, this option may be preferred, depending on the host computer and the size of map files.

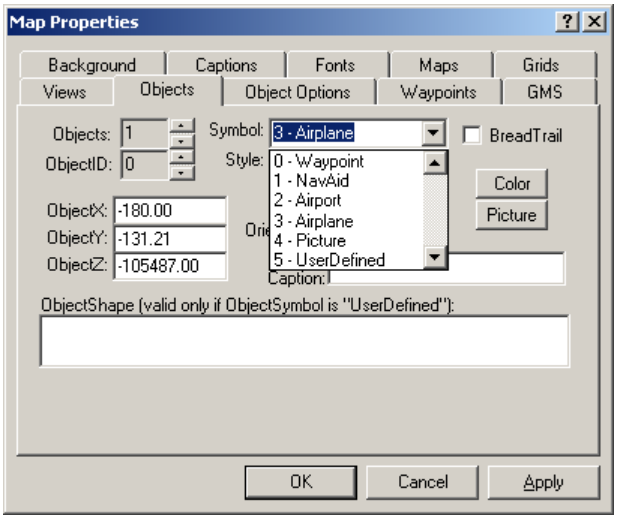
In Stationary mode, the visible display in Map View is established by panning coordinate positions set through the Views tab. The easiest way to locate the current object and map image is to select Floating mode (which brings the object and map to the center of the map pane), and then reselect Stationary mode.

### Zoom and Pan

If ZoomEnabled is selected, the display zooms relative to the current view's center by dragging the mouse (right mouse button depressed) over the map pane. Dragging up zooms out; dragging down zooms in. When PanEnabled is selected, the display is panned by dragging the mouse over the display with the left mouse button depressed. The zoom and pan options only apply in Stationary mode.

## *Map View Objects*

Incoming data is displayed through an airplane on a stationary or moving map. Airplane size is controlled through the scale textbox and color is selected through the palette linked to the color button. When Bread Trail is selected, a line will be drawn on the map tracing the path of flight. Objects may be configured further in the Object Options tab.



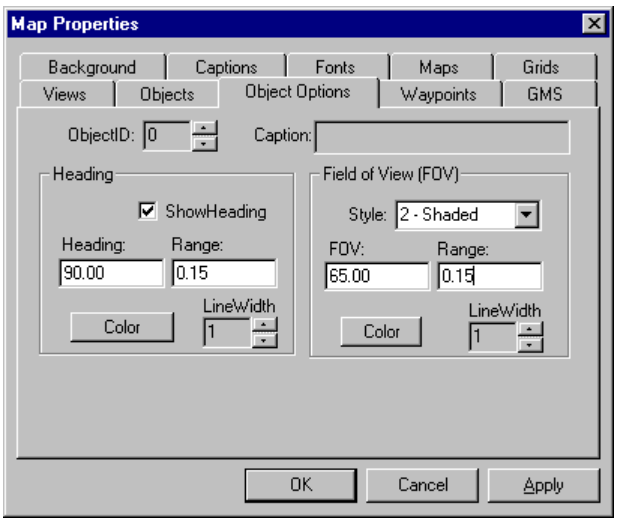

Initial longitude and latitude can be set through the ObjectX and ObjectY text boxes. This is replaced with incoming longitude and latitude values during a simulation.



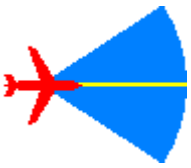
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Map View Object Options

The Map View object display can be enhanced through Object Options. The Heading indicator is a line marking the current heading of the aircraft with a choice of range, color, and line width. If data has been linked to the Heading property of the Map View window, the heading angle of the aircraft and heading indicator will be set by that value when the bus is running.



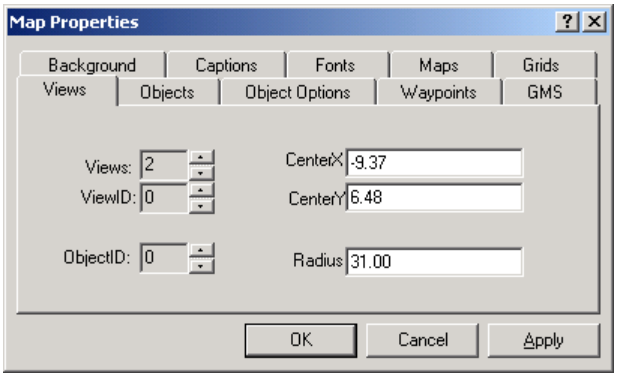

The "FOV" option shows the "field of view" in degrees. The angle, range, color, and line width can be adjusted. This field of view can be toggled on or off, marked by boundary lines, or shaded.



The image on the left illustrates an airplane object enhanced by the heading and field of view options selected in the above figure.

## Map View Views

In Map View, a "view" (configured in the Views tab) is the "line of sight" of an object. Each view is tied to an object (identified by ObjectID). The Radius defines how much of the map is shown from the center.




When Map View 'MappingMode' is set to 'floating', the X and Y coordinates of the view center are synchronized to match the coordinates of the object (i.e. airplane).



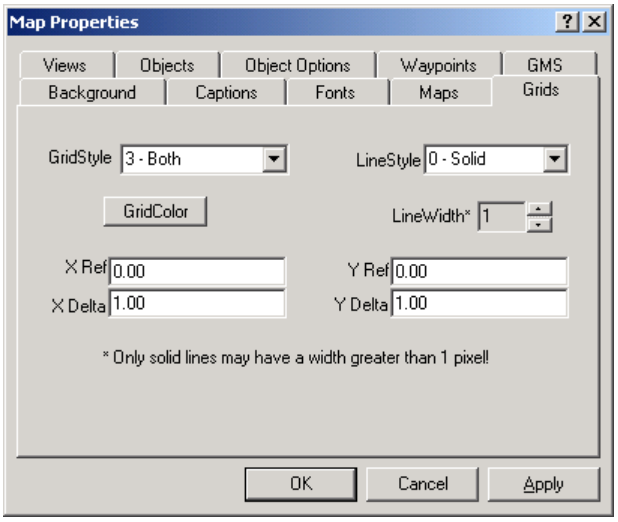
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Display Zoom

The zoom settings of the map view status are used to control the display radius. The various zoom settings will either zoom in or out of the maps. In addition to the supplied zoom values (i.e. '+2', 'Normal', '-1'), users can open the Map Properties and manually set the View's radius (the units of the longitude/latitude values determine the coordinate system). When using TerraServer to supply the photographic or topographical maps, CoPilot attempts to import maps to fill the Map View display area. The number of maps required depends on the radius of the map display and the map scale. Selecting the Auto-scale option in TerraServer properties usually produces the best resolution.

## *Map View Grids*

Rectangular grids may be superimposed over the map image through the options in the Grids tab. Grid style choices include horizontal (latitude), vertical (longitude), both, or none. The XRef and YRef properties set the reference position on the map for longitude and latitude. To work properly, the supplied map files must be level, with respect to latitude, and the boundaries accurate.



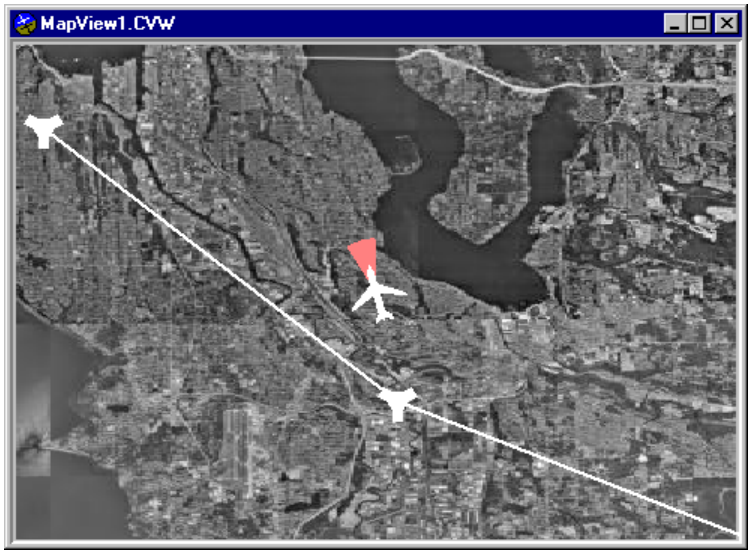

The XDelta and YDelta values set the distance in degrees between latitudinal and longitudinal grid lines. Values of less than 1 are invalid.

## *Map View Waypoints*

The map and flight path can be marked by five types of waypoints: airplane, airport, navigational aid, waypoint, or user-supplied picture. In the example below, two waypoints are visible on a TerraServer map, connected by a solid red line.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

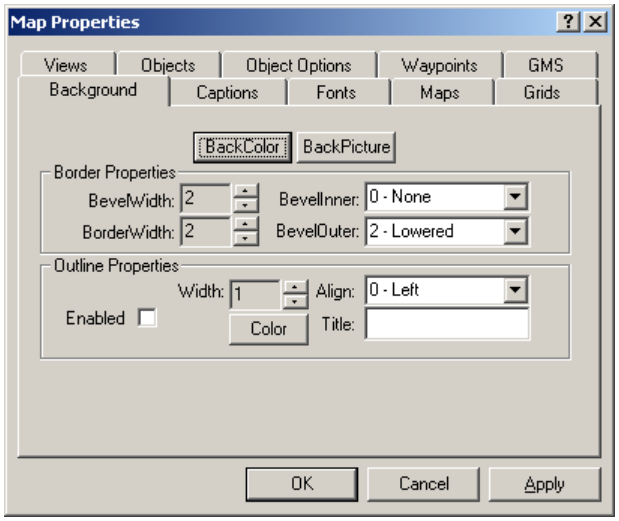



The symbol type, style, color, location, and scale may be set for each waypoint.

## *Other Map View Properties*

### Background

The BackColor button determines the background color. It is ignored if BackPicture is set. The BackPicture option allows a graphic to be displayed in the background. To clear the picture, select a background color with the BackPicture button.




The border and outline properties allow the user to create a frame around the display pane and label it.

### Captions

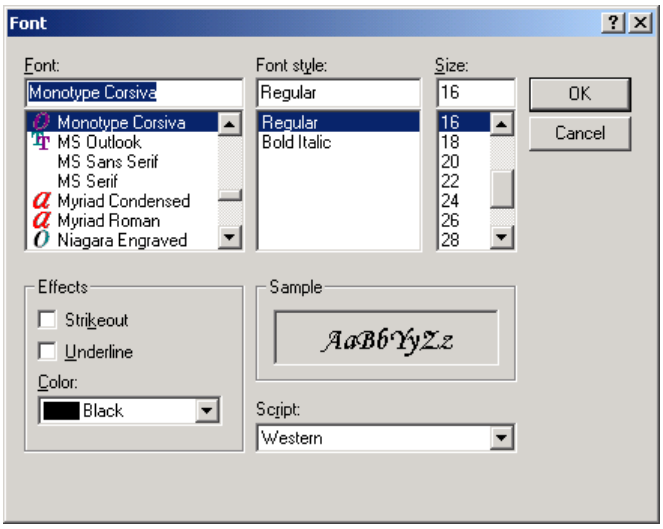
The Captions tab allows you to create text on the surface of your map. Use captions to give names to areas on the map, label objects, etc.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Fonts

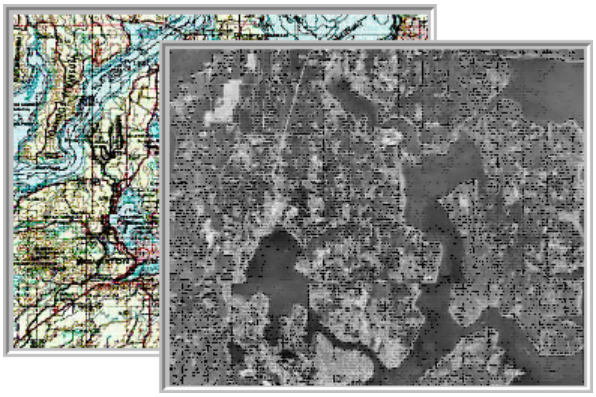
The Fonts tab allows the user to define a number of fonts for use within the Map View window. To define a set of fonts, first enter the number of fonts you wish to configure in the Fonts: text box, or click the up button. Next, select the font you wish to configure by entering a value in FontID.




Next, click the Set Font button to open the font dialog box. Choose the font, style, and size from the list boxes. You may also select effect, color, and character set. The sample box previews your current choices.

## *TerraServer Maps*

TerraServer® maps may be used instead of graphics images to generate a map background for Map View. TerraServer is a Microsoft® website providing access to a database of topographical and photographic maps for the United States and some other locations at no cost to the user. CoPilot determines which maps you need based on the longitude and latitude of the Map pane display, and downloads the maps for that area into a buffer. TerraServer requires Windows 2000® or above, Internet Explorer® 5.0 or above, and Internet access.

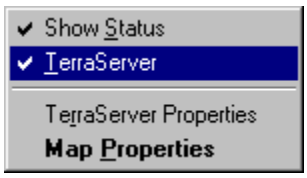


To link to TerraServer images:

- Right click the Map View window to access the context menu
- Enable **TerraServer** by selecting it. Notice that reopening the context menu will show a check mark next to TerraServer.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



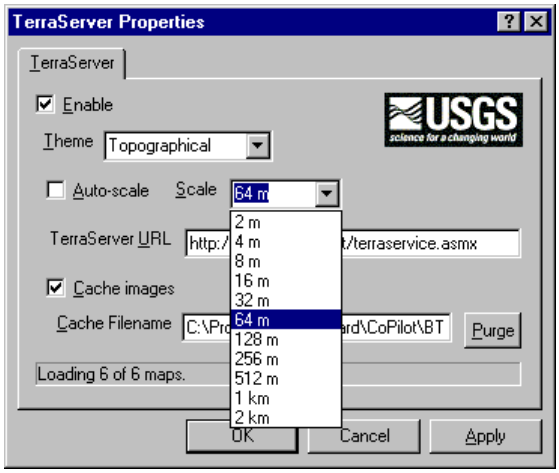

To change TerraServer Properties:

- Right click the Map View window to access the context menu
- Select **TerraServer Properties** to open the properties window

## *TerraServer Configuration*

### Topographical Images

TerraServer maps are digital raster graphics (DRG), that is, digitally scanned images of a USGS (U.S. Geological Survey) topographic maps. Printed maps are scanned at a minimum of 250 dots per inch (dpi) or higher and compressed in a TIFF file. They are not a substitute for aeronautical maps, but they do provide a recognizable background for tracking flight. Users may select a scale from two meters to 2 kilometers per pixel, but the best solution is usually achieved through automatic scaling.




### Photographic Images

The USGS provides black and white photographs of the United States. The photographs are digitized to one square meter per pixel and corrected for distortion. These images (known as digital orthophoto quadrangles or DOQs) cover an area of about 25 square miles. Up to sixteen overlapping images can be combined in a Map View window. Selecting an image scale that fits these constraints can be difficult since a very high resolution may be slow and exceed the 16-image limit. Auto-scaling is usually the best solution.

### TerraServer Status Information

When TerraServer is enabled, the same status information found at the bottom of the TerraServer Properties page can be found in the left section of the CoPilot status bar (bottom section of the CoPilot desktop).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Scripting Displays

CoPilot 5 introduced the CoPilot ATE (Automated Test Environment). This environment uses Python script and allows automated manipulation of CoPilot objects with greater control than ever offered before. The following section describes the features of CoPilot ATE. Microsoft® Visual Basic® scripting is still supported using a Script View.

### Python Script Editor Overview

The Python Script Editor is a critical component of CoPilot ATE projects, enabling users to automate many of the tasks associated with simulating and testing via CoPilot. The following section describes CoPilot ATE in more detail.

The CoPilot 5 automation model has been expanded to further enhance the capabilities of Scripting. The functionality available to Python Shell-Level Scripting supersedes the VB (Visual Basic) scripting that was available in CoPilot 4. Although the VB Script Views will still be supported, it is suggested that new development use the Python Shell-Level scripting instead.

### Script View

Script View is a CoPilot Professional component that allows users to extend the functionality of the CoPilot environment. The CoPilot 5 automation model has been expanded to further enhance the capabilities of Scripting. However, Script View which uses the Microsoft® Visual Basic® Scripting language (VBScript) has been superseded by the new Python Shell-Level Scripting described in the 'Shell-Level Scripting' section that follows. Although the VB Script Views will still be supported, it is suggested that new development use the ATE with Python scripting, and debugging facilities instead.

Note: See *Appendix H: Legacy CoPilot Support* for additional information about VB Scripting.

# Controls

Controls are used throughout CoPilot. Controls (such as labels, knobs and buttons) are used to display data and to edit data. CoPilot Professional adds the ability to customize the visualization of data using 'Quick Views' and 'Control Views.' The Control View, as described earlier in this section, is a highly customizable container for controls, gauges, dials, and virtual instruments. These control widgets use the ActiveX technology to allow embedding and linking. *Appendix E: Control View Library* on page *445* describes the controls included with CoPilot Professional.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Automated Test Environment (ATE)

## Overview

CoPilot Automated Test Environment (ATE) includes the concepts of shell-level scripting, test management, and other features to the software. ATE expands upon scripting concepts introduced with VB Scripting in the Script View in CoPilot 4.

Through the use of the Python Programming Language (www.python.org) in ATE, CoPilot is able to leverage several advantages from this powerful scripting language, including:

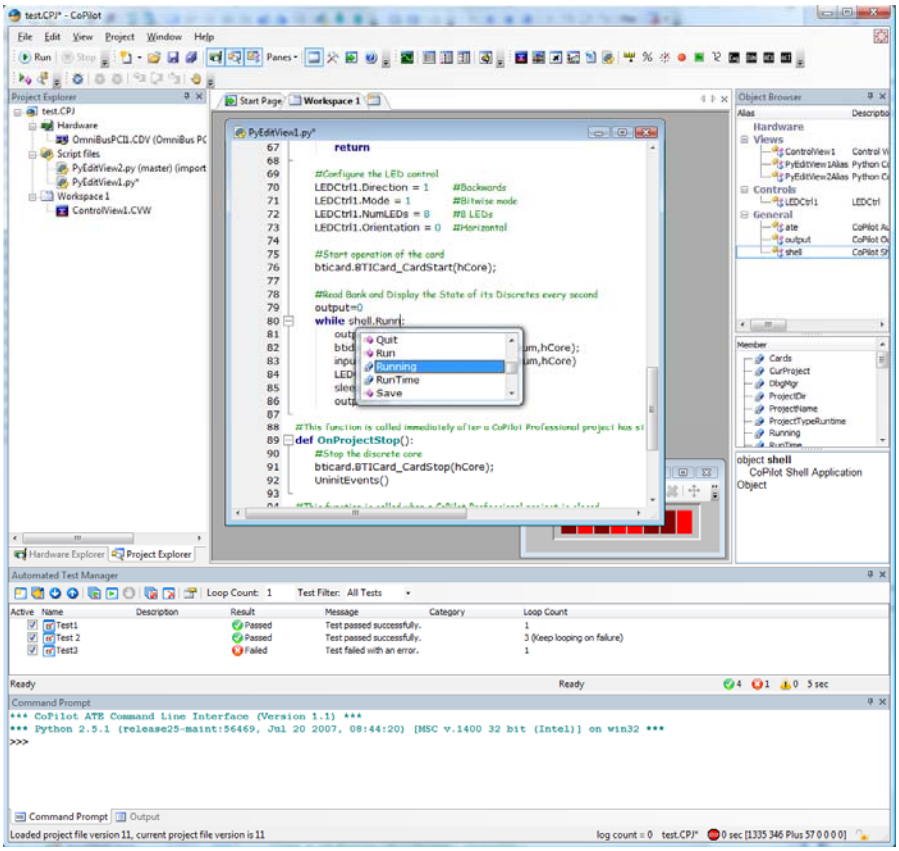
- Vastly improved script performance compared with VB Script in Script View
- Easier editing of code, with syntax error checking, Autocomplete, Intellisense, and a more flexible text editor
- A more feature-rich and powerful scripting language (such as 64-bit data types)
- Quicker code execution resulting in faster script calls
- The ability to run functions at any time instead of just on simulation start and stop (ability to control the CoPilot shell object)
- Much better error information when errors occur
- A large library of code to use included with Python by default
- Command line interface with the 'Command Prompt'
- Debugging functionality with 'Watch Window' from within CoPilot
- Integrated math function support with the Python math module

Using CoPilot ATE, users are able to control many of the aspects of CoPilot by writing simple commands using the published "Automation Model." By writing a chain of these commands in a Python "module," users can automate tasks ranging from simple macros to complex test frameworks.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CoPilot 5.0 introduces several UI elements that are shown in the following screen shot and provides full access of the CoPilot Automation Model to the ATE components.



*CoPilot showing several features of ATE: Python Edit View, Object Browser, Automated Test Manager, and Command Prompt*

- **Automated Test Manager** – A framework provided for running automated tests powered through Python scripts in CoPilot.
- **Output Pane** – Contains textual output from various portions of the CoPilot software.
- **Command Prompt** – Users may enter individual commands using the Automation model and the Python scripting language in this pane.
- **Object Browser** – Shows aliases accessible via CoPilot ATE scripting which represent various objects in CoPilot.
- **Python Script Editor** – Editor for writing scripts in the Python programming language that can control CoPilot operations.
- **Python Debugger** – Facilitates the debugging of Python code through breakpoints, controlled program execution, etc.
- **Watch Window** – provides the ability to watch (and edit) variables and object properties for debugging and tracing script execution.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Python Help

Extensive documentation for the Python scripting language exists. The following section describes how Python scripting is used in the CoPilot ATE environment. Use the Object Browser or auto-complete to show what properties and methods are available to an object (alias).

An online help utility is included and is accessed from the command prompt. Typing 'help()' in the command prompt will show basic help information. A list of available modules, keywords, and topics, is available by typing 'help("modules")', 'help("keywords")', or 'help("topics")'.

Typing *help("math")* shows the help information for the included math functions, such as acos, asin, cos, cosh, log, pi, pow, sin, sinh, sqrt, tan, tanh, and many more,

A Python tutorial can be found on the internet at: http://www.python.org/doc/current/tutorial/.

# ATE Projects

## Introduction

CoPilot ATE projects are created in the same way as any other project. In order to allow your project to interact and run scripts, you must first add a "Master script" to your project. The function of the Master script is described in the following section. Creating the Master script can be done in a number of ways:

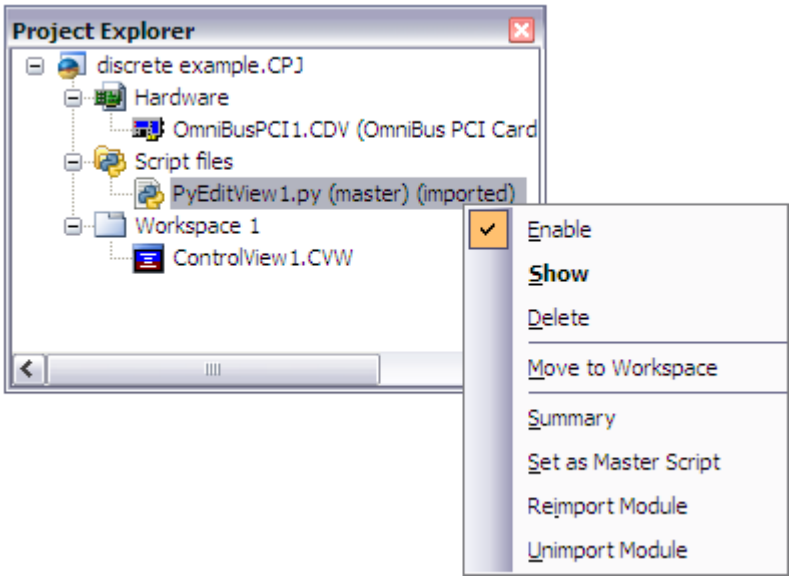
- **New Hardware or View Dialog** – By selecting **File | New | New Hardware or View**, then selecting "Python Script Editor" from the "Professional Views" tab. This will add a script to your project automatically configured as the master script.
- **Project Explorer** – By right-clicking on the "Script files" node, you can select to create a new Master script.
- **Automatically** – From the CoPilot Options dialog (Project | Options), you may select to have Master scripts created automatically with new projects.

## Scripts in Project Explorer

ATE Projects are controlled by Python modules (or scripts) contained in the project. These modules are managed in CoPilot in the same way as any other view. In the Project Explorer, Python scripts are hosted under the "Script files" node.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




As shown by the screen shot, Python scripts have a few extra options in their context menus:

| Item | Description |
|---|---|
| Set as Master Script | Tags the script as the Master Script for this project. See the "Master Script File" section below for more details on the function of a Master Script. |
| Import/Reimport Module | Imports the module, first checking the syntax. See "Importing modules" below. |
| Unimport Module | Unimports the module, available only if the module was already imported. See "Importing modules" below. |

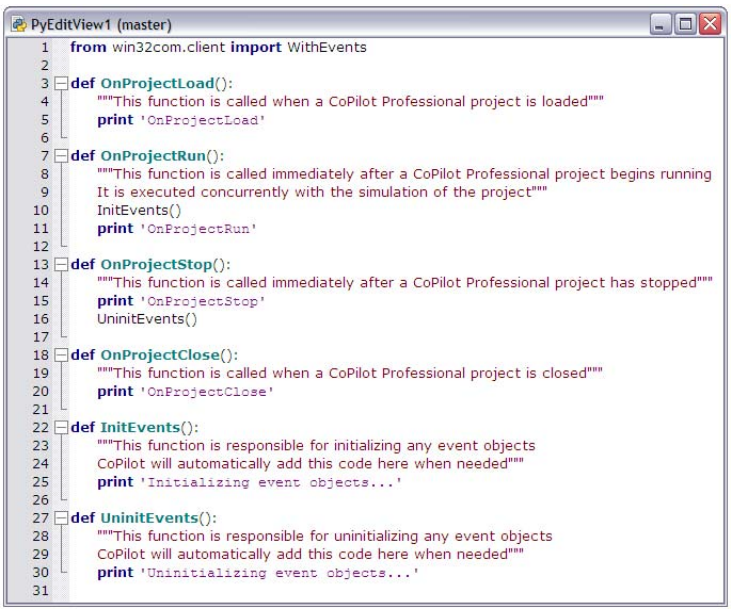
## Master Script File

The Master script is a file that contains script code and handles events. The Master script file instructions are written in the Python programming language to take specific actions when certain project related events occur. When a project is loaded, started, stopped, or closed, an associated function for that event is called in the specified Master script file. If the functions are not defined in the Master script file, then that function is skipped; it will not use alternate definitions from other user script files.

Note: Only one file may be set as the Master Script file at a time. When tests are run with the ATE Test Manager, the script file used by each test is temporarily made the Master Script while the test is running.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




When a Master script is created, it is automatically populated with 6 functions notated with the "def" keyword. The first four Master script functions (`OnProjectLoad`, `OnProjectRun`, `OnProjectStop`, `OnProjectClose`) are meant to be modified by the user to accomplish their tasks. The final two functions (`InitEvents` and `UninitEvents`) are not modified by the user typically and may be ignored in most cases. These two functions are responsible for setting up and tearing down connections to objects that handle events. Event handling is described in more detail in a later section.

Take a look at the first function defined automatically:

```
def OnProjectLoad():
   """This function is called when a CoPilot
Professional project is loaded"""
   print 'OnProjectLoad'
```

On the first line is the function definition. This is the function name CoPilot looks for to call into when the project containing this script is first loaded. If the name is changed, the function won't be called. The second line contains a docstring (designated by use of triple-double quotes) which describes the function in typical Pythonic fashion. Finally, the third line contains a print statement which will show up in the Output Pane when this function is called (the Output Pane is described in detail later).

---

Note: The call to `OnProjectLoad` can be skipped by simultaneously holding the (**Ctrl**+**Shift**) keys while the project is loading.

---

The Master script is also responsible for holding event handlers for aliases defined by the project. These event handlers will automatically insert a few lines of code into the `InitEvents` and `UninitEvents` functions. Aliases and event handlers will be described in more detail, later.

Finally, the `OnProjectRun` function is treated specially by CoPilot. The other three functions in the Master script are executed sequentially with other operations in CoPilot. In other words, CoPilot will wait for this script to finish before continuing to do any more processing. However, with the



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

`OnProjectRun` function, the code therein is executed *concurrently* with CoPilot processing. In other words, this code may continue to run in the background as the CoPilot simulation continues. The ability to continue running this code along side of CoPilot greatly enhances its capability to monitor databus activity and respond to various inputs dynamically, without having to rely on an event handler. Event handlers will be described in detail in a later section.

Note: The `OnProjectRun` function, while similar to the `OnScriptStart` function that was used within the Script View, executes in a different fashion. It is executed concurrently with the running of the project rather than before it.

## User Scripts Files

Each project may contain other script files in addition to the Master script file. These other script files are structured exactly the same as the Master script, except the Project does not directly interact with these script files. Instead, the user is responsible for calling the functions defined within these modules. Functions defined in these script files may be called from the following places:

**Command Prompt** – First, you must import the module by right-clicking the script file in the Project Explorer and selecting "Import Module"

Then, you may access functions from the module using the following syntax:

```
modulename.functionname()
```

Accessing globals of a module from outside of the module (i.e. shen viewing the global variable a watch window) using the following syntax:

```
modulename.globalname
```

While the project is running, you may only call functions that aren't defined in the Master script due to the threading model used to run the project event handlers. Otherwise, an error will be generated.

**Master script** - Other modules may be imported and called from any of the four project event handlers (described earlier). This uses the same syntax as calling into modules from the Command Prompt.

**Macros** – Scripts can be called as macros, which are short functions with no parameters used to automate a particular task. These will be described in more detail in the "Macro Execution" section.

**Other User script** files - Other modules may be imported and called from within another script file that is referenced via the Command Prompt, Master script, or a Macro. This uses the same syntax as calling into modules from the Command Prompt, but would be in the script of another module.

Note: In order to successfully import and run code, there must not be any syntax errors. Errors will be displayed in a message box on the screen and the offending line will be highlighted in the script that has the error.

It is also worth mentioning that a user script can be promoted to the Master script through its context menu in the Project Explorer. However, if the script doesn't define any of special Master script functions (`OnProjectLoad,`



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

`OnProjectRun, OnProjectStop, OnProjectClose`) it will be unable to perform any of the special tasks usually reserved for the Master script. Even if these functions are defined in other script files, CoPilot only attempts to call the Master script functions for the module specified as the Master.

## Running Scripts

As mentioned in the previous section, script code can be run from the Command Prompt Pane, via various scripts hosted in the current CoPilot project (both master and user modules), and finally through the Macro dialog.

Before any code is run, the code is checked for any syntax errors. These errors prevent the interpreter from being able to execute the instructions, so execution of the code will not begin. Errors of this type will be shown in a message box with a description and traceback provided by the interpreter. CoPilot will then highlight the line that caused the error to occur. If a syntax error is encountered from a command you entered in the Command Prompt Pane, the interpreter output will simply be displayed on the next line of the Command Prompt in red.

Once the code begins running, it is possible that the Python interpreter encounters a runtime exception for some reason, such as calling a function or importing a module that doesn't exist. These don't show up as syntax errors since the code is still syntactically correct. However, when the line is encountered and run by the interpreter, an exception will be passed back by the interpreter describing the problem. If your own code doesn't handle this exception with an exception handler, CoPilot will display the exception string and traceback, which will look similar to the syntax error display. Here again, the line that caused the exception will be highlighted in CoPilot where applicable or simply printed on the next line of the Command Prompt Pane if encountered there.

Some exceptions may be generated asynchronously, such as in an event handler or in a different thread you have created. Since these types of exceptions will not prevent a script from continuing to run, they will not generate a message box and highlight the offending line. Instead, the error(s) will be printed to the Output Pane.

A more thorough introduction to errors and exceptions in Python can be found at http://docs.python.org/tutorial/errors.html.

## Script Threading (Advanced Topic)

Python provides threading functionality through the `Threading` module. This is an advanced feature that is not explicitly supported by all of our CoPilot objects. While it may work fine for some applications, using event handlers or the concurrency provided by the `OnProjectRun` function is a better approach.

Should using the `Threading` module be unavoidable, it is necessary to initialize COM in the thread that is created in order to marshal COM calls into and out of the new thread. This is done with:

```
#Begin your thread
import pythoncom
pythoncom.CoInitializeEx(pythoncom.COINIT.APARTMENTTHREADED
#Your code
pythoncom. CoUninitialize()
#End your thread
```

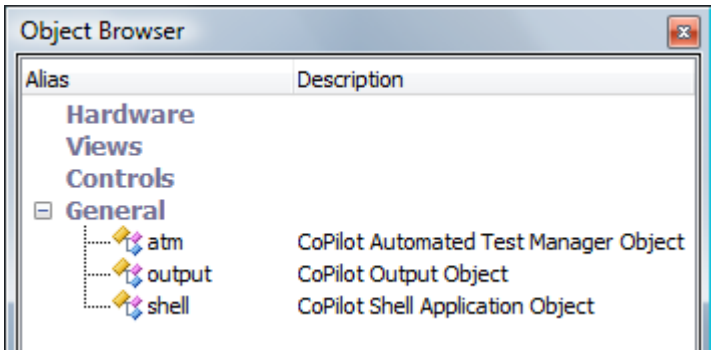


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

These two calls will setup and teardown COM for your thread.

## Script Alias References

Aliases in CoPilot are script-accessible representations of various objects in CoPilot. A list of the current aliases for the open project can be found via the Object Browser. This pane is described in more detail later in the "Object Browser" section.




Each alias has a name (used to reference the alias in your scripts) and a number of methods, properties, and events (referred to collectively as members). Accessing these members is as simple as typing a "." following the name of the alias in your script code. When you enter a . into a script or in the Command Prompt following an alias, the members will appear in a list to show your what is available for that alias.

There are four categories of aliases, as shown in the above screen shot. While aliases in each category are accessed and utilized in the same fashion from Python, the types of objects they represent and how they are added to the project are different.

- **Hardware Category** – Contains aliases that represent objects from the Hardware Explorer tree. These aliases are added to the project by dragging them from the Hardware Explorer tree into the Object Browser or via the context menu on the item in the Hardware Explorer.
- **Views Category** – Aliases in this category correspond to the various views in the project. These aliases are added to the project automatically. They may be renamed and the new name is saved in the aliases.py file associated with the project.
- **Controls Category** – This category hosts aliases for controls in the Control View. Aliases here are automatically added, and their names are always consistent with the "Name" property of the control they represent.
- **General Category** - Hosts aliases called "ate," "output," and "shell" that are always present regardless of whether a project is open or not. These aliases are important and described later in further detail. They may not be removed or renamed by the user.

Aliases are automatically saved in the Project directory whenever a project is saved, to make sure they are available the next time you open the project. The saved aliases are actually loaded by running a script called "aliases.py" which contains instructions for the project to load aliases. Modifying this file to load different aliases is possible, but is beyond the scope of this manual. Aliases are



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

most easily changed for a project by simply adding and removing them from the Object Browser and resaving the project.

## Built-in Aliases

### *Description*

CoPilot contains three built-in aliases, available for access from Python at any time. These three aliases cannot be deleted or renamed. Their purposes are outlined below.

### *The "shell" alias*

Perhaps the most important alias is one that you don't have to worry about adding. It is called "shell" and it represents the CoPilot application. From the shell object, you can traverse the entire CoPilot Automation Model to access any Automation object from CoPilot. You can use methods and properties on the shell object to control various aspects of the CoPilot software, such as opening and closing projects, running and stopping projects, accessing hardware statistics, and controlling the appearance of views in the View area.

Since the shell alias is always available, you can use it outside the confines of a CoPilot project to launch other projects, or use it to launch a sequence of projects in a test sequence.

### *The "atm" alias*

The "atm" alias provides the interface to the Automated Test Manager framework. Calls into the methods on this alias tell the Automated Test Manager when to run or stop its tests, as well as when a test has passed or failed, and provides logging capabilities to the active test. This alias is described in more detail in the "Automated Test Manager" section.

### *The "output" alias*

Many of the functions available in the Output Pane through its toolbar may be accessed through the "output" alias, as well as some new ones. Functions such as saving, clearing, printing, and switching the active output stream are available, as well as a method for creating brand new output streams to track different kinds of output separately. For more information, see the "Output Pane" section.

# Automated Test Environment Components

## Python Script Editor

The Python Script Editor is a view in CoPilot specifically for the maintenance and management of Python scripts. Since it is a view, it is saved along with the project. However, the views are special since they have the extension .py, which is the standard Python script extension. This allows them to be viewed and modified by other editors outside of CoPilot, though this is generally not recommended.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Text Editor Features

The Python Script Editor at its most basic level is a text editor, and supports basic word processor operations, accessible via the view's context menu as well as the "Edit" menu that appears when the Python Script Editor is the active view.

| Item | Shortcut Key | Description |
|---|---|---|
| Undo | Ctrl+Z | Undo the last action |
| Redo | Ctrl+Y | Redo the last undone action |
| Cut | Ctrl+X | Cut the selected text to the clipboard |
| Copy | Ctrl+C | Copy the selected text to the clipboard |
| Paste | Ctrl+V | Paste the text from the clipboard |
| Delete | Del | Delete the character in front of the caret |
| Select All | Ctrl+A | Select all text in the document |
| Zoom In | Ctrl+Mouse Wheel Up | Enlarges the text by one point, making it appear larger |
| Zoom Out | Ctrl+Mouse Wheel Down | Reduces the text by one point, making it appear smaller |
| Find | Ctrl+F | See description below |
| Find Next | F3 | See description below |
| Find Previous | Shift+F3 | See description below |
| Replace | Ctrl+H | See description below |
| Check Syntax | *None* | |
| Reimport Module | *None* | Imports the module, first checking the syntax. See "Importing modules" below |
| Unimport Module | *None* | Unimports the module, available only if the module was already imported. See "Importing modules" below. |
| Auto-Indent | *None* | See description below |
| Line Number | *None* | See description below |
| Code Folding | *None* | See description below |
| Expand / Collapse | Mouse click the ⊟ / ⊞ icons | Not in context menu. See code folding description below |
| Expand All / Collapse All | Ctrl+Shift+Mouse click the ⊟ / ⊞ icons | Not in context menu. See code folding description below |
| Indentation Guides | *None* | See description below |

*Python Script Editor Context Menu and Keyboard Shortcuts*

## Find and Replace Functionality

Using Find and replace in the Python Script Editor is much like the Find and Replace found in many word processors. It maintains a history of past search/replace terms in the drop down list, and also allows you to enter new search/replace terms. You can select to match entire words, to match the case, pick a search direction, or use a regular expression as your search pattern (advanced). To find out more about using regular expressions, please see *Appendix G: Regular Expressions*.

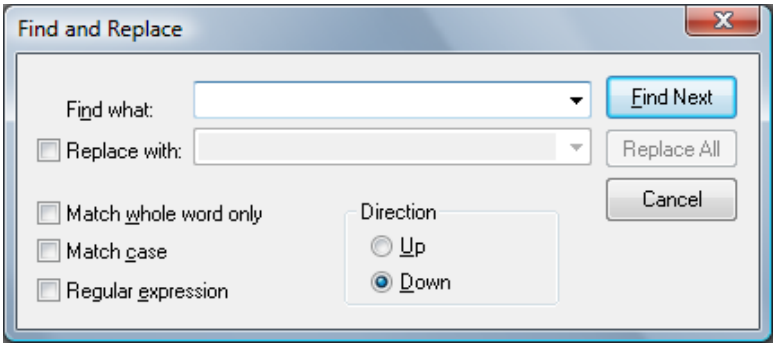
Starting the dialog with Find (**Ctrl**+**F**) versus Replace (**Ctrl**+**H**) changes the default state of the dialog. With Find, the "Replace with" combo box is grayed out and unchecked. The dialog effectively finds and replaces text based on the state of this check box, and the button text changes between "Find Next" and



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

"Replace Next." When in replace mode, the "Replace All" button also becomes available, allowing all occurrences of a text string to be replaced at once.

Once you select to "Find Next," or "Replace Next," the editor will find and either highlight or replace the next occurrence of the word based on your current caret location (could be before or after the caret depending on the search direction). If the search item is not found, a message box will appear you informing you of this. You may continue to find and replace occurrences using the last search term by using the "Find Next" and "Find Previous" commands. These commands will also perform replace functionality if that was the last operation performed.




## IDE Features

The Python Script Editor is also a powerful IDE, providing support for writing Python code quickly and efficiently.

| Item | Description |
|---|---|
| Add Watch | Adds the word under the mouse-click to the Watch Window with automatic scope. |
| Check Syntax | Check the syntax of the module. See "Syntax check" below. |
| Import Module | Imports the module, first checking the syntax. See "Importing modules" below. |
| Unimport Module | Unimports the module, available only if the module was already imported. See "Importing modules" below. |
| Auto-Indent | Turns auto-indent on/off. See "Auto-indent' below. |
| Line Numbers | Turns line numbers on/off. See "Line numbers" below. |
| Code Folding | Turns code folding on/off. See "Code folding" below. |
| Indentation Guides | Turns indentation guides on/off. See "Indentation guides" below. |

## Syntax highlighting

Code will be colored and formatted to help you to identify various keywords and different language features, such as strings, comments, variables, numbers, functions definitions, and classes. This helps make the code more readable and easier to edit quickly. It will also highlight text when you currently have a string open, to help you remember to close your string again before entering new code.

## Syntax check

It can be frustrating to write some code and attempt running only to find that there are syntax errors. The Python Script Editor will automatically check the syntax when you attempt to save, import, or run the "Check Syntax" command. If there is a syntax error, it will be displayed in a message box and the offending line will be highlighted in red by the editor.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Debugger

CoPilot 5.1 Introduced the Python Debugger, which provides functionality useful for debugging code, such as stepping and breakpoints. More information can be found in the "Python Debugger" section.

### Importing modules

Typically in Python, modules are imported using the import statement, by typing

```
import modulename
```

Modules include both modules that are included as a part of the standard Python distribution (such as math) and user created modules.

Since CoPilot manages the user created modules defined in a project, CoPilot is able to automatically import these files. However, standard Python modules must still be imported. Whenever you save a file defined in the Python Script Editor, CoPilot first attempts to import the file as a module and then notifies you if any syntax errors prevented the import from occurring successfully.

Once the module is imported, its functions and variables can be accessed without an import statement from anywhere within CoPilot. You can tell if a module has been imported by looking at its title bar or its name in the Project Explorer. If the name says "(imported)" next to it, you know that is available for direct access.

You may decide to update code in modules that have already been imported. To make sure that these code changes take effect when you access the module in the future, you will need to import it again. This can be done by saving the module again (import is automatic), or by selecting "Reimport Module" from the menu. Reimporting the module in either fashion will overwrite the old version with the new version.

If for some reason you wish to get rid of a module definition, you can select "Unimport Module" from the menu. After doing this, the module will no longer be directly available without an explicit "import" statement or by importing the module again through the menu.

### Autocomplete

Many IDEs provide help for discovering what methods and properties are available on a given object. When you type a supported object's name followed by a "." in the editor, autocomplete will give you a drop down list of all the methods and properties that object supports. Selecting one of these items with the mouse or by pressing enter will place that item string into the editor.

### Calltips

When calling into a supported function, a calltip will appear, that shows the parameters required by the function, and a description of the function if available. This gives you information about the proper parameters to pass into the function without thumbing or clicking through documentation.

### Tab nanny

Python is a tab-delimited language, meaning that the tabs serve a special purpose in the language syntax. Both spaces and tabs are valid delimiters; however they cannot be used interchangeably within the same script. The Tab



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

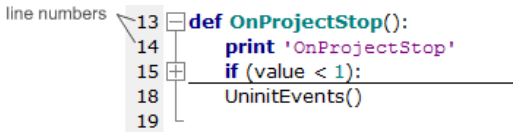
nanny will underline spaces or tabs with a blue pattern when they don't match the delimiter you had been using previously in the script.

### *Auto-indent*

Placing the correct number of tabs after every line of code is necessary and can become cumbersome. Fortunately, auto-indent will automatically indent the next line of code the correct number of times for you.

### *Line numbers*

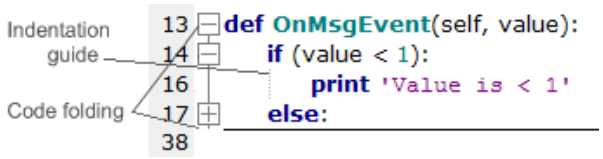
Line numbers are displayed along the left margin.




*Python Script Editor line numbers*

### *Indentation guides*

These provide a vertical dotted line in the editor in order to help you understand and manage the tab levels within your Python code.




*Python Script Editor indentation guide and code folding*

### *Code folding*

The code folding markers allow you to expand and collapse various sections of code. This is helpful for hiding code that you don't care to look at or may be distracting, as well as hiding comment strings that you don't need to see in order to fit more code on the screen. Look for the ⊞ and ⊟ icons on the left margin next to the line number (if visible). Clicking on a ⊟ will collapse that code section, whereas clicking on a ⊞ will expand it.

Note: Holding down (**Shift**+**Ctrl**) while clicking a ⊞ or ⊟ will expand all or collapse all.

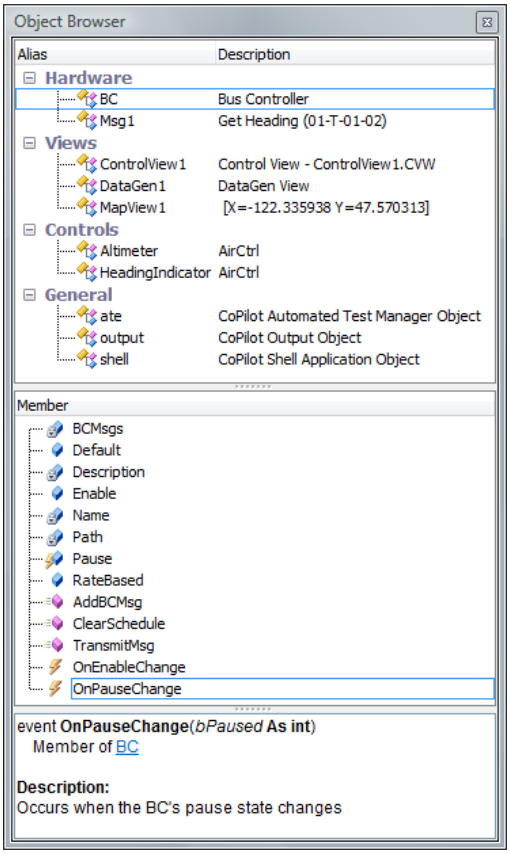
## Object Browser

The Object Browser is a pane in CoPilot used for tracking the aliases available for use with ATE, via the Python Script Editor or the Command Prompt Pane. It contains a list of all the aliases available to the currently opened project. These aliases represent various CoPilot objects, and have methods and function that operate on or query these objects. Some of these aliases also have events, which will be discussed in further detail later.

Note: The Watch Window Pane is often used in conjunction with the Object Browser to view the values of variables and particular object properties.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




## *Object Alias Assignment Mechanics*

Aliases are added either automatically by CoPilot or through a specific user action. For a user to add an alias themselves, they have three options:

- Drag an object into the Object Browser – After prompting for a name, the object will be added to the alias list.
- Via the Object Browser context menu – Selecting "Add New Alias" from the Object Browser context menu will allow you to select an alias from the "Select A Scriptable Object" dialog. Here you will see all the objects in CoPilot that are available to be an alias. Selecting one will prompt for a name and add it to the alias list.
- Via the Hardware Explorer context menu – Supported objects in the Hardware Explorer tree have "Assign Alias" in their context menus. Click this to give the alias a name and add it to the alias list.

Once an object is an alias, it is now directly accessible from Python script code in the Python Script Editor and Command Browser Pane. It will remain available as long as the project it is defined in remains open. It is important to note that not all properties and methods on an object are available while in Simulation mode. Usually this will be obvious from the description of the method or property. If it is not available, calling that member will result in a raised exception (see section "Running Scripts").

These aliases can be renamed from their context menu in the Object Browser or by double-clicking their name in the Object Browser. Trying to add the same object twice into the Object Browser will simply give you the chance to rename that alias, instead of adding a duplicate. The "shell," "ate," and "output" aliases



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

are special, permanent aliases that persist even outside of the existence of a project. They may not be renamed.

Valid names for aliases match naming conventions for many scripting languages, specifically:

- The name must begin with a letter or underscore
- It may contain only letters, numbers, and underscores

Attempting to add an illegal name will show an error message and prompt you to choose a different name.

Keep in mind that if you change the name of an alias, all references to that alias in Python scripts needed to be updated to the new name. This is not done automatically.

Aliases may be removed by pressing the "delete" key when they are selected in the Object Browser, or by deleting them from their own context menu. Note that aliases will automatically be removed when the object they represent is deleted. Aliases may also be deleted through the Python functions `DeleteAlias` and `DeleteAllAliases`.

Object Browser Overview

The Object Browser is a visual representation of the aliases that are currently available. It allows you to browse the various defined aliases and discover what types of members (methods, properties, or events) are available. Selecting an alias from the top panel (Object Panel) will display its members in the panel below it (Member Panel) as a list. Then, in turn, these members can be selected to display a description in the third and final panel (Description Panel). The Object Browser is an invaluable resource for discovering the functionality available to you through your aliases.

#### Object Panel

You can perform a few operations via the context menu of aliases in the Object Panel:

| Item | Description |
|---|---|
| Rename Alias | Allows you to rename the selected alias. |
| Remove Alias | Allows you to remove the selected alias. |
| Add Alias | Brings up a selection dialog that allows you to pick a new alias to add from a list of available objects. |

Changing the selection in the Object Panel Updates the contents of the Member Panel.

#### Member Panel

If there is an alias selected in the Object Panel, this Member panel will show all available properties, methods, and events for that alias. Definitions:

- **Property** – This is a value associated with the object. Sometimes these properties are read-only and may only be queried using the `alias.property` convention. Its value may be changed using `alias.property = newvalue`.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

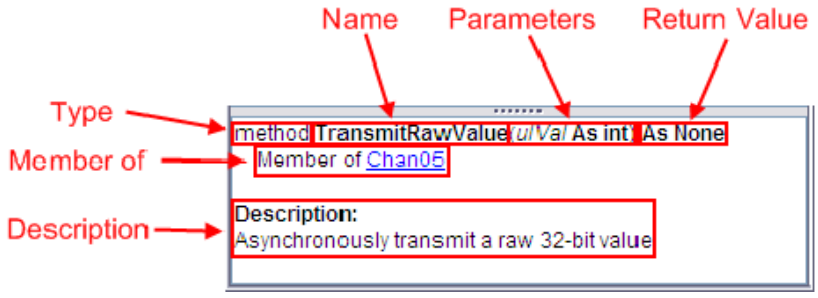
- **Read-Only Property** – Same as Property with the exception that the value can only be read. Attempts to write to this property results in an `AttributeError` exception.
- **Event Firing Property** – Same as Property with the added effect that an event will be fired when this property value changes.
- **Method** – **These can a**lso be thought of as functions which cause an alias to perform an action. Methods may have a list of *parameters* which must be passed in as *arguments* to the method. It is also possible that the method has no arguments. Finally, methods may return a value, so it is valid to assign the return value to a new variable such as `returnvalue = alias.method(arg1, arg2)`. Some methods return more than one value, and are called with the form `returnvalue1, returnvalue2 = alias.method(arg1, arg2)`.
- **Event** – Some objects support members called events. These are described more thoroughly in the following "Using Events" subsection.

The following table outlines the context menu operations available in the Member Panel:

| Item | Description |
|---|---|
| Show Member Properties | Filters the Member List to toggle showing/hiding all properties. |
| Show Member Methods | Filters the Member List to toggle showing/hiding all methods. |
| Show Member Events | Filters the Member List to toggle showing/hiding all events. |
| Sort Alphabetically | Sorts the member list alphabetically. |
| Sort By Member Type | Sorts the member list by type (i.e. property, method, and event). |
| Copy Target | Copies a string that concatenates the alias name with the member name (e.g. `aliasname.membername`) that can be pasted into a script or the Command Prompt Pane for use. |
| Add To Watch | Adds the selected alias/member name combination (e.g. `aliasname.membername`) to the Watch Window with a global scope |

### Description Panel

The bottom panel in the Object Browser is the Description Panel. As the member selection is changed in the Member Panel, the text in the Description Panel changes to provide help about the selected member.




*Object Browser's description panel*

Object Browsers



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

**Type** – This reflects the type of member it is: method, property, or event.

**Name** – The name of the member

**Parameters** (optional) – If it's a method or event, here the parameters for the function appear, as well as their types. The various Python data types available are described in the "Python Data Types" section.

**Return value** – The type of the value returned from the method, or the type of the property. Not applicable for events.

**Member of** – Shows the name of the alias that hosts the method

**Description** – Documentation describing the member in more detail, including warnings, when to use, and any sort of parameter constraints.

### *Using Object Events*

Aliases for some objects also support events. Events are handled differently than other member types, since they are not accessed directly from your script code. Instead, CoPilot helps you handle these events automatically.

An event is essentially a way for an object to notify script code when an important action has occurred, such as when a value goes out of range, a message goes stale, or an error is detected. These events will pass unnoticed unless you specify an *event handler*.

Event handlers are created through the context menu of the event in the Object Browser by clicking "Add Event Handler." If you have a Master script defined in your project, the event handler code will be automatically inserted into your Master script. If not, this operation will fail. Some of the code will be added to the `InitEvents` and `UninitEvents` functions. This code is responsible for connecting the event handlers to the object and should be left alone for the event to work properly.

By default, the `InitEvents` and `UninitEvents` functions are called within `OnProjectRun` and `OnProjectStop`, respectively. This serves to allow the events to be functional only when the project is running. With regards to hardware, this makes sense as most events only make sense when the simulation is running. However, you may wish to control when events are active or not yourself. This is done by changing where these two functions are called.

The actual event handler is held within a handler class associated with the alias. This allows multiple event handlers from the same alias to be grouped together in the same class. When you add the event handler, you will see code that looks something like this:

```
class BC_handler:
    def OnPauseChange(self, bPaused):
            raise NotImplementedError
```

As the code is right now, it will raise an exception of type `NotImplementedError`. This will print the exception and traceback into the Output pane every time the event occurs ("fires"). The exception should be replaced with your own code, such as the following:

```
class BC_handler:
    def OnPauseChange(self, bPaused):
            print "BC Paused =", bPaused
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note that the new code makes use of the parameter `bPaused`. Some events will have more parameters which are explained in the description (shown in the Description Panel). Some events will just have the `self` parameter, which isn't used. Now, instead of printing the exception and traceback whenever the event fires, it will print a line to the output such as:

```
BC Paused = True
```

Of course, this is just a simple example. The code can take any sort of action you desire.

## Python Data Types

The following table lists the various data types that can be encountered in CoPilot ATE:

| Type | Description |
|---|---|
| `array` | An array of values, represented as a list in Python (e.g. `[1,2,3]`) |
| `bool` | A Boolean with valid values `True` and `False` |
| `currency` | A floating point value with a fixed 2 places after the decimal point |
| `int` | An integer value |
| `float` | A floating point value, effectively the same as `double` in C |
| `long` | An integer value with no functional difference from the `int` data type (present for backward compatibility) |
| `None` | A type representing nothing, typically seen for functions that do not return a value |
| `object` | Accepts an object reference, commonly in CoPilot an alias is expected here |
| `str` | Basic string type, represent by surrounding text in quotes (e.g. `"text"`) |
| `time` | Python Time type |
| `unicode` | A Unicode variant of the string type, using 2-bytes per character and designated by preceding the string with a 'u' (e.g. u"text") |
| `variant` | The `variant` type accepts different types as input – documentation will indicate what type of input is valid |

## Python Debugger

### *Overview*

New for CoPilot 5.1 is the Python Debugger. The debugger allows users to debug their scripts using familiar methods common in many IDEs (Integrated Development Environments). Users are able to step through their code one instruction at a time, add breakpoints, and view current values of local variables.

The various commands available in the debugger are accessible through the Python Debugger Toolbar and the menu (Project | Python Debugger).

The following table lists the commands available to the debugger. These can be accessed through the toolbar or Project menu.

| Item | Shortcut Key | Description |
|---|---|---|
| Enable Debugger | None | Enables the debugger. Now any Python code that is run goes through the debugger, meaning breakpoints will be hit. |
| Run Debugger | F5 | Runs the active script, when the debugger is enabled. Only available if the active View is a Python Script Editor. |
| Stop Debugger | Shift+F5 | Stops the script that is being debugged, whether it is running or at a breakpoint. |



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| | | |
|---|---|---|
| Step Into | F11 | Steps into the next statement, including into a function call. |
| Step Out | Shift+F11 | Steps out from the current function, going to the next one up on the stack. If this function is on top of the stack, it will run the function to completion. |
| Step Over | F10 | Steps over to the next statement, skipping any function calls. |
| Breakpoint | F9 | Sets a breakpoint in a Python Script Editor file. |

### *Using the Debugger*

The first step is to enable it, by pressing the **Enable Debugger** button. After doing this, any Python code run from CoPilot is now under debugger control. This comes into play when the user sets breakpoints.

Breakpoints are set in Python Script Editor Views, and viewed in the margin next to the line number (if present). The user may set a breakpoint through the menu, toolbar, or the F9 shortcut.

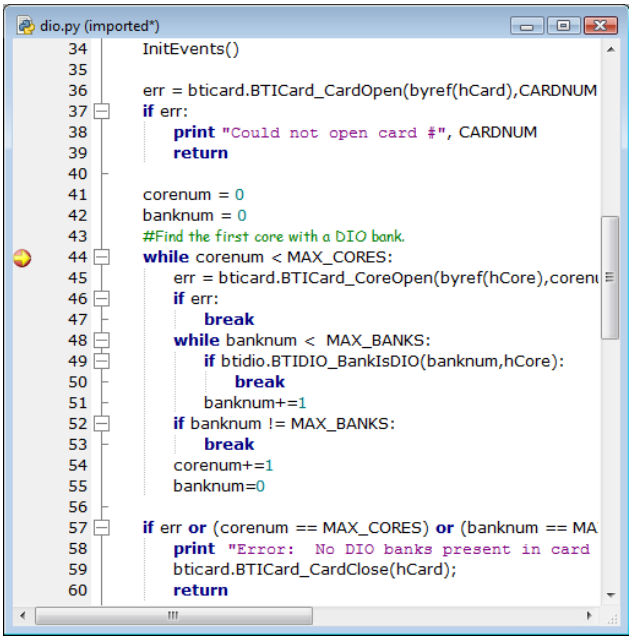
There are three possible symbols that can be seen in the margin when debugging. They are outlined and described below:

| Item | Description |
|---|---|
| Current Instruction | The active instruction. This is the next instruction that will be executed when the script is started again. |
| Active Breakpoint | This defines a point where program execution will halt and wait for user interaction. When the breakpoint is hit, the arrow symbol will overlay the breakpoint symbol, indicating this is the next instruction that will be run. |
| Inactive Breakpoint | Inactive breakpoints occur if the user attempts to modify the script while it is being run. Scripts currently cannot be edited while being debugged, so inactive breakpoints will be ignored. Debugging should not be continued with inactive breakpoints, since the code being executed may not match what is seen in the view. Stopping debugging will return the breakpoints back to their active state. |

Once a breakpoint has been set, the execution of Python code will stop ("break") where the breakpoint is located. In the following screenshot, the execution has stopped at the breakpoint:



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*Python Script Editor at a debugging breakpoint*

At this point, the user may use any command available to the debugger, such as moving directly to the next statement, continuing to the next breakpoint, or even stopping debugging altogether. While stepping through the code, you will see the arrow move to the various statements, depending on the control flow. Additionally, if the code calls into another module defined in the CoPilot project, the current statement will follow into that module and allow debugging to continue normally.

While the debugger is waiting for the next command, it is possible to perform some additional operations from the Command Prompt pane. Variables may be added to the Watch Window and modified on the fly. For more information on the Watch Window, see the related section below. Resuming execution can be done by selecting one of the other debug commands such as "Step Over."

# Automated Test Manager (ATM)

## *Overview*

The Automated Test Manager provides a framework for writing tests using Python. Using this framework and the CoPilot Automation Model, it is possible to write comprehensive tests using Ballard Hardware. Flexibility in the testing framework allow options like configurable tests sets, looping behavior, and logging capabilities. Here are the components of the Automated Test Manager:

**Automated Test Manager Pane** – This serves as a dashboard providing the interface for creating, reordering, and configuring tests. It also shows current test progress, including looping information and pass/fail counts.

**The "atm" Alias** – This is the control interface for the Automated Test Manager. Using the various methods on this alias, Python code can control the execution of tests, as well as report the status of their completion.

**Test Output** – The Output Pane holds an "ATM" stream which contains the output generated by tests and the Automated Test Manager. It serves as a record



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

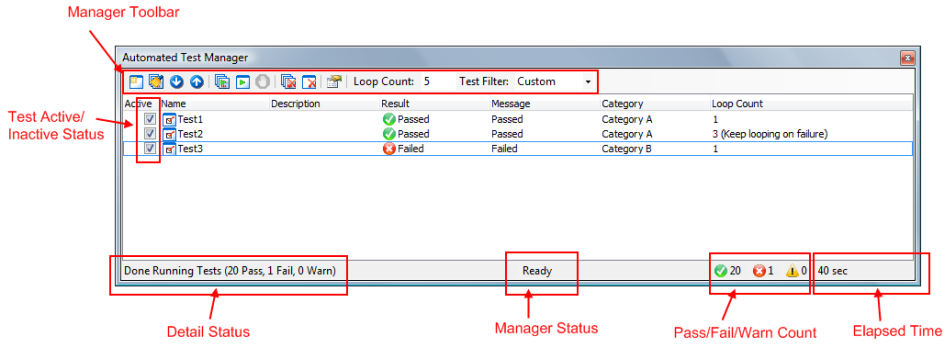
for what has occurred, and includes timestamps and running totals of passes and failures.

Like the other panes in CoPilot, the Automated Test Manager Pane may be accessed through **View | Other Windows** or the CoPilot Standard Toolbar (version 5) or the **Scripting** ribbon tab (version 6). It is only available in CoPilot Professional Version.

## *The Automated Test Manager Pane*

Like other panes in CoPilot, the Automated Test Manager pane can be docked, pinned, or floated freely depending on the situation. It serves as the platform for creating and managing tests, as well as a visual indicator of test progress when tests are running.

The image below shows the layout of the pane, as well as highlights some of the important features.




### Toolbar

The toolbar provides many of the functions available to the Automated Test Manager. They are described below.

| Item | Description |
|---|---|
| New Test | Creates a new test, launching the Test property page allowing you to configure it. More information in the "Creating Tests" section. This is the same as "Add Test…" from the context menu. |
| Clear Output | Clears any generated output from the Automated Test Manager, including any results, counts, and text in the Output Pane. This is the same as "Clear Test Results" from the context menu. |
| Move Down | Moves the selected test down one in the list, effectively making it run later in the test sequence. |
| Move Up | Moves the selected up down one in the list, effectively making it run earlier in the test sequence. |
| Run All | This begins running all the enabled tests through the specified loop counts. This is the same as "Run All Tests" from the context menu. |
| Run Selected | Runs the selected test one time. |
| Stop Test | If a test is currently running, this will stop the execution of any further tests after this test completes. This is the same as "Stop Running Tests" from the context menu. |
| Remove All | Removes all tests. This is the same as "Delete All Tests" from the context menu. |
| Remove Selected | Removes the selected test from the test list. |
| Properties | Accesses the Automated Test Manager property page. This is the same as "Manager Properties" from the context menu. |



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

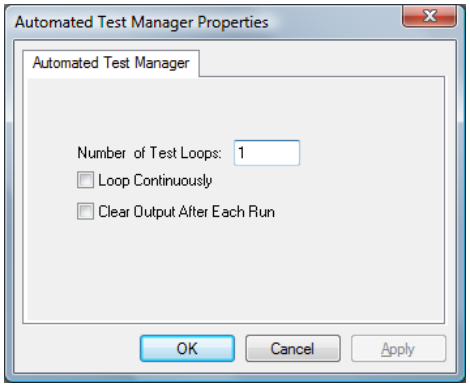
Besides the buttons there is the option to configure the Loop Count. This value specifies how many times all the tests will be run through. There is also the test filter, which will enable and disable tests based on their category. Selecting "All Tests" will enable all tests. Selecting "Custom" allows any tests to be selected. The other options are based on the configured categories for the current tests. Selecting an option will enable only tests of that category.

### Status Bar

At the bottom of the Test Manager Pane is the status bar. It contains the current state of the Manager and information about the running of the tests. The leftmost panel provides detailed status information regarding what state the Test Manager is currently in. The second panel provides an overall status, such as which loop the Manager is currently on. Next to the right is the overall count of Pass/Failures/Warnings for this test run. Finally is an elapsed of the current test run.

### Manager Property Page

The property page of the Manager is accessed through the "Properties" button on the toolbar or "Manager Properties" through the context menu. It is seen below:




*Automated Test Manager property page*

**Number of Test Loops** – Specifies the number of times all of the tests will run through. For example, if five is specified, the active tests will be looped through five times.

**Loop Continuously** – Check to loop the tests indefinitely, until the **Stop Test** button is pressed. This is the same property as the "Loop Count" shown in the Automated Test Manager toolbar. When selected, the infinity symbol will appear here and in the toolbar.

**Clear Output After Each Run** – Every time "Run All Tests" is performed, all of the generated output from the previous test runs will be deleted. Otherwise, new output will continue to append at the end of the log.

### Test List

The main content of the pane is the list of tests that are configured for the project. Each test has its own line in the list. The order in the list is the order that the tests will be run in. To change the order the tests are run in, use the toolbar or context menu and select "Move Test Up" or "Move Test Down." Other information about the tests can be seen in the various columns, including the result of the last run of the test. Tests may be enabled and disabled with the

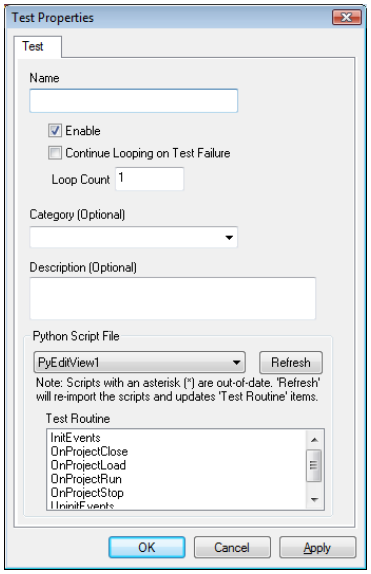


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

checkbox on the far left. Each test also has a context menu with test-specific operations.

## *Creating Tests*

A new test can be created by clicking the **New Test** button on the toolbar. A property page will now appear allowing you to configure the test that is being created.




*Test Properties in the ATE Automated Test Manager*

**Name** – The name of the test as it appears in the Test Manager pane and output. This is a required property.

**Enable** – Check to have the test enabled (default).

**Continue Looping on Test Failure** – If the test is being looped multiple times (Loop Count > 1), then checking this box will allow the test to continue onto its next run even if it fails. Otherwise, the next test will be run.

**Loop Count** – The number of times this test will be run before moving onto the next test in the sequence.

**Category** – An optional property that specifies the test category. This category is used for test filtering.

**Description** – An optional description of the test.

**Python Script Files** – Specifies which Python module contains the test procedure.

---

Note: The specified Python script file (module) is assigned as the master script while the test is running. After all tests are run, the original master script of the project is restored.

---

**Test Routine** – Specifies the Python function called for this test. It must have 0 parameters. There is more information in the "Running Tests" section to follow.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## *Running Tests*

Once one or more tests have been specified, the contents of the test must be written. The test actions are written in the script function specified.

Since the module specified for the test becomes the Master script, keep in mind that the `OnProjectRun` and `OnProjectStop` methods will be called. Any events being used for the test must also be initialized and uninitialized, as they are by default with these two methods.

---

Note: The specified Python script file (module) is assigned as the master script while the test is running. After all tests are run, the original master script of the project is restored.

---

The final important piece of writing test procedures is making sure to inform the Test Manager of the completion of the test, so the next test may be run. This is accomplished through the use of the "atm" (Automated Test Manager) alias. The alias has three methods that should be called just before the function is exited to notify the test completion status. They are:

**`Pass`** – Indicates the test has passed successfully, this is indicated in the Test Manager with a Pass icon.

**`Fail`** – Indicates the test has failed, this is indicated in the Test Manager with a Failure icon.

**`PassWithWarning`** – Indicates the test has passed, but some warnings have occurred, this is indicated in the Test Manager with a Warning icon.

The "atm" alias also supports a method called `Log`. This method has a string parameter which is the log message and a Boolean indicating whether the log message should be tagged with the time and test name or not.

Now that the test has been configured entirely, it is possible to run the test(s).

Run the entire sequence of tests with **Run All** . The Automated Test Manager will now proceed through the sequence of tests specified, obeying which tests are active, test loop criteria, and Manager loop criteria. As the tests are progressed through, the icons of the tests will change indicating which test is currently being run. These icon states are listed and described in the following table:

| Icon | Name | Description |
|---|---|---|
|  | Test | This is the default icon state, when no tests are currently being run. |
|  | Running Test | This icon indicates the test that is currently running. |
|  | Inactive Test | A test that has been disabled will have this icon while tests are running. |
|  | Pending Test | A test that is enabled that will be run has this icon. |
|  | Completed Test | After a test has been run, it will have this icon. |

The status bar will be updated continuously as new tests are run, keeping a running count of passes and failures as well as an elapsed time. All of the information that is visually tracked in the Test Manager pane is also tracked in the Output Pane in the "ATM" stream, as described in the following section.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### *Test Output*

The Automated Test Manager logs its actions through the "ATE" stream in the Output Pane. This output can be saved and printed for future reference as a log of how the tests were run. Output from individual tests through the methods on the "atm" alias are also recorded, along with a time-tag. This is useful for tracking custom information with regards to the testing.

The output may continue to append, or start anew each time the tests are run, depending on the "Clear Output After Each Run" state of the Test Manager.

## Command Prompt

### *Overview*

The Command Prompt Pane provides command line style access to the CoPilot Automation model via the Python interpreter. It allows you enter commands one at a time to call functions, start macros, or just test small pieces of code that you want to put in one of your scripts. Many tasks in CoPilot can be done through entering commands at the command prompt.

Commands entered at the Command Prompt are passed to the Python interpreter, executed, and their result passed back to the Command Prompt to be output as a string. For example, entering:

```
>>> 1 + 1
2
```

These commands, of course, may also act upon CoPilot objects through their aliases, such as:

```
>>> shell.ProjectName
u'MyProject'
```

It is possible to run functions from scripts you've written into the project, by simply calling them

```
>>> MyMacros.Macro1
```

Finally, if this is your first contact with the Python programming language, the entire tutorial can be run from within the CoPilot environment using the Command Prompt Pane. The Python tutorial is located at http://docs.python.org/tutorial/.

#### Features

- **Syntax highlighting** – Script code is colored and formatted to help you to identify various keywords and different language features, such as strings, comments, variables, numbers, functions definitions, and classes. It also highlights text when you currently have a string open, to help you remember to close your string again before entering new code. Finally, error output returned from the Python interpreter when a command is entered is shown in red for easy identification.
- **Autocomplete** – When a supported object's name is typed followed by a "." in the Command Prompt, you will be given a drop down list of all the methods and properties that object supports. Selecting one of these items with the mouse or by pressing enter will place that item string at the current location. Autocompletion for the Command Prompt



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

supports many objects, including Python built-ins and those you have defined earlier in your session at the Command Prompt.

- **Multiline input** – Commands aren't limited to just a single line. Some statements may require several lines to be completed successfully. In this case, pressing "Enter" will not send the command, but rather prompt you for the next line in the statement with a ... prompt. You finish your multiline command by pressing "Enter" on a blank line. Example:

```
>>> x = 1


>>> y = 2


>>> if x > y:
...       print "x is bigger"
... else:
...       print "y is bigger"
...
y is bigger
```

  Pay close attention to the tab level in your multiline commands. Using the wrong number of tabs can cause a syntax error. CoPilot attempts to provide the correct number of tabs automatically, but if your needs are different they can always be deleted or added to.

- **Command history** – Trying to access the history of commands in order to execute them again or modify them slightly (i.e. fix a syntax error) can be done by hitting the "up" and "down" arrow keys.

## *Context Menu Features*

The Command Prompt's context menu supports several features:

| Item | Shortcut Key | Description |
|---|---|---|
| Undo | Ctrl+Z | Undo the last action |
| Redo | Ctrl+Y | Redo the last undone action |
| Cut | Ctrl+X | Cut the selected text to the clipboard |
| Copy | Ctrl+C | Copy the selected text to the clipboard |
| Paste | Ctrl+V | Paste the text from the clipboard |
| Delete | Del | Delete the character in front of the caret |
| Select All | Ctrl+A | Select all text in the document |
| Zoom In | Ctrl+Mouse Wheel Up | Enlarges the text by one point, making it appear larger |
| Zoom Out | Ctrl+Mouse Wheel Down | Reduces the text by one point, making it appear smaller |

### Helper functions

CoPilot ATE introduces functions that can simplify some tasks, or present information to the user:



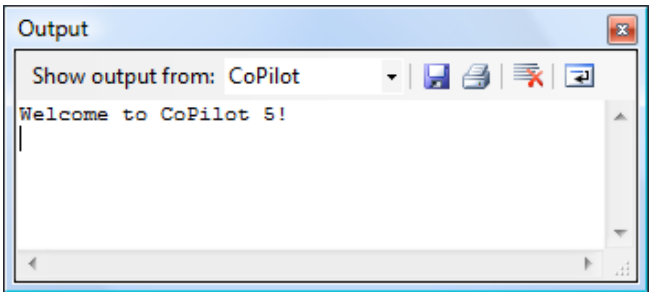
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **`help(object)`** – Placing a function name, object, or type as the parameter to this function will print the documentation associated with the object, if available
- **`PrintAliases()`** – This function will display all currently active aliases that are available.
- **`DeleteAlias(aliasname)`** – Deletes an alias based on the given name parameter.
- **`DeleteAllAliases()`** – Deletes all the aliases currently defined, excepting the built-in aliases in the "General" category.
- **`_ClearGencache()`** – This will reset the generated type library information Python uses to work with aliases. Generally this function shouldn't be called unless problems arise when accessing alias objects. Clearing the cache by calling this function can sometimes fix these problems.

# Output Pane

## *Output Pane Overview*

Output from CoPilot and Python are captured in the Output Pane. The Output Pane consists of a toolbar that allows you to switch between different output streams as well as performing a few different operations on those streams. An "output" alias provides access to the features of the Output Pane through scripting.




*Sample output in the Output Pane*

By default, the Output Pane holds three different default output streams, known as "CoPilot," "Python," and "ATM." In addition to the default output streams, users can create their own output streams.

It is important to note that the contents of the Output Pane are not preserved after a project is closed. If the contents need to be saved for later use, the "Save" button on the toolbar or `ExportToFile` method on the output object can save the text to a file.

## *CoPilot Output Stream*

This output stream is useful for keeping track of what sort of actions may be occurring behind the scenes as CoPilot processes a task. For instance, when a simulation is begun, the CoPilot output stream will announce the various steps in the process as they occur, to give progress updates. When stopping the simulation, the progress is also reported, as well as the total run time. The



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CoPilot output stream can also hold error output, such as reporting when certain views can't be run since the hardware is not keyed to support them.

### Python Output Stream

Any output from the Python interpreter that occurs during simulation from the running of a Python script will output to the Python output stream. In this stream, you will see the output of all `print` statements, as well as any runtime exceptions.

### ATM Output Stream

The Automated Test Manager pane will send its output to this stream. This output includes messages from the Automated Test Manager itself, as well as pass, fail, warning, and log messages output from the various tests.

### Other Output Streams

Additional user output streams can be created via the Command Prompt pane or a Python script. The command `output.CreateStream("test")` will create a new output stream called "test." The return value of this command is the stream ID for the "test" stream, and this stream ID is used to perform various operations upon this stream, most commonly via `output.Write(4, "This is some output")` where the first parameter is the stream ID and the second is the output you wish to write to the stream.

For more information about what can be done with the Output Prompt through Automation, consult the members and descriptions of the "output" alias from the Object Browser.

### Output Pane Toolbar

The toolbar at the top of the Output Pane has a few simple tasks available:

| Item | Description |
|---|---|
| Output Selector | Selects the active output stream which is displayed below. Initial options are "CoPilot" and "Python" though more can be added. |
| Save  | Allows you to save the active output stream to an external text file. |
| Print  | Shows the Print Preview Dialog and allows the output to be printed |
| Clear All  | Clears the contents of the active output stream. |
| Toggle Word Wrap  | Toggles word wrap for all output streams on and off. Turning on word wrap will automatically wrap text to the next line; otherwise the line will continue past the edge of the window and needs to be scrolled horizontally to see the rest of it. |

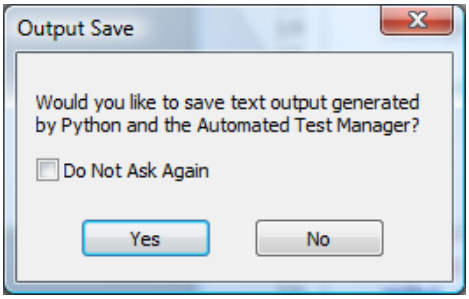
### Saving Contents

The Output Pane has several provisions for saving the contents to a text file. The toolbar has a **Save**  toolbar button, as well as the command `ExportToFile` accessible from Python code. Both of these commands allow the user to manually or programmatically save the contents of the file.

Additionally, if new text has been detected in either the "Python" or "ATE" output stream, CoPilot will ask you if you wish to save the new output through a dialog:



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*Prompt to save output text*

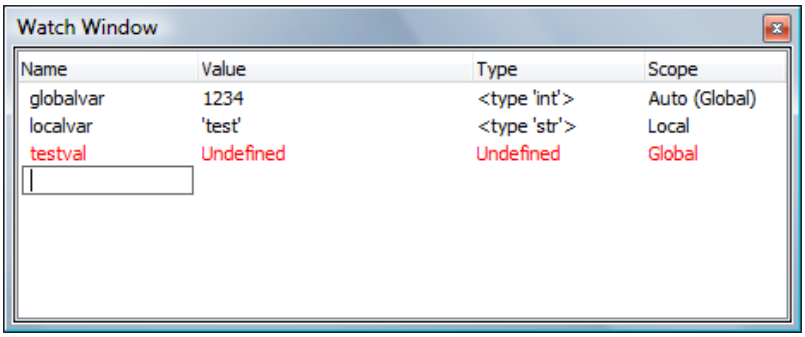
Selecting "Yes" will show save dialogs for each stream that has new content. Selecting "No" will not prompt you to save any content. By checking the box, you can prevent CoPilot from showing this dialog in the future.

# Watch Window Pane

## *Watch Window Pane Overview*

The Watch Window allows the user to track variables defined in Python as well as properties of aliases. Like a traditional watch window common in many IDEs, it can be used side by side with a debugger in order to track variables and modify their values on the fly. However, due to the scripting nature of Python, it may also be used without the debugger. This can be useful for tracking and modifying values as the project runs.

## *Details*




*An Example of entering a new value in the Watch Window Pane*

The Watch Window consists of a list of user-entered variable names. New values are added by double-clicking the empty line below the last item or through the context menu "Add New Watch Item." Once the item is added, it is evaluated to see if the variable is defined in the given scope (which is Auto by default). The scopes are defined as follows:

- **Auto** – This is the default option. The variable is looked for first in the built-in dictionary (which contains all the alias definitions), then the global dictionary (variables defined at the Command Prompt), then the local module dictionary (variables defined within Python modules, only visible when debugging).
- **Global –** Searches the built-in dictionary, and then the global dictionary. Does not search local module dictionaries. Adding the



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

module name before the variable name in the form "module.variable" will show variables defined globally within a particular module.

- **Local –** Searches only the local module dictionaries. Used while debugging to watch and modify locally defined values within module functions.

The following table is a list of the various commands available in the Watch Window's context menu:

| Item | Description |
|---|---|
| Scope Auto | Set the selected item's scope to "Auto" |
| Scope Global | Set the selected item's scope to "Global" |
| Scope Local | Set the selected item's scope to "Local" |
| Delete | Deletes the selected item |
| Delete All | Deletes all items |
| Add New Watch Item | Show a prompt that allows entry of a new watch item into the list |

The values displayed (and their associated types) are refreshed about twice a second. This grants additional flexibility over traditional "watch windows" because it allows variables to be tracked even when not actively debugging. When not actively debugging, watch variables defined as "Local" will be grayed out as they are inaccessible in the current mode. Undefined variables will appear in red.

Many variables may be modified on demand. Simply double-clicking on the "Value" column of the desired watch item will allow the value to be edited, if possible. The new value is immediately applied upon exiting the edit of the value. Note that the type of the value may be changed as well, as Python is a dynamically-typed language (e.g. an integer with the value of '1' may become a string with the value "one").

There are a few shortcuts in other areas of CoPilot for adding values to the Watch Window. In the Object Browser, an alias property may be right-clicked and selecting "Add to Watch" will add that alias/property combination to the Watch Window. Also, from the Python Editor, right-clicking on a word (variable name) and selecting "Add to Watch" will add that item to the Watch Window.

## CoPilot Macros

Like many of Microsoft's Office Programs, CoPilot allows you to create macros to accomplish some tasks automatically. Macros are just short Python functions with no parameters that perform a custom action upon CoPilot object(s). Examples of macros include configuring multiple data value, performing data analysis, or asynchronously transmitting a group of messages. Macros are used to combine multiple actions into a single step (like pushing a macro button on the macro toolbar).

Creating a macro is easy. In a new or existing Python module in the current project, add a new function definition:

```
def MyMacroFunction():
    print "This is my macro function."
```

This macro is now available for use within the Macro Dialog or the Macro Toolbar, described in the following sections. It will simply print the string indicated. Changing the action of the function is as simple as changing its code.

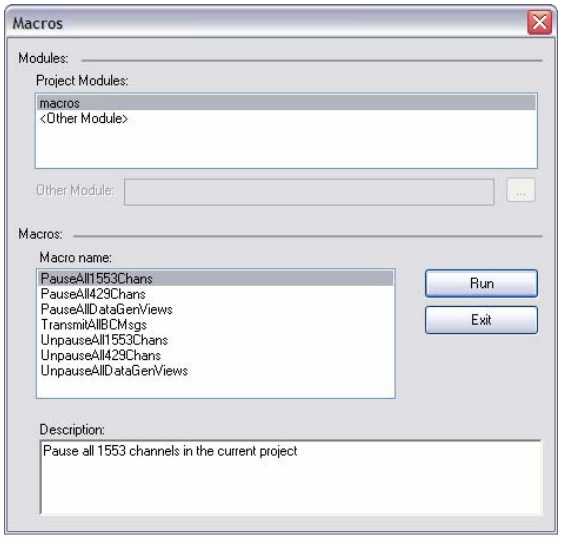


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

One thing to note is that in order for the Macro Dialog and Macro Toolbar to find the macro properly, the module must be imported by CoPilot. To do this, simply save the module, or use the context menu and select “Import Module.”

## Running Macros with the Macro Dialog

To access the Macro Dialog, select Project | Macros | Run Macro (version 5) or the **Macros** button on the **Scripting** ribbon tab. The following dialog will appear:




*Macro dialog*

The Macro Dialog consists of three areas. The first area allows you to select the Python module that contains the macro you wish to run. These modules are the modules that are defined in the current project and imported. It is also possible to select the “<Other Module>” list item which allows you to browse for an external module (any “.py” file) in another location.

Once the module has been selected, the actual macro must be chosen. In the second list, all of the functions with no parameters in the selected module are displayed. Selecting one of these will show a description string in the third and final area, which is generated from the function’s docstring.

The Macro Dialog is modeless, so it can be run multiple times and the effects of the macro are immediately visible in CoPilot. The Macro Dialog remains until the “Exit” button is pushed. Any errors that occur should present a message box with the error description. If the macro code itself generates errors or output, it will be displayed in the Output Pane.

## Accessing Macros to the Macro Toolbar (Version 5)

Another way of accessing macros in CoPilot 5 is through the Macro Toolbar. To show or hide the Macro Toolbar, right-click in the toolbar area and check/uncheck the “Macro Toolbar” item. The toolbar has two buttons by default, shown below.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

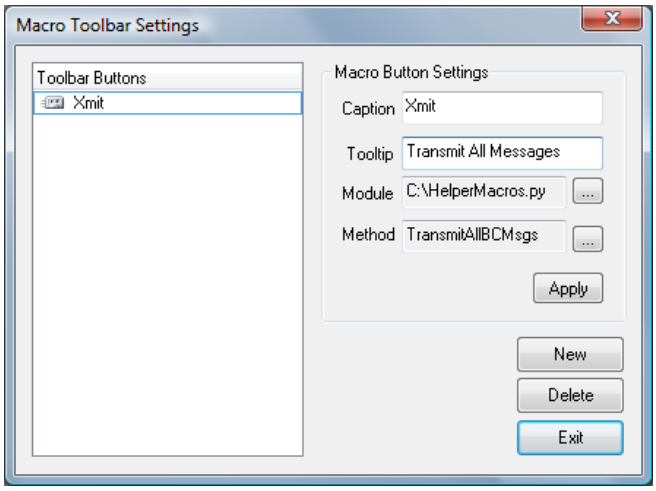



*The macro toolbar*

The Macro Dialog button will show the Macro Dialog, described in the previous section. Pressing the second button ("Toolbar Settings") will open the Toolbar Settings Dialog.

This dialog allows the creation of new buttons in the Macro Toolbar. To begin creating a new button, press the **New** button in the dialog. Now start by filling in a caption (What will appear for the button in the toolbar) and the tooltip. Next, select a module. This will bring up a file browser, where you can select a Python module from anywhere in the file system. Finally, choose the method within that module that you would like to run when the macro is called. The methods that show up here are all of the methods defined in the module with 0 parameters.

After completing the configuration of the macro button, press **Apply** to update the list on the left. You should see something similar to below:




*Macro toolbar settings*

Once added, the toolbar should now have the new item in it, as shown below:




Pressing the new item button (i.e. the 'Xmit' button shown above) in the toolbar will run the linked macro.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Project Security

## Overview

The CoPilot Project Security settings control user runtime permissions. These settings allow or restrict the ability of a user to perform various tasks. Setting a project password prevents other users from modifying the security settings and the restricted areas of the project.

## Configuring Project Security

CoPilot Professional projects can be locked for runtime operation or unlocked to allow unrestricted access. A password can be set for each project to prevent unauthorized configuration and data changes depending which project security settings are enabled. To change a project's security, first open the desired project, then select 'Security Settings' from the Project menu item.

### Using Project Security

The Project Access setting allows a project to be configured as either an unrestricted project or a runtime (locked) project with limited functionality. Unrestricted projects do not use any security settings.

#### *Unrestricted Project Access*

Unrestricted projects allow all available features without any limitations. This is the default security setting for a project. CoPilot Standard projects are always unrestricted; a CoPilot Professional license is required to use project security settings.

#### *Runtime Project Access*

A project configured for runtime access prevents tasks that have been disallowed. The available project security permissions are described in the *Project Security Options* section on page *432*.

### Project Passwords

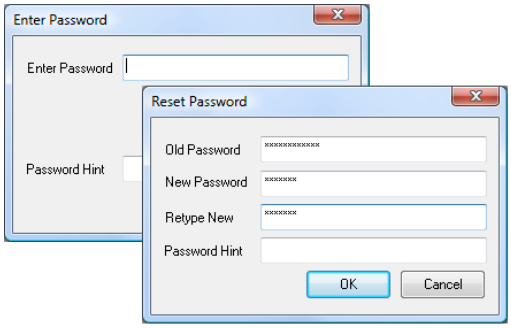
Users must log in to make project permission changes. The password for a project is set by clicking the 'Log In' button on the Project Security tab of the



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CoPilot Options. Passwords may be after being set by clicking the ‘Change Password’ button.

Note: Be sure to keep your project passwords in a safe place so they are not lost. The password hint serves as a reminder in case the password is forgotten.




*Setting a project password*

# Project Security Options

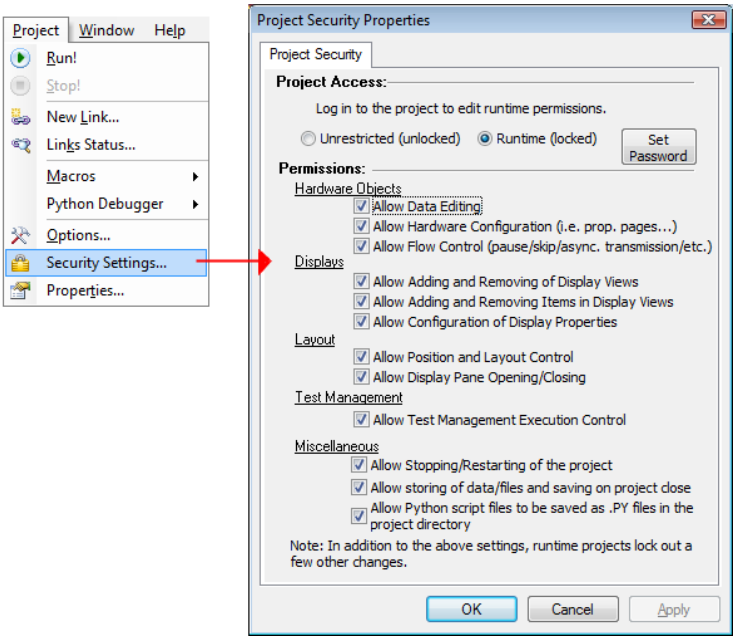
Projects with security settings enabled set with runtime access can limit permissions to specific features in CoPilot. These categories include:

- Hardware Objects – control of hardware device configuration and data
- Displays – Control of display view
- Layout – opening/closing display panes and views
- Test Management – control of Test Manager execution
- Miscellaneous – various other settings

The specific items available are shown in the Project Security property page. To configure a project’s security settings open the desired project, then select ‘Security Settings’ from the Project menu as shown in the following image.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*Project Security Settings property page*

Note: When the ATE Python files are saved as part of the project file and not as .PY files in the project directory, the .PY files are not created and the script code cannot be viewed from outside of CoPilot (i.e. with Windows Explorer).

## Securing Proprietary Python Code

Users may have proprietary code in Python modules to allow only certain users to have access. Combining a few of the security permissions can prevent unauthorized viewing of Python modules.

First, minimize any script files you wish to prevent the user from accessing. Next, place the project into the runtime (locked) state and set a project password. Then uncheck the option to 'Allow Python script files to be saved as .PY files in the project directory' so that the contents of the Python module are only accessible from within CoPilot. Finally, uncheck the 'Allow Position and Layout Control' to prevent users from showing hidden views.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Appendix A: CoPilot Standard and Professional Differences

CoPilot is available in two versions: CoPilot Standard and CoPilot Professional. A description of the CoPilot Professional features are first listed. Then, the table following provides a brief comparing some of the features available with CoPilot Standard to CoPilot Professional.

| CoPilot Professional- Features | |
|---|---|
| Cockpit View | Simulated aircraft cockpit with aircraft dynamics |
| Control View | Virtual controls, gauges, knobs and simulated aircraft instruments |
| Quick View | Special library of controls that quick-launch directly from a data field |
| Strip View | Strip charts of historical data with statistical summaries |
| Data Plot | Charts all data points of selected fields (in Monitor View) |
| Map View | Real-time moving map display with TerraServer maps engine |
| Automated Test Environment (ATE) | ATE allows the implementation of test scripts using the CoPilot object model, and airs in building entire test suites for verification or validation of systems and other uses |
| Test Manager | Sequence through tests and generate test results. The Test Manager reports the current and past test conditions with the ability to dynamically skip or include tests… |
| Python Shell-Level Scripting | Python shell-level scripting builds upon the functionality of the VB Script View by allowing the control of the CoPilot application to start, stop, and open projects, and manage tests via script routines. |
| Command Line Interface | Powerful command line interface to execute Python statements, run other scripts, and interact with objects at runtime |
| Script Debugger | Python script debugger with breakpoints and code stepping |
| Project Security | Project security settings to limit user access to specific features |
| VB Scripting | Scripting engine using VBScript routines to control and customize CoPilot with the VBScript View – See *Appendix H: Legacy CoPilot Support* |
| List Buffering | List of data values where the transmission of the next value is automatically advanced by the hardware to ensure data integrity |
| 429 Custom Schedule | Custom user scheduling where the exact order of messages, inter-message spacing, gaps, and gap errors can be created from scratch or seeded using the message transmit interval settings |

ATE related features



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following table compares features available with CoPilot Standard to CoPilot Professional.

| Feature | Standard | Professional |
|---|---|---|
| “Project” Backstage (Version 6 only) | X | X |
| Start Page (Version 5 only) | X | X |
| Theme Configuration | X | X |
| Configurable Toolbars (Version 5 only) | X | X |
| Activity View (Version 6 only) | X | X |
| Protocol Browser (Version 5 only) | X | X |
| Quick Tasks | X | X |
| Multi-Protocol Support | X | X |
| 1553, 429, AFDX/664, 708, Serial Support | X | X |
| Transmit/Receive | X | X |
| Time Tagging (IRIG) | X | X |
| Engineering Units & Database | X | X |
| Data Logging with Filtering | X | X |
| Multiple Schedules (1553, 429) | X | X |
| Rate-Based Schedules (1553, 429, AFDX) | X | X |
| Statistics | X | X |
| Error Injection | X | X |
| Multi-Device Support | X | X |
| Display Filtering | X | X |
| Advanced Data Views | X | X |
| Import/Export | X | X |
| Data Generators | X | X |
| Software Playback | X | X |
| Hardware Playback | X | X |
| Control View with Control Gallery | | X |
| Quick View | | X |
| Strip View | | X |
| Data Plot | | X |
| Map View | | X |
| Automated Test Environment (ATE) | | X |
| Automated Test Manager | | X |
| Python Shell-Level Scripting | | X |
| Python Debugger | | X |
| Command Prompt | | X |
| Watch Window | | X |
| Custom User Macros | | X |
| VB Scripting | | X |
| Project Security | | X |
| List Buffering | | X |
| 429 Custom Scheduling | | X |



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Appendix B: Hardware Object Icon Reference

## Object Icons

Icons in the following table will not appear by themselves. Instead, these are known as overlays, which means that they will appear on top of an existing icon. Generally, only one overlay is displayed at a time, with a few exceptions.

| State Overlays | | | | | | | |
|---|---|---|---|---|---|---|---|
| Overlay | State | Overlay | State | Overlay | State | Overlay | State |
|  | Software Playback |  | Problem |  | Monitor Enabled |  | Configuration Warning |
|  | Hardware Playback |  | Pause |  | Error Injection Enabled |  | Default Schedule |
|  | Custom Schedule |  | Free Running Schedule |  | Empty Custom Schedule |  | Record All Data |

The following icons appear across various protocols and card types.

| Common Icons | | | | |
|---|---|---|---|---|
| Object | ICON | State | ICON | State |
| **Core** |  | Normal |  | Disabled |
| **Sequential Monitor** |  | Normal |  | Disabled |

Icons in the following table represent the various hardware supported in the CoPilot environment.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| Hardware Icons | | | | | | |
|---|---|---|---|---|---|---|
| **Platform** | **ICON** | **Description** | **ICON** | **Description** | **ICON** | **Description** |
| **Ethernet** |  | Avionics BusBox | | | | |
| **Ethernet/ UBS** |  | OmniBusBox | | | | |
| **USB** |  | USB 1553<br>USB 429 |  | *BUSBox (v 5 only)* | | |
| **PCI** |  | LP1553-5<br>LP429-5<br>OmniBus PCI |  | LP1553-3 *(v 5 only)*<br>LP429-3 *(v 5 only)*<br>LP708-1 *(v 5 only)* |  | LP-AFDX-1 |
| **PCIe** |  | LE1553-5<br>LE429-5 | | | | |
| **PCMCIA** |  | CM1553-3 *(v 5 only)*<br>CM429-2 *(v 5 only)* | | | | |
| **cPCI** |  | OmniBus cPCI |  | LC1553-3 *(v 5 only)*<br>LC429-3 *(v 5 only)*<br>LC708-1 *(v 5 only)* |  | LC-AFDX-1 |
| **PMC** |  | OmniBus PMC |  | LM-AFDX-1 | | |
| **VME** |  | OmniBus VME | | | | |
| **ISA** |  | PC1553-3 *(v 5 only)*<br>PC708-1 *(v 5 only)* | | | | |
| **NIC** |  | AFDX NIC | | | | |
| **DEMO** |  | 1553 Demonstration Hardware<br>429 Demonstration Hardware |  | AFDX Demonstration Hardware | | |

These icons represent the various MIL-STD-1553 protocol objects.

| MIL-STD-1553 Icons | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Object** | **ICON** | **State** | **ICON** | **State** | **ICON** | **State** | **ICON** | **State** |
| **Channel** |  | Normal |  | Disabled | | | | |
| **Bus Controller** |  | Normal |  | Disabled | | | | |
| **Receive BC Messages** |  | Normal |  | Disabled |  | Timed out |  | Single Shot Enabled |
| **Transmit BC Message** |  | Normal |  | Disabled |  | Timed out |  | Single Shot Enabled |



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| | | | | |
|---|---|---|---|---|
| **RT->RT Message** |  Normal |  Disabled |  Timed out |  Single Shot Enabled |
| **RT (Simulated)** |  Normal |  Disabled |  Timed out | |
| **RT (External)** |  Normal |  Disabled |  Timed out | |
| **Receive SA** |  Normal |  Disabled |  Timed out | |
| **Transmit SA** |  Normal |  Disabled |  Timed out | |
| **Bus Monitor** |  Normal |  Disabled | | |
| **Field** |  |  |  Timed out | |

These icons represent the various ARINC 429 protocol objects.

| ARINC 429 Icons | | | | | | |
|---|---|---|---|---|---|---|
| **Object** | **ICON** | **State** | **ICON** | **State** | **ICON** | **State** |
| **Channel (EquipID not set)** |  | Normal |  | Disabled | | |
| **Channel (EquipID set)** |  | Normal |  | Disabled | | |
| **Message** |  | Normal |  | Disabled |  | SDI Enabled |
| **Field** |  | Normal |  | Disabled | | |

These icons represent the various AFDX/ARINC 664 protocol objects.

| AFDX Icons | | | | | | |
|---|---|---|---|---|---|---|
| **Object** | **ICON** | **State** | **ICON** | **State** | **ICON** | **State** |
| **Network (Redundant)** |  | Normal |  | Disabled | | |
| **Network (Independent)** |  | Normal |  | Disabled | | |
| **VL (Transmit)** |  | Normal |  | Disabled | | |
| **VL (Receive)** |  | Normal |  | Disabled | | |
| **Port (Sampling)** |  | Normal |  | Disabled | | |



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| | | | | | | |
|---|---|---|---|---|---|---|
| **Port (Queuing)** |  | Normal |  | Disabled | | |
| **FDS** |  | Normal |  | Disabled | | |
| **DS** |  | Normal |  | Disabled | | |
| **Field** |  | Normal |  | Disabled |  | Defined outside the port |

These icons represent the various ARINC 708 protocol objects.

| ARINC 708 Icons | | | | |
|---|---|---|---|---|
| **Object** | **ICON** | **State** | **ICON** | **State** |
| **Dataset** |  | Normal |  | Disabled |

These icons represent the various Serial (RS422) protocol objects.

| Serial (RS422) Icons | | | | |
|---|---|---|---|---|
| **Object** | **ICON** | **State** | **ICON** | **State** |
| **Serial Port** |  | Normal |  | Disabled |



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Appendix C: Supported Hardware and Protocols

CoPilot can operate most of Ballard Technology's MIL-STD-1553, ARINC 664/AFDX, ARINC 708, serial and discrete interface hardware. Please contact customer support if you device is not listed below to find out if CoPilot supports your particular interface card.

| CoPilot Supported Hardware | Generation | Capable Protocols |
|---|---|---|
| Box Products | | |
| (Ethernet, USB, Serial) | | |
| Avionics BusBox: AB1000 | 5G | 1553, 429, 664/AFDX, 708, discretes, serial, others |
| Avionics BusBox: AB2000 | 5G | 1553, 429, 664/AFDX, 708, discretes, serial, others |
| OmniBusBox* | 4G | 1553, 429, 664/AFDX, 708, discretes, serial, others |
| BUSBox *(v 5 only)* | 3G | 1553, 429, 708 |
| USB 1553 | 5G | 1553, discretes |
| USB 429 | 5G | 429, discretes |
| PCI Products | | |
| OmniBus PCI | 4G | 1553, 429, 664/AFDX, 708, discretes, serial, others |
| LP1553-5 | 5G | 1553, discrete |
| LP1553-3 *(v 5 only)* | 3G | 1553 |
| LP429-5 | 5G | 429, discrete |
| LP429-3 *(v 5 only)* | 3G | 429 |
| LP708-1 *(v 5 only)* | 3G | 708 |
| LP-AFDX-1 | NA | 664/AFDX |
| PCIe Products | | |
| OmniBus PCIe | 4G | 1553, 429, 664/AFDX, 708, discretes, serial, others |
| LE1553-5 | 5G | 1553, discrete |
| LE429-5 | 5G | 429, discrete |
| cPCI Products | | |
| OmniBus cPCI | 4G | 1553, 429, 664/AFDX, 708, discretes, serial, others |
| LC1553-3 *(v 5 only)* | 3G | 1553 |
| LC429-3 *(v 5 only)* | 3G | 429 |
| LC708-1 *(v 5 only)* | 3G | 708 |
| LC-AFDX-1 | NA | 664/AFDX |
| VME Products | | |
| OmniBus VME | 4G | 1553, 429, 664/AFDX, 708, discretes, serial, others |
| PMC Products | | |
| OmniBus PMC | 4G | 1553, 429, 664/AFDX, 708, discretes, serial, others |
| LM-AFDX-1 | NA | 1553, 429, 664/AFDX, 708, discretes, serial, others |
| PCMCIA Products | | |
| CM1553-3 *(v 5 only)* | 3G | 1553 |
| CM429-1 *(v 5 only)* | 2G | 429 |
| PC/ISA Products | | |
| PC1553-3 *(v 5 only)* | 3G | 1553 |
| PC708-1 *(v 5 only)* | 3G | 708 |

**For optimal performance with the OmniBusBox, an Ethernet connection is preffered.*



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CoPilot for 1553 provides multiple levels of hardware support for its various interface cards, as described below.

### Levels of MIL-STD-1553 Capability

CoPilot 1553 operates in conjunction with Ballard's 5G (PCI, PCIe, Ethernet, USB), 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 hardware interface devices can be ordered with varying levels of MIL-STD-1553 bus capabilities. These levels have been referred to as models in some of our previous hardware products. As an example, the PCMCIA board could be ordered as a CM1553-3B4 or a CM1553-3C to correspond to level B4 or level C functionality. The number of simultaneous terminals and additional hardware features of the MIL-STD-1553 levels are summarized in the table below.

| | Level A | Level B4 | Level B32 | Level C | Level D | Single (S) Function | Multi (M) Function |
|---|---|---|---|---|---|---|---|
| Bus Controller | ✓ | ✓ | ✓ | ✓ | ✓ | ✓++ | ✓ |
| Simultaneous Terminals | 1 | 4 | 32 | 32 | 32 | 32++ | 32 |
| Monitor | ✓ | ✓ | ✓ | ✓ | ✓ | ✓++ | ✓ |
| Filtering for Terminal | ✓ | ✓ | ✓ | ✓ | ✓ | ✓++ | ✓ |
| Filtering for Subaddress | | ✓ | ✓ | ✓ | ✓ | ✓++ | ✓ |
| Concurrent Terminal | | | | ✓ | ✓ | | ✓ |
| Protocol Error Injection | | | | ✓ | ✓ | | ✓ |
| Variable Transmit | | | | | ✓ | | |
| Zero Crossing Distortion | | | | | ✓ | | |

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

Note: Level D 1553 databuses are only available with select 4G/5G products; consult your hardware manual for supported devices.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Appendix D: Sample Projects

CoPilot ships with a variety of sample projects, showcasing some of the more advanced features CoPilot supports.

The sample projects can be found at "C:\UserName\My Documents\My CoPilot Projects\Samples\Projects." These examples are copied to your My Documents folder on the first run of CoPilot, and are therefore tracked separately for each Windows User, each user managing a separate version of the files.

The following is a list and description of the various available sample projects and the versions of CoPilot they are available in:

- **4-StateButton** – Uses VB scripting to interact with a button on a Control View. Pressing the button cycles through four different states (version 5 and 6).
- **429 Bit Walking** – Uses VB scripting to walk a single bit through the ARINC 429 data. Illustrates timer events and programming states. This example can be applied to other protocols besides ARINC 429 (version 5 only).
- **AirDynamicsSim (Python)** – Uses Python scripting in CoPilot ATE to control autopilot inputs of an Aircraft View. The current altitude, longitude, latitude, roll, pitch, etc. respond accordingly (version 5 and 6).
- **AirDynamicsSim (VBScript)** – Uses VB scripting to control autopilot inputs of an Aircraft View. The current altitude, longitude, latitude, roll, pitch, etc. respond accordingly (version 5 and 6).
- **BB1720** – ARINC 708 & 429, Same as WeatherRadarSystem except only one BUSBox-1720 board (version 5 only).
- **Conversion429To1553_ABB –** Converts 429 protocol data to 1553 data utilizing a multi-protocol Avionics BusBox.
- **Conversion429To1553_Demo –** Converts 429 protocol data to 1553 data utilizing two demo cards (429 and 1553).
- **Discrete Example** – Uses Python scripting to control an OmniBus Discrete Module And display the state of 8 discretes in a bank with an LED Control (version 5 and 6).
- **ExampleTimeScript** – ARINC 429, Uses VB scripting to convert the time of day (Now) into two differently formatted labels (version 5 only).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **ISOAlphaRx** – ARINC 429, Uses VB scripting to receive an ISO Alphabet No. 5 string (version 5 and 6).
- **ISOAlphaTx** – ARINC 429, Uses VB scripting to transmit an ISO Alphabet No. 5 string (version 5 and 6).
- **Output429MonData** – ARINC 429, Uses VB scripting loop through recorded monitor data in the Sequential Monitor Object (in the Hardware Explorer) (version 5 and 6).
- **WeatherRadarSystem** – ARINC 708 & 429, Contains LP708-1 and LP429-3 boards. Has a Control Panel with links hooked to the 2 429 control words for ARINC 708. Also has a 429-view to verify correct transmission values of 429 fields (version 5 and 6 -- version 6 utilizes two USB devices)
- **WilliamsburgSink** – ARINC 429, Uses VB scripting to simulate a sink during a bit-oriented file transfer using Williamsburg File Transfer Protocol (version 5 and 6).
- **WilliamsburgSource** – ARINC 429, Uses VB scripting to simulate a source during a bit-oriented file transfer using Williamsburg File Transfer Protocol (version 5 and 6).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Appendix E: Control View Library

## Controls Library Overview

### Controls Hosted in Control View Window

ActiveX controls used with CoPilot Professional are objects used to translate and display information. Active X controls are generally graphical; however, controls may perform tasks like modifying data or timing events without any graphical display. They are hosted in the CoPilot Control View container and linked to fields based on a drag and drop assignment or through a Project | Link operation. Controls can be adapted and customized through the control property window. Other controls installed and registered with the system may also be used by CoPilot.

### Included Controls

Many ActiveX control components are provided as part of the CoPilot Professional package. Purchasing an additional license to use these 3rd party controls is not required to use the controls within CoPilot. Although controls can be instantiated through a drag and drop operation, users can also select and modify controls in Design Mode. Available controls can be accessed through the Controls Palette as shown in the image below. Moving the mouse over the controls in the controls palette shows the tooltip with the control name.




The list of controls included with CoPilot Professional is continually expanding. This appendix lists some of the included controls and provides additional details



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

for some of these included controls. Refer to following sections in this appendix and to the help documentation listed below for additional information about the included controls. Refer to the documentation supplied with any user-added controls.

| Control | Help Documentation |
|---|---|
| GMS Aircraft Instruments | Air.hlp |
| GMS Alphanumeric LED | Numled.hlp |
| GMS Angular Gauge | Agauge.hlp |
| GMS Knob | Knob.hlp |
| GMS LED | Led.hlps |
| GMS Linear Gauge | LGauge.hlp |
| GMS Percent | Percent.hlp |
| GMS Selector Knob | Select.hlp |
| GMS Slider | Slide.hlp |
| GMS Strip Chart | Strip.hlp |
| GMS Toggle | Toggle.hlp |
| BT Data Modifier | CopXCtrls.chm |
| BT Binary Editor | CopXCtrls.chm |
| BT Background Text | CopXCtrls.chm |
| BT ConvertDoubles | CopXCtrls.chm |
| BT Label | CopXCtrls.chm |
| BT Picture | CopXCtrls.chm |
| BT Timer | CopXCtrls.chm |
| BT Msg/Input Box | CopXCtrls.chm |
| Xtreme Date Picker | Xtream Controls.chm |
| Xtreme Calendar | Xtream Controls.chm |
| Xtreme CheckBox | Xtream Controls.chm |
| Xtreme ColorPicker | Xtream Controls.chm |
| Xtreme ComboBox | Xtream Controls.chm |
| Xtreme CommonDialog | Xtream Controls.chm |
| Xtreme DateTimePicker | Xtream Controls.chm |
| Xtreme FlatEdit | Xtream Controls.chm |
| Xtreme GroupBox | Xtream Controls.chm |



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

| | | |
|---|---|---|
|  | Xtreme HexEdit | Xtream Controls.chm |
|  | Xtreme Label | Xtream Controls.chm |
|  | Xtreme ListBox | Xtream Controls.chm |
|  | Xtreme ListView | Xtream Controls.chm |
|  | Xtreme Month Calendar | Xtream Controls.chm |
|  | Xtreme ProgressBar | Xtream Controls.chm |
|  | Xtreme PushButton | Xtream Controls.chm |
|  | Xtreme RadioButton | Xtream Controls.chm |
|  | Xtreme ScrollBar | Xtream Controls.chm |
|  | Xtreme Slider | Xtream Controls.chm |
|  | Xtreme SyntaxEdit | Xtream Controls.chm |
|  | Xtreme Tab | Xtream Controls.chm |
|  | Xtreme TreeView | Xtream Controls.chm |
|  | Xtreme WebBrowser | Xtream Controls.chm |

Later in this appendix, several of the controls, including aircraft instruments, strip chart, and others, are described in detail.

Note: Several additional controls are included with CoPilot Professional. See documentation directory on the CoPilot CD for the CopXCtrls.chm file describing additional Ballard Technology ActiveX controls and 'Xtream Controls.chm' describing the included Xtream Suite ActiveX controls.

# Working with Controls

### Naming a Control

When ActiveX components are selected from the Controls Palette and assigned to a Control window, CoPilot automatically assigns a name to the control. That unique name is used to identify the control when it is linked to a field. The default name assigned is "Controln" where n is the number of controls that have been created in the project. While that simplifies construction, if a large number of controls are used, the designer may want to specify a unique name.

To change the descriptive name of a control:

1. Right click on the control to access the context menu for that control
2. Choose **Edit Name** from the context menu
3. Type the desired name for the control and click **OK**

### Editing Properties

All controls are defined through properties.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

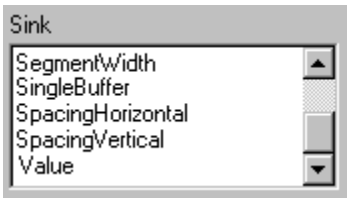
To access the properties of a control:

1. Select **Control | Design Mode** or right click on the control window and select **Design Mode**. (Notice the presence of dots when in Design Mode.)
2. Right click on an existing control and select **Properties**.

Properties can be separated into two classes: Fundamental properties and cosmetic or optional properties. In most cases, controls provide immediate feedback so you can see the results of property choices in the control display as you change values. For that reason, it is not necessary to describe cosmetic attributes in the property dialog box. Try them and observe the effects of various properties on the control.

### Creating Links Manually

The Control View ActiveX controls can be used as a data source or to display field data. When linking controls through the Project |Link operation, the data source will typically be a "ValueEngr" attribute of a receive field in the Hardware Explorer. The corresponding value property is identified in the Control Property table and the discussion of individual ActiveX components. Primary control properties are listed in the Control Property table.



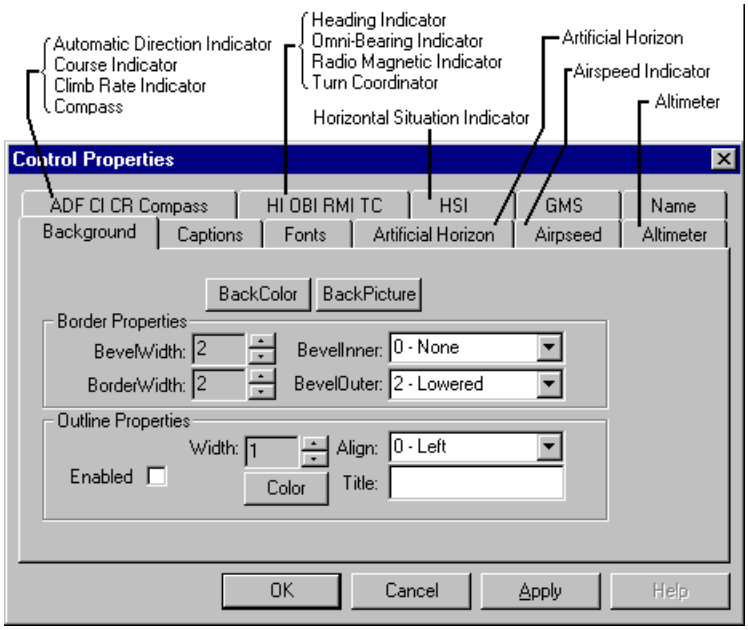

# Aircraft Instruments

### A Family of Aircraft Instruments

The Aircraft Instruments control is a composite control library of twelve frequently used flight instruments. Selection involves a two-step process: first, selecting the Aircraft Instrument button from the Controls Palette, then selecting an instrument through one of the Aircraft Instrument property tabs.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




Each of these Aircraft Instruments is described in the sections that follow.

## Aircraft Instrument Properties

### Value Properties

Each control is defined by one or more properties. In most cases the "engineering unit value," property of fields will be linked with control value properties shown in the table below.

**Frequently Used Control Properties**

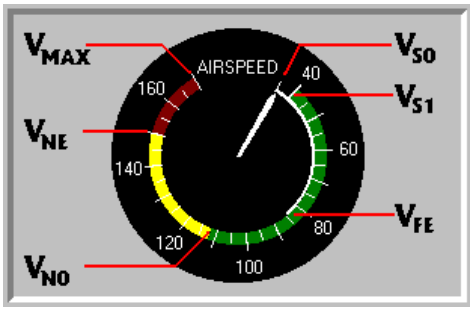
| *Property* | *Range* | *Property* | *Range* |
|---|---|---|---|
| **Airspeed Indicator** | | **Heading Indicator** | |
| Airspeed | 35 to 2000 | Heading | 0-360 |
| **Altimeter** | | **Horizontal Situation Indicator** | |
| Altitude | 0 to 32,767 | HSIBearing | 0 to 360 |
| AltBarometer | 20 to 40 | HSIBugHeading | 0 to 360 |
| | | HSICourseHeading | 0 to 360 |
| **Artificial Horizon** | | HSIHeading | 0 to 360 |
| AHRoll | 0 to 360 | | |
| AHPitch | 0 to 360 | **Numeric LED** | |
| AHHeading | 0 to 360 | Value | user defined |
| **Automatic Direction Finder** | | **Omni Bearing Indicator** | |
| ADFBearing | 0 to 360 | OBICourse | 0-360 |
| **Climb Rate Indicator** | | **Radio Magnetic Indicator** | |
| ClimbRate | + to -rate max | RMICompass | 0-360 |
| ClimbRateMax | 5000 | RMIBearing1 | 0-360 |
| | | RMIBearing 2 | 0-360 |
| **Compass** | | | |
| CompassHeading | 0 to 359 | **Strip Chart** | |
| | | Y (last property in menu) | user defined |
| **Course Indicator** | | | |
| CIActCourse | 0 to 360 | **Turn Coordinator** | |
| CIOrdCourse | 0 to 360 | TCTurn | -360 to 360 |
| | | TCInclinometer | -20 to 20 |



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Airspeed Indicator

The Airspeed Indicator control displays airspeed in context of aircraft flight characteristics. The airspeed indicator is found on the Airspeed tab.

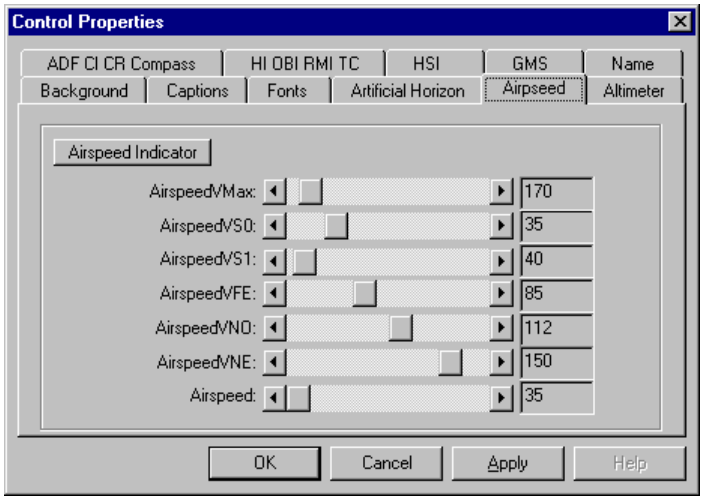



### Essential Properties

Data Value: The ValueEngr of a field is linked to Airspeed, the last variable or property on the property window below. Airspeed values can range between the limits defined by AirspeedVSO and AirspeedVMax.

### Optional Properties

Aircraft flight characteristics are defined through five airspeed properties.




- AirspeedVMax: Indicates maximum displayed speed.
- AirspeedVS0: Indicates stall speed with flaps down.
- AirspeedVS1: Indicates stall speed with flaps up.
- AirspeedVFE: Indicates max safe speed with flaps down (extended).
- AirspeedVNO: Indicates max structural or “not to exceed” speed.
- AirspeedVNE: Indicates never exceeds airspeed.



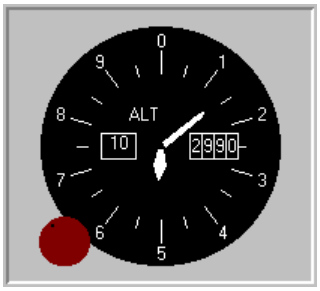
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Altimeter

The Altimeter indicator displays the altitude of the aircraft. In addition, it may display information on the barometric pressure settings.

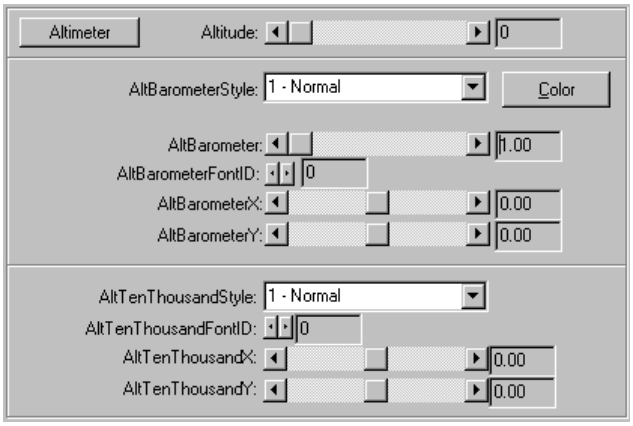
To create an Altimeter:

1. Select the **Aircraft Instruments** icon from the Palette and draw a rectangle in the Control View window
2. Right click on the control and choose **Properties**
3. Select the **Altimeter** tab and click the **Altimeter** button




### Essential Properties

Data Value: Altimeter fields are linked to the Altitude property. Barometric data is linked to the AltBarometer property.




### Experiment with Optional Properties

A variety of customizing options is available through the dialog box above. The easiest way to understand these options is to experiment and observe the effects on the Altimeter control.

## Artificial Horizon

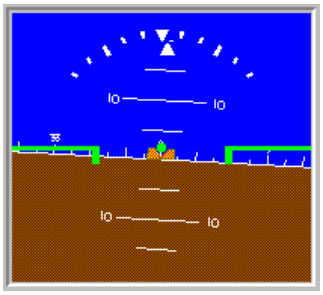
The Artificial Horizon graphically integrates the values of three data types: Roll (or bank), pitch, and heading.

To create an Artificial Horizon control:

1. Select the **Aircraft Instruments** icon from the Palette and draw a rectangle in the Control View window
2. Right click on the control and choose **Properties**
3. Select the Artificial Horizon tab and click the Artificial Horizon button

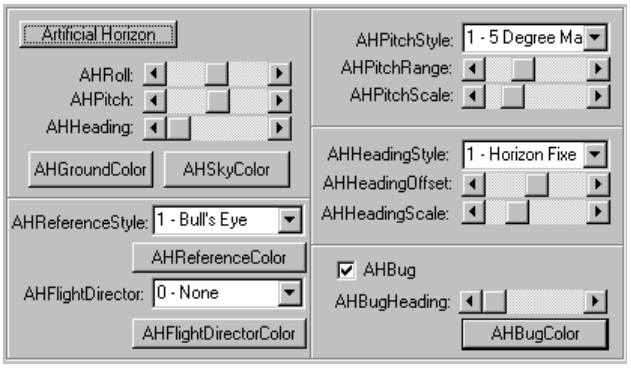


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



### Essential Properties

Data Value: The variable names of the three respective properties are shown on the property window: AHRoll, AHPitch, and AHHeading. These are connected through three separate link operations.




### Experiment with Optional Properties

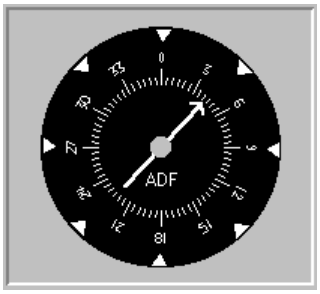
A variety of customizing options is available through the dialog box above. The easiest way to understand these options it to experiment and observe the effects on the Artificial Horizon control.

## Automatic Direction Finder (ADF)

The ADF control is used to display the direction of a homing beacon.

To create an Automatic Direction Finder control:

1. Select the Aircraft Instruments icon  from the Palette and draw a rectangle in the Control View window
2. Right click on the control and choose Properties
3. Select the **ADF CI CR Compass** tab and click the **ADF** button






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Essential Properties

Data Value: Incoming fields are linked to the ADFBearing property

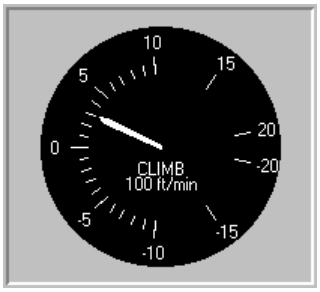



## Climb Rate Indicator (CRI)

The climb rate indicator displays aircraft rate of climb or descent in hundreds of feet per minute.

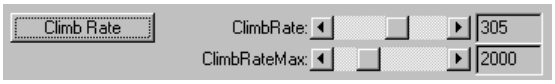
To create an Climb Rate Indicator control:

1. Select the **Aircraft Instruments** icon from the Palette and draw a rectangle in the Control View window
2. Right click on the control and choose **Properties**
3. Select the **ADF CI CR Compass** tab and click the **Climb Rate** button




### Essential Properties

Data Value: The Climb Rate control is defined by the ClimbRate property. The upper and lower rate is defined by the ClimbRateMax property.




## Compass

The Compass is a generic directional tool.

To create a compass control:

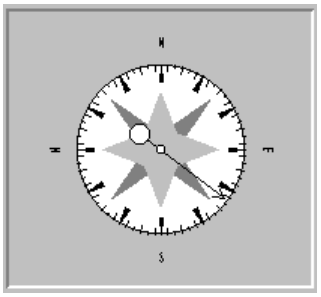
- Select the Aircraft Instruments icon from the Palette and draw a rectangle in the Control View window (the compass is the default control)

To reconfigure an Aircraft Instrument as the default compass:

1. Right click on the control and choose **Properties**
2. Select the **ADF CI CR Compass** tab and click the **Compass** button

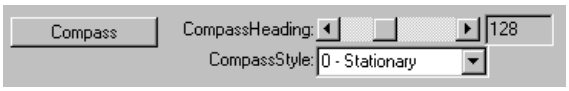


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



### Essential Properties

Data Value: Compass fields are linked to the CompassHeading property.



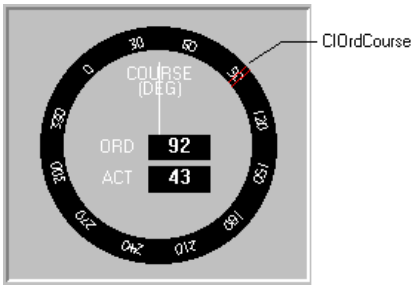

Optional Display: Compass style can be changed from Stationary to Floating through the CompassStyle property. When the floating style is selected, the needle points straight up and the background changes.

## Course Indicator

The Course Indicator is used to display the desired and actual aircraft course.

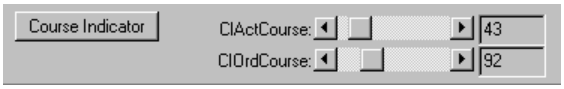
To create a Course Indicator control:

1. Select the **Aircraft Instruments** icon  from the Palette and draw a rectangle in the Control View window
2. Right click on the control and choose **Properties**
3. Select the **ADF CI CR Compass** tab and click the **Course Indicator** button




### Essential Properties

Data Value: Course Indicator fields are linked to the CIActCourse and CIOrdCourse.




CIActCourse defines the rotation of the outside circle. CIOrdCourse position the placement of the red marker on that circle.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Heading Indicator

The Heading Indicator is a simple form of aircraft heading display.

To create a Heading Indicator control:

1. Select the **Aircraft Instruments** icon from the Palette and draw a rectangle in the Control View window
2. Right click on the control and choose **Properties**
3. Select the **HI OBI RMI TC** tab and click the **Heading Indicator** button




#### Essential Properties

Data Value: The ValueEngr property of a field is linked to the Heading Indicator property of the Heading Indicator.

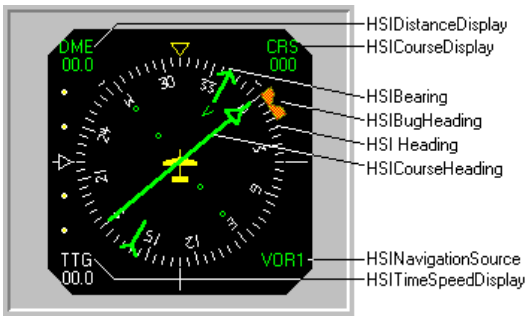



## Horizontal Situation Indicator (HSI)

The HSI control is a flexible control that integrates directional information from a variety of sources. The source values are also noted on the four corners of the HSI display (see figure below).

To create a Horizontal Situation Indicator control:

1. Select the **Aircraft Instruments** icon from the Palette and draw a rectangle in the Control View window
2. Right click on the control and choose **Properties**
3. Select the **HSI** tab and click the **HSI** button



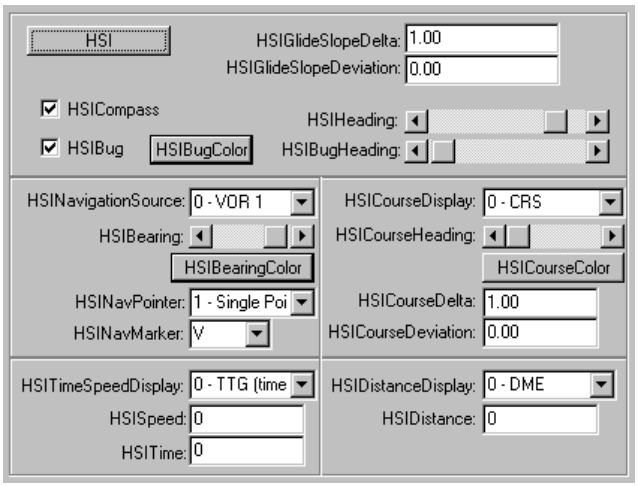



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Essential Properties

Data Value: Several of the properties noted on the properties can be defined through incoming fields, including:

- HSIBearing, represented by the open arrow,
- HSIBugHeading,
- HSICourseHeading, represented by the closed arrow and
- HSIHeading controls the rotation of the outer ring.




### Experiment with Optional Properties

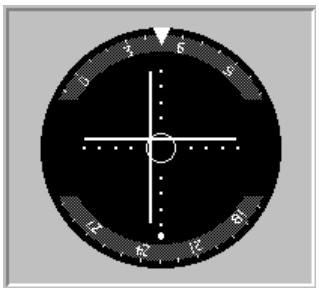
In addition to the four properties noted above, the HSI is defined through eighteen other display and data properties. Some of these can be linked to incoming 1553 fields; others will be defined by the user through the property window.

## Omni Bearing Indicator (OBI)

The Omni Bearing Indicator is a Radio Navigational aid that indicates deviation from the desired course.

To create a Omni Bearing Indicator control:

1. Select the **Aircraft Instruments** icon  from the Palette and draw a rectangle in the Control View window
2. Right click on the control and choose **Properties**
3. Select the **HI OBI RMI TC** tab and click the **OBI** button





Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Essential Properties

Data Value: Incoming fields are linked to the OBICourse property.

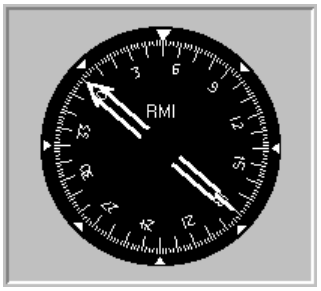



## Radio Magnetic Indicator (RMI)

The Radio Magnetic Indicator displays direction to a radio beacon against the background of aircraft heading.

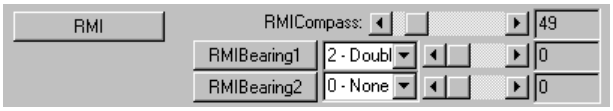
To create a Radio Magnetic Indicator control:

1. Select the **Aircraft Instruments** icon from the Palette and draw a rectangle in the Control View window
2. Right click on the control and choose **Properties**
3. Select the **HI OBI RMI TC** tab and click the **RMI** button




### Essential Properties

Data Value: Fields are linked to the RMICompass, RMIBearing1, and RMIBearing2 properties of the RMI control.




## Turn Coordinator

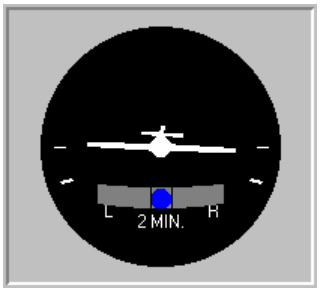
The Turn Coordinator is a simple instrument used to coordinate the amount of rudder and aileron used in a turn.

To create a Turn Coordinator control:

1. Select the **Aircraft Instruments** icon from the Palette and draw a rectangle in the Control View window
2. Right click on the control and choose **Properties**
3. Select the **HI OBI RMI TC** tab and click the **Turn Coordinator** button

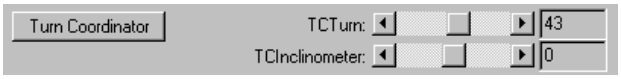


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




**Essential Properties**

Data Value: Fields are linked to the TCTurn and TCInclinometer properties of the Turn Coordinator control.




# General Purpose Controls

The list of controls included with CoPilot Professional is continually expanding. In addition to the strip chart and various aircraft controls described in the previous section, this section describes some of the other included controls. The following sections of this appendix briefly summarize some of the included controls. Each summary contains a description, the essential properties to link to, and other information as needed. Refer to the help documentation listed below for additional information about the included controls. Refer to the documentation supplied with any user-added controls.

## Alphanumeric LED

The  GMS Alphanumeric LED is one of the simplest of controls included with CoPilot although it can be adapted to display either numeric values or characters through the property window.




The control must be in Design Mode to access the properties dialog box. To reach the LED property window, right click the LED control and select Properties.

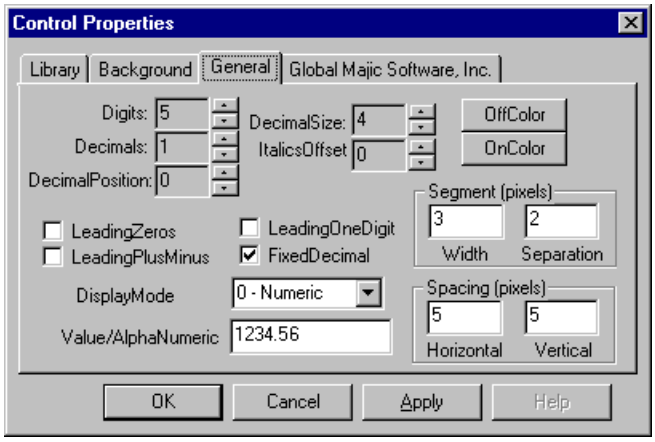
**Essential Properties**

When used as a sink, the ValueEngr of the field is linked to the Value property of the control (the last property in the properties list box).

The essential properties are listed on the General Tab.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




At design time, the “Digits” and “Decimals” values must be changed to accommodate the expected data range of the field. The most significant digits will not be displayed if too few digits are reserved.

### Optional Properties

The “FixedDecimal” check box forces the data to maintain a consistent decimal location.

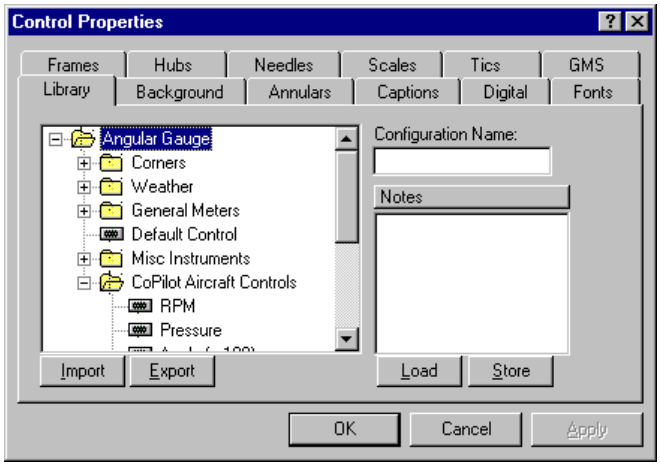
When using the string property of a field, set the display mode to “1 AlphaNumeric” and link to the “Value/AlphaNumeric” property.

The LED display can be adapted to various time formats by selecting one of the options from the tree on the Library tab or through colors and size parameters.

## Angular Gauge

### Description

The highly customizable  GMS Angular Gauge can be used to define transmit values or display transmit and receive data. Predefined controls can be accessed through the Library property.




Scales, needles, fonts, annulars, captions, and tics can be customized through the Angular Gauge property window.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Essential Properties

Data Value: Fields are linked to NeedleValue.

## Data Modifier

### Description

A  BT Data Modifier is used to linearly modify an input value. This control will multiply the input value by the scale value and add the offset to the result.

> If the source for this control is a data field in the Hardware Explorer, the results can be customized further by using a data interpreter.

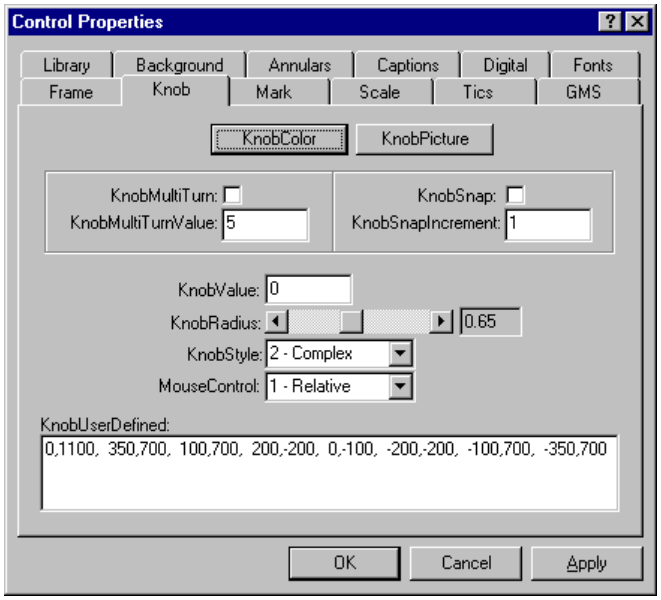
### Essential Properties

Data Value: Input fields are linked to _InVal (when the control is the sink). Output fields are linked to _OutVal (when the control is the source). The Scaling and Offset values can also be linked to dynamic field data. Set the parameters of this control in the property browser window.

## Knob

### Description

The  GMS Knob control is a highly customizable knob or dial. It is probably most useful to define transmit values.




This control is customizable through a selection of knob styles, scales, annulars, captions, and tics.

### Essential Properties

Data Value: Fields are linked to the _KnobValue property.

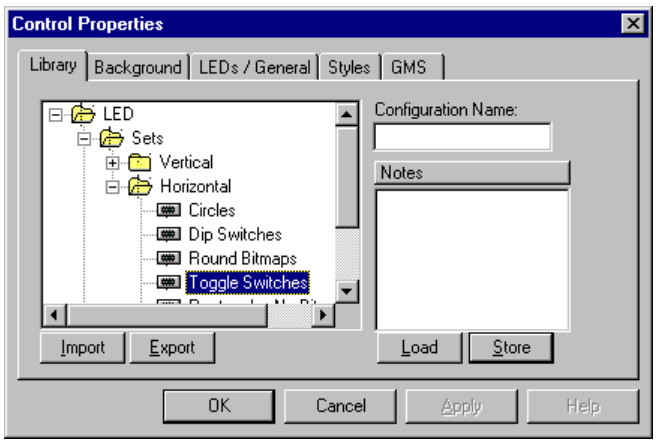


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# LED

### Description

The  GMS LED control provides a variety of Light-Emitting Diode (LED) interfaces. Families of predefined LEDs are available through the Library tab. Styles, size, color, and bitmaps are selected through the property window.



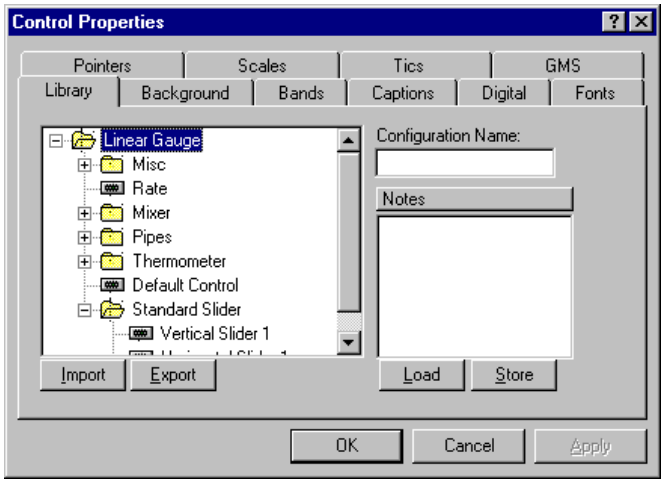

### Essential Properties

Data Value: The field is linked to the _Value of the LED control.

# Linear Gauge

### Description

The  GMS Linear Gauge can be configured as a slider for input or as a meter for output.




Users may select from a library of controls and/or modify a control through scales, bands, pointers, captions, and tics.

### Essential Properties

Data Value: Fields are linked to the PointerValue property.

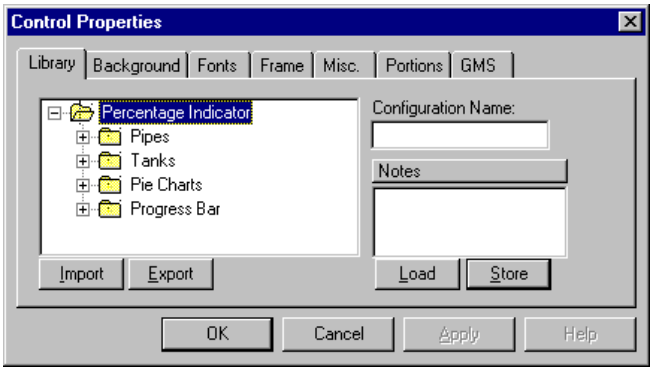


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Percent

### Description

The  GMS Percent control is typically used to show progress or indicate a level.




Since users may set values through the control it can be used for transmit or receive fields.

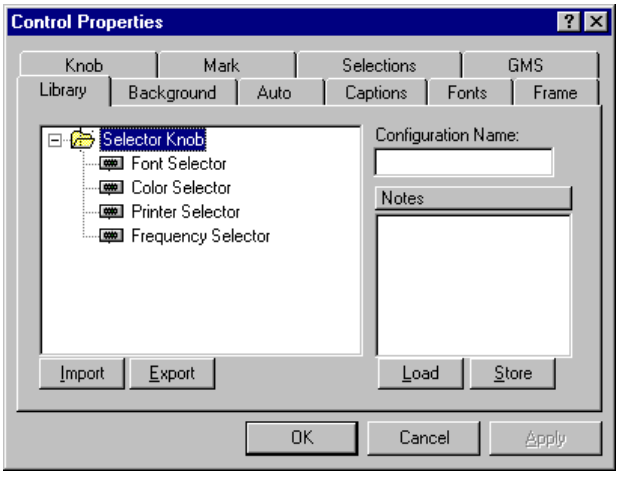
### Essential Properties

Data Value: Fields are linked to the _Portion value of the control. Field range must be set through the min/max property on the Miscellaneous tab.

## Selector Knob

### Description

The  GMS Selector Knob is used to choose between discrete options. Although selector knobs can be used as toggle switches, they are typically used when a larger number of states are involved (e.g., 1 Hz, 10 Hz, or 100 GHz bandwidths).




Selector knobs are customized through captions, knob forms, colors, lines, and offsets.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

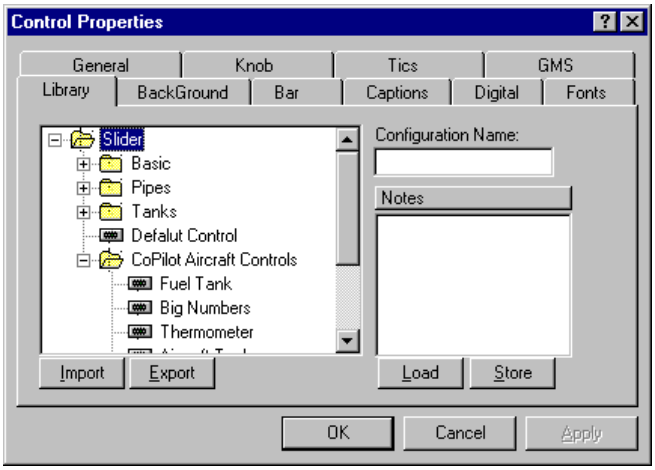
### Essential Properties

Data Value: Fields are linked to the _Value property.

## Slider

### Description

The  GMS Slider control is a versatile input or output control. The family of slider controls includes gauges and meters.




Sliders may be selected from the library of predefined slider forms or customized through bars, colors, bitmaps, knobs, captions, and tics.

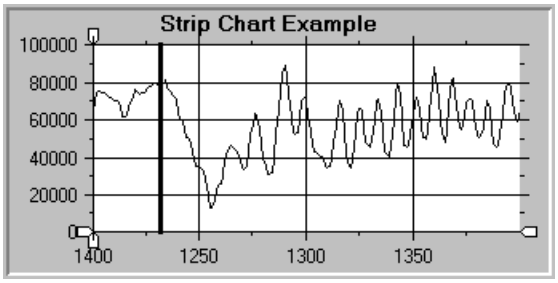
### Essential Properties

Data Value: Fields are linked to the _Value properties of Slider control.

## Strip Chart

### Description

The  GMS Strip Chart ActiveX Control is a two-dimensional charting control that provides an ideal interface for viewing a sequence of data. In many cases, the strip chart can be implemented with minimal change to control properties. Additional properties expand the display options but increase design complexity.






Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Fields Linked with Properties

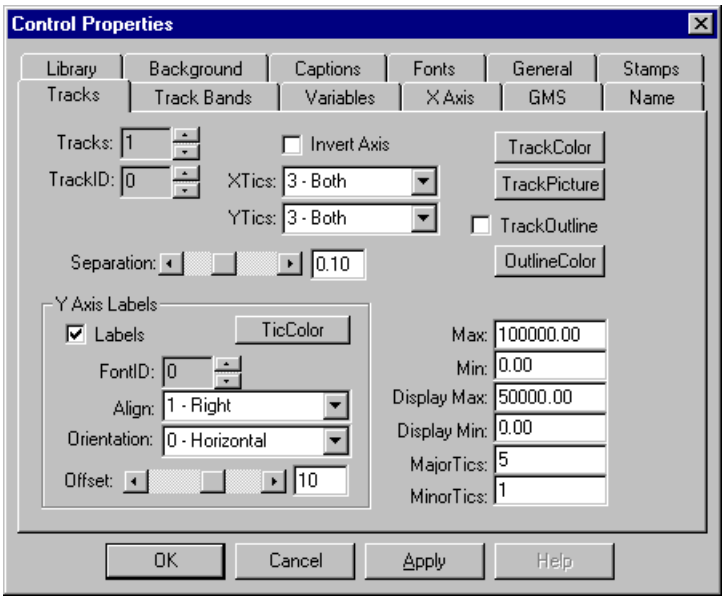
Fields are linked to ActiveX controls through properties. When CoPilot is running, the control is updated each time the value of the field changes. That works well for controls used to display current values like an altimeter or compass. A strip chart display can be confusing however. Since the counter only advances when the data value changes, the interval between values may not be uniform. Alternatively, data presented through the Strip View window, also available with CoPilot Professional, is plotted at even time intervals.

Panning and zooming functionality (available for strip chart controls and strip view displays) allow for quick review and in-depth analysis of data trends. The multi-tab property window associated with the strip view control provides tools for designing charts with full control of scales, captions, fonts, and ticks.

### Essential Properties

Data Value: Field data is linked to the "Y" property. This is the last property shown in the property window (not to be confused with the _Y at the top of the property list).

Data Ranges: In most cases, the Y-axis of the strip chart must be adjusted to fit the upper and lower limits of the field. Typically, the Min/Max values associated with field properties should be restated in the Min/Max text box on the Tracks tab. Display limits can be used to zoom in on the expected data values.




Positioning: Margins are set through the General tab. All margins are defined through slide bars based on a 0–1 scale.

### Optional Properties

Titling: Titling, title position, and font selection is defined thought the Captions and Fonts tab.

Display Mode: The graph moves forward continuously or wraps around depending of the Display Mode selection on the General tab.

Handles: The small tabs shown on the above chart allow users to zoom in on a portion of the X or Y-axis. The tabs are activated through the Handles text box on the General tab.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Line Control: The width, style, and color of variable lines are set through options on the Variables tab.

### Strip Chart versus Strip View

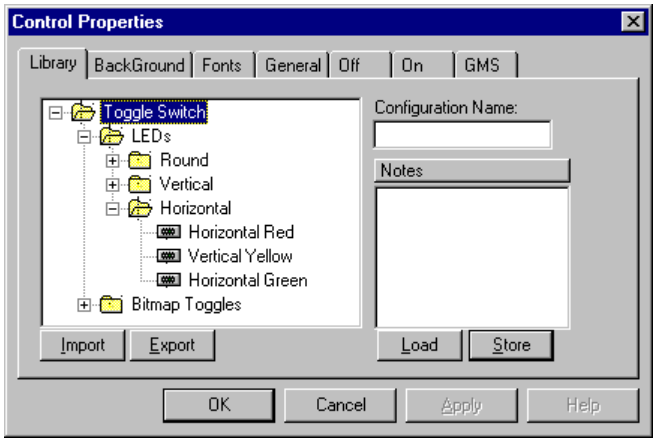
Since you can create a strip chart two different ways, it is important to understand the advantages and disadvantages of each. The Strip View is easiest to use since it is established through drag and drop methods and users can easily combine data for several fields into a common display. Scaling and titling is defined automatically and the display accurately reflects time. However, property options are limited.

The strip chart control available through the Control View window is more flexible since users have access to a wider range of properties and it can be combined with other controls in a comprehensive display of information.

## Toggle

### Description

The  GMS Toggle switches are typically used to set or display states (e.g., on/off, yes/no, true/false, etc.). Single bit values are most commonly used with this control.




Users can customize switches through selection or creation of bitmap forms, colors, and captions.

### Essential Properties

Data Value: Fields are linked to the _Value property of the control.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Appendix F: Script Examples

## Example Script Library

### Script Library

Ballard maintains a library of sample scripts for use with the CoPilot ATE and the Python Script Editor, as well as the Visual Basic Script View window. Comments in the header in each sample script describe the script objective, required objects, and tips on adapting the script for different objects or objectives. A read-me file in the scripts folder contains a list of the sample scripts plus general instructions for using each script.

### Locating Sample Scripts

The sample script library can be accessed from:

1. The CoPilot installation CD
2. The scripts folder (copied during first run for each user; located at "C:\UserName\My Documents\My CoPilot Projects\Samples\Scripts")
3. The following topics in this section, discuss selected sample scripts in detail
4. Contact technical support for additional samples and suggestions.

### Using Example Scripts

The following sections describe a few selected example scripts in detail. You can use these scripts as they are, or you can modify them to work better for you. You can also use them as a foundation for creating your own scripts.

## Example Script Details

### Selected Examples

The following sections describe a few selected example scripts in detail. Each of the examples detailed in this section are available both in Python and VBScript, though the shown example code is from the VBScript. The scripts are very similar in both languages so the text is applicable in both cases. These descriptions will teach you how to use, understand, and modify scripts. You can then apply that knowledge to other example scripts not covered in these sections.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: To view script examples in the following help topics, use the Previous/Next links at the top of the help window.

The example scripts in the following sections may be copied and pasted into the Script View or Python Script Editor windows by dragging your mouse across the text and using the **Ctrl**+**C** and **Ctrl**+**V** keystrokes.

### Python Versus VBScript

CoPilot 4 used VBScript through the Script View to accomplish automation tasks within CoPilot. With CoPilot 5 and 6, this functionality is still available, however it is recommended users choose to use Python (as included with CoPilot ATE) as their scripting language of choice in CoPilot.

Some of the CoPilot ATE with Python scripting advantages over VBScript in a Script View include:

- Easier editing of code, with syntax error checking, Autocomplete, Intellisense, and a more flexible text editor
- A more feature-rich and powerful scripting language (such as 64-bit data types)
- Quicker code execution resulting in faster script calls
- The ability to run functions at any time instead of just on simulation start and stop (ability to control the CoPilot shell object)
- Much better error information when errors occur
- A large library of code to use included with Python by default
- Command line interface with the 'Command Prompt'
- Debugging functionality with 'Watch Window' from within CoPilot

See the *Automated Test Environment (ATE)* section on page *399* for additional information.

### Importing a Sample Script into Script View (VBScript)

The CoPilot sample scripts, and any saved script file, can be imported directly into the scripting pane.

To import a script:

1. Right click in the Script pane to open the context menu, point to **Import Script**, and either…
   - Click **Overwrite** to replace the contents of the scripting pane with the imported file
   - Or click **Append** to insert the selected file at the end of the code in the scripting pane
2. In the Select Import File dialog, click the browse button to launch the Open File dialog
3. Browse to the script you wish to import, select it, and click OK to close the dialog and load the script

You can browse to the folder of sample scripts that are installed with CoPilot or to a script of your own.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#### VBScript Structure

Each script is composed of at least two subroutines: ScriptStart and ScriptStop. All subroutines are preceded by “Sub” and end with “End Sub.” Comments are preceded by a single apostrophe. Each example script is preceded by a brief explanation and the actual file name under which it is saved.

#### VBScript Subroutine Principles

The ScriptStart subroutine is called and run through when the Run button engages the CoPilot simulation. Other subroutines may be called from ScriptStart or may be triggered by events. ScriptStop is called when the simulation is halted and executes any necessary clean-up actions (e.g., saving files, closing applications, releasing resources, etc.).

#### VBScript Prerequisites

In order for a script to execute successfully, all the necessary conditions must be present. Objects referenced in the script must be present in CoPilot, properly configured, active, and added to the object pane of the Script View window. Be sure that the identifier of the object in the object pane matches the identifier in the script. (Objects may be renamed at any time.)

## Create a Binary String from a Decimal Number

This is a simple script illustrating the use of a subroutine and functions to transform a decimal value into a binary string. In this example, the decimal value is initialized to 27, but you could get the value from another source (for example, another field).

#### BinaryTextOut.txt

```
‘Global Variable
Const MyVal = 27    ‘Sample Value


Sub ScriptStart
       Dim MyBinaryString
       MyBinaryString = BinStr(MyVal)
       ‘Output results to output window in Script View
       View.OutputLine “Binary Output=” & MyBinaryString
End Sub
Sub ScriptStop
End Sub


Function BinStr(InNumber)
       While InNumber
              If InNumber AND 1 Then
                     BinStr = “1” & BinStr
              Else
                     BinStr = “0” & BinStr
              End IF
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
            'Left-shift by 1... or divide by 2
            InNumber = CLng(InNumber / 2)
      WEnd
End Function
```

## Increment Counter on Value Change

This example illustrates how a counter could be linked to a data field. This script sets up an event handler for the engineering value of a field. When the value changes, a counter is incremented via the script.

**ChangeCounter.txt**

```
'Globals
Dim bInitialized    'boolean-initialized flag
Dim dOldValEngr     '(double) previous data value
Dim lChangeCounter  '(long) counts the number of times the
value changes


Sub ScriptStart
      'Initialization of constants
      bInitialized = 0
      dOldValEngr = 0
      lChangeCounter = 0
End Sub
Sub ScriptStop
End Sub

'Event handler for engineering value of Field1
Sub Field1_ValueEngr
      If bInitialized = 0 Then
            bInitialized = 1
      End If
      If dOldValEngr <> Field1.ValueEngr Then
            lChangeCounter = lChangeCounter + 1
            'Store new value
            dOldValEngr = Field1.ValueEngr
            'Output new count
            View.OutputLine "lChangeCounter= " &
lChangeCounter
            'Output results to output pane in Script View
      End If
      MsgBox "lChangeCounter= " & lChangeCounter
End Sub
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Add Random Noise to Data Pattern

This example receives data from one source and transmits it to another with superimposed random noise. For example, the source could be a data field that is generating a sine wave (via the Data Generator). The retransmitted value would be the same sine wave garbled by random noise.

This example is based on ARINC 429 messages (labels). Modify this script for MIL-STD-1553 by making the following changes:

1. Change “Msg1.ValueEngr” and “Msg2.ValueEngr” to data fields (for example, “Field1.ValueEngr”).
2. Change all references in this script of “TimeTag” to “MsgTimeTag” (for example, “Sub Msg1_MsgTimeTag”).

**ChangingDataAddRndNoise.txt**

```
‘Globals
Const ERRORVAL = 50


Sub ScriptStart
   Randomize
   Msg2.ValueEngr = 0      ‘Initialized to 0
End Sub
Sub ScriptStop
End Sub


‘Event Handler for time-tag of Lx43Msg1 changes the data
Sub Msg1_TimeTag
       Value = (ERRORVAL * Rnd)   - (ERRORVAL/2)
       ‘Random Number between +/-ERRORVAL
       Msg2.ValueEngr = Value + Msg1.ValueEngr
End Sub
```

## Generate Data from Monitor Record Count

This example changes the data value of Field1 to the count in the Sequential Monitor when the Sequential Monitor count changes.

**ChangingDataFromMonCount.txt**

```
Sub ScriptStart
End Sub


‘Event Handler for the count of Mon1
Sub Mon1_Count
   ‘Set the data value of Field1
   Field1.ValueEngr = Mon1.Count
End Sub
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
Sub ScriptStop
   View.OutputLine “There were “ & CStr(Mon1.Count) & “
items recorded.”
End Sub
```

## Generate Random Data Values

This example changes the data value of Msg1 to a random value (between MINVAL and MAXVAL) when the time-tag of Msg1 changes. In this example, Msg1 is a ARINC 429 message (label). This can be modified for MIL-STD-1553 by making the following changes:

1. Rename your 1553 BCMsg object to Msg1 in the Script View object pane.
2. Change all references in this script of “TimeTag” to “MsgTimeTag,” even in “Sub Msg1_TimeTag.”
3. Change the “Msg1.ValueEngr = Value” line to work with a data field (for example, “Field1.ValueEngr = Value”).

**ChangingDataRndGen.txt**

```
‘Globals
Const MAXVAL = 200
Const MINVAL = -200

Sub ScriptStart
       Randomize
End Sub
Sub ScriptStop
End Sub

‘Event Handler for Timetag of Msg1 changes the data
Sub Msg1_TimeTag
   Value = MINVAL + ( (MAXVAL-MINVAL) * Rnd)
       ‘Random Number between MINVAL and MAXVAL
   Msg1.ValueEngr = Value
End Sub
```

## Generate Sine Wave Data Values

This example changes the data value of Msg1 to a sinusoidal value (between MINAMP and MAXAMP) when the time-tag of Msg1 changes. The period is assigned by the PERIOD constant in milliseconds (10,000 ms = 10 sec). In this example, Msg1 is a ARINC 429 message (label). This can be modified for MIL-STD-1553 by making the following changes:

1. Rename your 1553 BCMsg object to Msg1 in the Script View object pane.
2. Change all references in this script of “TimeTag” to “MsgTimeTag,” even in “Sub Msg1_TimeTag.”



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3. Change the “Msg1.ValueEngr = Value” line to work with a data field (for example, “Field1.ValueEngr = Value”).

**ChangingDataSinGen.txt**

```
‘Globals
Const MAXAMP = 100          ‘Max Amplitude
Const MINAMP = -100         ‘Min Amplitude
Const PERIOD = 10000        ‘Period in milliseconds
Const PI     = 3.14159
Dim MyStartTime
Sub ScriptStart
   ‘Initialization of Constants
   MyStartTime = 0
End Sub
Sub ScriptStop
End Sub


‘Event Handler for time-tag of Msg1 changes the data
Sub Msg1_TimeTag
   If MyStartTime = 0 Then
       MyStartTime = CLng(Msg1.TimeTag)
       Angle = 0.0
   Else
       deltaMiliSec=(CLng(Msg1.TimeTag)-
CLng(MyStartTime))/1000
       deltaMiliSec = deltaMiliSec MOD PERIOD
       Angle = CDbl( (deltaMiliSec/PERIOD) * 2.0 * PI)

End If
   Value = (MINAMP+MAXAMP) + (MAXAMP-MINAMP)/2.0 *
Sin(Angle)
   Msg1.ValueEngr = Value
End Sub
```

## Create a Simulated Monitor File in Excel

This script simulates some of the functions of a Bus Monitor (or Sequential Monitor). First, an Excel worksheet is created and formatted to receive the data. Then, the time-tag from the specified 1553 message is copied into the first column. The engineering value of the data field of the message is copied into the second column. For this first row, the delta values will be null because they have nothing to compare against.

Starting with the second row, the delta value of the message time-tag is entered into the third column. (The formula is entered and Excel calculates the value.) The delta value of the field data is entered into the fourth column.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The BCMsg1_MsgTimeTag subroutine is triggered by the time-tag of the specified message. This function will continue to run and copy data to the Excel file as long as the time-tag event that triggers it continues.

When the CoPilot simulation is stopped, ScriptStop is called, which closes Excel. You will be prompted to save the Excel file before exit.

**Export1553Mon2Excel.txt**

```
'Globals
Dim objExcel
Dim ExcelSheet
Dim iRowCount
Sub ScriptStart
	Field1.ValueEngr = 0
	'Open Excel and make it visible
	Set objExcel = CreateObject("Excel.Application")
	objExcel.Visible = True
	objExcel.UserControl = True
	'Create new workbook
	objExcel.Workbooks.Add
	Set ExcelSheet = objExcel.ActiveSheet
	objExcel.DisplayAlerts = False
	'Delete the worksheets not being used
	objExcel.Worksheets("Sheet2").Delete
	objExcel.Worksheets("Sheet3").Delete
objExcel.DisplayAlerts = True
	'Rename the worksheet
	ExcelSheet.Name = "Data1"
	'Write column names
	iRowCount = 1
	ExcelSheet.Cells(iRowCount, 1).Value = "MsgTimeTag"
	ExcelSheet.Cells(iRowCount, 2).Value = "Data"
	ExcelSheet.Cells(iRowCount, 3).Value = "Delta Time
(ms)"
	ExcelSheet.Cells(iRowCount, 4).Value = "Delta Value"
End Sub
Sub ScriptStop
	ExcelSheet.Application.Quit
	'You will be prompted for save information
End Sub
Sub BCMsg1_MsgTimeTag
	iRowCount = iRowCount+1
	ExportData(iRowCount)
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
End Sub
Sub ExportData(iRow)
	ExcelSheet.Cells(iRow, 1).Value = BCMsg1.MsgTimeTag
	ExcelSheet.Cells(iRow, 2).Value = Field1.ValueEngr
	If (iRow > 2) Then
		ExcelSheet.Cells(iRow, 3).Value = CStr( “=(a” 
& iRow & “ - a” & (iRow-1) & “)/1000” )
		ExcelSheet.Cells(iRow, 4).Value = CStr( “=b” & 
iRow & “ - b” & (iRow-1) )
	End If
End Sub
```

## Create and Display a Text File

This script creates a file, writes text to the file, and returns the file contents to the output window in Script View. All file access is within the TextStreamTest subroutine. To be able to access the file system object from multiple locations, move the “Dim...” line outside of all Subs (to make the variables global) and move the TextStreamTest close line (ts.Close) to the ScriptStop subroutine.

**FileReadWrite.txt**

```
Sub ScriptStart
	‘Display the return value of TextStreamTest in the 
output window
	View.Output TextStreamTest(“test1.txt”)
End Sub
Sub ScriptStop
End Sub


Function TextStreamTest(filename)
	Const ForReading = 1, ForWriting = 2, ForAppending = 
8
	‘Appending->Open a file and write to the end of the 
file.
	Const TristateUseDefault = -2, TristateTrue = -1, 
TristateFalse = 0
	Dim fso, f, ts
	Set fso = CreateObject(“Scripting.FileSystemObject”)
fso.CreateTextFile filename	‘Create a file.
	Set f = fso.GetFile(filename)
	Set ts = f.OpenAsTextStream(ForWriting, 
TristateUseDefault)
	‘Output text to a file
	ts.Write “CoPilot Script”
	ts.Write “Output File”
	ts.WriteBlankLines(1)
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
        'Writes "1" carriage return at the end of the line
        ts.WriteLine Date & " " & Time
        'Writes the specified text with a carriage return
        ts.Write "Hello World"
        ts.Close
        'Now read the file back

Set ts = f.OpenAsTextStream(ForReading, TristateUseDefault)
        'Set the function return value to the file contents
        TextStreamTest = ts.ReadAll
        ts.Close

End Function
```

## Trigger Monitor by Limit Conditions

### Monitor Control Example

The following illustrates how the MonitorControl.txt script might be used to record bus activity only during takeoff and landing. The script below controls the Sequential Monitor pause function based on the value of the Altitude field. The script refers to two CoPilot objects (a data field and a Sequential Monitor), so we need to add them to the Script View window.

To assign the monitor and field objects:

1. Drag the Sequential Monitor object from the Hardware Explorer into the Script View object pane
2. Drag the Altitude field object from the Hardware Explorer into the Script View object pane

Objects are given a default name as they are added to the object pane. Since script refers to these as Mon1 and Field1, click the name of each object and change the names to Mon1 and Field1.

The script below includes three subroutines:

- Sub ScriptStart is called when the CoPilot Run  button is clicked. This subroutine places the monitor in a ready position (releasing the monitor from pause state if necessary).
- Sub Field1_ValueEngr is an event handler. It is called each time the ValueEngr property of Field1 is changed. If the engineering units value of Field1 (Altitude) is less than 5,000 feet, the monitor records. If greater than 5,000 feet, the monitor is paused.
- Sub ScriptStop is called when the CoPilot Stop  button is clicked.

### MonitorControl.txt

```
'Globals
Const UPPERLIMIT = 5000
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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

Sub Field1_ValueEngr
	If Field1.ValueEngr <= UPPERLIMIT Then
		Mon1.Pause = False
	Else
		Mon1.Pause = True
	End If
End Sub
```

## Trigger Monitor by Record Count

This sample script pauses the Sequential Monitor when a set number of records has been recorded. The Sequential Monitor must be present in the Hardware Explorer and active.

**MonitorCount.txt**

```
‘Globals
Const MAXCOUNT = 500

Sub ScriptStart
	‘Make sure the monitor is not set to pause on start
	Mon1.Pause = False
End Sub
Sub ScriptStop
End Sub

Sub Mon1_Count
	If CDbl(Mon1.Count) >= MAXCOUNT Then
		Mon1.Pause = True
		‘Pause the monitor when above MAXCOUNT
	End If
End Sub
```

## Save Data to Excel

This is a simple example of how to write CoPilot data to Microsoft Excel. This could easily be incorporated into another script, using data values from that



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

routine. To write to Excel in the background without obscuring the CoPilot screen, set ExcelSheet.Application.Visible to false.

If the file name in your script already exists, an Excel message box will appear asking for permission to overwrite the old file. This will halt CoPilot until cleared by clicking yes or no. To prevent this event from pausing CoPilot, choose a unique file name or delete or move the existing file.

**SaveDataToExcel.txt**

```
Sub ScriptStart
End Sub


Sub ScriptStop
       Dim objExcel
       Dim ExcelSheet
       ‘Create the Excel application object
       Set objExcel = CreateObject(“Excel.Application”)
       objExcel.UserControl = True
       ‘Create a spreadsheet
       Set ExcelSheet = CreateObject(“Excel.Sheet”)
       ExcelSheet.Application.Visible = True
       ‘Add data from CoPilot and save
       ExcelSheet.ActiveSheet.Cells(1,1).Value =
Field1.ValueEngr
       ExcelSheet.SaveAs “C:\COPILOT PROJECTS\EXAMPLE.XLS”
       ExcelSheet.Application.Quit
       Set objExcel = Nothing
       Set ExcelSheet = Nothing
End Sub
```

## Access a File Open or File Save Dialog

This script creates a common “Open” dialog box and opens the file specified by user input. Use the dialog to browse for the file you wish to open. To modify this script to create a “Save As” dialog, replace “.showopen” with “.showsave.”

**SaveLoadFileDialog.txt**

```
Sub ScriptStart
       set oCmnCtrl = createobject(“MSComDlg.CommonDialog”)
       with oCmnCtrl
              .Filter = “All Files (*.*)|*.*|Text Files
(*.txt)|*.txt “
              .FilterIndex = 2
              ‘filter is initialized to item 2: *.txt files
              .MaxFileSize = 260
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
            .flags = cdlOFNHideReadOnly +
cdlOFNFileMustExist
            ‘file must NOT be read-only and must exist to
show up
            .showopen()
            ‘show the “Open” dialog or use .showsave() for
“Save As” dialog
      end with
      openedFile = oCmnCtrl.Filename
      ‘This is the filename of the file selected
      set oCmnCtrl = nothing
      ‘free the oCmnCtrl object


      If openedFile <> NULL Then
      ‘To do: Now open the file name specified by
“openedFile”
      End If
End Sub
Sub ScriptStop
End Sub
```

## Open a Web Page

This example opens Internet Explorer® to the Ballard Technology home page. Instead of Ballard’s site, other websites or documents (such as PDF, GIF, JPG, WAV, AVI, DOC, XLS, etc.) can be opened via Internet Explorer.

**WebBrowse.txt**

```
Dim ie
Sub ScriptStart
      ‘Create the Internet Explorer object
      Set ie = CreateObject(“internetExplorer.application”)
      ‘Display new web page
      ie.Navigate(“www.ballardtech.com”)
      ie.Visible = True
End Sub
Sub ScriptStop
      Set ie = Nothing
End Sub
```

## Create a Web Page

This example creates a web page with text and a hyperlink to the Ballard Technology home page. This page is saved to the host computer hard drive. Internet Explorer is launched and the web page with hyperlink is displayed.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This script demonstrates how to access and manipulate the Internet Explorer object. Web developers can expand this example to implement their HTML code from within CoPilot.

**WebPageOutput.txt**

```
Dim ie 'Internet Explorer Object
Dim strOutputFileName
strOutputFileName = "C:\test1.html"


Sub ScriptStart
	'Create the Internet Explorer object
	Set ie = CreateObject("InternetExplorer.application")
	'Define file constants
	Const ForReading = 1, ForWriting = 2, ForAppending =
8
	'Appending->Open a file and write to the end of the
file.
	Const TristateUseDefault = -2, TristateTrue = -1,
TristateFalse = 0
	'Open file for writing
	Dim fso, f, ts
	Set fso = CreateObject("Scripting.FileSystemObject")
	fso.CreateTextFile strOutputFileName   ' Create a
file.
	Set f = fso.GetFile(strOutputFileName)
	Set ts = f.OpenAsTextStream(ForWriting,
TristateUseDefault)
	'Write HTML markup to a file
	ts.Write "<HTML>"
	ts.Write " <HEAD>"
	ts.Write " <TITLE>CoPilot Scripting with
I.E.</TITLE>"
	ts.Write " </HEAD>"
	ts.Write "<BODY>"
	ts.Write "<H1>ScriptStart Text and Markup</H1>"
	ts.WriteBlankLines(1)
	ts.Close
	'Open file for appending
	Set fso = CreateObject("Scripting.FileSystemObject")
	Set f = fso.GetFile(strOutputFileName)
	Set ts = f.OpenAsTextStream(ForAppending,
TristateUseDefault)
	'Write HTML markup to a file
	ts.Write "<P>ScriptStop Text and Markup</P>"
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
        ts.WriteBlankLines(1)
        ts.Write “<P>Cool Link   “
        ts.Write “<A
href=””http:\\www.ballardtech.com””>Ballard Technology</A>”
        ts.Write “</P>”
        ts.Write “</BODY>”
        ts.Write “</HTML>”
        ts.Close
        ‘Display new web page
        ie.Navigate(strOutputFileName)
        ie.Visible = True
End Sub
Sub ScriptStop
        Set ie = Nothing
End Sub
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Appendix G: Regular Expressions

## Purpose

This section on regular expressions applies to Python scripting in ATE (Automated Test Environment) that can be used for searching for patterns rather than literals. The following syntax section lists several types with examples.

## Syntax

1. char

matches itself, unless it is a special character (metachar): `. \ [ ] * + ^ $ ( )`

2. `.`

matches any character.

3. `\`

matches the character following it, except:

`\a`, `\b`, `\f`, `\n`, `\r`, `\t`, `\v` match the corresponding C escape char, respectively BEL, BS, FF, LF, CR, TAB and VT;
Note that \r and \n are never matched because in the CoPilot Python Script Editor, regular expression searches are made line per line (stripped of end-of-line chars).

when followed by a digit 1 to 9 (see [8]);

when followed by a left or right angle bracket (see [9]);

when followed by d, D, s, S, w or W (see [10]);

when followed by x and two hexa digits (see [11]);

Backslash is used as an escape character for all other meta-characters, and itself.

4. `[set]`

matches one of the characters in the set. If the first character in the set is ^, it matches the characters NOT in the set, i.e. complements the set. A shorthand S-



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

E (start dash end) is used to specify a set of characters S up to E, inclusive. The special characters `]` and `-` have no special meaning if they appear as the first chars in the set. To include both, put `-` first: `[-]A-Z]` (or just backslash them).

| example | match |
|---|---|
| [-]\|] | matches these 3 chars, |
| []-\|] | matches from ] to \| chars |
| [a-z] | any lowercase alpha |
| [^-]] | any char except - and ] |
| [^A-Z] | any char except uppercase alpha |
| [a-zA-Z] | any alpha |

5. `*`

any regular expression form [1] to [4] (except [7], [8] and [9] forms of [3]), followed by closure char (`*`) matches zero or more matches of that form.

6. `+`

same as [5], except it matches one or more. Both [5] and [6] are greedy (they match as much as possible).

7.

a regular expression in the form [1] to [12], enclosed as `(`*`form`*`)` matches what *form* matches. The enclosure creates a set of tags, used for [8] and for pattern substitution. The tagged forms are numbered starting from 1.

8.

a \ followed by a digit 1 to 9 matches whatever a previously tagged regular expression ([7]) matched.

9. `\< \>`

a regular expression starting with a `\<` construct and/or ending with a `\>` construct, restricts the pattern matching to the beginning of a word, and/or the end of a word. A word is defined to be a character string beginning and/or ending with the characters A-Z a-z 0-9 and _. Scintilla extends this definition by user setting. The word must also be preceded and/or followed by any character outside those mentioned.

10. `\l`

a backslash followed by d, D, s, S, w or W, becomes a character class (both inside and outside sets `[]`).

d: decimal digits

D: any char except decimal digits

s: whitespace (space, \t \n \r \f \v)

S: any char except whitespace (see above)

w: alphanumeric & underscore (changed by user setting)

W: any char except alphanumeric & underscore (see above)

11. `\xHH`



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

a backslash followed by x and two hexa digits, becomes the character whose Ascii code is equal to these digits. If not followed by two digits, it is 'x' char itself.

12.

a composite regular expression xy where x and y are in the form [1] to [10] matches the longest match of x followed by a match for y.

13. ^ $

a regular expression starting with a ^ character and/or ending with a $ character, restricts the pattern matching to the beginning of the line, or the end of line. [anchors] Elsewhere in the pattern, ^ and $ are treated as ordinary characters.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Appendix H: Legacy CoPilot Support

## CoPilot VB Scripting

With the introduction of CoPilot ATE in version 5.0, Script Views has been deprecated. It is suggested that new applications use the Python scripting language with the Python Script Editor, however, the Visual Basic Script Views are still supported and can access the updated CoPilot Object Model.

Some of the CoPilot ATE with Python scripting advantages over VBScript in a Script View include:

- Easier editing of code, with syntax error checking, Autocomplete, Intellisense, and a more flexible text editor
- A more feature-rich and powerful scripting language (such as 64-bit data types)
- Quicker code execution resulting in faster script calls
- The ability to run functions at any time instead of just on simulation start and stop (ability to control the CoPilot shell object)
- Much better error information when errors occur
- A large library of code to use included with Python by default
- Command line interface with the 'Command Prompt'
- Debugging functionality with 'Watch Window' from within CoPilot

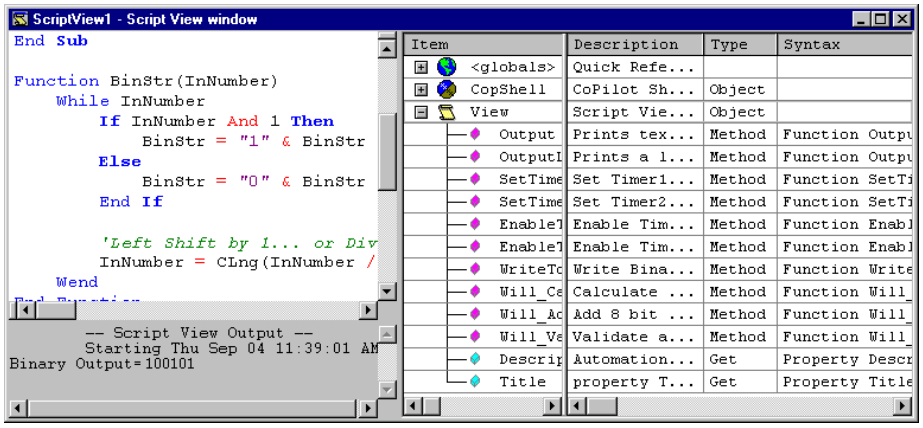
See the *Automated Test Environment (ATE)* section on page *399* for additional information.

#### Customize CoPilot with Scripting

Script View is a CoPilot Professional component that allows users to extend the functionality of the CoPilot environment. Scripts could be used to 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 can also be used to transfer information between CoPilot and other applications (using OLE Automation). Scripts are defined using Microsoft® Visual Basic® Scripting language (VBScript).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




### VBScript

Scripts are written using VBScript, which is a subset of Microsoft® Visual Basic®. If you are already familiar with Visual Basic or Visual Basic for Applications® (VBA), VBScript will be easy to use. Less experienced users will be able to understand and develop useful applications by adapting the scripting examples included with CoPilot. Additional information, tutorials, and examples of VBScript are widely available on the Internet.

### Access the Script View Window

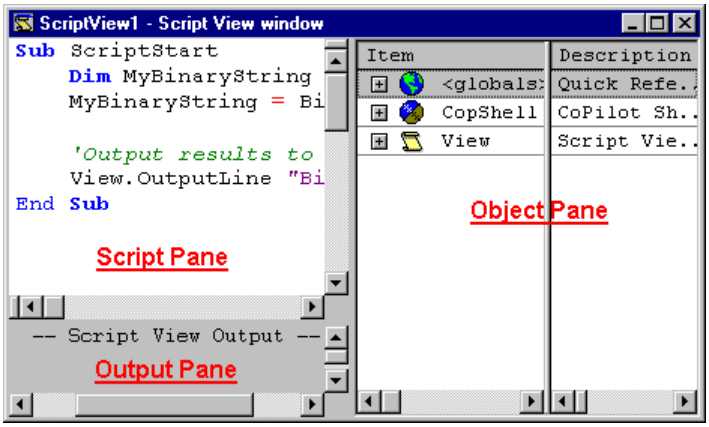
To create a Script View window:

1. Select the File | New Hardware or View command to open the New Hardware or View dialog
2. Select Script View in the Common Views tab and click OK

Script View windows can be saved as part of a CoPilot project (with the File | Save As command).

### Script View Panes

The Script View window is divided into three panes: the script pane and script output pane on the left, and the script objects pane on the right. These panes may be resized as needed by dragging the pane edges while holding the left mouse button.





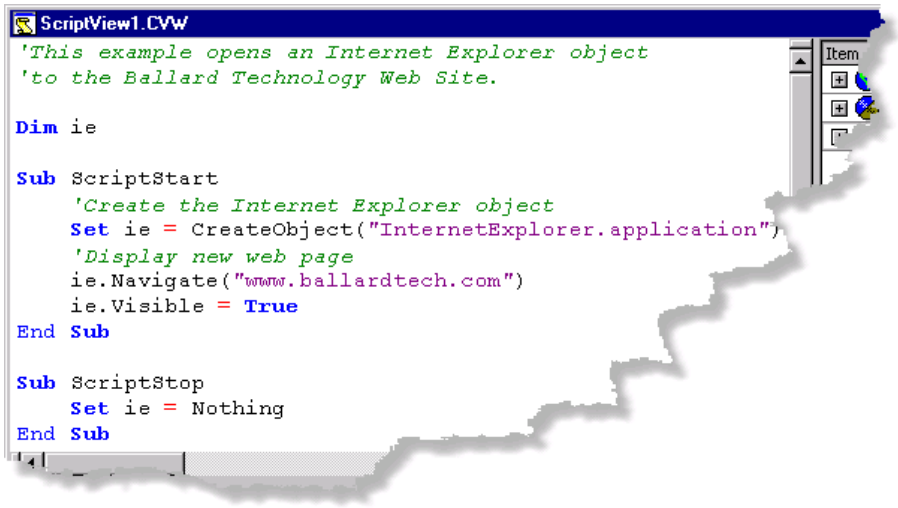

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Script Pane

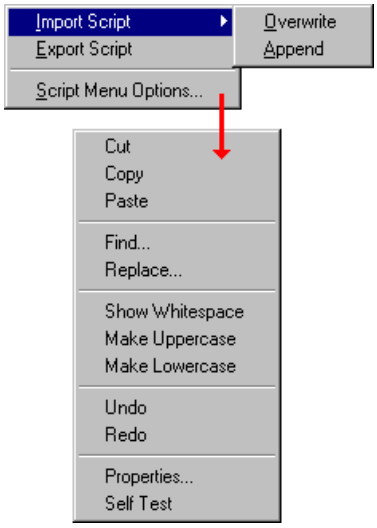
### Code Window

Procedures are recorded and edited through the script pane. Type directly in the pane, paste in a script from the provided examples and edit it, or import a script from the included library. The text is color-coded by type. The default colors (described below) can be changed by the user.

- Reserved words are blue (for example, “While” and “Sub ScriptStart”).
- Comments (preceded by a single quote) are in green
- Operators are red
- Strings (encased in double quotes) are purple




The right-click context menu in the script pane provides access to a number of options and commands, including Export and Import Script. To expand the submenu, point to Script Menu Options… (see figure below).




### Importing a Sample Script

The CoPilot sample scripts, and any saved script file, can be imported directly into the scripting pane.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To import a script:

1. Right click in the Scripting Pane to open the context menu, point to **Import Script**, and either…
   - Click **Overwrite** to replace the contents of the scripting pane with the imported file
   - Or click **Append** to insert the selected file at the end of the code in the scripting pane
2. In the Select Import File dialog, click the browse button to launch the Open File dialog
3. Browse to the script you wish to import, select it, and click OK to close the dialog and load the script

You can browse to the folder of sample scripts that are installed with CoPilot (see Sample Script Library) or to a script of your own.

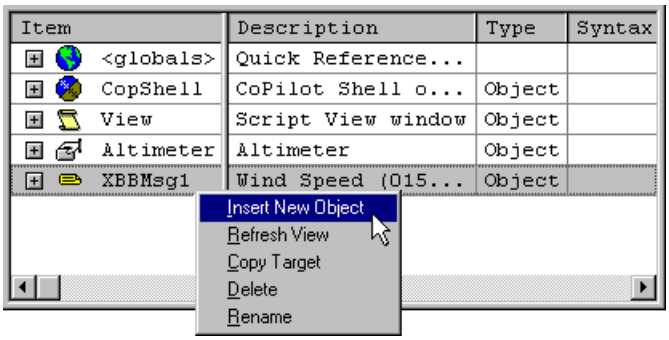
## Script Objects

### CoPilot Objects

Scriptable CoPilot objects include ARINC 429 and MIL-STD-1553 messages, fields, Sequential Monitors, Control View controls, certain CoPilot views, and many other objects.

The key to scripting is access to CoPilot objects and properties. In order for a script to access a CoPilot object, the object must be added to the Object Pane. An exception to this rule is when objects are created from within the script (for example, a non-CoPilot object associated with an external application).

The "globals" object (see figure below) is actually a quick VBScript reference guide built in to the CoPilot Script View window. The globals, CoPilot Shell, and Script View objects are always present.




Note: The options displayed in the Objects pane **context menu** change to reflect the commands available for the selected object (see figure above).

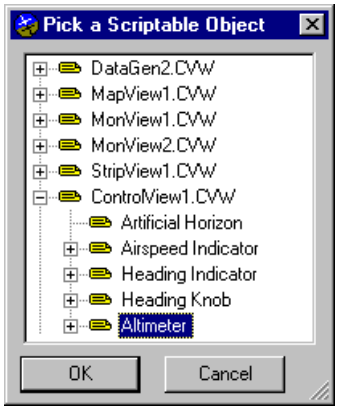
To add a CoPilot object to the Object Pane:

1. Right click in the Object Pane and choose **Insert New Object** from the context menu (see figure above)
2. In the Pick a Scriptable Object dialog (see figure below), browse to and select an object



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3. Click **OK** to load the object into the Object Pane and close the dialog
4. Alternatively, scriptable objects in the Hardware Explorer can be added by dragging and dropping them into the Object Pane

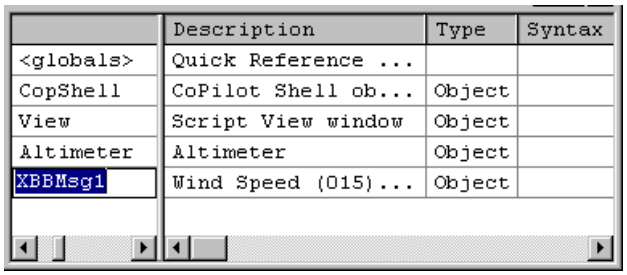



### Referencing and Renaming Objects

When an object is added to the Objects pane, a default name appears in the Item column of the Object pane. It is often useful to rename objects with a more descriptive name so that object references in the script are self-documenting. Similarly, sample scripts reference objects by generic names. You can change either the object name in the Script Pane or the object name in the Object Pane to match the script.

To rename an object:

- Click twice on the name and edit it (a cursor and highlighting will appear)
- Alternatively, right click on the object and choose Rename from the context menu



| | Description | Type | Syntax |
|---|---|---|---|
| <globals> | Quick Reference ... | | |
| CopShell | CoPilot Shell ob... | Object | |
| View | Script View window | Object | |
| Altimeter | Altimeter | Object | |
| XBBMsg1 | Wind Speed (015)... | Object | |

## Script Object Properties and Methods

*property: A characteristic or attribute of an object (e.g., Visible = True).*

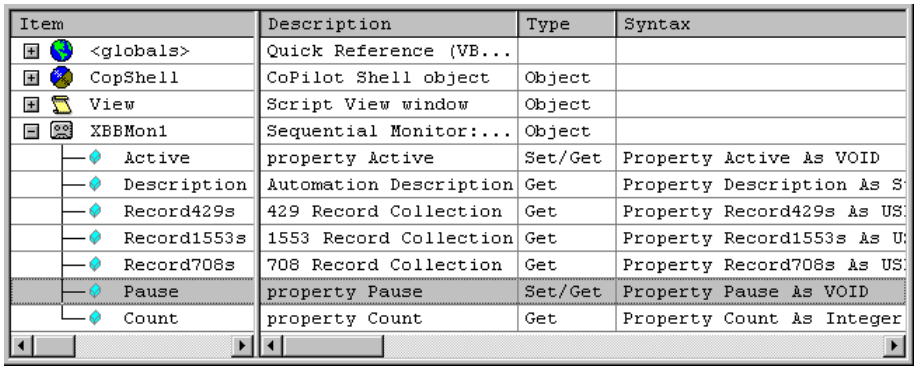
*method: A procedure, function, or routine associated with an object (e.g., ShowPropertyPage).*

### Properties and Methods

When an object in the Object Pane is expanded, its properties and methods are listed below it (see figure below). The Description column provides a brief explanation of each property or method. The Type column classifies each entry as a Get and/or Set property or as a method. The Syntax column provides the calling syntax (e.g., the data type of a property, the parameters of a method, etc.). The figure below shows the properties of a Sequential Monitor object.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




Tip: If part of the content of a row are obscured, hover the mouse over the cell and a pop-up will appear with the complete string.

- Properties of an object are marked with a light blue icon
- "Get" properties can be read by the script
- "Set" properties can be modified by the script
- A small "E" is included in the icon of properties that can trigger scripting events.
- Items marked with a magenta icon are methods that generally perform a task, such as activating a window

Tip: A quick way to use a property or method in your code is to right click on it in the Object pane, choose Copy Target from the context menu, then paste it into the Scripting pane.

### Object Event Handlers

*event: In the context of scripting, an event is anything that fires a notification that can then trigger an event handler to execute (for example, when a value property changes).*

Properties icons marked with an E symbol can trigger scripting events. When a triggering property changes (either from script or from any other mechanism), it can cause an event handler subroutine to be executed.

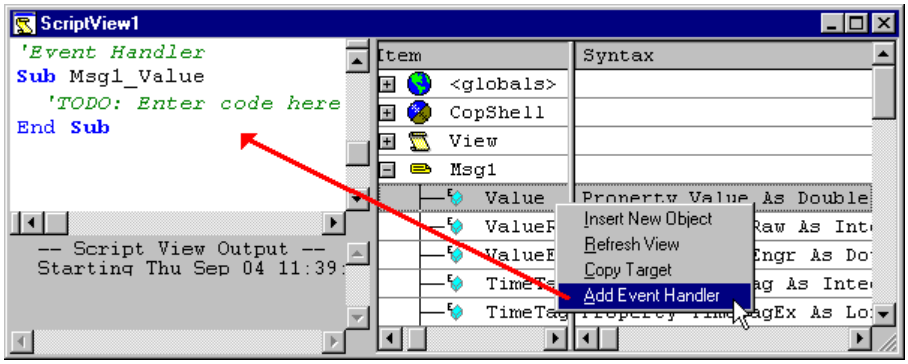
The Object pane has a menu shortcut for creating event handler subroutines in the Script pane.

To automatically create an event handler:

1. Right click on a property in the Object pane that can trigger an event (small "E" on blue property icon; see figure below) and choose **Add Event Handler** from the context menu
2. In the Script pane, add your code within the new subroutine



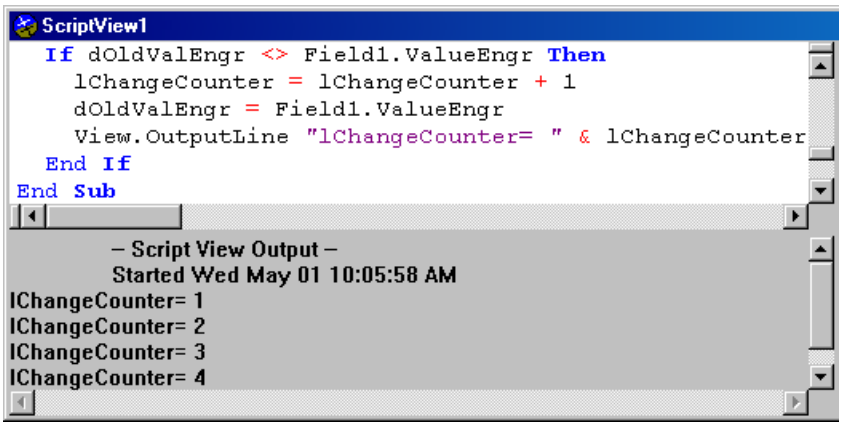
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




Tip: If an event handler for a property already exists in the Script pane, executing the Add Event Handler command will auto-scroll the Script pane to that routine (to prevent two event handlers for the same property from accidentally being coded).

## Script Output

The Output pane may be used to output information, view results, time-stamp an event, debug the script, etc. While a script is running, error messages (if any) are also displayed in the output pane. To send a string or value to the output pane, use the "Output" or "OutputLine" method. In the example below, an "if" statement is used to increment a globally defined counter each time the value of Field1 changes. The counter is displayed in the output pane.



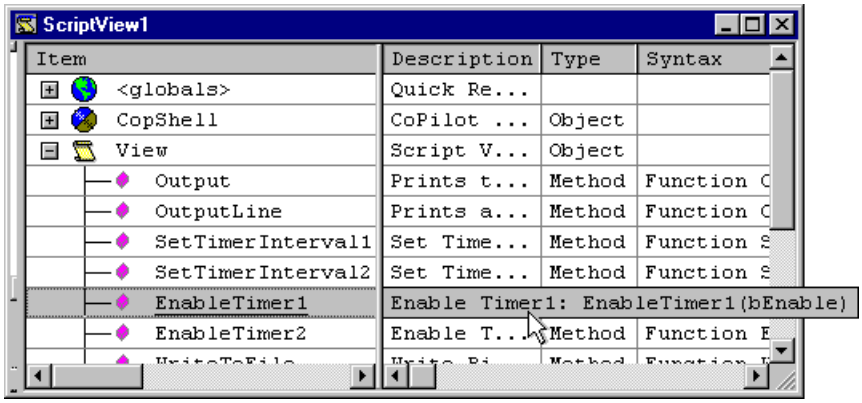

## Script Timers

### Built-In Timers

Timers can be used in scripts to trigger event handler routines at predetermined intervals. For example, script timers could be used to sample bus traffic by toggling monitor recording at selected intervals, compare values by taking a sampling of all data on the bus at timed intervals, or many other possible applications. Each of the two script timers are engaged individually through the ♦ EnableTimer method. The interval for each timer is set through the ♦ SetTimerInterval method. These methods are associated with the View object listed in the Object pane (see figure below).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




When a timer's interval elapses, the event handler routine OnScriptTimer# (where # is 1 or 2) is called. A subroutine must exist for each of the enabled timers (although it does not have to contain any code). A timer subroutine is illustrated in the example below.

### Timer Example

The script below illustrates the use of a single timer firing once every ten seconds (10,000 milliseconds). Timer 1 is enabled and the timer interval is defined in the ScriptStart routine. In this example, the OnScriptTimer1 routine is executed and the time-tag for BCMsg1 is written to the output window when the timer is fired.

```
Sub ScriptStart
    'Initialize Timers
    View.EnableTimer1(True)
    View.SetTimerInterval1(10000)  ' Ten Seconc
    ....
    ....
End Sub

Sub OnScriptTimer1
    Dim curTime
    curTime = CLng(BCMsg1.MsgTimeTag)
    View.OutputLine "Timer1: Time = " & CStr(c
End Sub
```

```
-- Script View Output --
Started Wed Nov 13 01:38:17 PM
Timer1: Time = 9.073878
Timer1: Time = 19.256783
Timer1: Time = 29.539721
Timer1: Time = 39.542577
Timer1: Time = 49.685475
Timer1: Time = 59.908394
End
```

Note: The interval shown in the output window (above) is slightly larger than the ten-second target. Timer intervals can be affected by the processing time of the routine and the speed of the host processor.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Scripting Principles

### Event-Oriented

VBScript is an event-driven, interpreted language and all routines are triggered by events. You can write a procedure for any scriptable event in CoPilot. The exception is the ScriptStart and ScriptStop routines, which are tied to the CoPilot Run and Stop buttons (events).

### Order of Execution

The basic components used in creating script in the Script View window include global statements and definitions, the ScriptStart routine, the ScriptStop routine, event handlers, and functions.

The location and sequence of these subroutines within the script is not important because they are always processed in the following order:

1. Globals
2. ScriptStart
3. Event handlers and functions
4. ScriptStop

When the CoPilot simulation is run , globals are executed first, and then the ScriptStart routine is called. While the simulation is running, event handler subroutines are processed whenever their respective event fires. The ScriptStop subroutine is called when CoPilot is stopped .

---

Note: The ScriptStart routine is executed before hardware devices (cards) are started.

---

### Syntax

All subroutines must begin with the prefix “Sub,” followed by their name and arguments (if any) within parentheses. Each subroutine must end with the line “End Sub.” Function blocks must be enclosed by the “Function” and “End Function” statements.

Condition statements (such as “if” or “while”) do not require a prefix, but must have an end line (for example, “End If”).

The beginning and ending statements for each block of code encapsulate it and establish scope.

### Globals

The global section is any code located outside of “Sub” and “Function” blocks. Globals usually appear at the top of a script to avoid confusion. Variables, constants, and statements that are declared or initialized globally are available in all subroutines.

### ScriptStart and ScriptStop

ScriptStart and ScriptStop must be included in each CoPilot script. ScriptStart is triggered by the Run  event and is often used to initialize data. ScriptStop is triggered by the Stop  event and is typically used to output results (to a file or the Output pane), reset data values, and release objects created within the script.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Event Handlers

Event handler subroutines are only executed in response to a triggering event. An event handler is named by combining the object name, underscore, and the property name. In the example below, subroutine "Field1_ValueEngr" is triggered by the "ValueEngr" property of the "Field1" object.

```
Sub Field1_ValueEngr
    'Output the value
    View.OutputLine Field1.ValueEngr
End Sub
```

Tip: To automatically add an event handler routine, right click on an object property in the Object pane that can trigger an event (small "E" icon) and choose **Add Event Handler** from the context menu.

All event handlers (and any subroutines or functions they call) should be written to exit as quickly as possible because processing time is shared with the CoPilot user-interface thread.

Note: Modifying a property within its event handler subroutine may cause an infinite loop to occur.

### Subroutines

A subroutine that is not an event handler must be called from another subroutine or function (and does not return a value, as functions do).

```
Sub ExportData
    Sheet.Cells(1,1).Value = Msg1.TimeTag
    Sheet.Cells(1,2).Value = Msg1.ValueEngr
End Sub
```

### Functions

Functions must are called from another subroutine or function. Functions return a single output to the calling routine through the function name.

```
Function DegF (Celsius)
    DegF = Celsius * 1.8 + 32
End Function
```

### For Further Reference

A help file with reference information about VBScript is included on the CoPilot installation CD (script56.chm). You may browse the CD and copy it onto your hard drive. Additional information, tutorials, and examples of VBScript are widely available on the Internet.

## Script Execution

### Running Scripts

Code created in a Script View window is executed when the CoPilot simulation is started. Multiple scripts can be run simultaneously by using multiple Script View windows, but this is not recommended.

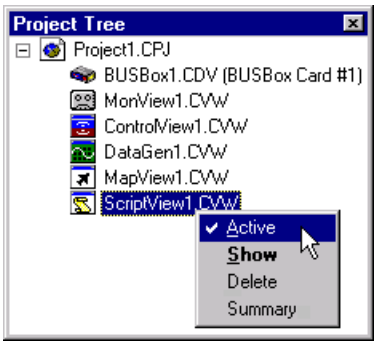


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note that closing a Script View window with the title bar minimize button hides it from view, but the display is still present in the Project Explorer and will run during simulation.

To prevent a script from running during simulation:

1. Right click on the Script View icon in the Project Explorer tree to open the context menu
2. Click the **Active** option to clear the checkmark and deactivate the Script View (see figure below)




Deactivating the view in this way will effectively pause the Script View window.

### Saving and Loading Script View Windows

To save a Script View window, select File | Save. To add a Script View in any CoPilot project, choose File | Open Hardware or View and browse to the location of your saved Script View component.

## Sample Script Library

### Library Features

Ballard maintains a library of sample scripts for use in the CoPilot Script View window. Comments in the header in each sample script describe the script objective, required objects, and ways to adapt the script. A README.TXT file in the Scripts folder contains a list of the sample scripts plus general instructions for using the each script.

### Locating Sample Scripts

The sample script library can be accessed from:

- The CoPilot installation CD
- The CoPilot Projects folder (copied during installation; default path is C:\CoPilot Projects\Scripts)
- *Appendix F: Script Examples*, which contains a few sample scripts discussed in detail
- Ballard's CoPilot Scripts page on the Ballard website at http://www.ballardtech.com/CoPilot_Scripting.asp

Visit the CoPilot Scripting page on the Ballard website often for an up-to-date listing of sample scripts.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### Importing a Sample Script

The CoPilot sample scripts, and any saved script file, can be imported directly into the scripting pane.

To import a script:

1. Right click in the Scripting Pane to open the context menu, point to **Import Script**, and either…
   - Click **Overwrite** to replace the contents of the scripting pane with the imported file
   - Or click **Append** to insert the selected file at the end of the code in the scripting pane
2. In the Select Import File dialog, click the browse button to launch the Open File dialog
3. Browse to the script you wish to import, select it, and click OK to close the dialog and load the script

You can browse to the folder of sample scripts that are installed with CoPilot or to a script of your own.

### Sample Script Applications

Scripts may be used for a variety of tasks in CoPilot. The list below illustrates where a script might be useful.

Data Generation:

- Conditional value based on incoming values, error conditions, etc.
- Create result (e.g., volume = length x width x height)

Data Translation:

- Custom data interpreter (partition raw data and instantiate engineering unit fields in the tree)
- Links between avionics protocols (e.g., translate and insert MIL-STD-1553 message into ARINC 429 labels)

Monitor Data:

- Start/Stop monitor (e.g., if altitude < *n*)
- Sample data over time (e.g., save a 5-second sample once a minute)
- Create markers (i.e., bookmarks) that can be placed in monitor files through scripts, then add monitor search feature to advance forward and backward to markers.

External Links:

- Place data/view information outside CoPilot (e.g., Excel® or Access® file, tables, histograms through pre-programmed graphics, etc.)
- Place results into standard reports in Excel for specified tests (time, date, project name, min/max values, errors, times, pass, fail, etc.)
- Place log file in Excel that lists time, date, and results for a sequence of runs
- Accumulate list of messages with errors in Excel or other file
- Trigger external signal (e.g., sounds, voice, lights, screen effects, etc.)



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Access an external database containing ranges or reference values for purpose of comparison

CoPilot Macros:

- Bring a specific CoPilot display to the foreground if specified conditions occur
- Save selected information on exit
- Construct and present run summaries, run time notes, dates, etc. on exit
- Create special CoPilot effects that can be engaged by script (e.g., screen effects, sounds, icons for status bar, etc.).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

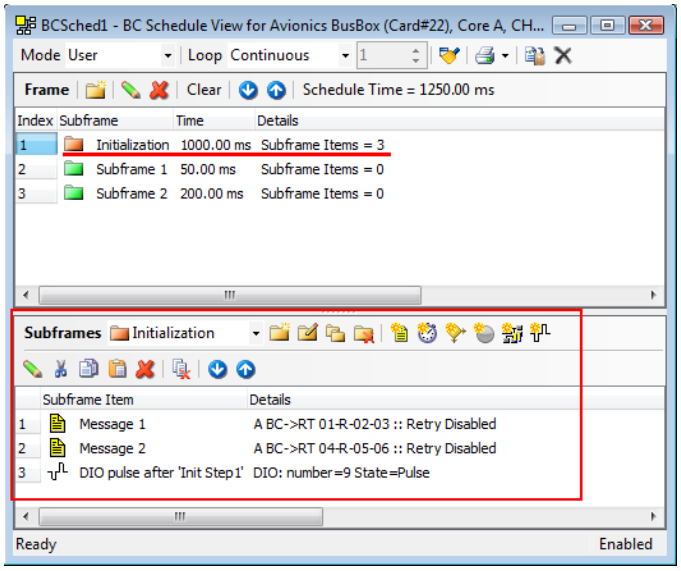
# Appendix I: Custom 1553 Schedule Examples

This appendix contains examples of custom MIL-STD-1553 schedules to illustrate how to create initialization routines, reuse subframe schedules, and change user defined operational schedule modes.

## Initialization Routines

An initialization routine is defined as a group of schedule operations, such as initialization messages, that are only called during the start-up sequence of the schedule.

As shown in the following image, there are 3 subframes. The first scheduled subframe contains the initialization messages and a following digital output pulse.

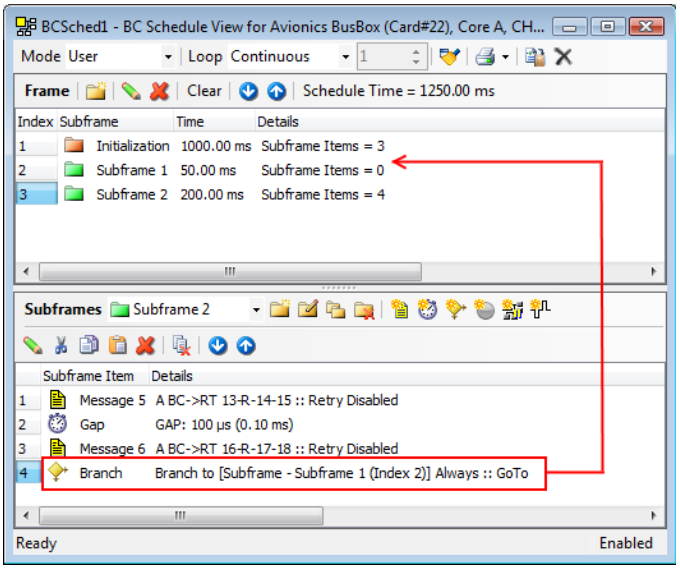



*BC Schedule with an initialization subframe*

A branch at the end of the last subframe (‘Subframe 2’)is added to skip the ‘Initialization’ subframe after the schedule runs through the first pass.



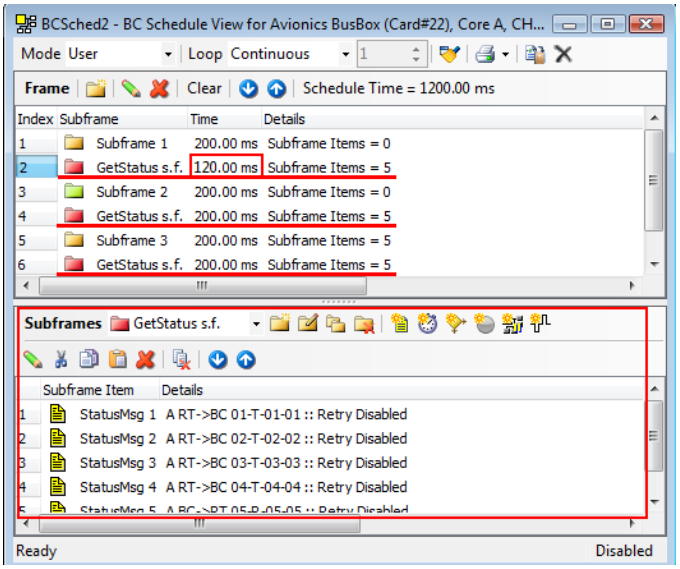
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com




*Unconditional Branch in a BC schedule used to skip the initialization subframe*

# Reusable Subframe Schedules

Subframes created in the BC Schedule contain references to messages, gaps, branches and other schedule entries (opcodes). The major frame can reference a subframe multiple times without having to recreate each instance of the desired subframe. This example creates four subframes where 'GetStatus s.f.' is referenced three times by the schedule. This example also shows the ability to modify the frame time for individual instances of a repeated subframe as shown with the frame time of 150.00 ms for schedule index 2.




*BS Schedule referencing the same subframe multiple times*



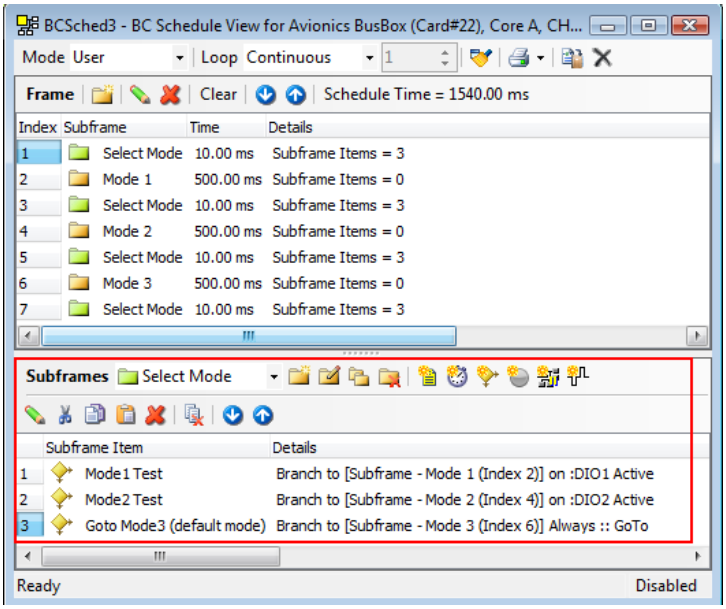
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Schedule Swapping and Schedule Modes

The following examples switch schedule operation between ‘virtual schedules’ where the virtual schedule consists of one or more group(s) of subframes. The first example uses DIO to switch between three different schedule modes. DIO schedule branching is supported by 4G and 5G hardware devices. The second example uses the status word values of a status message to determine when to change modes. Combinitions of the following examples (such as using a combination of conditional branching with script event handlers) may also be done to perform operations such as sending different bus controller messages when the SRQ bit in the status is set based on the value of a dataword.

## DIO Controlled Schedule

Control the operation of the 1553 Bus Controller with software to switch between schedules. This can be accomplished several ways, but one of those ways is to use DIO schedule branching. This example creates the ‘Select Mode’ subframe used to test the state of the two input discrete and change the operational mode of the BC. The schedule for each of the three different operational modes in this example are contained in unique subframes: ‘Mode 1,’ ‘Mode 2,’ and ‘Mode 3.’ On start, the first ‘Select Mode’ subframe acts as an initialization frame begins the BC schedule in the proper mode. After the mode subframe completes, the DIO values are tested and the mode is changed if necessary; otherwise, the current mode continues execution.




*BS Schedule using a subframe to test DIO values and switch operating modes*

## Software Controlled Schedule

To control a schedule via software, the schedule is first constructed as described by the DIO controlled schedule above. However, this requires hardware that supports either discrete that can operate as BOTH inputs and outputs (as are on available with most 5G devices) or output signals may be externally wired to the input lines that the schedule will branch on. Then via scripting, when a message



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

is received, a data value changes, or some other condition, the output discretes are set or cleared to control the schedule.

## Status Word Controlled Schedule

Similar to the DIO controlled schedules described above, a status word controlled schedule simply branches on the previous status word condition as opposed the the DIO state.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Appendix J: Revision History

The following revisions have been made to this manual:

**Rev Preliminary Date: May 30, 2008**

Preliminary release of this manual. Combined CoPilot 1553, CoPilot 429, and CoPilot AFDX manuals into a single document.

**Rev Date: July 1, 2008**

Major revisions and additions to this manual to document version 5.0 features.

**Rev A Date: January 21, 2009**

Documentation of new version 5.1 features and other minor revisions.

**Rev B Date: July 24, 2009**

Documentation of new version 5.12 features and other minor revisions.

**Rev C Date: February 18, 2010**

Documentation of new features and new hardware support and other minor revisions to the CoPilot software up through version 5.17.

**Rev D Date: January 04, 2012**

Documentation of new CoPilot version 6 features.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Glossary of Terms

## AFDX

Part 7 of ARINC 664 defines a deterministic network known as AFDX, which stands for Avionics Full Duplex Switched Ethernet. IEEE Standard 802.3 (Ethernet) is an integral part of the AFDX specification. The AFDX extensions to IEEE 802.3 address the special requirements (quality of service, reliability, etc.) of an aircraft environment. AFDX is a new standard providing much higher data rates than existing avionics databus protocols.

## ARINC 429

ARINC 429 is a commercial avionics protocol defining a local area network for transfer of digital data between avionics system elements.

## ARINC 664

ARINC 664 is a multi-part specification defining an Ethernet data network for aircraft installations. Part 7 of the ARINC 664 specification defines deterministic AFDX networks.

## ARINC 708

ARINC 708 describes the characteristics of an airborne pulse Doppler Weather Radar system intended for installation in commercial transport type aircraft.

## ATE

Automated Test Environment.

## Bandwidth Allocation Gap (BAG)

The Bandwidth Allocation Gap (BAG) defines how often a virtual link (VL) on an AFDX network can transmit.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## BCD

BCD, or binary-coded decimal, is a method of using binary digits to represent the decimal digits 0 through 9. In this format, four binary digits (bits) are allocated to each decimal digit.

## BNR

BNR, or two's complement binary notation, is a type of encoding that stores data as a binary number. Negative numbers are encoded as the two's complement of positive values. The first bit is used as the sign bit to indicate positive/negative sense.

## Broadcast

A broadcast message is a mode code message transmitted by the Bus Controller using a unique broadcast address (RT 31) as a means of transmitting to all of the terminals connected to the databus with a single message. RTs do not transmit a status word in response to broadcast commands, but may be polled by the BC to confirm reception.

## Bus Monitor

The terminal assigned the task of receiving bus traffic and extracting selected information to be used at a later time. (See also Sequential Monitor)

## Concurrent Monitoring

Concurrent monitoring is the ability of a Ballard board to actively monitor a MIL-STD-1553 databus at the same time it is simulating a Bus Controller and one or more Remote Terminals.

## Data Accept

In a weather radar system, data can be uniquely identified (or addressed) through the use of an Indicator address in the Data Accept field (bits 50 and 51) of the message header. In a system with two display units, data can be accepted or rejected by an "Indicator" (display unit) based on the Data Accept field. This allows two displays (or in rare cases two systems) to time-share the same ARINC 708 databus.

| Bit 50 | Bit 51 | | Field |
|---|---|---|---|
| 0 | 0 | Do Not Accept Data | 0 |
| 1 | 0 | Indicator 1 Accept Data | 1 |
| 0 | 1 | Indicator 2 Accept Data | 2 |
| 1 | 1 | All Indicators Accept Data | 3 |

## Data Set (DS)

An AFDX Functional Data Set (FDS) contains up to 4 different Data Sets, each holding related field definitions. The status of the Data Set is indicated by the FDS Functional Status Set (FSS).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Dataset

A dataset  is a CoPilot object (located in the Hardware Explorer) that holds a sequential collection of ARINC 708/453 1600-bit messages. Each dataset object holds radar display configuration settings (default or user-defined). Datasets can also contain time-tag and error information (for recorded data) and user-assigned formatting such as bookmarks and breakpoints.

## Design Mode

Control View windows support two operational modes: Design mode and Run mode. In Design mode, the user can create, move, and customize controls and configure the Control View window. For example, in Run mode a mouse click on a control might toggle its value; in Design mode the same mouse click would simple select the control so it could be repositioned. When CoPilot enters Simulation mode, all Control View windows are automatically shifted to Run mode.

## Discretes

Digital controls lines used for signaling, synchronization, triggering, and setting/receiving status from external equipment. Discretes are also referred to as digital inputs/outputs (or DIO for short).

## Docking Pane

Dockable windows used for display and configuration that may be moved around or undocked from the CoPilot application to change the display layout. Docking panes include the Hardware Explorer, Project Explorer, Protocol Browser (version 5), Command Line, and many others.

## Dynamic Bus Control

A mode code transmission from the Bus Controller to a Remote Terminal offering control of the bus. When the RT responds with an acceptance status word, control passes to the RT and it becomes the new BC.

## Functional Data Set (FDS)

An AFDX Frame contains a payload, which after 32-bits of reserved space, contains 1 or more Functional Data Sets. Each Functional Dataset has up to 4 Data Sets (DS), which contain field definitions. At the beginning of the Functional Data Set is a 32-bit Functional Status Set (FSS).

## Functional Status Set (FSS)

The AFDX Functional Status Set defines four functional status bytes, each corresponding to one of the four data sets. These bytes reflect the state of their respective data sets:

- No Data
- Normal Operation
- Functional Test



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- No Computed Data

## Edit Mode

Edit mode is the CoPilot state that exists when the Stop button is depressed. Data initialization, configurations, and other settings that need to be defined before run time are specified in Edit mode. There is no interaction with the databus or Ballard board until Simulation mode is initiated through the Run button.

## Engineering Units

Converting the ones and zeros that come across the databus into recognizable "engineering units" can include defining the data by data field, assigning a meaningful name to the field, decoding the information with an interpreter, and associating units of measurement with the data.

## Error Injection

Error injection is a method of modifying BC and RT message so they do not comply with the MIL-STD-1553 specification. Non-compliant messages are typically used to test the response of a hardware system to bus faults.

## Hardware Explorer

The Hardware Explorer is a view of the hardware associated with the project. The icon in the Hardware Explorer represents state for some objects as shown in the *Appendix B: Hardware Object Icon Reference*. Depending on the databus protocol(s) supported you will see a hierarchical branch structure containing Bus Controllers, Remote Terminals, Equipment, Labels, Networks, VLs, Ports, fields or other items specific to that protocol. The Hardware Explorer is used to specify and display the definitions and structure of the information being communicated over the databus.

## List Buffer

List buffers are lists of data values for messages where items in the list are automatically advanced by the hardware synchronous to the transmission of the message (i.e. via schedule).

## Network (AFDX)

An AFDX network defines a system for interconnections similar to Ethernet. However, the ARINC 664/AFDX networks are redundant networks for reliability and quality of service (QOS) requirements. Ballard Technology, Inc. hardware devices offer independent network operation for test purposes of interacting with more than one system.

## Message (ARINC 664/AFDX)

A message, as defined and used in CoPilot is synonymous with an instance of an ARINC 664/AFDX port on the databus. The payload data and associated data interpreters are defined by the port 'message'.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Message (ARINC 708)

A message, as defined by ARINC 708, is a 1,600-bit word preceded and followed by a sync pulse. The 708 word is divided into a 64-bit header and 512 three-bit range bins holding display data.

## Message (MIL-STD-1553)

A message, as defined by MIL-STD-1553, is the transmission of a command word and (if they are specified) status and data word(s). There are six kinds of 1553 messages: BC to RT, RT to BC, RT to RT, and three types of mode code messages.

## Mode Code

Mode Control Commands are a series of BC commands defined by MIL-STD-1553B to aid the Bus Controller in the management of the multiplex databus and the electrical control of the Remote Terminals (for example, Reset Remote Terminal).

## Pane

Dockable windows used for display and configuration that may be moved around or undocked from the CoPilot application to change the display layout. Docking panes include the Hardware Explorer, Project Explorer, Protocol Browser (version 5), Command Line, and many others.

## Radix

The radix is the base of a system of numbers (such as 10 of the decimal system and 16 of hexadecimal). For example, the number 10 in hexadecimal is equivalent to 16 decimal or 20 octal.

## Run Mode

Control View windows support two operational modes: Design mode and Run mode. In Run mode (or User mode), the number and position of controls are fixed and any input from the user (such as a mouse click) triggers the appropriate response from the control. For example, if a control is a data source, in Run mode the user can modify the data stream by using the mouse to turn a dial or flip a switch. When CoPilot enters Simulation mode, all Control View windows are automatically shifted to Run mode.

## Scripting

Scripting in CoPilot is the act of writing scripts (or modules) that perform automated tasks in CoPilot. These scripts can be written with VBScript using the Script View in CoPilot Professional, or using the Python Script Editor provided with CoPilot ATE. Development using Python in ATE is recommended due to better functionality, performance, and user-friendliness. Scripts manipulate CoPilot objects by using the defined and documented “CoPilot Automation Model”



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## Security

Security in CoPilot refers to project security. CoPilot Professional includes the ability to change security levels of a certain aspects of a CoPilot project. This can range from runtime configuration restrictions to locking down projects to allow only a subset of the CoPilot functionality. See the "Project Security" section of Part 3 of this manual for additional information.

## Sense

Sense with CoPilot interpreters is an annotation indicating the direction. Positive Sense is the direction of a positive value, such as up, foreward, north, etc. Negative Sense is the direction of a negative value, such as down, backward, south, etc.

## Sequential Monitor

If you have a newer Ballard hardware (4G or 5G), you will have a Sequential Monitor in addition to the Bus Monitor. The Sequential Monitor does not represent a physical terminal; rather, it allows a single point of configuration with the capability to concurrently monitor/record the bus traffic from multiple avionics databuses on the same hardware core. When both monitors are present for a 1553 databus, the Bus Monitor controls the flow of 1553 traffic into the Sequential Monitor, where it collects and can be viewed by opening a 1553 Monitor View window.

## Shell

The shell is the object used in CoPilot ATE to access the CoPilot Automation model. The shell objects supports many methods and properties for controlling CoPilot operation automatically, as well as providing access to the Hardware Explorer tree via the Cards collection (`shell.Cards`).

## Shell-Level Scripting

Writing scripts that perform project configuration and manage CoPilot using the shell alias can be referred to as Shell-level scripting. This type of scripting offers the flexibility of allowing users to run their code at any time during simulation or edit mode, including running commands directly from the Command Prompt.

## Simulation Mode

Simulation mode is the CoPilot operational state that exists while the Run button is depressed. During Simulation mode, data is transmitted or received on the databus through the Ballard hardware, all active CoPilot objects and displays are animated, and menus change to reflect the commands available during run time. CoPilot returns to Edit mode when the Stop button is engaged.

## Software Playback

CoPilot enters Software Playback mode when playback has been enabled and the Run button is pressed. CoPilot uses prerecorded data during playback and does not communicate with the databus or the Ballard board. Otherwise,



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

playback features and user interface are similar to those in Simulation mode. CoPilot returns to Edit mode when the Stop  button is engaged.

## Start Page

The Start Page in version 5 provides an easy way to access or create projects, access help documents and tutorials, and display detected hardware information. It is commonly the first page that comes up when CoPilot is started, though this feature can be turned off. The Start Page was replaced with **Backstage** in version 6.

## Time-Tag

The time-tag is a value derived from an internal clock on the Ballard hardware device. Each message in the Bus Monitor has a time-tag attached. A linear time-tag is sequential and a delta time-tag is incremental. Time-tags are range between 32-bits and 64-bits depending on the hardware. For example, 4G and 5G products use 64-bit values to represent the IRIG time.

## View

View windows are used to display, organize and modify data. CoPilot categories view windows into Standard and Professional views.

## Virtual Link (VL)

A Virtual Link is defined as a special one-to-many connection in the ARINC 664/AFDX protocol. A VL is identified by the VL number (or the destination MAC address).

## Word

A MIL-STD-1553 word is 20 bits, consisting of a 3-bit sync, 16 bits of data, and 1 parity bit. There are three types of words: command, status, and data.

## Workspace

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. Use the Window menu to tile, cascade, or select/open view windows and organize Workspaces.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE INTENTIONALLY BLANK.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# Index


.

.CDV 56
.CPB 56
.CPJ 56
.CVW 56
.MDB 212, 256
.MON 56
.PY 56

1

100 kHz 226
1553
  copy fields 159
  database 157
  field database 155
  field save 157
  message rate 52
1553 hardware playback 330
1553 view 102
1553 view conversion 203

3

377 224
3G hardware 32, 209, 239, 441

4

429
  custom schedule 228
  message range 52
  message rate 52
  overview 217
  rate-based schedule 226
  skip message 233
  tolerance 226
  transmit 52
429 channel
  receive options 223
  transmit option 224
429 hardware playback 332
429 user scheduling 228
429 view 102
429View 251
453 269
4G
  hardware features 66
4G hardware 32, 127, 147, 208, 239, 441

5

5G
  hardware features 66
5G hardware 32, 127, 147, 208, 239, 441

A

absolute word count error 210
Access a File Open or File Save Dialog 478
access database 212, 256
acos 401
action scope 49
active breakpoint 417
activex controls 367, 447
ActiveX controls 381, 445, 458
activity 307
add alias 413
add control 375, 377
add control to form 377
add hardware 63
add port 310
add random noise to data pattern 471
add to watch 414, 428
add VL 310
add watch 409
advanced features 21, 323
advanced hardware features 66
AFDX 32, 299
AFDX database 316
AFDX options 53
Aircraft Instrument Properties 449
aircraft instruments control 446
aircraft instruments control 448
aircraft instruments library 448
  airspeed indicator 450
  altimeter 451
  artificial horizon 451
  automatic direction finder (ADF) 452
  climb rate indicator (CRI) 453
  compass 453
  course indicator 454
  heading indicator 455
  horizontal situation indicator (HSI) 455
  omni bearing indicator (OBI) 456
  radio magnetic indicator (RMI) 457





Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

turn coordinator 457
Aircraft Instruments Library 449
airspeed indicator 450
alarm 224, 233, 364
alias 413
assignment 412
aliases
reference 406
all messages 212
all sync 147
alphanumeric LED 458
alphanumeric LED control 446
alternate bus 132, 134
altimeter 451
amplitude
parametrics 260
amplitude modulated
IRIG 66
amplitude modulation 113
amplitude modulation
transmit 69
analysis 248, 300
analyze 42
analyzer 48
analyzer toolbar 48, 109
angular gauge 459
angular gauge control 446
append 172, 245, 305
approach 96
ARINC 419 31
ARINC 429 31, 217, 313
database 243
object hierarchy 219
overview 217
schedule options 52
ARINC 429 schedule 227
ARINC 429 auto detect 220
ARINC 429 channels 220
ARINC 429 hardware playback 332
ARINC 429 options 52
ARINC 429 view conversion 251
ARINC 453 269
ARINC 575 31
ARINC 664 options 53
ARINC 664/AFDX 32, 299
object hierarchy 299
ARINC 708 32, 507
object hierarchy 263
ARINC database 238, 256
ARINC schedule export 231
ARINC schedule import 231
ARINC view 251
array 416
artificial horizon 451
ASCII 313
asin 401
asynchronous messages 135, 225
asynchronous tranmsission 131
asynchronous transmission 135, 225
1553 201
ATE 168
alias assignment 412
autocomplete 410, 423
auto-indent 411
automated test manager 400
breakpoints 417
calltips 410
code folding 411
collapse all 411
command history 424
command prompt 362, 400, 423
command prompt pane 44
data types 416
debugger 400, 416
events 415
expand all 411
help 401
helper functions 424
import modules 410
macros 428
multiline input 424
object browser 44, 400, 411
object model 360
output 425
output pane 44, 400
projects 401
Python script editor 400
syntax check 409
tab nanny 411
test manager 418
test manager pane 45
watch window 400, 427
watch window pane 45
ATE (automated test equipment) 399
atm
automated test manager 422
ATM 418
ATM (automated test manager) 400
atm alias 407, 418
atm output stream 422
ATM output stream 418
author’s name 51
auto busy 147
auto detect
429 messages 232, 236
ARINC 429 equipment 220, 224
auto detect equipment ID 259
warning 259
auto detect hardware 51
auto detect subaddress 147
auto detect VL 300
auto detect VLs 316



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

auto detection 238, 316
  parity error 234
auto hide 45
Auto RSN 319
auto scope 427
auto scroll 76, 80
auto speed 222
autocomplete 410, 423
auto-complete 401
auto-detect 242
auto-detect hardware 513
auto-detect labels 223
auto-detection 300
auto-indent 409, 411
automate tasks 399
automated test manager 400, 407, 418, 422
automated test manager loops 420
automated test manager pane 45
automated test manager properties 420
automatic direction finder (ADF) 452
automatic master script file creation 50
automatic start 50
automation 359
automation examples 363
automation model 359, 360, 399, 400, 407
avionics full duplex switched Ethernet 299
avionics level discretes 32

## B

background color 395
background picture 395
background processing 404
background text control 446
backstage 42
backwards 80
BAG 300, 315
Ballard Technology 34
Bandwidth Allocation Gap 315
bandwidth sharing 315
BC 115, 168
  external trigger 117
  pause 117
  properties 117
  schedule 118
  schedule stepping 117
  sync 118
BC context menu 116
BC controller
  pause 116
BC message 128
  bus 132
  configuration 130
  context menus 130
  continuous 132
  message editor 133
  retries 134
  single shot 132
  skip 132
BC message properties 131
BC messages 129
  configuration 116
  data 135
  rate 135
BC pause
  schedule 128
BC schedule 115, 118
  BC message 123
  branch 125
  default schedule 118
  editor 122
  export 121
  frame 123
  gap 124
  import 121
  message 123, 128
  rate-based schedule 118
  user schedule 118, 120
  view 122
BC schedule errors 120
BC schedule export 121
BC schedule import 121
BCD 162, 218, 237, 313
BCD interpreter 163
BCSched 120
big endian 52, 153
binary 92, 204, 252
Binary Coded Decimal 218, 313
binary editor 93
binary editor control 446
binary radix 189
binary string example 469
bit count error 210
BIT tests 65
blend 319
blending 319
BLS file format 140
BNR 162, 218, 313
BNR 48-bit 162
BNR 64-bit 162
BNR interpreter 163
bookmark filter 249
bookmark prompt 50
bookmarks 80, 193
bool 416
boolean 313
branch 125
  DIO 127, 128
branch error 127
breakpoint 80, 417
  hardware playback 332
  software playback 347



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

breakpoints 193, 417
  708 playback 268
  for software playback 342
  hardware playback 333
broadcast
  disable 114
  RT31 114
broadcast mode 114
broken link 351
broken links 252
broken sink 351
broken source 351
btimdb 213, 258, 317
btimdb.exe 89
built-in aliases 407
bus A 132
bus B 132
bus control error injection 318
bus controller 115
  configuration 116, 117
  schedule 115
bus monitor 168
  1553 169
bus monitor versus sequential monitor 169
bus scheduling 115
bus termination
  configuration 114

## C

calendar control 446
calltips 410
candidate equipment 221
CAPS 163
capture filter 74, 141
  network 304
  port 304
  VL 304
capture filters 175, 304
  1553 176
  ARINC 429 245
CDU 264
CDV file extension 56
change interpreter 161, 241
change password 432
change sequential monitor interpretations 79
channel
  ARINC 429 220
channel parity 243
channel parity error 243
channel pause 223
  1553 113
channel sample rate 113, 223
channel speed 222
channel stop
  1553 113
channel summary 231
check box control 446
check syntax 409
clear channel 223
clear pause on run 245
climb rate indicator (CRI) 453
code editor 407
code folding 409, 411
collapse all 411
collapse children 50
collapse node 50
color 204, 252
color picker control 446
column
  save 80
column settings 80
column width 206, 255
combo box control 446
command history
  ATE 424
command line 44
command prompt 44, 91, 362, 400, 401, 423
  autocomplete 423
  command history 424
  multiline input 424
  syntax highlighting 423
command prompt pane 44
command words 128
comment 235
common dialog control 446
compare record 78
compare records 250
compass 453
complex ramp 97
concurrency 404
concurrent monitoring 72, 170, 302
concurrent recording 72, 302
conditional branch 125
  DIO 127, 128
configure view 80
continuous message 132
continuous schedule 123
control 447
  data modifier 97
control add 375
control database 375
control definition 378
control grab handles 375
control library 385, 458
  angular gauge 459
  data modifier 460
  GMS Percent 462
  knob 460
  LED 461
  linear gauge 461
  slider 463



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

control link direction 375
control links 374
control palette 376
control property browser 375
control save 378
control selection 375
control selection gallery 377, 378
control type 377
control view 374
  429 messages 236
control view controls 445
control view customization 375
control view library
  aircraft instruments 448
  alphanumeric LED 458
  selector knob 462
  strip chart 463
  toggle switch 465
Control View Library of Controls
  GMS Linear Gauge 461
  GMS Percent 462
  GMS Slider 463
controls
  aircraft instruments 448
controls 367, 374, 381, 445, 447
controls
  strip chart 463
conventions 35
convert doubles control 446
CoPilot
  automation 359
  compatible hardware 61
  object model 360
  projects 53
CoPilot automation 359
CoPilot automation model 400
CoPilot Conventions 59
CoPilot enviornment 40
CoPilot hardware keys 61
CoPilot Menus 47
CoPilot Object Model 360
CoPilot operation 39
CoPilot options 50
copilot output stream 425
CoPilot Panes 43
CoPilot professional 21, 323, 445
CoPilot Professional 208, 256, 315, 367
CoPilot shell 407
*CoPilot Shell* 40
CoPilot standard 21, 323
CoPilot Standard 367
CoPilot Views 46
CoPilot Workspaces 43
copxctrls 447
copy 242
copy fields 159
copy target 414
copying engineering units 88
cos 401
cosh 401
course indicator 454
CPJ file 50
CPJ file extension 56
CRC 319, 320
create a simulated monitor file in excel 473
Create a Web Page 479
Create and Display a Text File 475
create field 158
create new field
  1553 154
create project 42
create test 421
create view 72
creating new project 54
creating OLE documents 105
currency 416
current insturction 417
custom schedule 118, 227, 228
custom script interpreter 163, 164, 218
custom speed 222
  parametrics 259
custom toolbar 49
custom user display 188
custom user display mode 185
customer support 37
customization 49
cutput pane
  copilot output stream 425

## D

data
  parity 244
data buffer 239
data collection modes 172, 245, 305
data count 201
data edit 91
data editing
  SA 150
data editor 150
data field
  ARINC 429 219
  interpreters 161
data fields 151
  create 1553 154
  editing 158
data generator
  complex ramp 97
  data list 97
  delay 96
  period 96
  phase 96



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

type 95
data initialization 151
data list 96, 227
data modifier 460
data modifier control 97, 446
data parity 219
data plot 79, 196
data recovery 51, 74
data replay 325
data sets 300
Data Sets 310
data types 217
database 83, 88, 89, 213, 237, 238, 256, 258, 316, 317, 375
1553 212
ARINC 429 243
loading definitions 256
RT load 141
RT save 141
saving definitions 213, 257
sharing 213, 258
database export
1553 212
database import 140
1553 212
dataMars 140
dataset 269
date picker control 446
date time picker control 446
debugger 400, 410, 416
watch window pane 427
debugger breakpoints 417
DEC 163
decimal interpreter 163, 166
decimal label display 233
decimal radix 189
default BC message rate 135
default frame time 118
default message rate 52, 120
default schedule 118, 119, 226
default VL definition 316
define engineering units 84
delay 96, 236
hardware playback 333
delete all links 351
DeleteAlias 425
delta display 190, 248
delta mode 247
demo card 37
demonstration hardware 37
description panel 413
design mode 374
detail pane 307
detect channel speed 222
detect hardware 51
detect labels 223
detected hardware 42, 51, 513
diagnostic tests 65
dials 374
digital I/O 32
digital input 68
digital input/output 32
digital output 68
DIN 67
DIO 125
schedule 127
sync lines 67
trigger lines 67
DIO lines 32
DIO pulse 127
DIO schedule branch 125, 127
direct memory access 302
direct-coupled 114
direction 95
disable Auto RSN 319
disable auto-detect 242
discrete 32, 125
avionics level 32
ttl 32
discrete interpreter 163, 166, 218
discrete wiring 33
discretes 32
display customization 49
display filter 191
display filtering
ARINC 429 248
display filters 74, 77, 78, 175, 190, 307
display pane 43
display radix 189
display settings 80, 247
display window 46
display windows 101, 369
distortion error 210
DMA 302
DMA mode 302
docking pane
auto hide 45
docking stickers 45
docking panes 41, 43
docking stickers 45
document 57
documentation 360
double precision 312
DoubleToValue 165
DOUT 67
drag and drop 242
drag and drop to copy 88
drag-drop 308, 313
DS 300, 310
duty cycle 96
dynamic bus control 147



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## E


edit 91
edit BC message data 133
edit data
  1553 field 168
  429 fields 241
  429 message 239
edit engineering units 85, 314
edit fields 158
edit links 351
edit message
  monitor view 195
edit schedule 122
efficiency 247
ELS file format 140
enable interval test 235
enable message 233
end index 80, 328
  software playback 347
engage software playback 339
engineering unit 151, 204, 252, 312
engineering unit editor 91, 92
engineering unit interpreters 161
engineering units 83, 238, 241
  editor 85
engineering units database definition 88
engineering view 201, 251, 301, 313
  429 messages 236
  AFDX 313
engineering view icons 207
environment options 50
equipment id 236
equipment ID 217, 220, 239
equipment ID detection 259
equipment identifier 217, 220
error
  engineering view 207, 252
error control 211, 212
error definition 209
error detection 243
error display 207, 252, 385
error filter 249
error injection 66, 311, 312, 318, 320
  1553 error types 209
  absolute word count 210
  AFDX 319
  ARINC 429 234, 259
  bit count 210
  broadcast 114
  bus control 318
  control 211
  frame 318
  gap 210
  inter-message gap time 259
  inverted sync 210
  manchester 210
  overview 208
  parametric frame frequency 309
  parity 210, 259
  relative word count 210
  response time 208
  response time 142
  RSN 318
  schedule 318
  sync 210
  tagging messages 210
  view window 209
  word count 210
  zero crossing distortion 210
error injection overview 208
error injection view 209
Ethereal 303
evaluation 37
even parity 219
event 233
event handler 415
event handling 224
events 214, 260, 320
  ATE 415
example
  binary string 469
example script library 467
example scripts 467
  custom interpreter 165
examples 363
Excel 303
exception 405
expand all 411
export 78, 121, 190, 231, 248, 303
export filter 78
export hardware 69
export monitor data 197
export range 78
ExportToFile 426
ExtDIORd 127
ExtDIORd 33
ExtDIOWr 127
ExtDIOWr 33
extern trigger input 132
external input 234
external RT 136, 147
external sync 67, 69, 118, 132, 225, 232, 234
external sync mode 118
external sync output 132
external sync outputs 33
external trigger 67, 68, 117, 132, 212, 225, 232, 234
external trigger inputs 33
external trigger step 234
EXTRIG 67




Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## F


fail 422
FDS 300, 310
field 312
  1553 152
  ARINC 429 219
field context menu 160
field data 161
field definitions
  1553 152
field properties 160
field view 102
fields
  1553 151
  ARINC 429 240
file extensions 56
file indexing 51
file recovery 51, 74
file summary 57
files properties 57
filter 190, 245
  properties 176
  recording controls 178
filtering
  capture 74
  display 74, 77, 78
filters 176
fire default filter message event 224
FireReceivedDefaultFilterMsg 224
flat edit control 446
float 416
floating point 312
form display 186
form display mode 185
form mode 80
frame 123, 311
frame errors 318
frame time 118
frame time frequency 319
free-running schedule 227, 228
full display 186
full display mode 185
function type 95
functional data sets 300
Functional Data Sets 310


## G


gain 265
gallery 378
gap 124
gap error 210
gauges 374
general purpose controls 458
generate data from monitor record count 471
generate random data values 472
generate sine wave data values 472
generation 441
generic labels 237
getting started guides 36
global options 50
global scope 427
global variable 404
goto 77, 125
  monitor view 194
grab handles 375
grid 375
grids
  map view 394
group box control 446


## H


hardware 61
hardware capability
  1553 109
hardware configuration 64
hardware definition 69
hardware detection 42
hardware explorer 44, 62
  monitor filtering 176
hardware generation 441
hardware import/export 69
hardware installation 62
hardware keys 61
hardware level
  1553 109
hardware playback 80, 325
  limiting 328
  pausing 333
  preparing the data source 327
  running 332
  setting up 330
  step 334
hardware playback export file 328
hardware playback resume 334
hardware playback vs. software playback 325
hardware settings
  advanced 66
hault 234
heading indicator 455
heart-beat 227
heavily loaded buses 180
help 401
  Python 401
help documentation 360
helper functions 424
hex 92, 204, 252
  Python 485
hex edit control 447
hex editor 93





Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

hex pane 307
hex radix 189
hexadecimal 311
hide code 411
high speed 222
horizontal situation indicator (HSI) 455

## I

I/O 32
ICD 83, 213, 258, 317
icons 437
IEEE 32-bit interpreter 167
IEEE 64-Bit interpreter 167
IEEE 754 167
IEEE interpreter 163
import 79, 121, 231
import hardware 69, 219
import module 402, 409, 410
import RT from database 140
inactive breakpoint 417
indentation guides 409, 411
independent network 308
index
  708 playback 268
index files 73
indexing 51, 74
InitEvents 415
input 132
input control 446
input line selection 68
input trigger 234
installation 62
installation requirements 29
instruction 417
int 416
integer 312
interface control document 83, 213, 258, 317
interface ID 320
inter-message gap time 259
interpreter 238, 312
  1553 151
  1553 BCD 162
  1553 BNR 162
  1553 BNR 48-bit 162
  1553 BNR 64-bit 162
  1553 CAPS 163
  1553 custom script 163
  1553 DEC 163
  1553 decimal 163
  1553 discrete 163
  1553 IEEE 754 32-bit 163
  1553 IEEE 754 64-bit 163
  1553 MILSTD1750 163
  1553 simple scalar 163
  BCD 163
  BNR 163
  change 161, 241
  custom script 164
  decimal 166
  IEEE 32-bit 167
  IEEE 64-bit 167
interpreter definition 161
interpreters
  discrete 166
  simple scalar 167
interval 207, 226, 253
interval mode 247
interval test 235
inverse bus enable 320
inverse bus order 320
inverted sync error 210
inverting parity 259
IP header 320
IP layer 320
IRIG 113
IRIG amplitude modulated signaling 66
IRIG timer 66
IsInputDiscrete 33
IsOutputDiscrete 33

## J

jitter 300

## K

keep-alive 227
keyboard shortcut 416
keystroke 122, 468
keywords 401
knob 460
knob control 446

## L

label 218
label 277 237
label 355 237
label auto detection
  parity error 234
label control 446, 447
label sample rate 223
labels 377 224
LDU 31
LED 461
LED control 446, 458
legalized SA 150
libpcap 303
licensing 61
limit display 248
limiting hardware playback 328



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

line numbers 409, 411
line selection 68, 69
linear display 190, 248
linear gauge 461
Linear Gauge 461
linear gauge control 446
link status 351
linking fields 349
linking messages 349
linking objects 349
links 374
  broken 252
list box control 447
list buffer 227, 232
  ARINC 429 239
list view control 447
little endian 51, 153
load 316
  database 256
load field
  1553 database 155
load project 42
load schedule 121, 231
local scope 428
lock display for edit 311, 312
locked 431
log 401
log file 175
  sequential monitor 247
logical errors 318
long 416
lookup detected VLs 316
lookup label 238
loop 95
  708 playback 268
loop continuously 420
loop count 95, 420
loop schedule 123
loops 420
low speed 222
LRU 31

## M

MAC layer 320
macro dialog 428, 430
macro toolbar 429
macros 428
major frame 123
major frame time 124
manchester error 210
map view 388, 389
  maps 391
  object options 393
  objects 392
  views 393
map view context menu 390
map view grids 394
map view properties 390, 395
map view waypoints 394
maps 396
master script 50, 402
  OnProjectLoad() 403
master script file 402
master script file creation 50
master script functions 403, 405
master script restore 422
math functions 401
math module 410
max delay 236
maximum allowable transmit range 226
maximum delta time 235
maximum transmit range 52
MC 113
MC sync 147
measured min time 235
measured transmit interval 235
member panel 413
menu bar 47
menus 47
message 128
  1553 128
  ARINC 429 232
message assignment 242
message box control 446
message configuration 130
message data
  1553 135
message editor 133
  1553 128
message hints 242
message interval 235
message list 129
message list display 269
message options
  ARINC 429 233
message parity error 243
message processing 242
message properties 232
message range 52
message rate 52
  BC 135
message retries 134
message retry 133
message sample rate 113, 223
message single-shot 119
message skip 119, 233
message timeout 233
message timing 227
message view
  1553 201
MIB 311



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MIL-STD-1553 30, 109
MIL-STD-1553 database 212
MIL-STD-1553 options 51
MIL-STD-1553 properties 51
MIL-STD-1553 Protocol 30
MIL-STD-1553 view conversion 203
MIL-STD-1553A 30
MIL-STD-1553B 30, 109, 142
MILSTD1750 interpreter 163
minimum delta time 235
minor frame 123
mode 265
mode code 144, 147
  SA 0 113
  SA 31 113
mode code configuration 113
mode code legalization 141, 144
  RT 137
mode code sync 147
Model B 72, 301, 302, 318
modify link 351
modifying fields 152
modules 401
MON file extension 56
mondata file 72
monitor
  selective 141
  summary 248
  summary export 248
monitor all 150, 245
monitor bookmark 50
monitor capture filters 175
  1553 176
monitor configuration 171
monitor data 72
monitor data collection modes 172, 245, 305
monitor data collection settings 247
monitor display filter 191
monitor display filters 175
monitor filters
  1553 176
  capture filters 175, 176, 178
  display filters 175, 190
monitor options 223
monitor pause 244
monitor plot 79
monitor protocol 44, 208, 255
monitor record 303
monitor record comparison 250
monitor recording modes 174, 246, 306
monitor statistics 181
monitor view
  multi-select 78, 199
monitor view 72, 75
monitor view 180
monitor view 181
monitor view
  Configuration 183
monitor view
  context menu 183
monitor view
  1553 184
monitor view
  display mode 185
monitor view
  display format 186
monitor view
  display formats 186
monitor view
  user-defined displays 188
monitor view
  data radices 189
monitor view
  time-tag 190
monitor view
  export 190
monitor view
  analysis 190
monitor view
  bookmarks 193
monitor view
  breakpoints 193
monitor view
  search 194
monitor view
  navigation controls 194
monitor view
  goto 194
monitor view
  step 194
monitor view
  recording controls 195
monitor view
  message editing 195
monitor view
  data plot charts 196
monitor view
  export 197
monitor view
  import records 199
monitor view
  import options 199
monitor view
  printing records 200
monitor view
  time-tag 248
monitor view
  export 248
monitor view
  analysis 248
monitor view 303
monitor view 303



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

monitor view 307
monitor view 307
monitor view 308
monitor view display filters 190
monitor view form display 186
monitor view full display 186
monitor view print options 200
monitor view radix 189
monitor view tool bar
  1553 184
monitor view toolbar 307
month calendar control 447
most recently used 51
moving map 388
moving map data 396
MRU 42, 51
multiple BC schedules 120
multiple control selection 375
multiple custom schedules 230
multiple schedules 119, 226
multi-select 199
  control view 375
  monitor view 78, 199
My CoPilot Temp Data directory 74

## N

naming controls 447
ndionum 127
negative sense 512
network statistics 309
network view 300
Network View 318
  error injection 309
  statistics 309
new link 350
new test 419
no operation 128
None 416
non-standard speed 222
normalization factor 315
notice 2 30, 109

## O

object 416
object browser 400, 406, 411
  ATE 44
object events 415
object hierarchy
  ARINC 429 219
  ARINC 664/AFDX 299
  ARINC 708 263
object model 360
object options 393
object panel 413
object reference 406
octal 92, 204, 252
octal editor 93
octal radix 189
odd parity 219
OLE automation 359, 364
OLE documents 103
OLE Documents 106
omni bearing indicator (OBI) 456
OnProjectClose() 403
OnProjectLoad() 403
OnProjectRun 415, 422
OnProjectRun concurrency 404
OnProjectRun() 403
OnProjectStop 415, 422
OnProjectStop() 403
Opaque 313
Open a Web Page 479
operating limit error 385
operating limit max 165
operating limit min 165
OperatLimMax 165
OperatLimMin 165
optimization 72
optimize columns 80
options 432
other resources 35
out of range 207, 252
output 132
  ATE 44
  script view 493
output alias 407
output line selection 68
output pane 400, 405, 425
  other output stream 426
  Python output stream 426
  toolbar 426
  user output stream 426
output polarity 68
output stream
  atm 422
  ATM 418
output stream saving 426
output synchronization 225, 234
overwrite 172, 245, 305

## P

packet blending 318, 319
palette 376
pane window 43
panes 41, 43
parametric 222
parametric amplitude 69, 113
parametric frame frequency 319
parametric frame time frequency 318



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

parametrics 259, 260
parity 31, 32, 142, 243
  ARINC 429 219, 234
  inverting 259
parity as data 219, 244
parity error 210, 224, 234, 243, 244
  429 259
parity rules 244
pass 422
pass with warning 422
password 431
pattern test 233
pause 244
  708 playback 268
  BC 117
  MIL-STD-1553 112
  software playback 347
pause channel
  1553 113
pause hardware playback 333
pause message 233
pause software playback with breakpoints 342
payload 300, 310
payload data 320
payload errors 318
percent control 446
percentage indicator 462
performance 75
performance optimization
  recording 72
performance setting 242
period 96
periodic message 135
permissions 432
phase 96
pi 401
picture control 446
pin polarity 68
pinouts 65
playback 77, 80, 267
  backwards 80
  breakpoint 80
  hardware 325
  single step 81
  start and end indexes 80
playback end index 328
playback file 56, 200, 239, 327, 328
playback speed 81
playback start index 328
plot data 79
polarity 68
port 300, 310
port view 91, 301
  field display 313
Port View 310, 311, 318
  configuration 310
  configuration mode 312
  data edit 311
  engineering units 312
  error injection mode 312
  hex editor 311
  Packet Edit & Display 311
  summary mode 311
positive sense 512
post analysis
  software playback 346
pow 401
preparing the data source for hardware playback 327
PrintAliases 425
professional 367
professional views 369
progress bar control 447
project 53
project access
  security 431
project create 42
project explorer 44
project file 50
Project File Management 56
Project Files 56
project load 42
project open 55
project options 51
project run 58
project save 55
project security 431
project start 50
project tab 42
prompt 120
properties
  map view 395
property browser 375
property page 64
proprietary 433
protocol
  MIL-STD-1553 109
protocol browser 44, 201, 208, 251, 255
  statistics 44
protocol errors 318
protocol options 51
protocols 30
push button control 447
PyQT 364
Python 399
  help 401
  proprietary information 433
  regular expressions 483
Python  help 401
Python code editor 407
Python debugger 400, 410, 416
Python output stream 426
Python script editor 400, 407



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Python script file 56
Python scripting 399
Python scripts 50
Python syntax 483
Python watch window 400

## Q

queuing ports 315
quick controls 208
quick tasks 42
quick view 208
  control library 385
quick view control configuration 386
quick view displays 384

## R

radar display 269
radio magnetic indicator (RMI) 457
radix display 80
radix editor 93
ramping 96
random 96
range 52, 78
range max 165
range min 165
RangeMax 165
RangeMin 165
rate 52, 118, 119, 135, 226, 227
  transmission 235
rate based schedule 52
rate-based schedule 118, 119, 226
rate-based scheduling 133
raw editor 91, 93
raw hex display 269
reapply interpreters 79
reapply sequential monitor interpretations 79
receive host enable 302
receive options
  429 223
recent projects 42, 51
record 71, 244, 302
record filtering 249
recording controls
  monitor view 195
recording databus traffic 71
recording modes 174, 246, 306
recover data 51, 74
redundant 308
redundant bus 134
redundant network 308, 319
reference documentation 360
refresh rate 205, 253
regular expressions
  Python 483
reimport module 402
relative transmission rate 315
relative word count error 210
remote terminal 136
remove alias 413
rename alias 413
repeat mode 319
replay 325
reset all counts 309
resize column 206, 255
response time 142, 147, 201
  error injection 142, 208
response time error generation 142
restore master script 422
restricted 431
resume hardware playback 334
retransmission 224
retry 134
  BC message 133
  bus 134
  conditions 134
reverse word order 52, 153
round-robin 300, 315
RSN 305, 318, 319
RSN error 319
RSN error injection 319
RSN errors 318
RT 136
  mode code legalization 137
RT configuration 140
RT configuration options 146
RT context menu 140
RT database configuration 147
RT filter 191
RT initialization 136
RT mode code configuration 144
RT Properties 141
RT status word 141
RT status word configuration 144
RT subaddress selection 143
RT summary 208
RTs context menu 139
RTs properties
  TSA window 138
run macro 428
run script 405
run selected 419
run software playback 341
running a project 58
running hardware playback 332
runtime 431

## S

SA 136
SA 0 mode code 113



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SA 31 mode code 113
SA configuration 148
SA context menu 149
SA data 150
SA filter 191
SA properties 150
SA summary 208
SA sync 147
sample rate
  1553 113
  ARINC 429 223
sample script library 497
save 316
  database 213, 257
  output stream 426
save as default 80
Save Data to Excel 477
save EquipID to database 223
save fields
  1553 database 157
save schedule 121, 231
save to database 316
saving monitor data 178
saving monitor files 178
saving OLE documents 105
saving sequential monitor data 247
sawtooth 96
schedule 118, 300, 315
  1553 118
  429 226
  BC 115
  BC pause 128
  branch 115, 125
  continuous 123
  custom 115, 118
  default 115
  DIO 128
  edit 122
  frame 115, 123
  gap 115, 124
  message 128
  no operation 128
  rate-based 115
  retry 115
  target rate 226
  trigger start 117
  view 116
schedule branch 127
schedule editing 122
schedule error handling 120
schedule errors 318
schedule export 121, 226, 231
schedule frame only 120
schedule hault 234
schedule import 121, 226, 231
schedule message 123
schedule mode 118, 119
schedule options 52
schedule print 121, 231
schedule settings 123
schedule stepping 117, 225
schedule switching 119
schedule trigger 225
schedule view 228
scheduled message 135
schedules 119
scheduling 225, 227
  ARINC 429 225
  custom 228
  default 226
  free-running 228
  unschedulable messages 230
scheduling slow messages 227
scope 49, 427
script
  master script 402
  run 405
  skip OnProjectLoad() 403
  user scripts 404
script details 467
script editor 400, 407
script examples 467
script file 56
script files 401, 404
script interpreter 163
Script Object Properties and Methods 491
script objects 490
script output 425, 493
Script Pane 489
script reference 406
script threading 405
script view 487
Script View 398
script view execution 496
script view timers 493
scripting 314, 318, 399, 487
  software playback 345
scripting events 214, 260, 320
scripting principles 495
scroll bar control 447
SDI 218, 240, 257
search 77, 308
  monitor view 194
  sequential monitor 249
secondary bus 132
security 431, 512
segmenting monitor files 179
select custom schedule 230
select multiple controls 375
select output 426
select schedule mode 230
selection gallery 378



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

selection grab handles 375
selective sync 147
selector knob 462
selector knob control 446
self test 65
self-test 224
sense 512
sequential monitor 71, 72, 75, 244, 302, 305, 307
  1553 169
  auto scroll 76
  capture filter 74, 75, 304
  data recovery 51, 74
  display filter 74, 77, 78, 307
  display filter toolbar 77, 78
  display settings 76
  export 78, 308
  goto & step toolbar 77
  import 79
  network capture filter 304
  pause 76
  performance 72
  playback 77
  playback toolbar 77
  port capture filter 304
  print 76
  print preview 76
  reapply interpreters 79
  search 77
  search toolbar 77
  set as next message 81
  split view 76
  standard toolbar 76
  status bar 76
  step 81
  synchronization 78
  time correlation 77, 78
  timed run 75
  toolbar 76
  VL capture filter 304
sequential monitor file 72
sequential monitor view 307
  configure view 80
  display settings 80
sequential monitoring
  ARINC 429 244
sequential record 303
serial 32
set as next message 81
set software playback start and end index 344
setting up hardware playback 330
shell 407
shell alias 407
shift key 92
short name 235
shortcut key 416
shortcut keys 199, 403, 408, 424, 468
show member events 414
show member methods 414
show member properties 414
show toolbar labels 51
sign status matrix 218
signal modulation
  amplitude 66
  pulse width 66
signal pinouts 65
simple scalar interpreter 163, 167
simulate RT 147
simulated RT 136
simulation 311
sin 401
sine 96
single precision 312
single shot message 132
single step 81
single-shot 318
single-shot enable 234
single-shot message 119
sinh 401
sink control type 377
sink link 375
sink object 351
skip 233
  708 playback 268
  hardware playback 332
skip message 119, 132
skip OnProject load() 403
slider 463
Slider 463
slider control 446, 447
software playback 80, 325, 337, 347
  breakpoint 347
  engage 339
  examples 345
  operation 338
  pause 347
  pause with breakpoints 342
  post analysis 346
  rate control 341
  run 341
  scripting 345
  share data files 347
  source 340
  specify resume point 343
  start and end index 344
software playback examples 345
software playback operation 338
software playback speed 341
software scheduling 227, 235
software timer scheduling 227, 235
software updates 37
sort 301, 414
source control type 377



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

source destination identifier 218
source link 375
source object 351
specify the software playback resume point 343
speed 222
  708 playback 268
  auto 222
  custom 222
  high 222
  low 222
  non-standard 222
speed detection 222
sqrt 401
square 96
SSM 218
standard 367
standard deviation 248
standard module 410
start index 80, 328
  software playback 347
start page 42, 51, 513
stastics
  protocol browser 44
statistics 309, 311
statistics display 181
statistics mode 80
status 420
status word 141, 144
step 77, 225
  708 playback 268
  hardware playback 332, 334
  monitor view 194
stop channel
  1553 113
stopped BC 118
str 416
string 313
StringToValue 165
strip chart control 446, 463
strip view 372
Strip View Properties 373
stutter mode 319
subaddress 136
subaddress auto detection 147
subaddress configuration 148
subaddress legalization 144, 150
subaddress summary 208
subaddress view
  1553 201
sub-VL 300, 315
suggested control 379
summary results
  monitor 248
support and service 37
supported protocols 30
switching schedules 119
sync 225, 234
  MC 147
  mode code 147
  SA 147
sync all 147, 234
sync error 210
sync mode 118
sync out lines 67
sync output
  BC message 132
sync output line selection 69, 118, 132, 225, 234
sync polarity 68
sync pulse 225
sync selective 147
synchronization 78, 113, 118, 225, 234
synchronization pulse 67
SYNCOUT 67
syntax edit control 447
syntax errors 404
syntax highlighting 409
system time 190, 248

## T

tab control 447
tab nanny 411
tag for error 150, 211
tagging messages 211
tagging messages for error injection 210
tan 401
tanh 401
task 109
tasks
  1553 109
  hardware explorer 109
temporary directory 73
temporary files 73
terminal simulation assignment 111, 138
termination 114
TerraServer configuration 397
TerraServer maps 396
test manager 45, 418
  loops 420
  output log 423
test output 423
test routine 421
theme 51
third-party ActiveX controls 381
third-party automation 364
threading module 405
tilt 265
time 416
time correlation 77, 78, 199
time correlation sink 78
time correlation source 78
time display 80



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

time synchronization 77, 78
timed run 174, 247, 306
  sequential monitor 75
timeout 204, 207, 252, 385
  message 133
timeout alarm 233
timeout value 233
timer control 446
timers
  script view 493
TimeSyncViews method 78
time-tag 513
time-tag format 190, 248
time-tags 190, 248
  IRIG 66
timing errors 318
timing factors 300
TKinter 364
toggle control 446, 465
tolerance 52, 226
toolbar
  ARINC 429 48
  macro 429
  MIL-STD-1553 48, 109
  output pane 426
toolbar labels 51
topics 401
T-R Unit 264
traceback 405
transformer-coupled 114
transmission order 320
transmission rate 133
transmit amplitude 69, 113
transmit equipment ID 224
transmit interval 52, 226, 227, 235
transmit message 131
transmit message interval 235
transmit option
  ARINC 429 224
transmit options 52
transmit range 52, 226
transmit rate 52, 218, 315
transmit schedules 318
transmit speed 218
TransmitMessage() 225
TransmitMsg() 135
TransmitRawValue() 225
TransmitRawValueArray() 225
tree view control 447
trigger 225
trigger event 224, 233, 364
trigger input
  BC message 132
trigger input line selection 68, 117, 132, 225, 234
trigger lines 67
trigger mode 117
Trigger Monitor by Limit Conditions 476
Trigger Monitor by Record Count 477
trigger start schedule 117
triggering 113, 234
triggers 318
TSA 111, 138
ttag 201
ttl discretes 32
turn coordinator 457
tutorial 401
type
  data generator 95

## U

UDP header 320
UDP layer 320
UI (user interface) 40
unauthorized viewing 433
unicode 416
unimport module 402, 409
UninitEvents 415
units 235
unrestricted 431
unschedulable messages 230
use default schedule 120
use software timer scheduling 235
user defined schedule 118
user interface 51
user output stream 426
user schedule 118, 119, 120, 122, 228
user scripts 404

## V

valid alias names 413
ValueToDouble 165
ValueToString 165
variable scope 427
variable transmit amplitude 69
variable watching 427
variables 406
variance 248
variant 416
VB
  execution 496
VB script 399, 487
  output 493
  timers 493
VB scripting principles 495
view data 201
view engineering units 85
view port 311
view refresh 205, 253
view window 46
view1 188



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

view2 188
viewing ARINC 429 data 251
viewing monitor records 72
views 46, 369, 393
virtual instruments 374
virtual links (VLs) 304, 307
Virtual Links (VLs) 308, 309, 310, 311
visual theme 51
VL detection 300
VL view 301, 311
VL View 318

## W

watch pane 45
watch window 45, 91, 400, 404
watch window pane 427
watch window scope 427
waypoints
  map view 394
web browser control 447
widgets 364, 374
Williamsburg 31
wired 33
word count error 210
workspace 43
workspace tab position 51
workspaces 43
wrap-around 224
WxWidgets 364

## X

XML 69
  hardware import/export 69
XML database import 89
XML hardware import 219
XML import/export 89, 213, 258, 317
xtream controls 447

## Z

zero crossing distortion error 210



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

**Artisan Technology Group** is an independent supplier of quality pre-owned equipment

## Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

## We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

## Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP