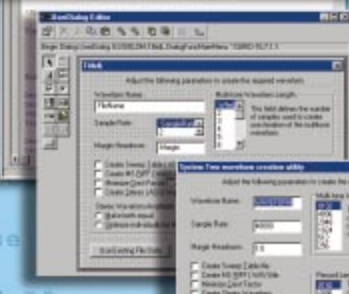


**APWIN BASIC EXTENSIONS
REFERENCE FOR SYSTEM TWO CASCADE**



APWIN BASIC Extensions Reference for System Two Cascade



Version 2
August, 1999

APWIN Basic Extensions Reference for System Two Cascade

Copyright 1993-1999 Audio Precision, Inc.

All rights reserved.

Version 2 August, 1999

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the publisher.

Audio Precision®, System One®, System One + DSP®, System Two®, FASTTEST®, APWIN®, Portable One®, and Dual Domain® are registered trademarks of Audio Precision, Inc. Windows™ is a trademark of Microsoft Corporation.

Published by:



Audio Precision, Inc.
PO Box 2209
Beaverton, Oregon 97075-2209
U.S. Toll Free: 1-800-231-7350
Tel: (503) 627-0832 Fax: (503) 641-8906
email: techsupport@audioprecision.com
Web: www.audioprecision.com

Printed in the United States of America
Audio Precision Part # 8211-0083

CONTENTS

APWIN Basic Extensions Reference	1
Introduction	1-1
Manual Conventions	1-2
System Panels	2
Analog Analyzer	3
AP.Anlr.ChACoupling	3-1
AP.Anlr.ChAFreqRdg	3-2
AP.Anlr.ChAFreqReady	3-2
AP.Anlr.ChAFreqSettling	3-3
AP.Anlr.ChAFreqTrig	3-4
AP.Anlr.ChAImpedance	3-4
AP.Anlr.ChAInput	3-5
AP.Anlr.ChALevelRdg	3-5
AP.Anlr.ChALevelReady	3-6
AP.Anlr.ChALevelSettling	3-7
AP.Anlr.ChALevelTrig	3-7
AP.Anlr.ChARange	3-8
AP.Anlr.ChARangeAuto	3-8
AP.Anlr.ChBCoupling	3-8
AP.Anlr.ChBFreqRdg	3-9
AP.Anlr.ChBFreqReady	3-10
AP.Anlr.ChBFreqSettling	3-10
AP.Anlr.ChBFreqTrig	3-11

AP.Anlr.ChBImpedance	3-11
AP.Anlr.ChBInput	3-12
AP.Anlr.ChBLevelRdg	3-13
AP.Anlr.ChBLevelReady	3-13
AP.Anlr.ChBLevelSettling	3-14
AP.Anlr.ChBLevelTrig	3-14
AP.Anlr.ChBRange	3-15
AP.Anlr.ChBRangeAuto	3-15
AP.Anlr.FuncBPBRFreq	3-16
AP.Anlr.FuncBPBR Tuning	3-17
AP.Anlr.FuncDetector	3-17
AP.Anlr.FuncFilter	3-18
AP.Anlr.FuncFilterHP	3-19
AP.Anlr.FuncFilterLP	3-20
AP.Anlr.FuncInput	3-21
AP.Anlr.FuncMode	3-22
AP.Anlr.FuncRange	3-23
AP.Anlr.FuncRangeAuto	3-24
AP.Anlr.FuncRdg	3-25
AP.Anlr.FuncReady	3-26
AP.Anlr.FuncSettling	3-26
AP.Anlr.FuncTrig	3-27
AP.Anlr.PhaseMode	3-27
AP.Anlr.PhaseRdg	3-27
AP.Anlr.PhaseReady	3-28
AP.Anlr.PhaseSettling	3-29
AP.Anlr.PhaseTrig	3-29
AP.Anlr.RdgRate	3-30
AP.Anlr.RefChAdBr	3-30
AP.Anlr.RefChBdBr	3-32
AP.Anlr.RefdBm	3-32
AP.Anlr.RefdBrAuto	3-33
AP.Anlr.RefFreq	3-35
AP.Anlr.RefFreqAuto	3-35
AP.Anlr.RefWatts	3-36
AP.Anlr.WFDetector	3-37
AP.Anlr.WFFilter	3-38
Application	4
AP.Application.AppDir	4-1
AP.Application.ClearCurrentError	4-1
AP.Application.CopyPanelToClipboard	4-2
AP.Application.DisplayDataOnTestOpen	4-3

AP.Application.DoReadings	4-4
AP.Application.GetCurrentErrorString	4-6
AP.Application.Input	4-6
AP.Application.MacroDir	4-7
AP.Application.Name	4-8
AP.Application.NewData	4-9
AP.Application.NewMacro	4-10
AP.Application.NewTest	4-10
AP.Application.Output	4-11
AP.Application.Page	4-11
AP.Application.PanelClose	4-12
AP.Application.PanelOpen	4-14
AP.Application.Quit	4-16
AP.Application.Restore	4-17
AP.Application.SetWatchDogTimer1	4-18
AP.Application.SetWatchDogTimer2	4-19
AP.Application.SysType	4-20
AP.Application.TestDir	4-21
AP.Application.TestName	4-21
AP.Application.ThrowErrors	4-22
AP.Application.Version	4-23
AP.Application.Visible	4-24
AP.Application.VisibleAll	4-25
AP.Application.VisibleBarGraphs	4-26
AP.Application.VisibleDataEditor	4-26
AP.Application.VisibleGraph	4-26
AP.Application.VisibleMacroEditor	4-27
AP.Application.VisiblePanels	4-27
AP.Application.WorkingDir	4-28
Auxiliary Instrument	5
AP.Aux.Reading1Rdg	5-1
AP.Aux.Reading1Ready	5-2
AP.Aux.Reading1Settling	5-3
AP.Aux.Reading1Trig	5-4
AP.Aux.Reading2Rdg	5-4
AP.Aux.Reading2Ready	5-4
AP.Aux.Reading2Settling	5-5
AP.Aux.Reading2Trig	5-5
AP.Aux.Reading3Rdg	5-6
AP.Aux.Reading3Ready	5-6
AP.Aux.Reading3Settling	5-7
AP.Aux.Reading3Trig	5-7

AP.Aux.Reading4Rdg	5-8
AP.Aux.Reading4Ready	5-8
AP.Aux.Reading4Settling	5-9
AP.Aux.Reading4Trig	5-9
AP.Aux.SetReading1	5-9
AP.Aux.SetReading2	5-10
AP.Aux.SetReading3	5-10
AP.Aux.SetReading4	5-10
AP.Aux.Setting1	5-11
AP.Aux.Setting2	5-11
AP.Aux.Setting3	5-11
AP.Aux.Setting4	5-11
Bar Graph	6
AP.BarGraph.AxisAutoScale	6-1
AP.BarGraph.AxisIncrement	6-3
AP.BarGraph.AxisLeft	6-3
AP.BarGraph.AxisLogLin	6-4
AP.BarGraph.AxisRight	6-4
AP.BarGraph.Comment	6-5
AP.BarGraph.CommentShow	6-6
AP.BarGraph.DigitsOnly	6-7
AP.BarGraph.Id	6-7
AP.BarGraph.Max	6-8
AP.BarGraph.Min	6-8
AP.BarGraph.New	6-9
AP.BarGraph.Reset	6-9
AP.BarGraph.TargetLower	6-10
AP.BarGraph.TargetRange	6-10
AP.BarGraph.TargetUpper	6-11
AP.BarGraph.Title	6-11
Status Bits	7
AP.Bits.ChAAudioModeRdg	7-1
AP.Bits.ChAAuxBitsRdg	7-1
AP.Bits.ChACategoryRdg	7-3
AP.Bits.ChAChModeRdg	7-4
AP.Bits.ChAChNumRdg	7-5
AP.Bits.ChAClockAccuracyRdg	7-7
AP.Bits.ChACopyrightRdg	7-8
AP.Bits.ChACrcRdg	7-9
AP.Bits.ChADestinationRdg	7-10
AP.Bits.ChAEmphRdg	7-11

AP.Bits.ChAFlag0_5Rdg	7-12
AP.Bits.ChAFlag6_13Rdg	7-14
AP.Bits.ChAFlag14_17Rdg	7-14
AP.Bits.ChAFlag18_21Rdg	7-15
AP.Bits.ChAFreqModeRdg	7-16
AP.Bits.ChALocalAddressRdg	7-17
AP.Bits.ChAModeRdg	7-17
AP.Bits.ChAOriginRdg	7-18
AP.Bits.ChARefSignalRdg	7-19
AP.Bits.ChASampleFreqRdg	7-20
AP.Bits.ChASourceNumRdg	7-22
AP.Bits.ChAStatusXferToString	7-24
AP.Bits.ChATimeOfDayRdg	7-26
AP.Bits.ChAUserBitsRdg	7-27
AP.Bits.ChAWordLengthRdg	7-28
AP.Bits.ChAXmitStatus	7-29
AP.Bits.ChBAudioModeRdg	7-29
AP.Bits.ChBAuxBitsRdg	7-30
AP.Bits.ChBCategoryRdg	7-31
AP.Bits.ChBChModeRdg	7-31
AP.Bits.ChBChNumRdg	7-32
AP.Bits.ChBClockAccuracyRdg	7-33
AP.Bits.ChBCopyrightRdg	7-34
AP.Bits.ChBCrcRdg	7-35
AP.Bits.ChBDestinationRdg	7-35
AP.Bits.ChBEmphRdg	7-36
AP.Bits.ChBFlag0_5Rdg	7-37
AP.Bits.ChBFlag6_13Rdg	7-37
AP.Bits.ChBFlag14_17Rdg	7-38
AP.Bits.ChBFlag18_21Rdg	7-38
AP.Bits.ChBFreqModeRdg	7-39
AP.Bits.ChBLocalAddressRdg	7-40
AP.Bits.ChBModeRdg	7-40
AP.Bits.ChBOriginRdg	7-41
AP.Bits.ChBRefSignalRdg	7-41
AP.Bits.ChBSampleFreqRdg	7-42
AP.Bits.ChBSourceNumRdg	7-43
AP.Bits.ChBStatusXferToString	7-44
AP.Bits.ChBTimeOfDayRdg	7-44
AP.Bits.ChBUserBitsRdg	7-45
AP.Bits.ChBWordLengthRdg	7-45
AP.Bits.ChBXmitStatus	7-46
AP.Bits.Cons.AudioMode	7-46

AP.Bits.Cons.Category	7-47
AP.Bits.Cons.Channels	7-48
AP.Bits.Cons.ChNum	7-48
AP.Bits.Cons.ClockAccuracy	7-49
AP.Bits.Cons.Copyright	7-49
AP.Bits.Cons.Emphasis	7-49
AP.Bits.Cons.SampleFreq	7-50
AP.Bits.Cons.SourceNum	7-50
AP.Bits.Mode	7-51
AP.Bits.Pro.AudioMode	7-51
AP.Bits.Pro.AuxBits	7-52
AP.Bits.Pro.ChMode	7-52
AP.Bits.Pro.CrcEnable	7-53
AP.Bits.Pro.Destination	7-53
AP.Bits.Pro.Emphasis	7-53
AP.Bits.Pro.Flag0_5	7-54
AP.Bits.Pro.Flag6_13	7-54
AP.Bits.Pro.Flag14_17	7-55
AP.Bits.Pro.Flag18_21	7-55
AP.Bits.Pro.FreqMode	7-56
AP.Bits.Pro.LocalAddress	7-56
AP.Bits.Pro.LocalAddressAuto	7-57
AP.Bits.Pro.Origin	7-58
AP.Bits.Pro.RefSignal	7-58
AP.Bits.Pro.SampleFreq	7-58
AP.Bits.Pro.TimeOfDay	7-59
AP.Bits.Pro.UserBits	7-60
AP.Bits.Pro.WordLength	7-60
AP.Bits.XmitChannel	7-61
RS-232	8
AP.CommA.Break	8-1
AP.CommA.CD Holding	8-1
AP.CommA.CD Timeout	8-2
AP.CommA.CommEvent	8-2
AP.CommA.CommId	8-3
AP.CommA.CommPort	8-4
AP.CommA.CTSHolding	8-5
AP.CommA.CTSTimeout	8-5
AP.CommA.DSRHolding	8-6
AP.CommA.DSRTimeout	8-6
AP.CommA.DTREnable	8-6
AP.CommA.Handshaking	8-7

AP.CommA.InBufferCount	8-8
AP.CommA.InBufferSize	8-8
AP.CommA.Input	8-8
AP.CommA.InputLen	8-9
AP.CommA.Interval	8-9
AP.CommA.NullDiscard	8-9
AP.CommA.OutBufferCount	8-10
AP.CommA.OutBufferSize	8-10
AP.CommA.Output	8-11
AP.CommA.ParityReplace	8-11
AP.CommA.PortOpen	8-11
AP.CommA.RThreshold	8-12
AP.CommA.RTSEnable	8-12
AP.CommA.Settings	8-13
AP.CommA.SThreshold	8-15

Computes 9

AP.Compute.Avg.Apply	9-1
AP.Compute.Avg.Data	9-1
AP.Compute.Avg.PostSweep	9-2
AP.Compute.Avg.Start	9-3
AP.Compute.Avg.Stop	9-3
AP.Compute.Center.Apply	9-3
AP.Compute.Center.Data	9-4
AP.Compute.Center.PostSweep	9-5
AP.Compute.Center.Start	9-5
AP.Compute.Center.Stop	9-6
AP.Compute.Clear.All	9-6
AP.Compute.Delta.Apply	9-6
AP.Compute.Delta.Data	9-7
AP.Compute.Delta.FileName	9-8
AP.Compute.Delta.PostSweep	9-8
AP.Compute.Equalize.Apply	9-9
AP.Compute.Equalize.Data	9-10
AP.Compute.Equalize.FileName	9-10
AP.Compute.Equalize.PostSweep	9-11
AP.Compute.Invert.Apply	9-11
AP.Compute.Invert.Data	9-12
AP.Compute.Invert.Horizontal	9-12
AP.Compute.Invert.PostSweep	9-13
AP.Compute.Linearity.Apply	9-13
AP.Compute.Linearity.Data	9-14
AP.Compute.Linearity.PostSweep	9-15

AP.Compute.Linearity.Start	9-15
AP.Compute.Linearity.Stop	9-16
AP.Compute.Max.Apply	9-16
AP.Compute.Max.Data	9-17
AP.Compute.Max.PostSweep	9-18
AP.Compute.Max.Start	9-18
AP.Compute.Max.Stop	9-19
AP.Compute.Min.Apply	9-19
AP.Compute.Min.Data	9-20
AP.Compute.Min.PostSweep	9-20
AP.Compute.Min.Start	9-21
AP.Compute.Min.Stop	9-21
AP.Compute.Normalize.Apply	9-22
AP.Compute.Normalize.Data	9-23
AP.Compute.Normalize.Horizontal	9-23
AP.Compute.Normalize.PostSweep	9-24
AP.Compute.Normalize.Target	9-24
AP.Compute.Sigma.Apply	9-25
AP.Compute.Sigma.Data	9-25
AP.Compute.Sigma.PostSweep	9-26
AP.Compute.Sigma.Start	9-27
AP.Compute.Sigma.Stop	9-27
AP.Compute.Smooth.Apply	9-27
AP.Compute.Smooth.Auto	9-28
AP.Compute.Smooth.Data	9-28
AP.Compute.Smooth.Passes	9-29
AP.Compute.Smooth.PostSweep	9-29
Data	10
AP.Data.AddRowToEnd	10-1
AP.Data.ColLimitError	10-2
AP.Data.ColLowerLimitError	10-3
AP.Data.ColName	10-3
AP.Data.ColNumOf	10-4
AP.Data.ColSize	10-4
AP.Data.ColUnit	10-5
AP.Data.ColUpperLimitError	10-6
AP.Data.DeleteRow	10-6
AP.Data.Id	10-7
AP.Data.InsertRowAfter	10-8
AP.Data.InsertRowBefore	10-9
AP.Data.LimitError	10-9
AP.Data.LowerLimitError	10-11

AP.Data.UpdateDisplay 10-12
 AP.Data.UpperLimitError 10-13
 AP.Data.Value 10-14
 AP.Data.XferToArray 10-15

DCX-127 11

AP.DCX.Ch1DcLevel 11-1
 AP.DCX.Ch1DcOutput 11-1
 AP.DCX.Ch2DcLevel 11-2
 AP.DCX.Ch2DcOutput 11-2
 AP.DCX.DigInFormat 11-2
 AP.DCX.DigInRdg 11-4
 AP.DCX.DigInRdgRate 11-4
 AP.DCX.DigInReady 11-5
 AP.DCX.DigInScale 11-5
 AP.DCX.DigInSettling 11-6
 AP.DCX.DigInTrig 11-6
 AP.DCX.DigOut 11-7
 AP.DCX.DigOutFormat 11-7
 AP.DCX.DigOutScale 11-8
 AP.DCX.DmmMode 11-8
 AP.DCX.DmmOffset 11-9
 AP.DCX.DmmRange 11-10
 AP.DCX.DmmRangeAuto 11-10
 AP.DCX.DmmRdg 11-11
 AP.DCX.DmmRdgRate 11-12
 AP.DCX.DmmReady 11-12
 AP.DCX.DmmScale 11-13
 AP.DCX.DmmSettling 11-13
 AP.DCX.DmmTrig 11-13
 AP.DCX.GateDelay 11-14
 AP.DCX.PortAOutput 11-14
 AP.DCX.PortBOutput 11-15
 AP.DCX.PortCOutput 11-15
 AP.DCX.PortDOutput 11-16

Digital Generator 12

AP.DGen.BurstInterval 12-1
 AP.DGen.BurstLevel 12-2
 AP.DGen.BurstOnTime 12-2
 AP.DGen.ChAAmpl 12-3
 AP.DGen.ChAEqAmpl 12-4
 AP.DGen.ChAFreq 12-5

CONTENTS

AP.DGen.ChAInvert	12-5
AP.DGen.ChAOutput	12-6
AP.DGen.ChBAmpl	12-6
AP.DGen.ChBEqAmpl	12-7
AP.DGen.ChBFreq	12-7
AP.DGen.ChBInvert	12-8
AP.DGen.ChBOutput	12-8
AP.DGen.ChBTrackA	12-8
AP.DGen.DitherType	12-9
AP.DGen.DualAmplRatio	12-10
AP.DGen.EqCurve	12-10
AP.DGen.Freq	12-11
AP.DGen.IMCenterFreq	12-11
AP.DGen.IMFreq	12-12
AP.DGen.IMHighFreq	12-12
AP.DGen.Offset	12-13
AP.DGen.Output	12-13
AP.DGen.Phase	12-13
AP.DGen.RefdBr	12-14
AP.DGen.RefFreq	12-15
AP.DGen.RefVFS	12-15
AP.DGen.StepRate	12-16
AP.DGen.Wfm	12-16
AP.DGen.WfmName	12-17
Events	13
APEvent_OnAuxSetting1	13-1
APEvent_OnAuxSetting2	13-2
APEvent_OnAuxSetting3	13-2
APEvent_OnAuxSetting4	13-3
APEvent_OnDcxProgramControllInput	13-3
APEvent_OnError	13-4
APEvent_OnSweepEnd	13-5
APEvent_OnSweepNestEnd	13-6
APEvent_OnSweepNestStart	13-6
APEvent_OnSweepReverseChannels	13-6
APEvent_OnSweepStart	13-8
APEvent_OnSweepStep	13-8
APEvent_OnSweepStepEnd	13-9
APEvent_OnSweepTrigger	13-9
APEvent_OnWatchDogTimeout	13-9
File	14

AP.File.AppendData	14-1
AP.File.ExportASCIIData	14-1
AP.File.ExportGraphic	14-2
AP.File.ImportASCIIData	14-4
AP.File.ImportDOSS1Test	14-4
AP.File.OpenData	14-5
AP.File.OpenMacro	14-6
AP.File.OpenTest	14-7
AP.File.OpenWfm	14-8
AP.File.SaveAll	14-9
AP.File.SaveDataAs	14-9
AP.File.SaveTest	14-10
AP.File.SaveTestAs	14-10
AP.File.SaveWfmAs	14-12
Analog Generator	15
AP.Gen.Ampl	15-1
AP.Gen.BurstInterval	15-2
AP.Gen.BurstLevel	15-3
AP.Gen.BurstOnTime	15-3
AP.Gen.ChAAmpl	15-4
AP.Gen.ChAEqAmpl	15-5
AP.Gen.ChAFreq	15-5
AP.Gen.ChAInvert	15-6
AP.Gen.ChAOutput	15-7
AP.Gen.ChBAmpl	15-8
AP.Gen.ChBEqAmpl	15-8
AP.Gen.ChBFreq	15-8
AP.Gen.ChBInvert	15-9
AP.Gen.ChBOutput	15-9
AP.Gen.ChBTrackA	15-10
AP.Gen.Config	15-11
AP.Gen.DACBandwidth	15-12
AP.Gen.DACSampleRate	15-13
AP.Gen.DualAmplRatio	15-13
AP.Gen.EqCurve	15-14
AP.Gen.Freq	15-14
AP.Gen.FreqAccuracy	15-16
AP.Gen.IMCenterFreq	15-16
AP.Gen.IMFreq	15-17
AP.Gen.IMHighFreq	15-19
AP.Gen.Impedance	15-19
AP.Gen.Output	15-20

CONTENTS

AP.Gen.Phase	15-21
AP.Gen.RefdBm	15-21
AP.Gen.RefdBr	15-23
AP.Gen.RefdBrAuto	15-23
AP.Gen.RefFreq	15-24
AP.Gen.RefFreqAuto	15-25
AP.Gen.RefWatts	15-26
AP.Gen.Wfm	15-27
AP.Gen.WfmName	15-30
Graph	16
AP.Graph.Comment	16-1
AP.Graph.CommentShow	16-2
AP.Graph.CopyToSweepPanel	16-3
AP.Graph.CursorPosition	16-4
AP.Graph.CursorRow	16-6
AP.Graph.CursorsOn	16-6
AP.Graph.CursorValue	16-7
AP.Graph.OptimizeIndividually	16-7
AP.Graph.OptimizeLeft	16-7
AP.Graph.OptimizeRight	16-8
AP.Graph.OptimizeTogether	16-8
Log	17
AP.Log.AddEntry	17-1
AP.Log.AddEntryWithoutTimeDate	17-2
AP.Log.Clear	17-2
AP.Log.Data	17-2
AP.Log.Enable	17-3
AP.Log.ErrorMessage	17-3
AP.Log.FileActivity	17-3
AP.Log.FileName	17-4
AP.Log.GraphTitle	17-4
AP.Log.PassFailMessages	17-4
AP.Log.PrintLogFile	17-5
AP.Log.TestName	17-5
AP.Log.View	17-6
Macro	18
AP.Macro.IsRunning	18-1
AP.Macro.Name	18-2

Print	19
AP.Print.Data	19-1
AP.Print.Graph	19-1
AP.Print.LoadFromTest	19-2
AP.Print.TrackGraph	19-2
Prompt	20
AP.Prompt.FontSize	20-1
AP.Prompt.Hide	20-1
AP.Prompt.IsUp	20-2
AP.Prompt.Position	20-2
AP.Prompt.Show	20-3
AP.Prompt.ShowWithContinue	20-3
AP.Prompt.ShowWithContinueAndStopSweep	20-3
AP.Prompt.Text	20-4
Regulation	21
AP.Reg.IsRunning	21-1
AP.Reg.SourceHigh	21-2
AP.Reg.SourceId	21-2
AP.Reg.SourceIteration	21-3
AP.Reg.SourceLow	21-4
AP.Reg.SourceOperation	21-4
AP.Reg.SourceStepSize	21-5
AP.Reg.Start	21-6
AP.Reg.StartNoWait	21-6
AP.Reg.SweepEnable	21-6
AP.Reg.TargetId	21-7
AP.Reg.TargetTolerance	21-7
AP.Reg.TargetToleranceMode	21-8
AP.Reg.TargetValue	21-8
AP.Reg.TimeOut	21-9
Digital Input/Output	22
AP.S2CDio.ChAPeakRdg	22-1
AP.S2CDio.ChAPeakReady	22-2
AP.S2CDio.ChAPeakTrig	22-2
AP.S2CDio.ChBPeakRdg	22-3
AP.S2CDio.ChBPeakReady	22-3
AP.S2CDio.ChBPeakTrig	22-4
AP.S2CDio.DelayRdg	22-4
AP.S2CDio.DelayReady	22-5

AP.S2CDio.DelaySettling	22-5
AP.S2CDio.DelayTrig	22-6
AP.S2CDio.FlagChAInvalidRdg	22-6
AP.S2CDio.FlagChBInvalidRdg	22-7
AP.S2CDio.FlagCodingRdg	22-8
AP.S2CDio.FlagConfidenceRdg	22-8
AP.S2CDio.FlagInvalidRdg	22-9
AP.S2CDio.FlagLockRdg	22-9
AP.S2CDio.FlagParityRdg	22-9
AP.S2CDio.InBitsDisplay	22-10
AP.S2CDio.InDeEmp	22-10
AP.S2CDio.InFormat	22-11
AP.S2CDio.InImpedance	22-12
AP.S2CDio.InInput	22-12
AP.S2CDio.InJitterBW	22-13
AP.S2CDio.InJitterDetector	22-14
AP.S2CDio.InJitterMode	22-14
AP.S2CDio.InMonitorMode	22-14
AP.S2CDio.InResolution	22-15
AP.S2CDio.InScaleFreq	22-15
AP.S2CDio.JitterRdg	22-16
AP.S2CDio.JitterReady	22-16
AP.S2CDio.JitterSettling	22-17
AP.S2CDio.JitterTrig	22-18
AP.S2CDio.OutCableSim	22-18
AP.S2CDio.OutCM	22-19
AP.S2CDio.OutCMAmpl	22-19
AP.S2CDio.OutCMFreq	22-20
AP.S2CDio.OutFormat	22-20
AP.S2CDio.OutInvert	22-22
AP.S2CDio.OutJitterAmpl	22-22
AP.S2CDio.OutJitterEqCurve	22-23
AP.S2CDio.OutJitterFreq	22-23
AP.S2CDio.OutJitterType	22-24
AP.S2CDio.OutNoise	22-24
AP.S2CDio.OutNoiseAmpl	22-24
AP.S2CDio.OutParityError	22-25
AP.S2CDio.OutPreEmp	22-25
AP.S2CDio.OutRate	22-26
AP.S2CDio.OutResolution	22-27
AP.S2CDio.OutRiseFall	22-27
AP.S2CDio.OutRiseFallTime	22-27
AP.S2CDio.OutSendInvalid	22-28

AP.S2CDio.OutVoltage	22-28
AP.S2CDio.RateRdg	22-29
AP.S2CDio.RateReady	22-30
AP.S2CDio.RateSettling	22-30
AP.S2CDio.RateTrig	22-31
AP.S2CDio.RefRate	22-31
AP.S2CDio.VoltageRdg	22-31
AP.S2CDio.VoltageReady	22-33
AP.S2CDio.VoltageSettling	22-33
AP.S2CDio.VoltageTrig	22-34
DSP Audio Analyzer	23
AP.S2CDsp.Analyzer.ChACoupling	23-1
AP.S2CDsp.Analyzer.ChAFreqRdg	23-1
AP.S2CDsp.Analyzer.ChAFreqReady	23-3
AP.S2CDsp.Analyzer.ChAFreqSettling	23-3
AP.S2CDsp.Analyzer.ChAFreqTrig	23-4
AP.S2CDsp.Analyzer.ChALevelRdg	23-4
AP.S2CDsp.Analyzer.ChALevelReady	23-5
AP.S2CDsp.Analyzer.ChALevelSettling	23-6
AP.S2CDsp.Analyzer.ChALevelTrig	23-6
AP.S2CDsp.Analyzer.ChARange	23-6
AP.S2CDsp.Analyzer.ChARangeAuto	23-7
AP.S2CDsp.Analyzer.ChBCoupling	23-7
AP.S2CDsp.Analyzer.ChBFreqRdg	23-8
AP.S2CDsp.Analyzer.ChBFreqReady	23-9
AP.S2CDsp.Analyzer.ChBFreqSettling	23-10
AP.S2CDsp.Analyzer.ChBFreqTrig	23-10
AP.S2CDsp.Analyzer.ChBLevelRdg	23-11
AP.S2CDsp.Analyzer.ChBLevelReady	23-11
AP.S2CDsp.Analyzer.ChBLevelSettling	23-12
AP.S2CDsp.Analyzer.ChBLevelTrig	23-12
AP.S2CDsp.Analyzer.ChBRange	23-13
AP.S2CDsp.Analyzer.ChBRangeAuto	23-13
AP.S2CDsp.Analyzer.FuncBPBRFreq	23-14
AP.S2CDsp.Analyzer.FuncBPBRTuning	23-15
AP.S2CDsp.Analyzer.FuncChARange	23-16
AP.S2CDsp.Analyzer.FuncChARangeAuto	23-17
AP.S2CDsp.Analyzer.FuncChARdg	23-17
AP.S2CDsp.Analyzer.FuncChAReady	23-18
AP.S2CDsp.Analyzer.FuncChASettling	23-19
AP.S2CDsp.Analyzer.FuncChATrig	23-19
AP.S2CDsp.Analyzer.FuncChBRange	23-20

AP.S2CDsp.Analyzer.FuncChBRangeAuto	23-20
AP.S2CDsp.Analyzer.FuncChBRdg	23-21
AP.S2CDsp.Analyzer.FuncChBReady	23-22
AP.S2CDsp.Analyzer.FuncChBSettling	23-23
AP.S2CDsp.Analyzer.FuncChBTrig	23-23
AP.S2CDsp.Analyzer.FuncDetector	23-23
AP.S2CDsp.Analyzer.FuncFilter	23-25
AP.S2CDsp.Analyzer.FuncFilterHP	23-26
AP.S2CDsp.Analyzer.FuncFilterLP	23-26
AP.S2CDsp.Analyzer.FuncMode	23-27
AP.S2CDsp.Analyzer.FuncPhaseMode	23-29
AP.S2CDsp.Analyzer.InputFormat	23-30
AP.S2CDsp.Analyzer.RdgRate	23-30

Digital Data Analyzer 24

AP.S2CDsp.BitTest.ChADataRdg	24-1
AP.S2CDsp.BitTest.ChADataReady	24-2
AP.S2CDsp.BitTest.ChADataTrig	24-2
AP.S2CDsp.BitTest.ChAErrRdg	24-3
AP.S2CDsp.BitTest.ChAErrReady	24-4
AP.S2CDsp.BitTest.ChAErrTrig	24-4
AP.S2CDsp.BitTest.ChBDataRdg	24-5
AP.S2CDsp.BitTest.ChBDataReady	24-6
AP.S2CDsp.BitTest.ChBDataTrig	24-6
AP.S2CDsp.BitTest.ChBErrRdg	24-7
AP.S2CDsp.BitTest.ChBErrReady	24-8
AP.S2CDsp.BitTest.ChBErrTrig	24-8
AP.S2CDsp.BitTest.DisplayError	24-9
AP.S2CDsp.BitTest.FreezeOnError	24-10
AP.S2CDsp.BitTest.RdgRate	24-10
AP.S2CDsp.BitTest.Wfm	24-11

Multitone Audio Analyzer 25

AP.S2CDsp.FastTest.Ch1Rdg	25-1
AP.S2CDsp.FastTest.Ch1Ready	25-2
AP.S2CDsp.FastTest.Ch1Source	25-2
AP.S2CDsp.FastTest.Ch1Trig	25-3
AP.S2CDsp.FastTest.Ch2Rdg	25-3
AP.S2CDsp.FastTest.Ch2Ready	25-4
AP.S2CDsp.FastTest.Ch2Source	25-5
AP.S2CDsp.FastTest.Ch2Trig	25-6
AP.S2CDsp.FastTest.FFTLength	25-6
AP.S2CDsp.FastTest.FreqRes	25-8

AP.S2CDsp.FastTest.InputFormat 25-8
 AP.S2CDsp.FastTest.Mode 25-9
 AP.S2CDsp.FastTest.PhaseDisplay 25-13
 AP.S2CDsp.FastTest.Processing 25-13
 AP.S2CDsp.FastTest.TrigDelay 25-15
 AP.S2CDsp.FastTest.TrigSource 25-15

FFT Spectrum Analyzer 26

AP.S2CDsp.FFT.AcquireLength 26-1
 AP.S2CDsp.FFT.Averages 26-1
 AP.S2CDsp.FFT.AveragesType 26-3
 AP.S2CDsp.FFT.Ch1Rdg 26-4
 AP.S2CDsp.FFT.Ch1Ready 26-5
 AP.S2CDsp.FFT.Ch1Source 26-6
 AP.S2CDsp.FFT.Ch1Trig 26-6
 AP.S2CDsp.FFT.Ch2Rdg 26-7
 AP.S2CDsp.FFT.Ch2Ready 26-7
 AP.S2CDsp.FFT.Ch2Source 26-8
 AP.S2CDsp.FFT.Ch2Trig 26-9
 AP.S2CDsp.FFT.InputFormat 26-9
 AP.S2CDsp.FFT.StartTime 26-9
 AP.S2CDsp.FFT.SubtractDC 26-10
 AP.S2CDsp.FFT.TransformLength 26-11
 AP.S2CDsp.FFT.TrigDelay 26-11
 AP.S2CDsp.FFT.TrigLevel 26-12
 AP.S2CDsp.FFT.TrigPolarity 26-13
 AP.S2CDsp.FFT.TrigSensitivity 26-13
 AP.S2CDsp.FFT.TrigSource 26-13
 AP.S2CDsp.FFT.WfmDisplay 26-15
 AP.S2CDsp.FFT.Window 26-17

Harmonic Analyzer 27

AP.S2CDsp.Harmonic.Ch1FreqRdg 27-1
 AP.S2CDsp.Harmonic.Ch1FreqReady 27-2
 AP.S2CDsp.Harmonic.Ch1FreqSettling 27-3
 AP.S2CDsp.Harmonic.Ch1FreqTrig 27-3
 AP.S2CDsp.Harmonic.Ch1FundRdg 27-4
 AP.S2CDsp.Harmonic.Ch1FundReady 27-4
 AP.S2CDsp.Harmonic.Ch1FundSettling 27-5
 AP.S2CDsp.Harmonic.Ch1FundTrig 27-5
 AP.S2CDsp.Harmonic.Ch1Source 27-6
 AP.S2CDsp.Harmonic.Ch1Sum1Harmonics 27-6
 AP.S2CDsp.Harmonic.Ch1Sum1Rdg 27-7

AP.S2CDsp.Harmonic.Ch1Sum1Ready	27-9
AP.S2CDsp.Harmonic.Ch1Sum1RelUnits	27-10
AP.S2CDsp.Harmonic.Ch1Sum1Settling	27-10
AP.S2CDsp.Harmonic.Ch1Sum1Trig	27-10
AP.S2CDsp.Harmonic.Ch1Sum2Harmonics	27-11
AP.S2CDsp.Harmonic.Ch1Sum2Rdg	27-12
AP.S2CDsp.Harmonic.Ch1Sum2Ready	27-13
AP.S2CDsp.Harmonic.Ch1Sum2RelUnits	27-13
AP.S2CDsp.Harmonic.Ch1Sum2Settling	27-14
AP.S2CDsp.Harmonic.Ch1Sum2Trig	27-14
AP.S2CDsp.Harmonic.Ch2FreqRdg	27-15
AP.S2CDsp.Harmonic.Ch2FreqReady	27-16
AP.S2CDsp.Harmonic.Ch2FreqSettling	27-17
AP.S2CDsp.Harmonic.Ch2FreqTrig	27-17
AP.S2CDsp.Harmonic.Ch2FundRdg	27-17
AP.S2CDsp.Harmonic.Ch2FundReady	27-18
AP.S2CDsp.Harmonic.Ch2FundSettling	27-19
AP.S2CDsp.Harmonic.Ch2FundTrig	27-19
AP.S2CDsp.Harmonic.Ch2Source	27-19
AP.S2CDsp.Harmonic.Ch2Sum1Harmonics	27-20
AP.S2CDsp.Harmonic.Ch2Sum1Rdg	27-21
AP.S2CDsp.Harmonic.Ch2Sum1Ready	27-23
AP.S2CDsp.Harmonic.Ch2Sum1RelUnits	27-23
AP.S2CDsp.Harmonic.Ch2Sum1Settling	27-24
AP.S2CDsp.Harmonic.Ch2Sum1Trig	27-24
AP.S2CDsp.Harmonic.Ch2Sum2Harmonics	27-25
AP.S2CDsp.Harmonic.Ch2Sum2Rdg	27-26
AP.S2CDsp.Harmonic.Ch2Sum2Ready	27-26
AP.S2CDsp.Harmonic.Ch2Sum2RelUnits	27-27
AP.S2CDsp.Harmonic.Ch2Sum2Settling	27-27
AP.S2CDsp.Harmonic.Ch2Sum2Trig	27-28
AP.S2CDsp.Harmonic.Freq	27-28
AP.S2CDsp.Harmonic.InputFormat	27-29
AP.S2CDsp.Harmonic.Selectivity	27-30
AP.S2CDsp.Harmonic.Tuning	27-30

Digital Interface Analyzer 28

AP.S2CDsp.Intervu.AmplVsTime	28-1
AP.S2CDsp.Intervu.AudioMonitor	28-2
AP.S2CDsp.Intervu.Averages	28-3
AP.S2CDsp.Intervu.JitterDetection	28-4
AP.S2CDsp.Intervu.TrigErrorCoding	28-6
AP.S2CDsp.Intervu.TrigErrorConfidence	28-6

AP.S2CDsp.Intervu.TrigErrorLock 28-7
 AP.S2CDsp.Intervu.TrigErrorParity 28-7
 AP.S2CDsp.Intervu.TrigPolarity 28-9
 AP.S2CDsp.Intervu.TrigSource 28-10
 AP.S2CDsp.Intervu.Window 28-12

Quasi-Anechoic Acoustical Tester 29

AP.S2CDsp.MLS.Ch1Rdg 29-1
 AP.S2CDsp.MLS.Ch1Ready 29-2
 AP.S2CDsp.MLS.Ch1Source 29-3
 AP.S2CDsp.MLS.Ch1Trig 29-3
 AP.S2CDsp.MLS.Ch2Rdg 29-4
 AP.S2CDsp.MLS.Ch2Ready 29-4
 AP.S2CDsp.MLS.Ch2Source 29-5
 AP.S2CDsp.MLS.Ch2Trig 29-6
 AP.S2CDsp.MLS.InputFormat 29-6
 AP.S2CDsp.MLS.TimeDelay 29-6
 AP.S2CDsp.MLS.TimeDisplay 29-8
 AP.S2CDsp.MLS.TrigSource 29-9
 AP.S2CDsp.MLS.WfmDisplay 29-9
 AP.S2CDsp.MLS.WindowETime 29-10
 AP.S2CDsp.MLS.WindowStart 29-11
 AP.S2CDsp.MLS.WindowStop 29-12

System Two DSP Program & Reference 30

AP.S2CDsp.Program 30-1
 AP.S2CDsp.RefCh1dBr 30-5
 AP.S2CDsp.RefCh2dBr 30-6
 AP.S2CDsp.RefFreq 30-7
 AP.S2CDsp.RefVFS 30-7

Serial Interface 31

AP.SI.RecClock.Bit.Freq 31-1
 AP.SI.RecClock.Bit.Invert 31-1
 AP.SI.RecClock.Bit.Source 31-1
 AP.SI.RecClock.Frame.Freq 31-2
 AP.SI.RecClock.Frame.Invert 31-2
 AP.SI.RecClock.Frame.Shift 31-2
 AP.SI.RecClock.Frame.Source 31-3
 AP.SI.RecClock.Frame.StartPosition 31-3
 AP.SI.RecClock.Frame.StopPosition 31-3

AP.SI.RecClock.Master.Freq	31-3
AP.SI.RecClock.Master.Invert	31-4
AP.SI.RecClock.Master.Source	31-4
AP.SI.RecClock.Ratio.BitToSubFrame	31-4
AP.SI.RecClock.Ratio.MasterToBit	31-5
AP.SI.RecClock.Ratio.SubFrameToFrame	31-5
AP.SI.RecClock.Ratio.SystemToMaster	31-5
AP.SI.RecClock.SubFrame.Freq	31-5
AP.SI.RecClock.SubFrame.Invert	31-6
AP.SI.RecClock.SubFrame.Source	31-6
AP.SI.RecClock.SubFrame.Width	31-6
AP.SI.RecClock.System.Freq	31-6
AP.SI.RecClock.System.Invert	31-7
AP.SI.RecClock.System.Source	31-7
AP.SI.RecData.ChASubFrame	31-7
AP.SI.RecData.ChBSubFrame	31-8
AP.SI.RecData.Format.Decode	31-8
AP.SI.RecData.Format.LSBFirst	31-9
AP.SI.RecData.Format.NumBits	31-9
AP.SI.RecData.Format.StartPosition	31-9
AP.SI.TransClock.Bit.Freq	31-10
AP.SI.TransClock.Bit.Invert	31-10
AP.SI.TransClock.Bit.Source	31-10
AP.SI.TransClock.Frame.Freq	31-11
AP.SI.TransClock.Frame.Invert	31-11
AP.SI.TransClock.Frame.Shift	31-11
AP.SI.TransClock.Frame.Source	31-12
AP.SI.TransClock.Frame.StartPosition	31-12
AP.SI.TransClock.Frame.StopPosition	31-12
AP.SI.TransClock.Master.Freq	31-12
AP.SI.TransClock.Master.Invert	31-13
AP.SI.TransClock.Master.Source	31-13
AP.SI.TransClock.Ratio.BitToSubFrame	31-13
AP.SI.TransClock.Ratio.MasterToBit	31-14
AP.SI.TransClock.Ratio.SubFrameToFrame	31-14
AP.SI.TransClock.Ratio.SystemToMaster	31-14
AP.SI.TransClock.SubFrame.Freq	31-14
AP.SI.TransClock.SubFrame.Invert	31-15
AP.SI.TransClock.SubFrame.Source	31-15
AP.SI.TransClock.SubFrame.Width	31-15
AP.SI.TransClock.System.Freq	31-16
AP.SI.TransClock.System.Invert	31-16
AP.SI.TransClock.System.Source	31-16

AP.SI.TransData.ChASubFrame	31-17
AP.SI.TransData.ChBSubFrame	31-17
AP.SI.TransData.Format.Encode	31-17
AP.SI.TransData.Format.LSBFirst	31-18
AP.SI.TransData.Format.NumBits	31-18
AP.SI.TransData.Format.StartPosition	31-18
Speaker	32
AP.Speaker.Mode	32-1
AP.Speaker.Source	32-2
Sweep	33
AP.Sweep.AbortTime	33-1
AP.Sweep.Append	33-2
AP.Sweep.CopyData1To2	33-3
AP.Sweep.CopyData2To1	33-4
AP.Sweep.CreateGraph	33-5
AP.Sweep.CreateTable	33-5
AP.Sweep.Data1.AutoDiv	33-5
AP.Sweep.Data1.Autoscale	33-6
AP.Sweep.Data1.Bottom	33-6
AP.Sweep.Data1.Div	33-7
AP.Sweep.Data1.Id	33-7
AP.Sweep.Data1.Limits	33-7
AP.Sweep.Data1.LogLin	33-9
AP.Sweep.Data1.Top	33-9
AP.Sweep.Data2.AutoDiv	33-10
AP.Sweep.Data2.Autoscale	33-10
AP.Sweep.Data2.Bottom	33-11
AP.Sweep.Data2.Div	33-11
AP.Sweep.Data2.Id	33-11
AP.Sweep.Data2.Limits	33-12
AP.Sweep.Data2.LogLin	33-12
AP.Sweep.Data2.Top	33-13
AP.Sweep.Data3.Id	33-13
AP.Sweep.Data3.Limits	33-14
AP.Sweep.Data4.Id	33-14
AP.Sweep.Data4.Limits	33-15
AP.Sweep.Data5.Id	33-16
AP.Sweep.Data5.Limits	33-16
AP.Sweep.Data6.Id	33-17
AP.Sweep.Data6.Limits	33-17
AP.Sweep.GraphType	33-18

AP.Sweep.IsRunning	33-18
AP.Sweep.PreSweepDelay	33-19
AP.Sweep.Recompare	33-19
AP.Sweep.Repeat	33-20
AP.Sweep.Reprocess	33-21
AP.Sweep.Rettransform	33-23
AP.Sweep.ReverseChannels	33-24
AP.Sweep.SinglePoint	33-25
AP.Sweep.Source1.AutoDiv	33-25
AP.Sweep.Source1.Div	33-26
AP.Sweep.Source1.EndOn	33-26
AP.Sweep.Source1.Id	33-27
AP.Sweep.Source1.LogLin	33-28
AP.Sweep.Source1.MinLevel	33-28
AP.Sweep.Source1.MinLevelId	33-29
AP.Sweep.Source1.Multiply	33-29
AP.Sweep.Source1.Spacing	33-30
AP.Sweep.Source1.Start	33-30
AP.Sweep.Source1.Steps	33-31
AP.Sweep.Source1.StepSize	33-31
AP.Sweep.Source1.Stop	33-32
AP.Sweep.Source1.Table	33-32
AP.Sweep.Source2.Id	33-33
AP.Sweep.Source2.LogLin	33-33
AP.Sweep.Source2.Multiply	33-34
AP.Sweep.Source2.Start	33-34
AP.Sweep.Source2.Steps	33-34
AP.Sweep.Source2.StepSize	33-35
AP.Sweep.Source2.Stop	33-35
AP.Sweep.Spectrum	33-36
AP.Sweep.Start	33-37
AP.Sweep.StartNoWait	33-38
AP.Sweep.StartWithAppend	33-38
AP.Sweep.StartWithRepeat	33-39
AP.Sweep.Stereo	33-39
AP.Sweep.Stop	33-39
AP.Sweep.Timeout	33-40
AP.Sweep.Waveform	33-41
Switcher	34
AP.SWR.ChABIn	34-1
AP.SWR.ChABInOut	34-1
AP.SWR.ChABOut	34-2

AP.SWR.ChAIn	34-2
AP.SWR.ChAInOut	34-2
AP.SWR.ChAOut	34-3
AP.SWR.ChBIn	34-3
AP.SWR.ChBInOut	34-4
AP.SWR.ChBOffset	34-4
AP.SWR.ChBOut	34-4
AP.SWR.Mode	34-5
AP.SWR.OutOffset	34-7
Sync/Ref Input	35
AP.Sync.DelayRdg	35-1
AP.Sync.DelayReady	35-2
AP.Sync.DelaySettling	35-2
AP.Sync.DelayTrig	35-3
AP.Sync.FrameLock	35-3
AP.Sync.Freq	35-4
AP.Sync.FreqRdg	35-5
AP.Sync.FreqReady	35-6
AP.Sync.FreqTrig	35-7
AP.Sync.Impedance	35-7
AP.Sync.OutDelay	35-8
AP.Sync.OutDelayFromRef	35-8
AP.Sync.OutFramesSyncInvert	35-9
AP.Sync.OutJitterClockOutputs	35-9
AP.Sync.OutOfRangeRdg	35-10
AP.Sync.Source	35-10
AP.Sync.SourceInput	35-10
Appendix A Settling Algorithm	A
Description	A-1
Settling Parameter Discriptions	A-1
Appendix B Parameter ID# List	B
Appendix C FFT Window Descriptions	C
Appendix D Extensions Error Codes	D
Errors	D-1
General Errors	D-2
General Warnings	D-7
Exception Errors	D-8

CONTENTS

DSP Errors D-10
DSP Warnings D-12

Appendix E Analog Filter ID# List E
Numerical Listing E-1

APWIN Basic Extensions Reference

Introduction

This chapter of the manual is divided into three sections.

The first section consists of APWIN system panels listed alphabetically by panel title. Each page consists of one or more panels and the commands applicable to each panel.

The second section consists of an alphabetical listing of all APBASIC extensions.

The third section consists of technical reference information for the command extensions sorted alphabetically. Each command contains many of the following parts.

Part	Description
<i>Syntax:</i>	Programming usage information.
<i>Command type:</i>	Method or Property
<i>Data type:</i>	Setting Data Type.
<i>Result:</i>	Query Data Type.
<i>Description:</i>	Technical Information.
<i>See also:</i>	Commands related to the current command that may contain relevant information.
<i>Example:</i>	Example procedure/macro
<i>Example Output:</i>	When an example program produces output to the immediate window of the Procedure/Macro Editor or output to a file a sample of what the output will be shown in this location.
<i>Comments:</i>	Additional information relating to the example procedure/macro.

Manual Conventions

This manual uses the following typographic conventions.

Example	Description
<i>event,</i> <i>var, arg</i>	For the syntax part of each command, italicized words indicate placeholders where the user must enter additional information.
FILENAME.TXT	Words in all CAPITOL letters indicate file names.
Sub Main AP.Gen.Amp = 1.0 End Sub	This font is used in all example macros and code modules.
[<i>expression list</i>]	In syntax, items inside square brackets are optional.
{ <i>While</i> <i>Until</i> }	In syntax, braces and a vertical bar indicate a choice between two or more items.
Command	For the syntax part of each command, the bold characters identify the part of the command that must be entered.
AP.Prompt. _ Text "Example"	The line continue character (_) is used to indicate that the code from one line to the next should be typed on one line.

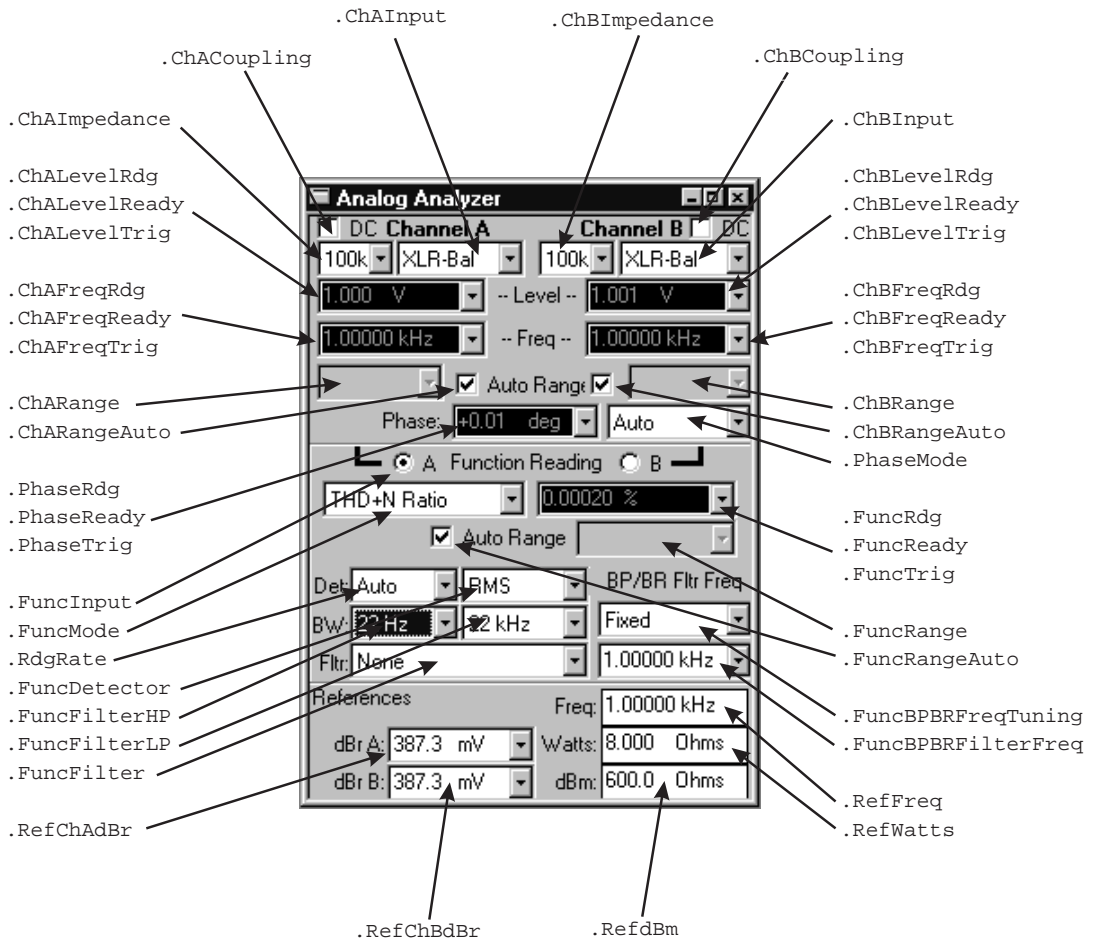
System Panels

Analog Analyzer

All commands on this page start with the following:

AP.Anlr

Example: AP.Anlr.ChACoupling

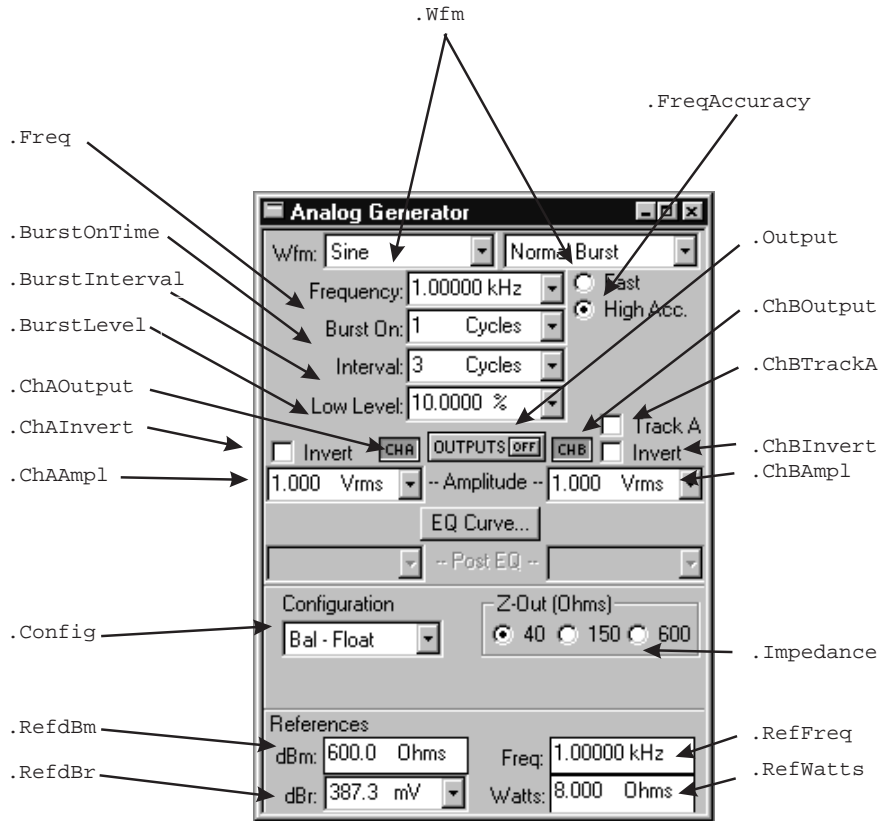


Analog Generator ...

All commands on this page start with the following:

AP.Gen

Example: AP.Gen.Freq

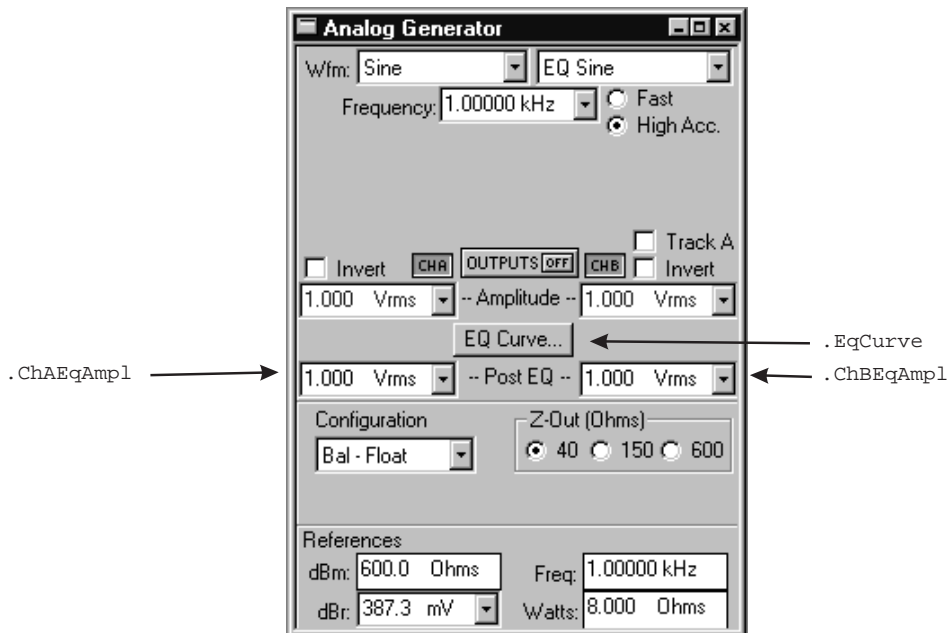


Analog Generator Continued ...

All commands on this page start with the following:

AP.Gen

Example: AP.Gen.IMFreq

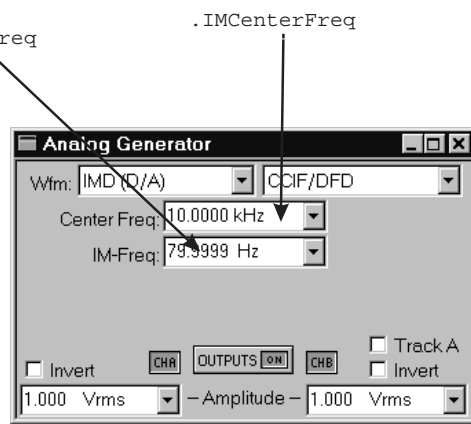
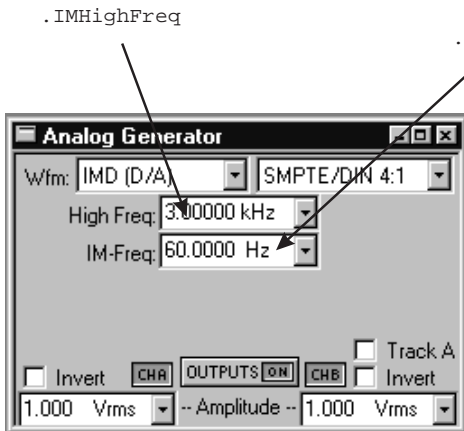
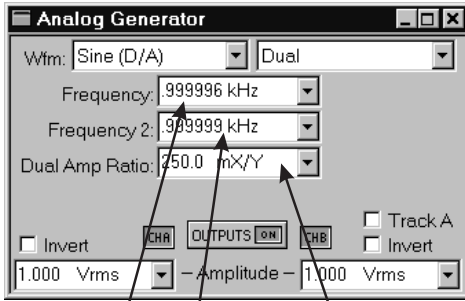
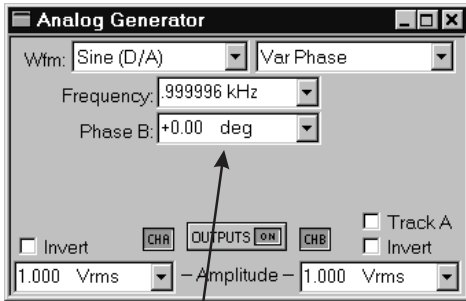


Analog Generator Continued

All commands on this page start with the following:

AP.Gen

Example: AP.Gen.Phase

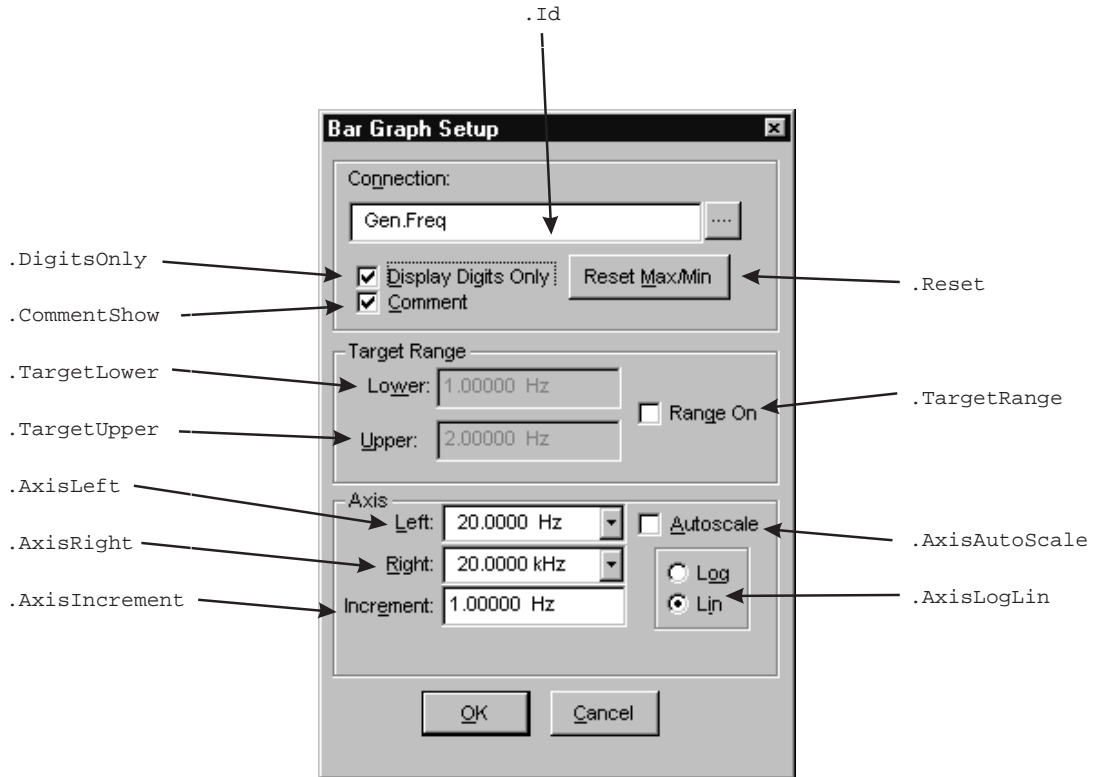


BarGraph

All commands on this page start with the following:

AP.Bar

Example: `AP.Bar.FastTest.InputFormat`

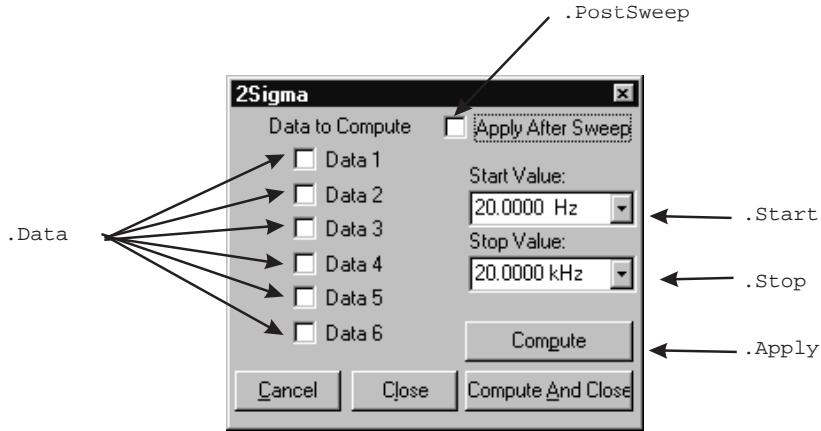


Computes ...

All commands for the top diagram start with the following:

AP.Compute.Sigma

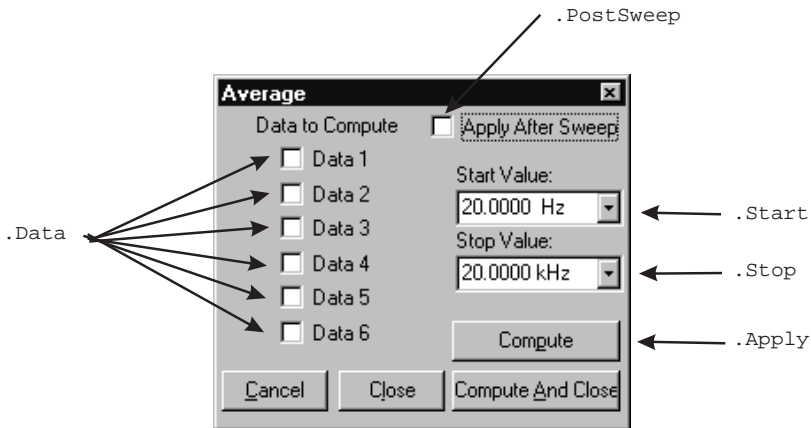
Example: AP.Compute.Sigma.Apply



All commands for the bottom diagram start with the following:

AP.Compute.Avg

Example: AP.Compute.Avg.Apply

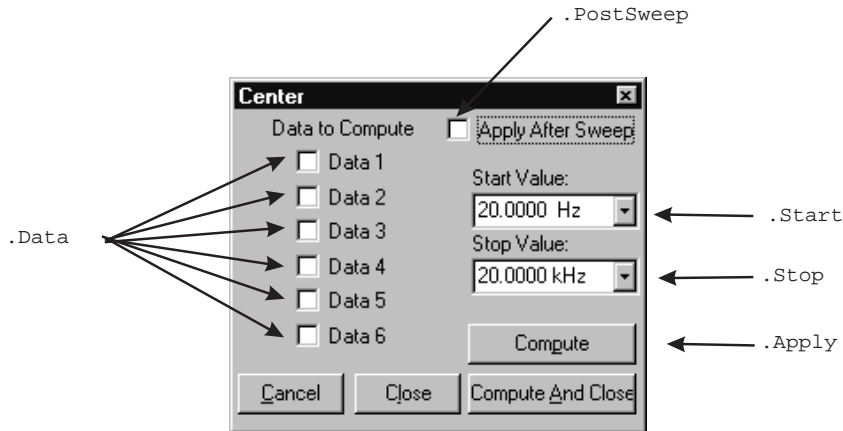


Computes Continued ...

All commands for the top diagram start with the following:

AP.Compute.Center

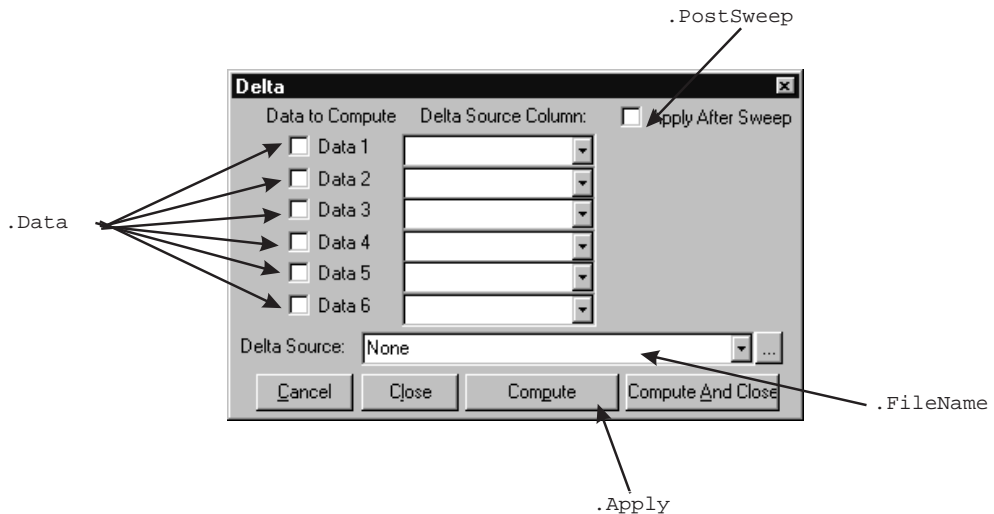
Example: AP.Compute.Center.Apply



All commands for the bottom diagram start with the following:

AP.Compute.Delta

Example: AP.Compute.Delta.Apply

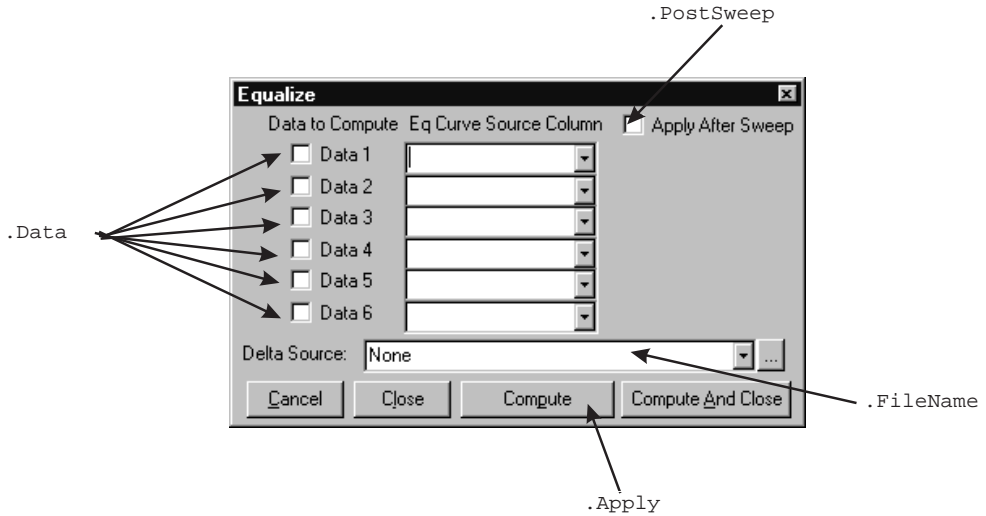


Computes Continued ...

All commands on this page start with the following:

AP.Compute.Equalize

Example: AP.Compute.Equalize.Apply

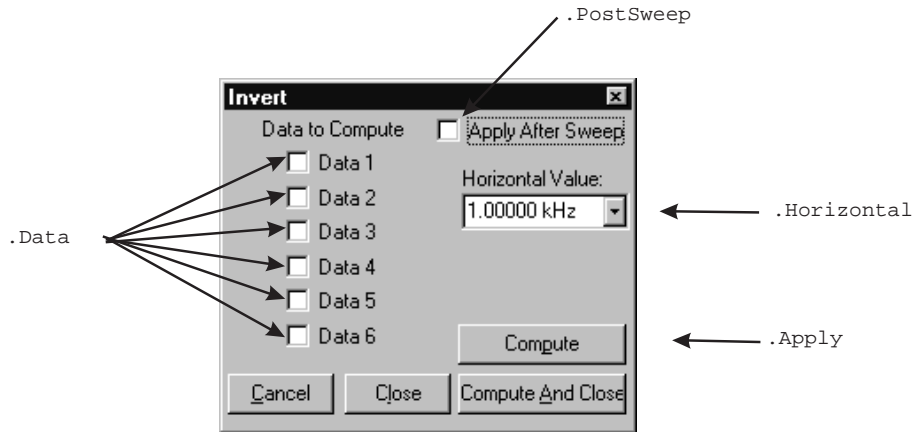


Computes Continued ...

All commands for the top diagram start with the following:

AP.Compute.Invert

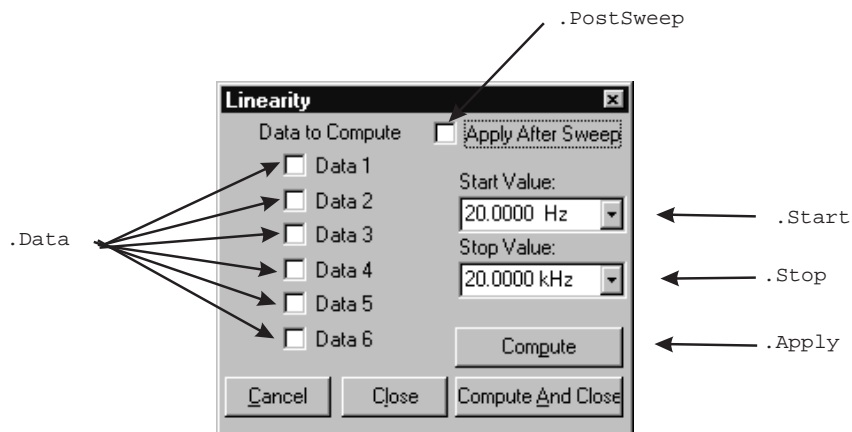
Example: AP.Compute.Invert.Apply



All commands for the bottom diagram start with the following:

AP.Compute.Linearity

Example: AP.Compute.Linearity.Apply

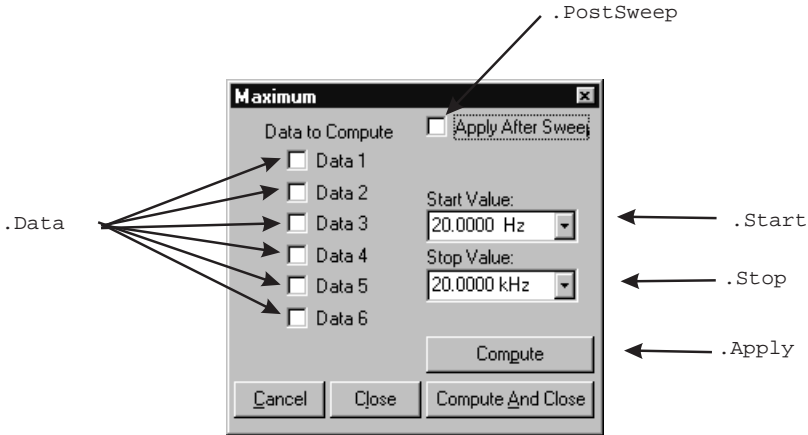


Computes Continued ...

All commands for the top diagram start with the following:

AP.Compute.Max

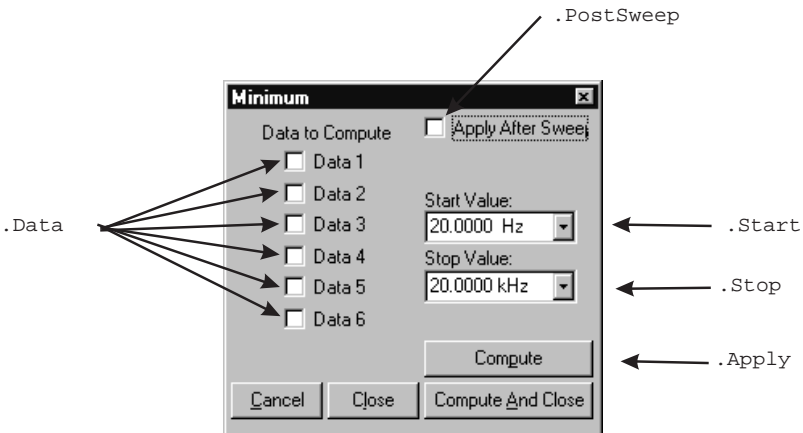
Example: AP.Compute.Max.Apply



All commands for the bottom diagram start with the following:

AP.Compute.Min

Example: AP.Compute.Min.Apply

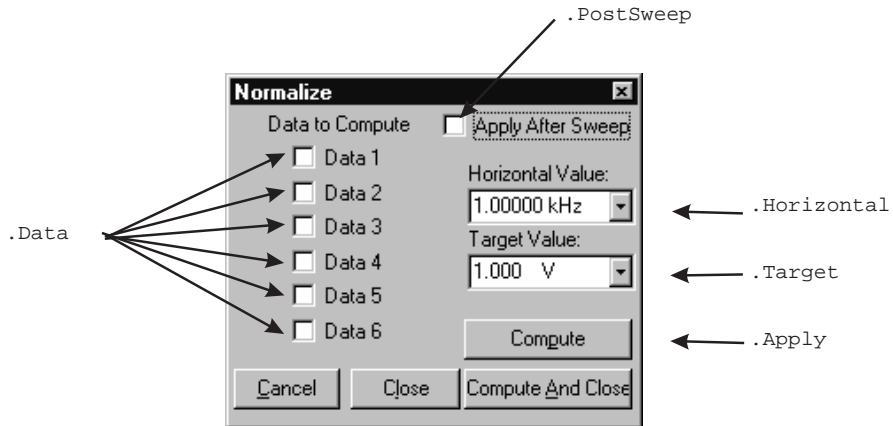


Computes Continued ...

All commands for the top diagram start with the following:

AP.Compute.Normalize

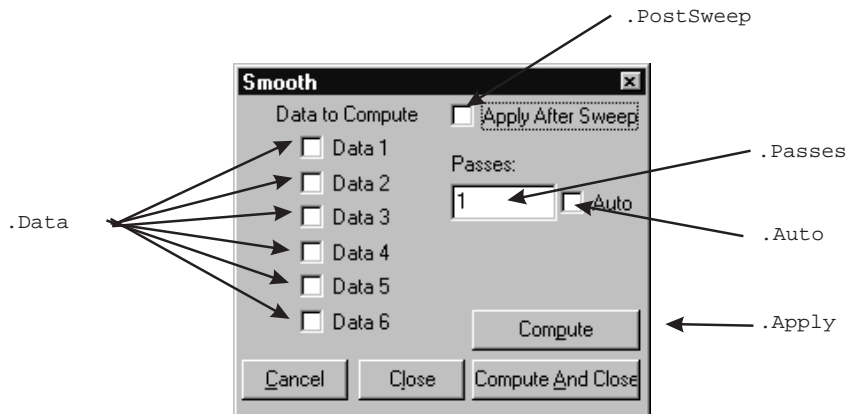
Example: AP.Compute.Normalize.Apply



All commands for the bottom diagram start with the following:

AP.Compute.Smooth

Example: AP.Compute.Smooth.Apply

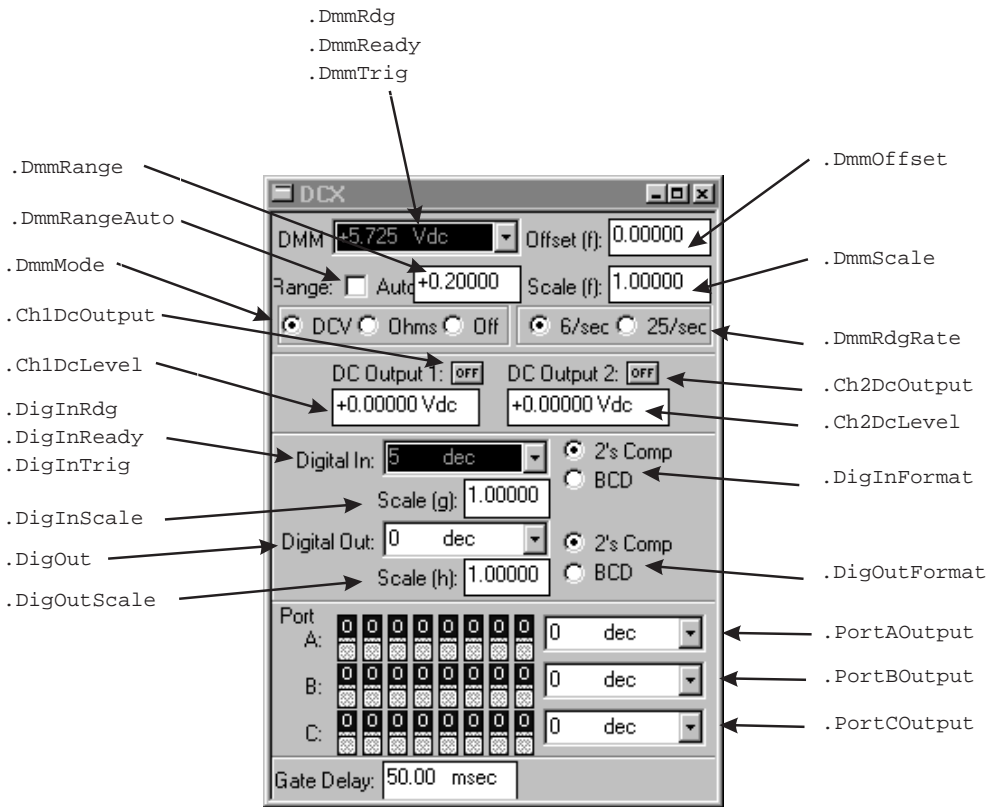


DCX-127

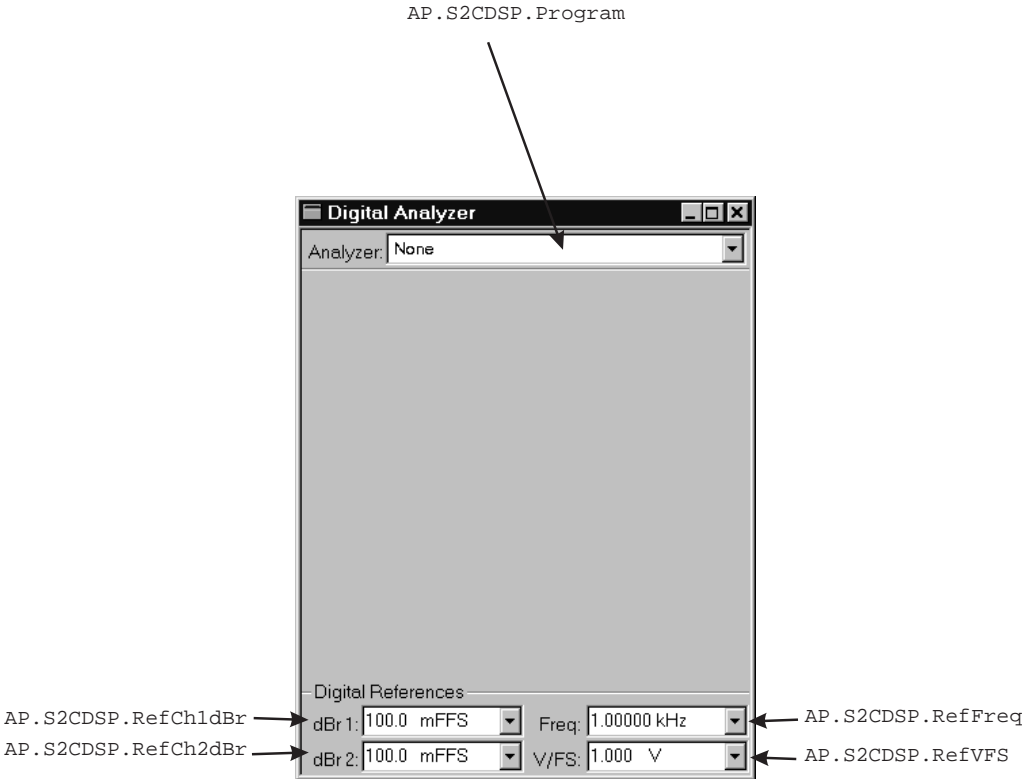
All commands on this page start with the following:

AP.DCX

Example: AP.S2CDsp.FastTest.InputFormat



Digital Analyzer Panels

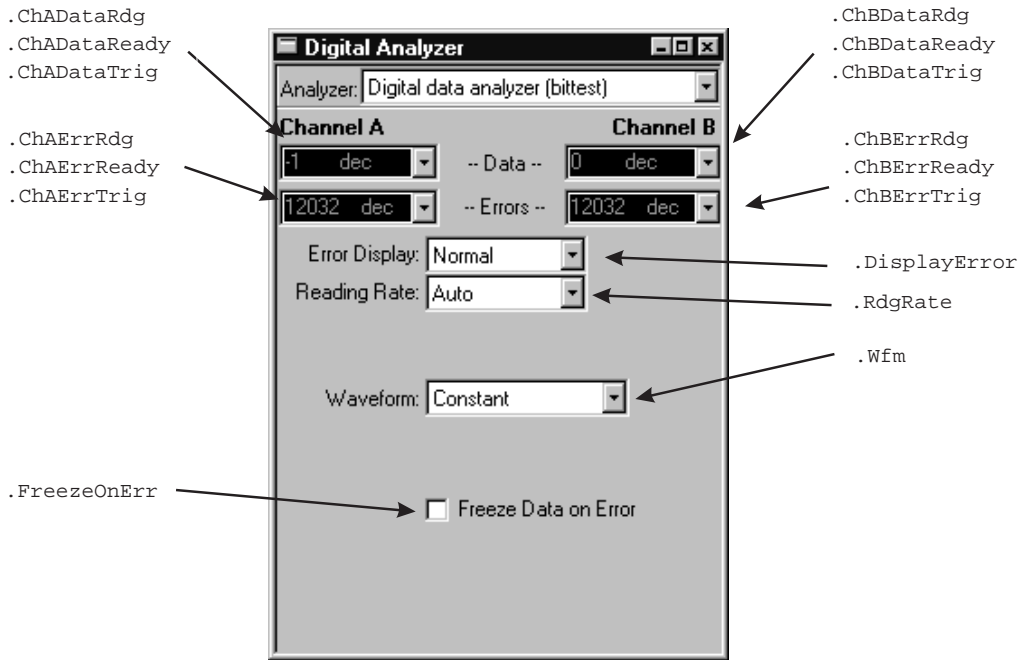


Digital Data Analyzer (BITTEST)

All commands on this page start with the following:

AP.S2CDsp.BitTest

Example: AP.S2CDsp.BitTest.ChADataRdg

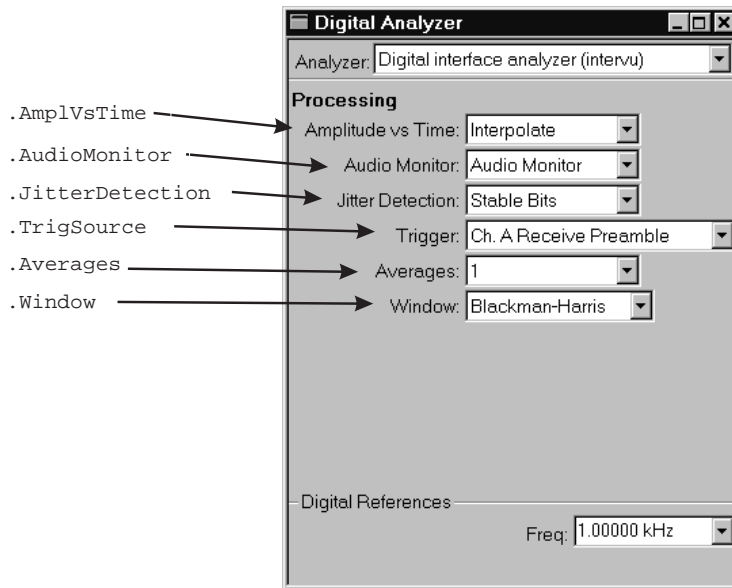


Digital Interface Analyzer (INTERVU)

All commands on this page start with the following:

AP.S2CDsp.Intervu

Example: `AP.S2CDsp.Intervu.AmplVsTime`

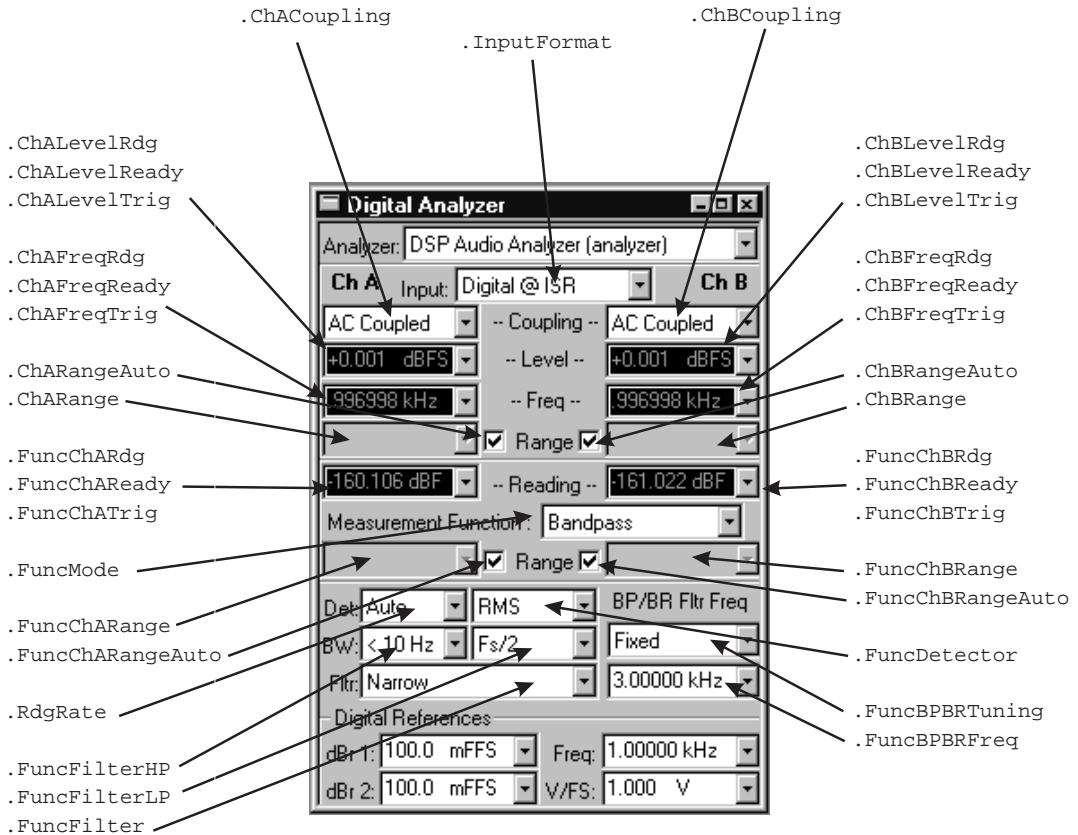


DSP Audio Analyzer (ANALYZER)

All commands on this page start with the following:

AP.S2CDsp.Analyzer

Example: AP.S2CDsp.Analyzer.ChALevelRdg

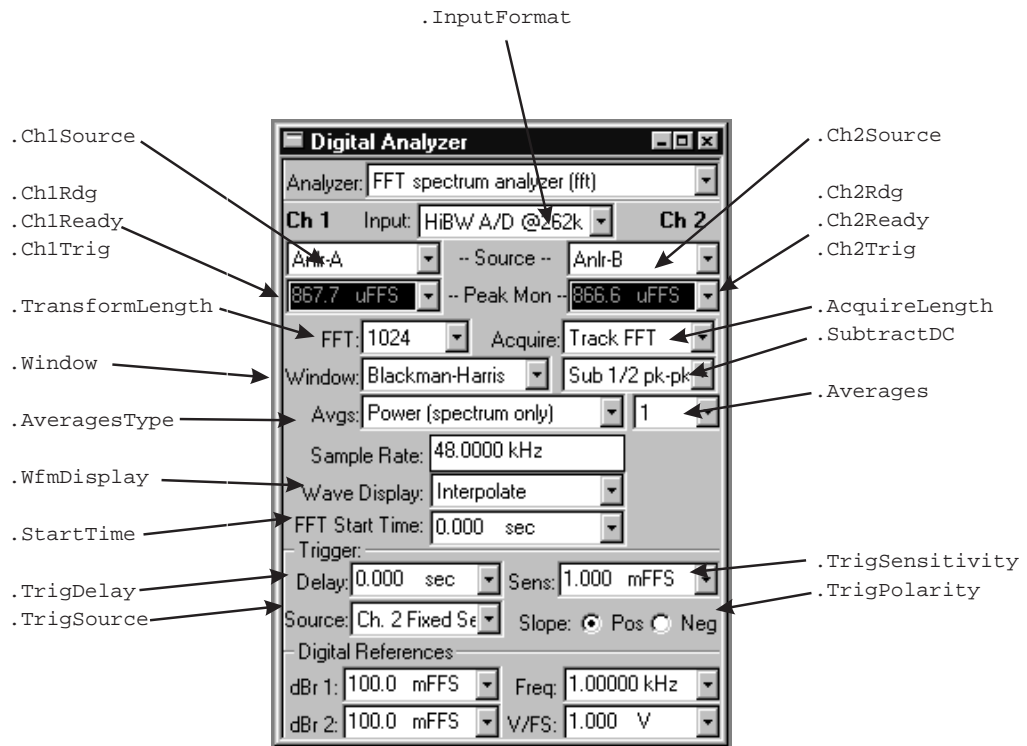


FFT Spectrum Analyzer (FFT)

All commands on this page start with the following:

AP.S2CDsp.FFT

Example: AP.S2CDsp.FFT.InputFormat

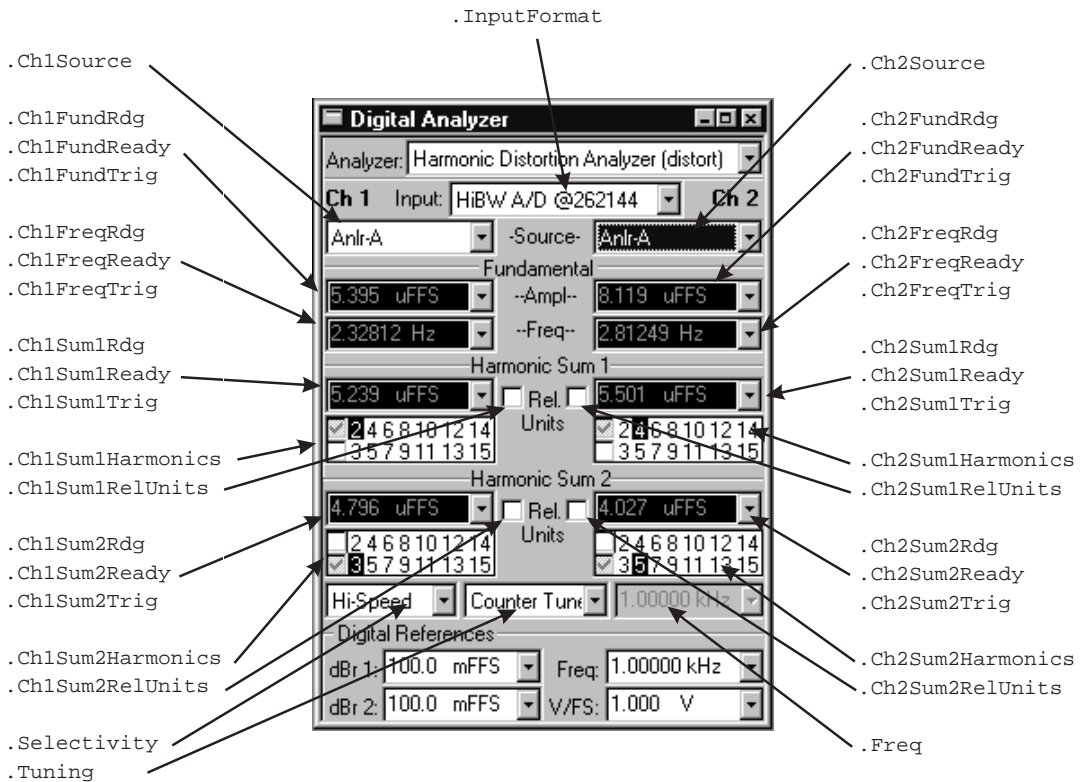


Harmonic Distortion Analyzer (DISTORT)

All commands on this page start with the following:

AP.S2CDsp.Harmonic

Example: AP.S2CDsp.Harmonic.InputFormat

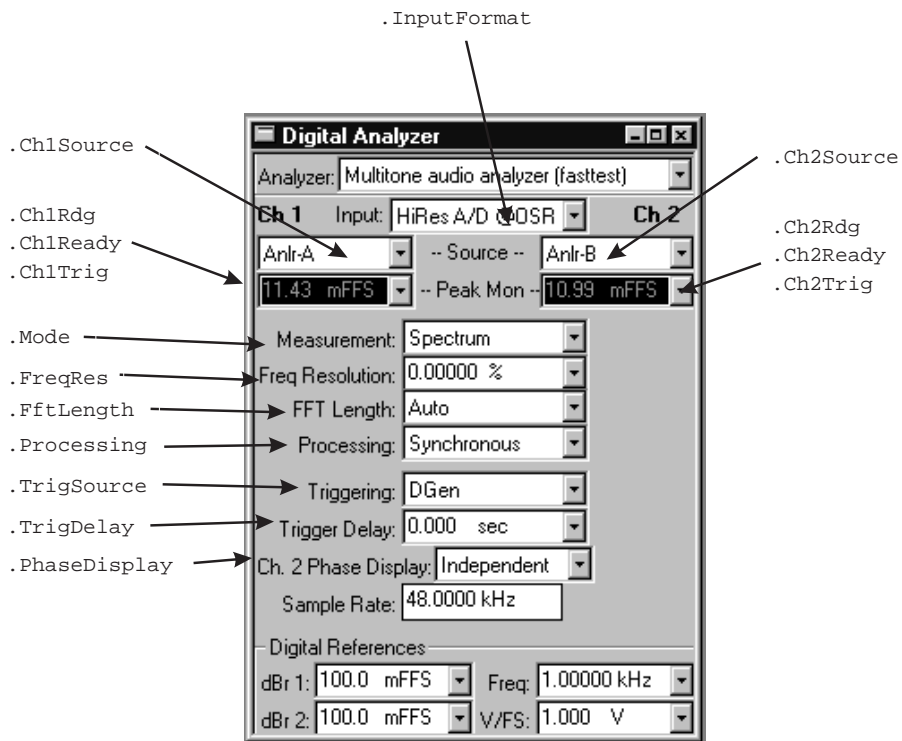


Multitone Audio Analyzer (FASTTEST)

All commands on this page start with the following:

AP.S2CDsp.FastTest

Example: AP.S2CDsp.FastTest.InputFormat

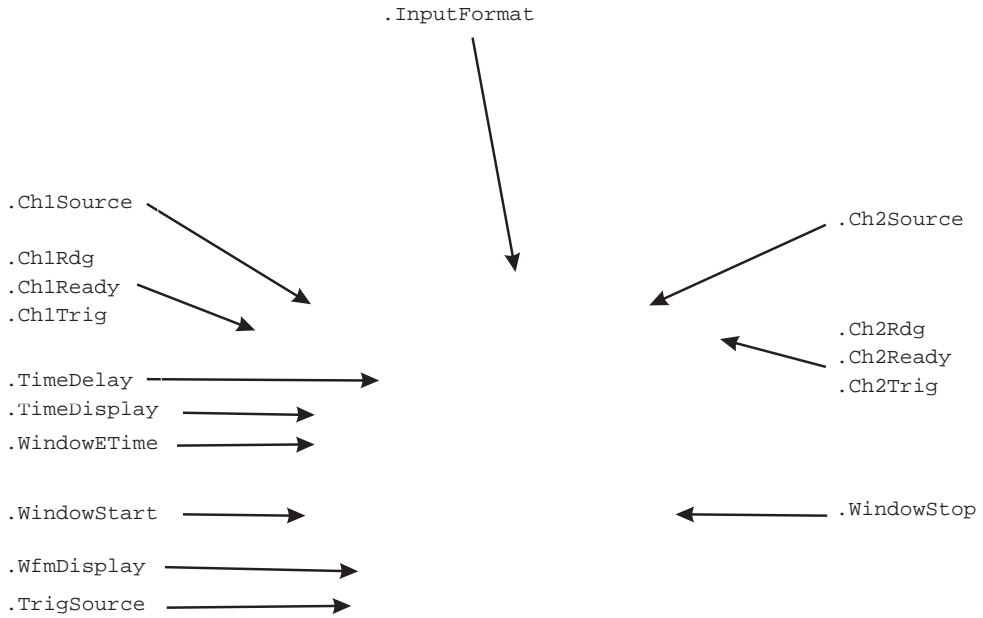


Quasi-Anechoic Acoustical Tester (MLS)

All commands on this page start with the following:

AP.S2CDsp.Mls

Example: AP.S2CDSP.Mls.InputFormat

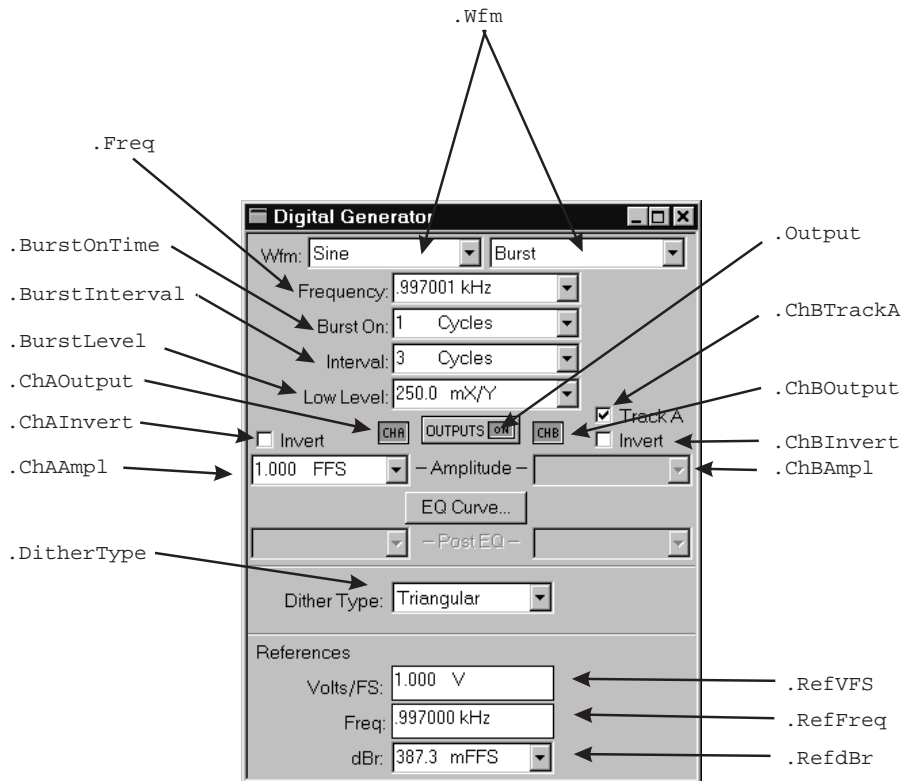


Digital Generator ...

All commands on this page start with the following:

AP.Dgen

Example: AP.Dgen.Freq

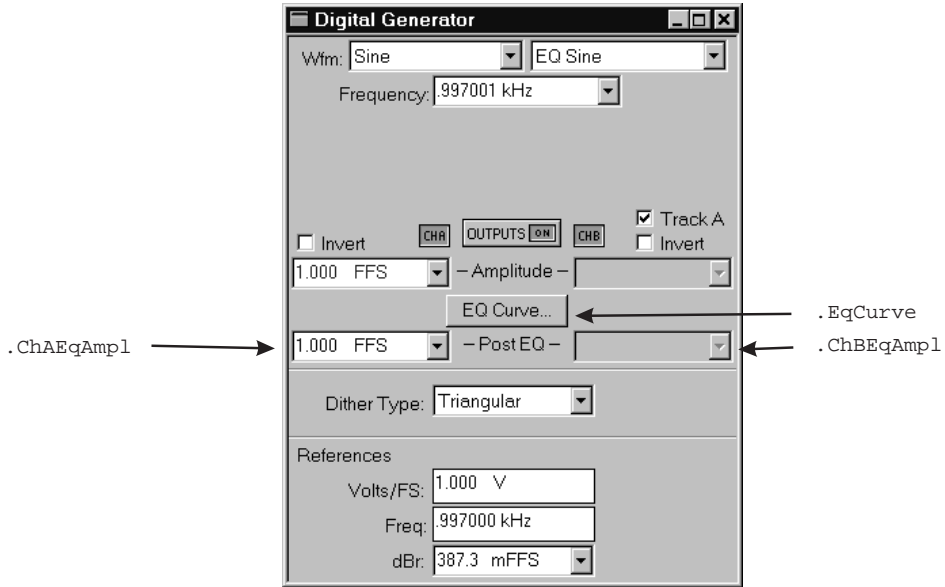


Digital Generator Continued ...

All commands on this page start with the following:

AP.Dgen

Example: AP.Dgen.ChAEqAmp1

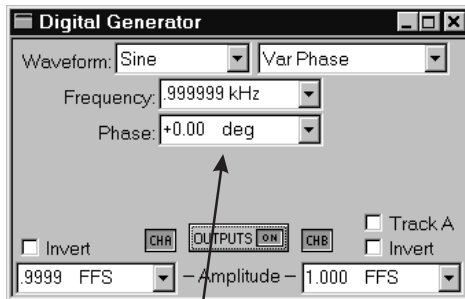


Digital Generator Continued ...

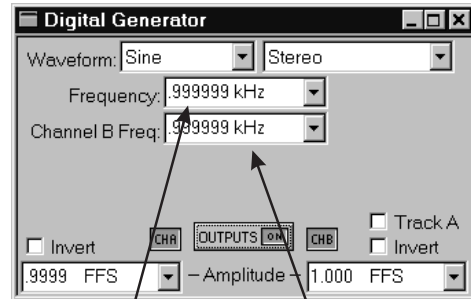
All commands on this page start with the following:

AP.Dgen

Example: AP.Dgen.phase

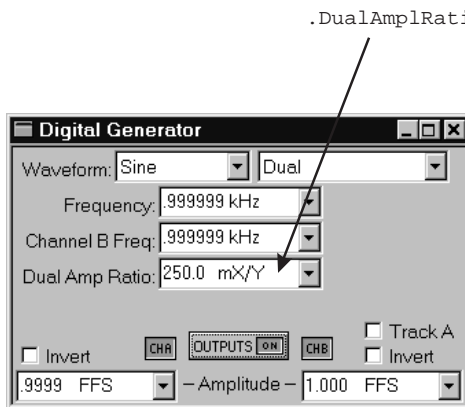


.Phase

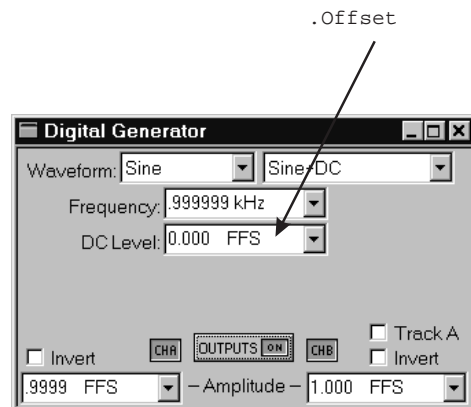


.ChAFreq

.ChBFreq



.DualAmplRatio



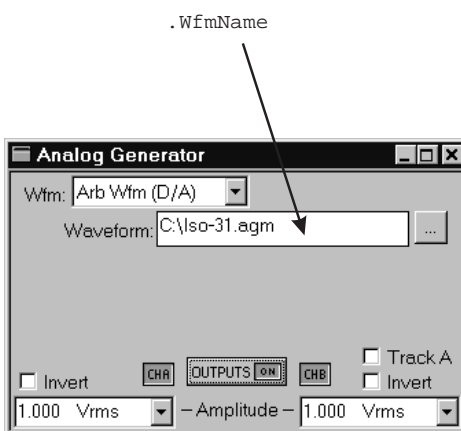
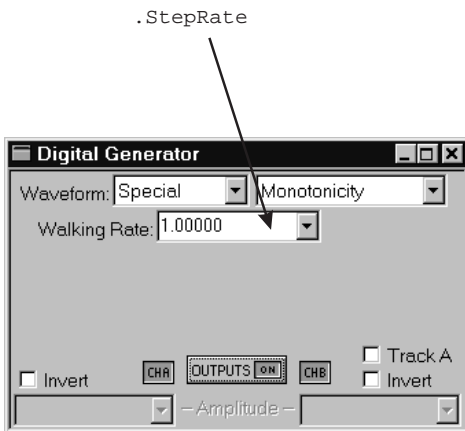
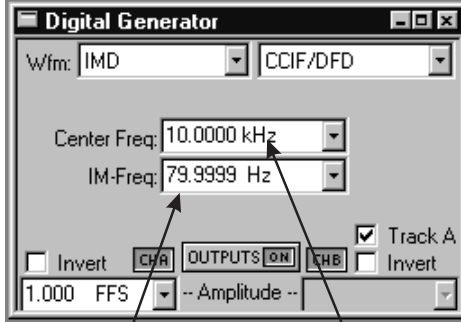
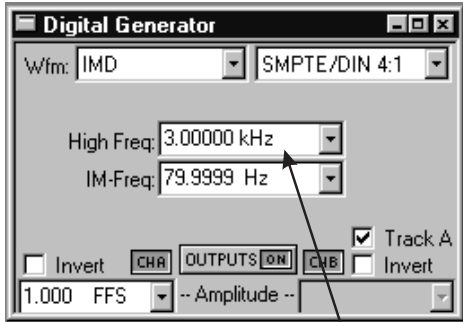
.Offset

Digital Generator Continued

All commands on this page start with the following:

AP.Dgen

Example: AP.Dgen.IMFreq

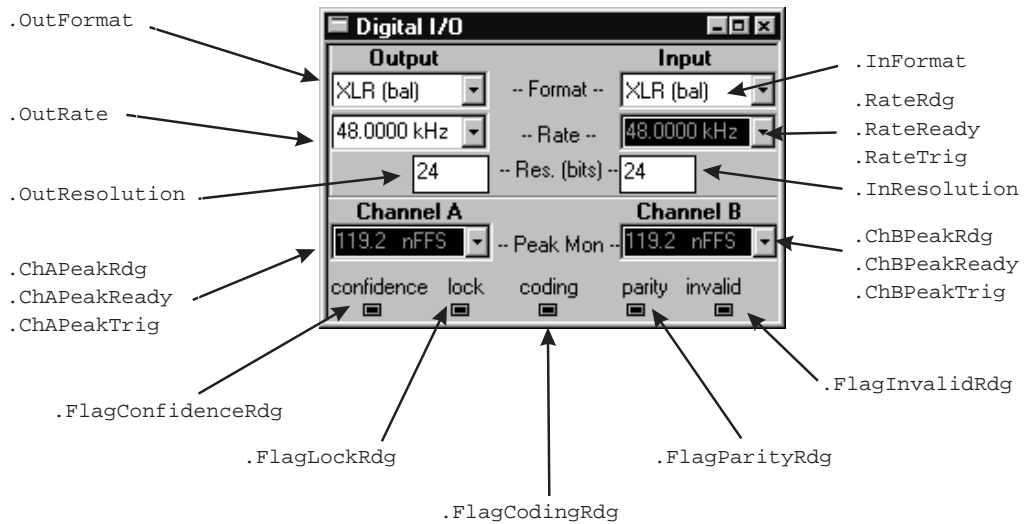


Digital IO Parameters - Input/Output small panel view...

All commands on this page start with the following:

AP.S2CDio

Example: AP.S2CDio.OutFormat

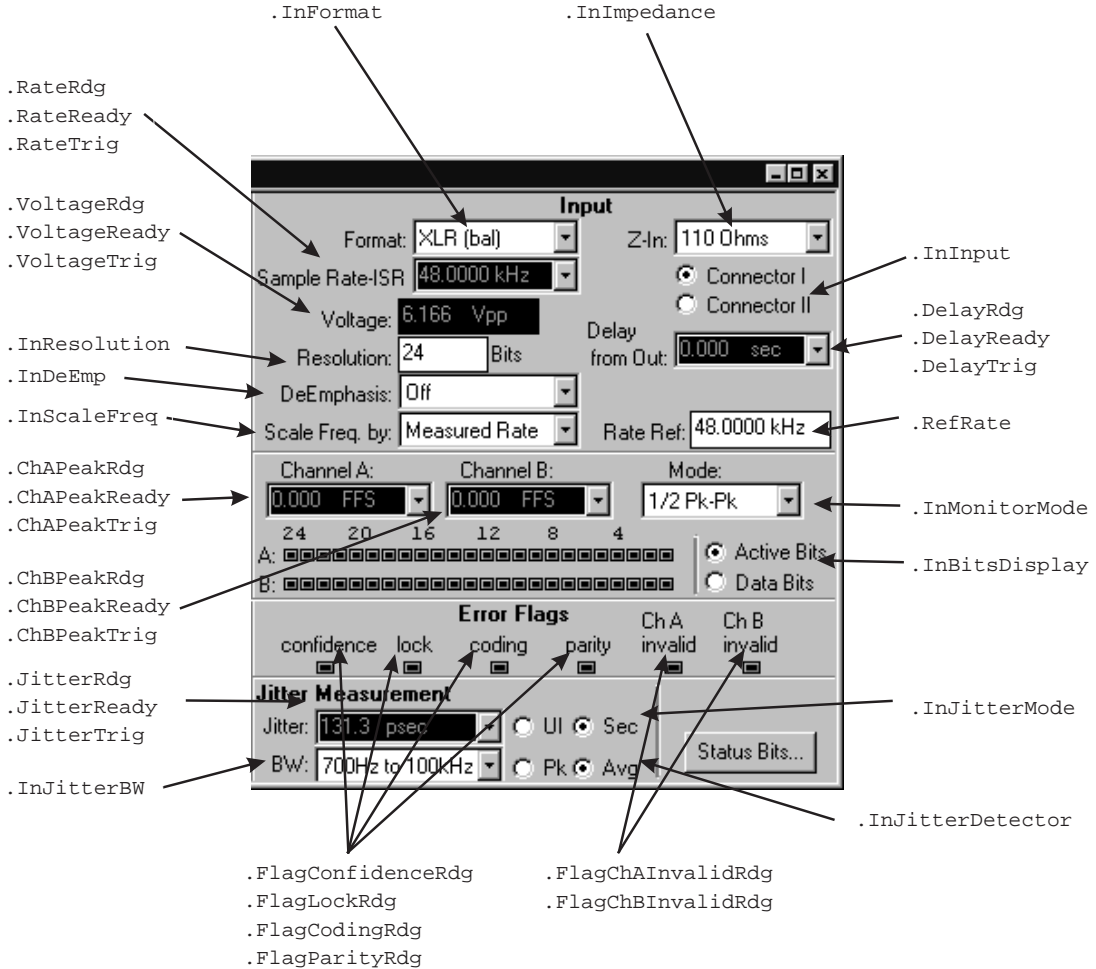


Digital IO Parameters - Input ...

All commands on this page start with the following:

AP.S2CDio

Example: AP.S2CDio.InFormat

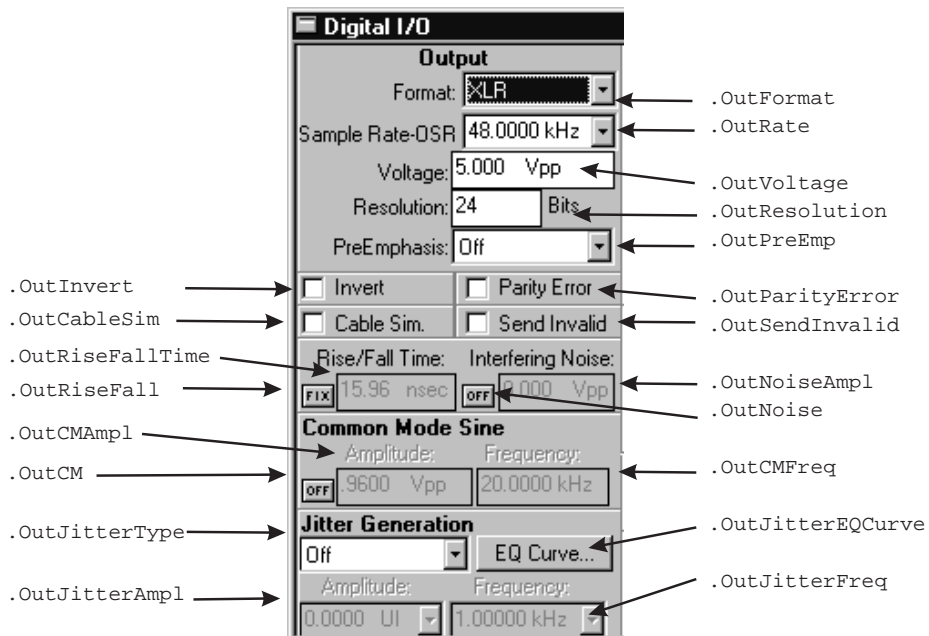


Digital IO Parameters - Output Continued

All commands on this page start with the following:

AP.S2CDio

Example: AP.S2CDio.OutFormat

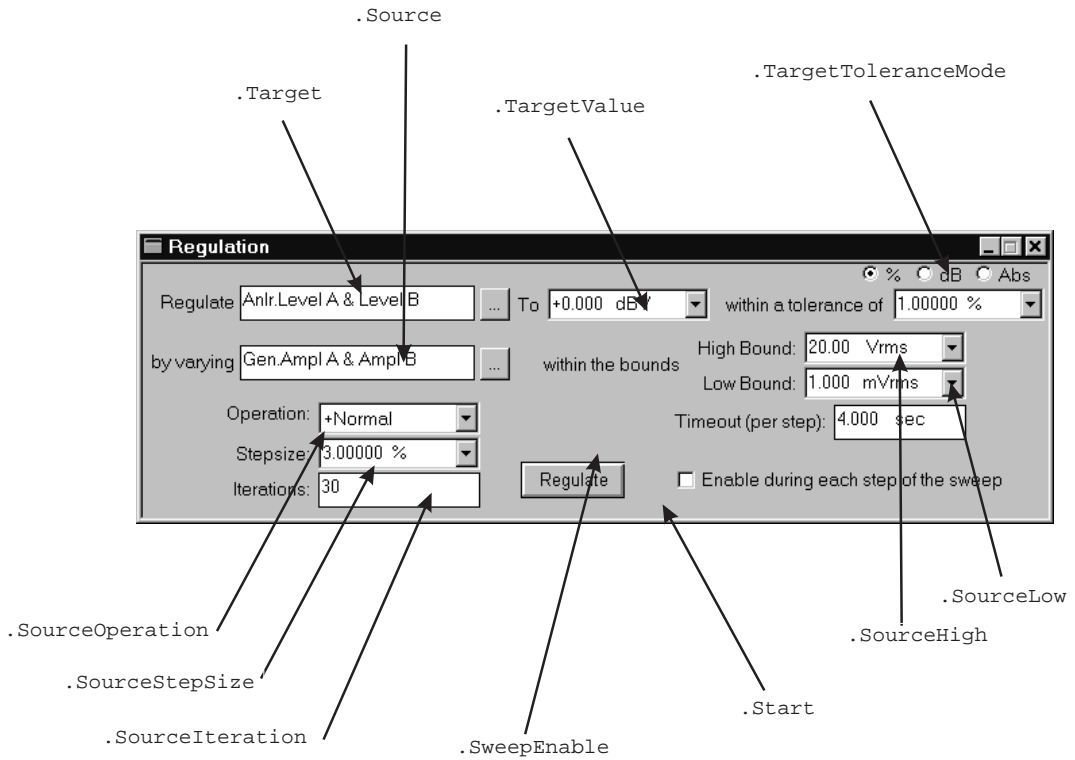


Regulation

All commands on this page start with the following:

AP.Reg

Example: AP.Sweep.Data1.Id

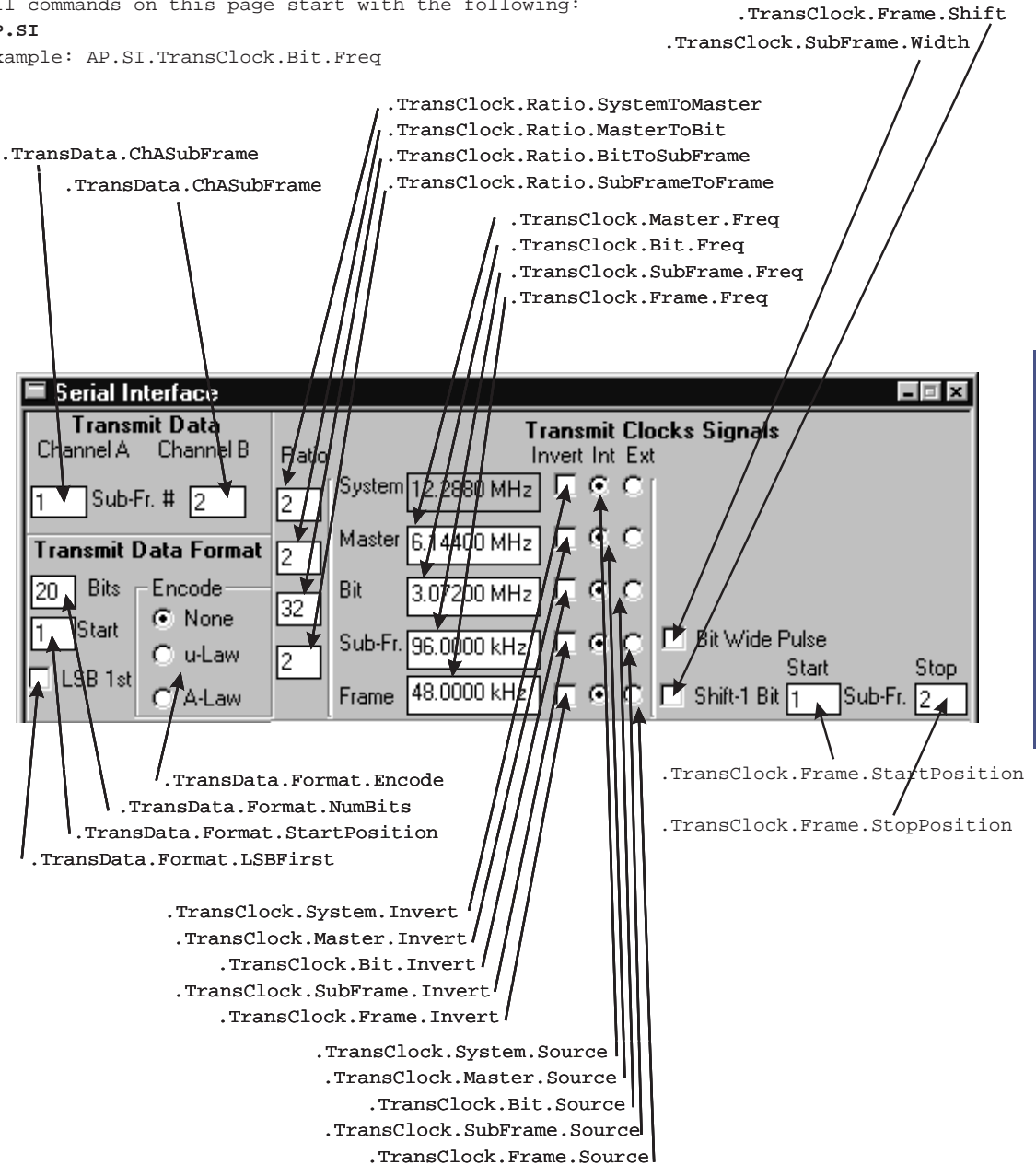


Serial Interface - Transmit ...

All commands on this page start with the following:

AP.SI

Example: AP.SI.TransClock.Bit.Freq



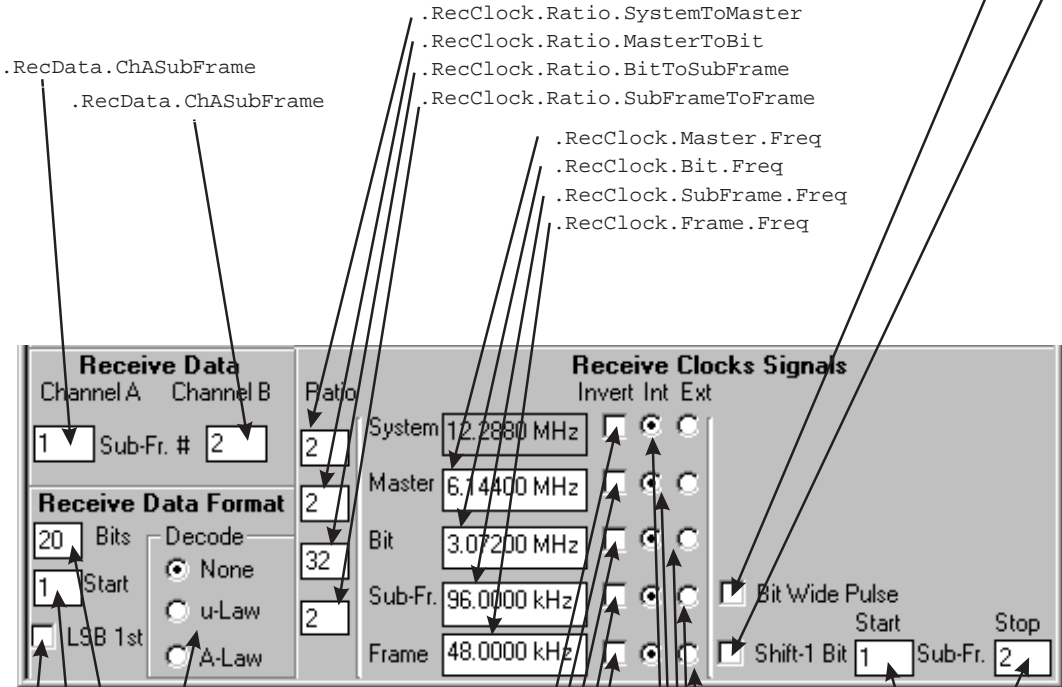
Serial Interface - Receive Continued

All commands on this page start with the following:

AP.SI

Example: AP.SI.RecClock.Bit.Freq

.RecClock.Frame.Shift
.RecClock.SubFrame.Width



.RecData.ChASubFrame
.RecData.ChASubFrame

.RecClock.Ratio.SystemToMaster
.RecClock.Ratio.MasterToBit
.RecClock.Ratio.BitToSubFrame
.RecClock.Ratio.SubFrameToFrame

.RecClock.Master.Freq
.RecClock.Bit.Freq
.RecClock.SubFrame.Freq
.RecClock.Frame.Freq

2 System Panels

.RecData.Format.Encode
.RecData.Format.NumBits
.RecData.Format.StartPosition
.RecData.Format.LSBFirst

.RecClock.System.Invert
.RecClock.Master.Invert
.RecClock.Bit.Invert
.RecClock.SubFrame.Invert
.RecClock.Frame.Invert

.RecClock.System.Source
.RecClock.Master.Source
.RecClock.Bit.Source
.RecClock.SubFrame.Source
.RecClock.Frame.Source

.RecClock.Frame.StartPosition
.RecClock.Frame.StopPosition

Settling

The screenshot shows a 'Settling' configuration window with several sections: Analog Analyzer, DCX, DIO, and DSP Audio Anlr. Each section contains a table of settings with columns for Tolerance, Floor, Points, Delay, and Algorithm. Arrows point from various property names to specific settings in the window.

Property Name	Setting	Tolerance	Floor	Points	Delay	Algorithm
AP.Anlr.FuncSettling	Amplitude	3.00000 %	100.0 nV	3	30.00 msec	Flat
AP.Anlr.ChALevelSettling	Level A:	1.00000 %	10.00 uV	3	30.00 msec	Flat
AP.Anlr.ChBLevelSettling	Level B:	1.00000 %	10.00 uV	3	30.00 msec	Flat
AP.Anlr.ChAFreqSettling	Frequency A:	0.50000 %	250.000 uHz	2	20.00 msec	Flat
AP.Anlr.ChBFreqSettling	Frequency B:	0.50000 %	250.000 uHz	2	20.00 msec	Flat
AP.Anlr.PhaseSettling	Phase:		+0.20 deg	2	20.00 msec	Flat
AP.DCX.DmmSettling	DCV	0.20000 %	+0.00050 Vdc	3	30.00 msec	Flat
AP.DCX.DigInSettling	Digital In:	0.00000 %	0 dec	3	30.00 msec	Flat
AP.S2CDio.RateSettling	Sample Rate:	0.50000 %	100.000 mHz	3	30.00 msec	Flat
AP.S2CDio.VoltageSettling	Voltage:	3.00000 %	10.00 mVpp	3	30.00 msec	Flat
AP.Sync.DelaySettling	Delay, In from Ref In:	0.01000 %	70.00 nsec	3	30.00 msec	Flat
AP.S2CDio.DelaySettling	Delay from Out:	0.01000 %	70.00 nsec	3	30.00 msec	Flat
AP.S2CDio.JitterSettling	Interface Jitter:	3.00000 %	3.000 nsec	3	100.0 msec	Exponential
AP.S2CDsp.Analyzer.ChBLevelSettling	DSP Audio Anlr.Ampl B	1.00000 %	1.000 uFFS	3	30.00 msec	Flat
AP.S2CDsp.Analyzer.ChBLevelSettling	DSP Audio Anlr.Level A	1.00000 %	1.000 uFFS	3	30.00 msec	Flat
AP.S2CDsp.Analyzer.ChALevelSettling	DSP Audio Anlr.Ampl A	1.00000 %	1.000 uFFS	3	30.00 msec	Flat
AP.S2CDsp.Analyzer.ChAFreqSettling	DSP Audio Anlr.Freq A	0.50000 %	10.0000 mHz	1	2.000 msec	Flat
AP.S2CDsp.Analyzer.ChBLevelSettling	DSP Audio Anlr.Level B	1.00000 %	1.000 uFFS	3	30.00 msec	Flat
AP.S2CDsp.Analyzer.ChBFreqSettling	DSP Audio Anlr.Freq B	0.50000 %	10.0000 mHz	1	2.000 msec	Flat

Speaker

All commands on this page start with the following:

AP.Speaker

Example: AP.Speaker.Source

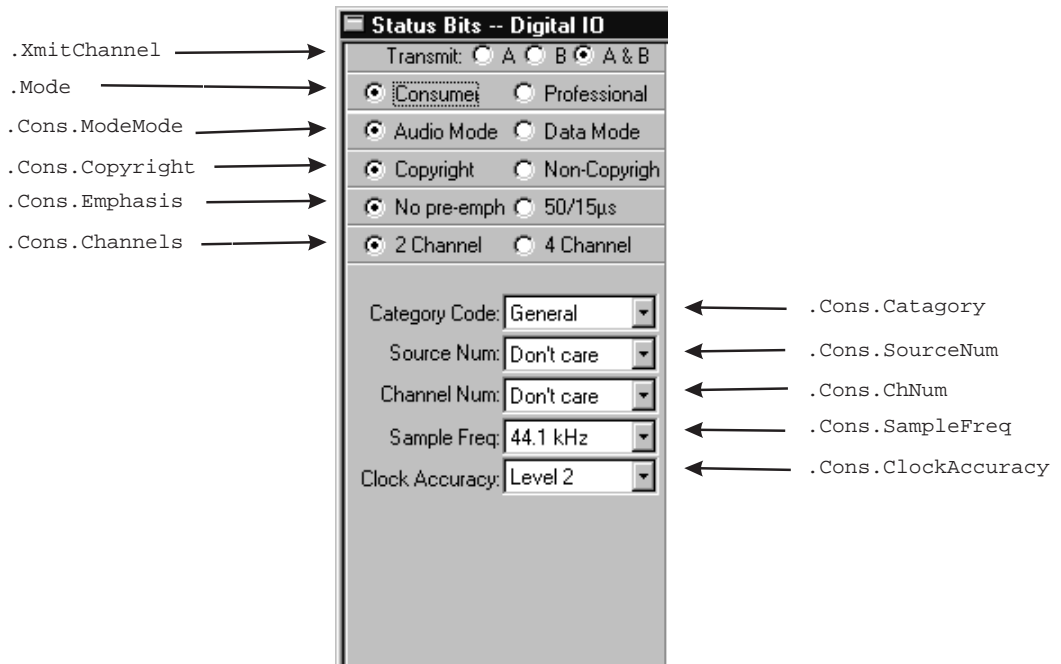


Status Bits — Digital IO - Transmit Consumer

All commands on this page start with the following:

AP.Bits

Example: AP.Bits.Mode

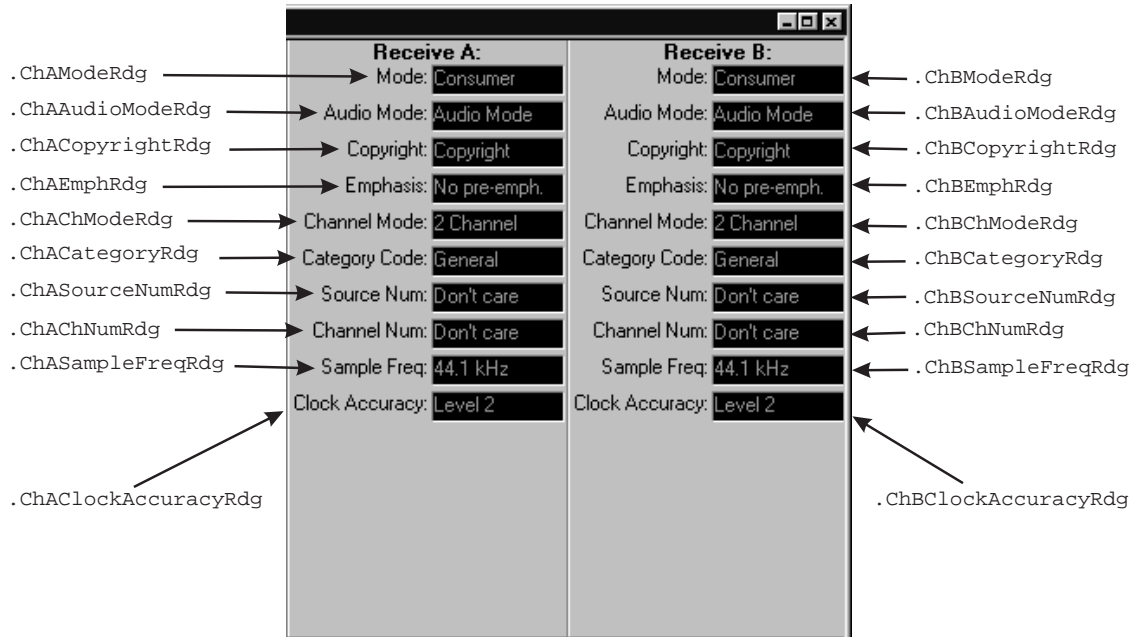


Status Bits — Digital IO - Receive Consumer

All commands on this page start with the following:

AP.Bits

Example: AP.Bits.

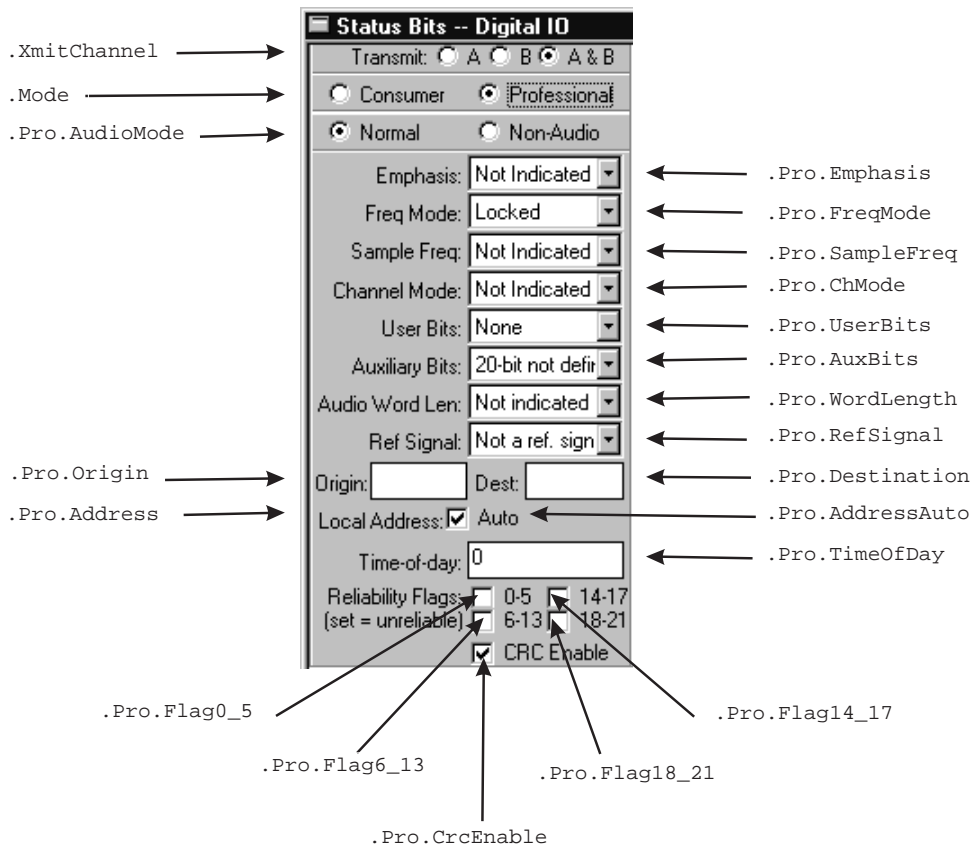


Status Bits — Digital IO - Transmit Professional

All commands on this page start with the following:

AP.Bits

Example: AP.Bits.XmitChannel

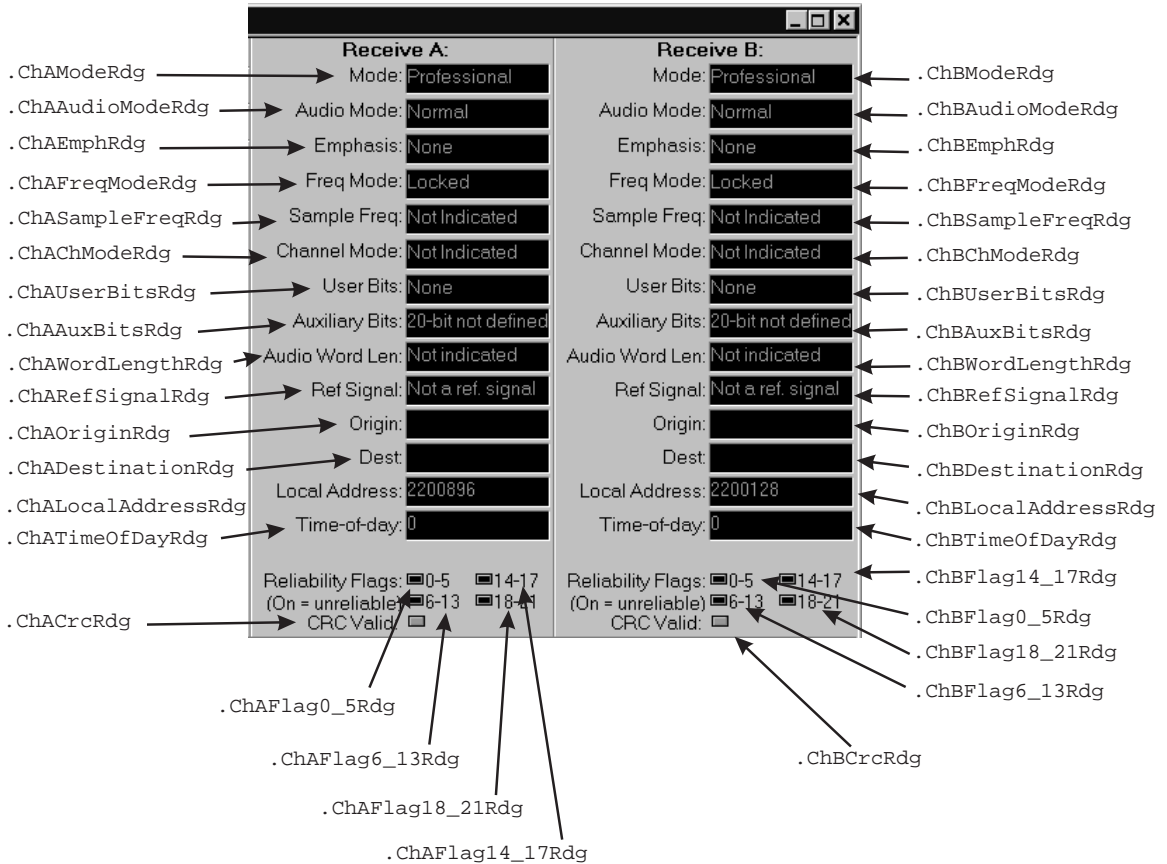


Status Bits — Digital IO - Receive Professional

All commands on this page start with the following:

AP.Bits

Example: AP.Bits.

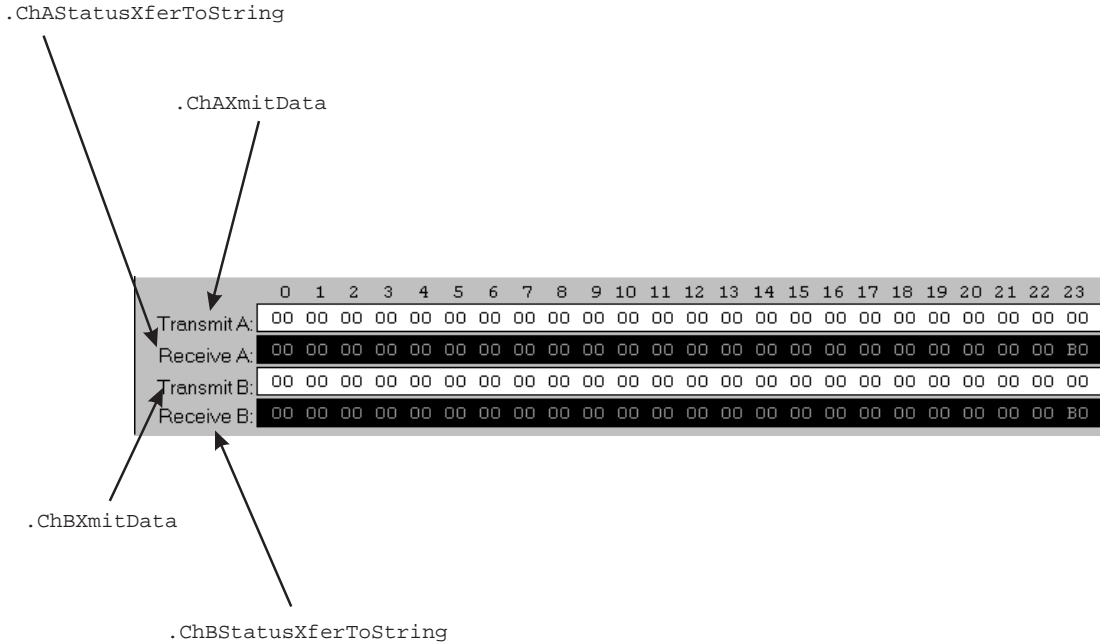


Status Bits — Digital IO - Receive Professional

All commands on this page start with the following:

AP.Bits

Example: AP.Bits.ChAStatusXferToArray

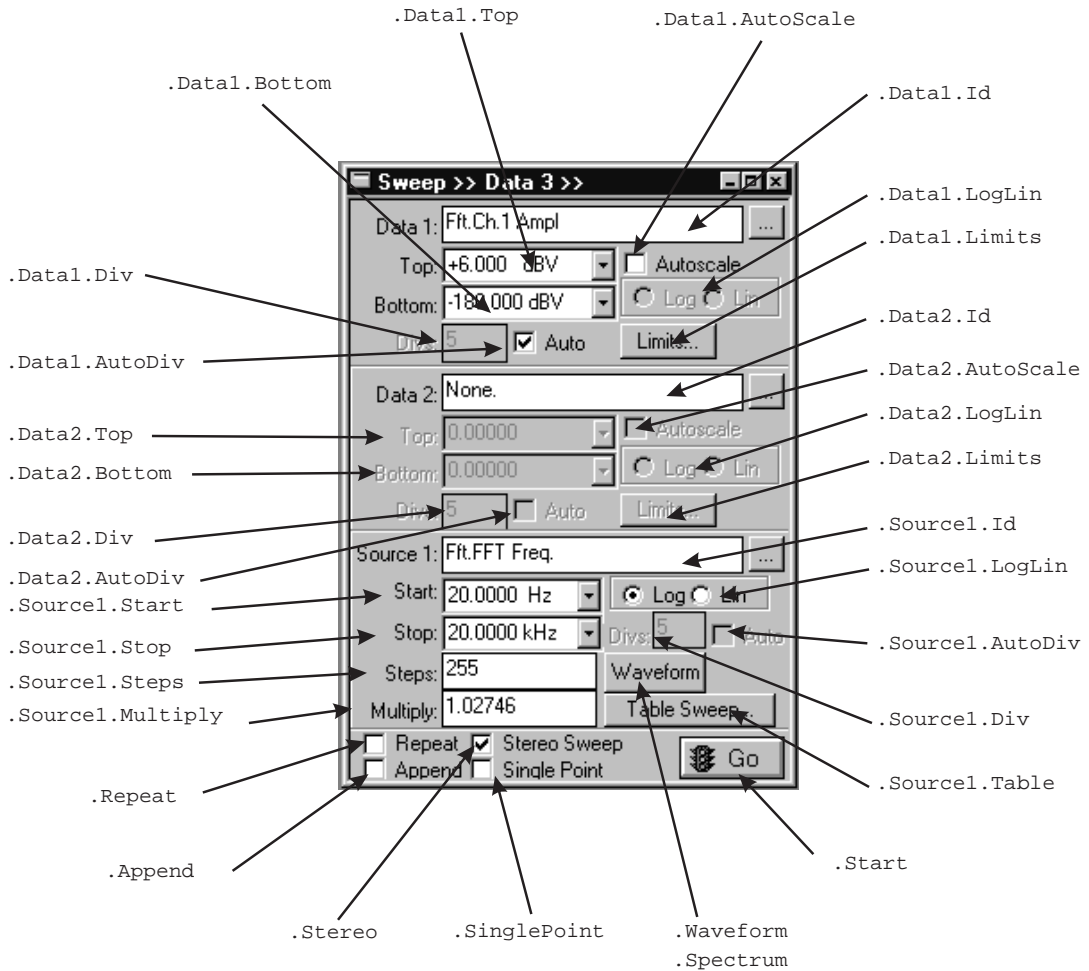


Sweep ...

All commands on this page start with the following:

AP.Sweep

Example: AP.Sweep.Data1.Id

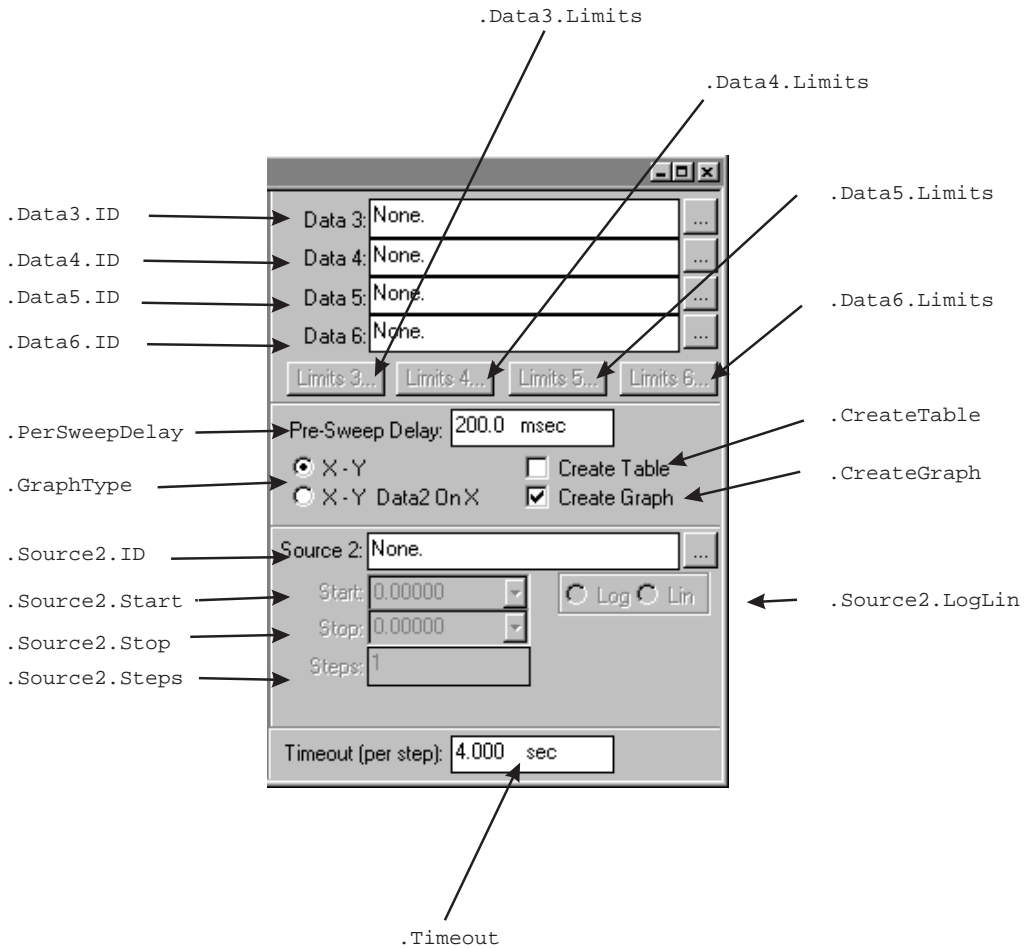


Sweep Continued ...

All commands on this page start with the following:

AP.Sweep

Example: AP.Sweep.Data1.Id



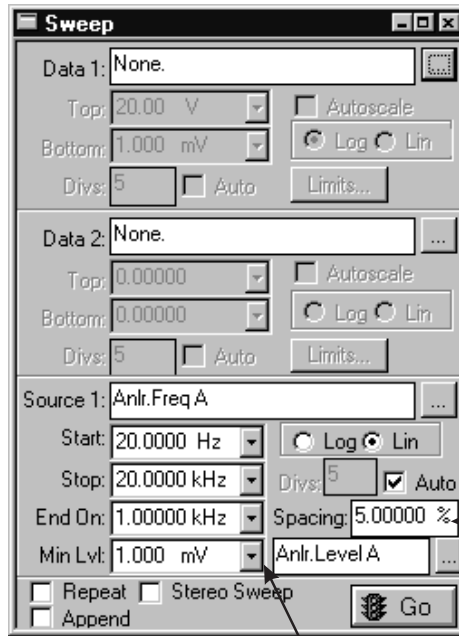
Sweep Continued ...

All commands on this page start with the following:

AP.Sweep

Example: AP.Sweep.Data1.Id

.Source1.EndOn →
.Source1.MinLevel →



.Source1.Spacing →

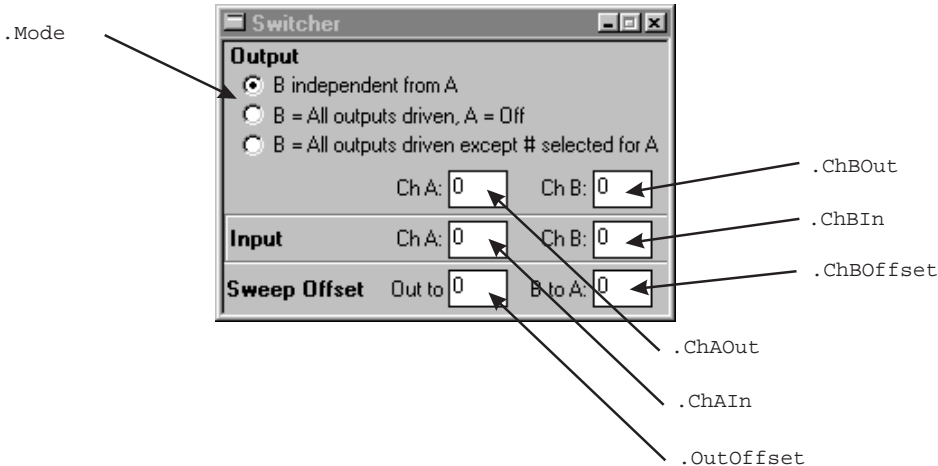
.Source1.MinLevelId →

Switcher

All commands on this page start with the following:

AP.SWR

Example: AP.SWR.Mode

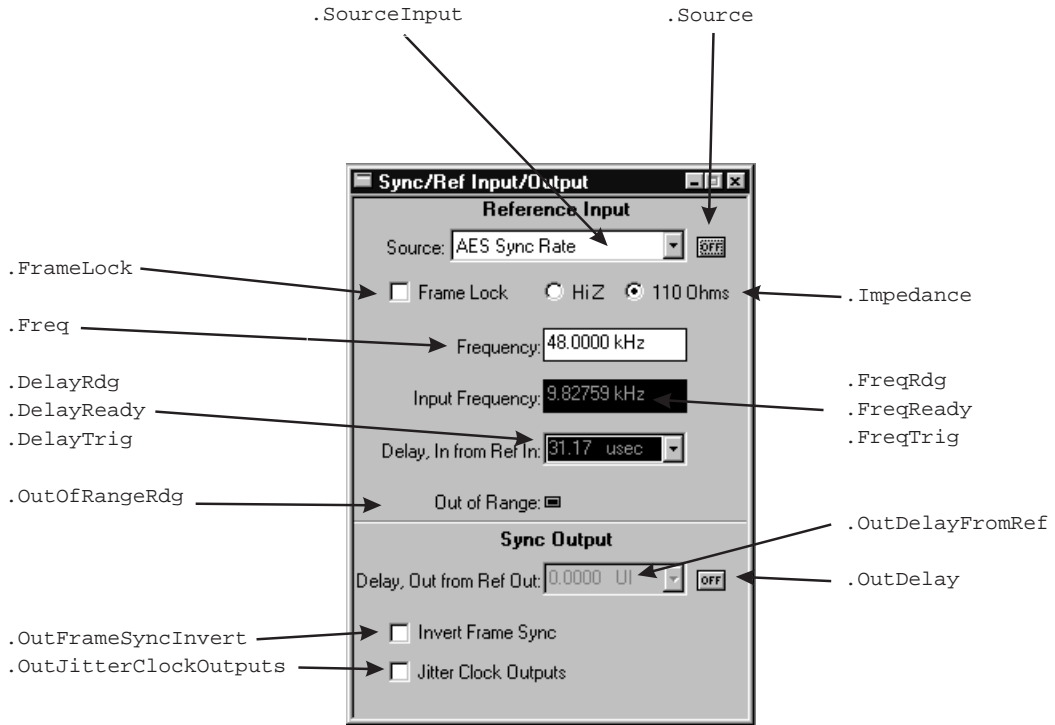


Sync

All commands on this page start with the following:

AP.Sync

Example: AP.Sync.Source



User Notes

User Notes

User Notes

Analog Analyzer

AP.Anlr.ChACoupling

Property

Syntax `AP.Anlr.ChACoupling`

Data Type Boolean

True DC coupled.
False Not DC coupled.

Description This command sets channel A Input Coupling to DC. This enables System Two to DC couple the input to the A/D converter for improved CMRR at low frequencies and increased low frequency measurement capability. By DC coupling the Analog Analyzer and DSP Audio Analyzer inputs DC Volts can be measured.

See Also `AP.Anlr.ChBCoupling`

Example

```
Sub Main
  AP.Application.NewTest          'Reset panels
  AP.Application.PanelClose(apbPanelAnlrSmall)
  AP.Application.PanelClose(apbPanelAnalogGenSmall)
  AP.Application.PanelOpen(apbPanelAnlrLarge)
  AP.S2CDsp.Program = 1 'Select DSP Audio Analyzer
  AP.Application.PanelOpen(apbDSPPanelLarge)
  AP.S2CDsp.Analyzer.InputFormat = 1 'Select A/D Input
  AP.Anlr.ChACoupling = True
  AP.Anlr.ChBCoupling = True
  AP.S2CDsp.Analyzer.ChACoupling = True
  AP.S2CDsp.Analyzer.ChBCoupling = True
  'Get readings.
  Wait 1
  Reading1 = AP.S2CDsp.Analyzer.ChALevelRdg("V")
  Reading2 = AP.S2CDsp.Analyzer.ChBLevelRdg("V")
  Debug.Print "Channel A DC Level = "; _
    Format(Reading1, "#.0000");" VDC"
  Debug.Print "Channel B DC Level = "; _
    Format(Reading2, "#.0000");" VDC"
End Sub
```

Example Output Channel A DC Level = 9.1081 VDC
Channel B DC Level = .0004 VDC

AP.Anlr.ChAFreqRdg

Property

Syntax `AP.Anlr.ChAFreqRdg(ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
<i>Unit</i>	The following units are available: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM.

Description This command returns a settled reading for the channel A Frequency meter and zeros the ready count.

See Also `AP.Anlr.ChAFreqReady`, `AP.Anlr.ChAFreqSettling`, `AP.Anlr.ChAFreqTrig`

Example

```
Sub Main
    AP.Application.NewTest          'Reset panels
    AP.Gen.Output = 1
    AP.Anlr.ChAInput = 1
    AP.Anlr.ChAFreqSettling(.5, .0002, "Hz", 3, .03, 1)
    AP.Anlr.ChAFreqTrig          'Trigger new reading.
    Do
        Ready = AP.Anlr.ChAFreqReady 'Get status.
    Loop Until Ready > 0
    Reading1 = AP.Anlr.ChAFreqRdg("Hz") 'Get reading.
    Debug.Print "Frequency A = ";Format(Reading1, _
        "#.0000");" Hz"
End Sub
```

Example Output Frequency A = 1002.9112 Hz

AP.Anlr.ChAFreqReady

Property

Syntax `AP.Anlr.ChAFreqReady`

Data Type Integer

0 Reading not ready.
 >0 Reading ready.

Description This command returns the Frequency A settled reading ready count. Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.Anlr.ChAFreqRdg` or `AP.Anlr.ChAFreqTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.Anlr.ChAFreqRdg` command will be guaranteed to return quickly.

See Also `AP.Anlr.FuncInput`, `AP.Anlr.ChAFreqRdg`,
`AP.Anlr.ChAFreqSettling`, `AP.Anlr.ChAFreqTrig`

Example See example for `AP.Anlr.ChAFreqRdg`.

AP.Anlr.ChAFreqSettling

Method

Syntax `AP.Anlr.ChAFreqSettling(ByVal Tolerance As Double, ByVal Floor As Double, ByVal FloorUnit As String, ByVal Points As Integer, ByVal Delay As Double, ByVal Algorithm As Integer)`

Parameters See Appendix A for Settling Algorithm and parameter name descriptions.

Description This command sets the settling parameters for the `AP.Anlr.ChAFreqRdg` command.

See Also `AP.Anlr.FuncInput`, `AP.Anlr.ChAFreqRdg`,
`AP.Anlr.ChAFreqReady`, `AP.Anlr.ChAFreqTrig`

Example See example for `AP.Anlr.ChAFreqRdg`.

AP.Anlr.ChAFreqTrig

Method

Syntax	<code>AP.Anlr.ChAFreqTrig</code>
Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP.Anlr.ChAFreqRdg</code> command. The reading in progress is aborted.
See Also	<code>AP.Anlr.FuncInput</code> , <code>AP.Anlr.ChAFreqRdg</code> , <code>AP.Anlr.ChAFreqReady</code> , <code>AP.Anlr.ChAFreqSettling</code>
Example	See example for <code>AP.Anlr.ChAFreqRdg</code> .

AP.Anlr.ChAImpedance

Property

Syntax	<code>AP.Anlr.ChAImpedance</code>						
Data Type	Integer						
	<table> <tr> <td><code>0</code></td> <td>300 ohms</td> </tr> <tr> <td><code>1</code></td> <td>600 ohms</td> </tr> <tr> <td><code>2</code></td> <td>100k ohms</td> </tr> </table>	<code>0</code>	300 ohms	<code>1</code>	600 ohms	<code>2</code>	100k ohms
<code>0</code>	300 ohms						
<code>1</code>	600 ohms						
<code>2</code>	100k ohms						
Description	This command selects one of the available termination impedances for the Analyzer channel A input.						
See Also	<code>AP.Anlr.ChBImpedance</code>						
Example	<pre> Sub Main AP.Application.NewTest 'Reset panels AP.Gen.Impedance = 2 'Set generator output Z to _ 600 ohms. AP.Gen.ChAAmpl("dBm") = 0 AP.Gen.Output = 1 AP.Anlr.ChARangeAuto = 0 'Set input ranging to fixed. AP.Anlr.ChARange("V") = 2.5 'Set input range to _ 2.5 Volts. AP.Anlr.ChAInput = 0 'Set anlr input to INPUT(XLR). AP.Anlr.ChAImpedance = 1 'Set cha A input Z to _ 600 ohms. AP.Anlr.FuncSettling(1, .000002, "V", 4, .05, 1) AP.Anlr.FuncTrig 'Trigger new reading. </pre>						

```

Do
    Ready = AP.Anlr.FuncReady 'Get status.
Loop Until Ready > 0
Reading1 = AP.Anlr.FuncRdg("dBm") 'Get settled _
    reading.
Debug.Print "Channel A Amplitude =";Format _
    (Reading1, "#.0000");" dBm"
Anlr.ChARangeAuto = 1 'Set input ranging to auto.
End Sub

```

Example Output Channel A Amplitude = 0.0079 dBm

AP.Anlr.ChAInput

Property

Syntax	<code>AP.Anlr.ChAInput</code>						
Data Type	Integer						
	<table> <tr> <td>0</td> <td>XLR-Bal</td> </tr> <tr> <td>1</td> <td>BNC-Unbal</td> </tr> <tr> <td>2</td> <td>GenMon</td> </tr> </table>	0	XLR-Bal	1	BNC-Unbal	2	GenMon
0	XLR-Bal						
1	BNC-Unbal						
2	GenMon						
Description	This command selects the Analog Analyzer channel A Input.						
See Also	<code>AP.Anlr.ChBInput</code>						
Example	See example for <code>AP.Anlr.ChAImpedance</code> .						

AP.Anlr.ChALevelRdg

Property

Syntax	<code>AP.Anlr.ChALevelRdg(ByVal Unit As String)</code>	
Data Type	Double	
Parameters	Name	Description
	<i>Unit</i>	The following units are available : V, dBu, dBV, dBr A, dBr B, dBg A, dBg B, dBm, W.

Description This command returns the channel A level meter settled reading. For System One the `AP.Anlr.FuncInput` command must be set to channel A.

See Also `AP.Anlr.FuncMode`, `AP.Anlr.FuncInput`

Example

```
Sub Main
  AP.Application.NewTest
  AP.Gen.Output = 1
  AP.Anlr.ChAInput = 1
  AP.Anlr.ChALevelSettling(1, .000025, "V", 3, .03, 1)
  AP.Anlr.ChALevelTrig          'Trigger new reading.
Do
  Ready = AP.Anlr.ChALevelReady    'Get status.
  Loop Until Ready > 0
  Reading1 = AP.Anlr.ChALevelRdg("V") 'Get settled _
  reading.
  Debug.Print "Level A Amplitude = ";Format(Reading1, _
    "#.0000");" V"
End Sub
```

Example Output Level A Amplitude = 0.9957 V

AP.Anlr.ChALevelReady

Property

Syntax `AP.Anlr.ChALevelReady`

Data Type Integer

0 Reading not ready.
>0 Reading ready.

Description This command returns the Level A settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.Anlr.ChALevelRdg` or `AP.Anlr.ChALevelTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.Anlr.ChALevelRdg` command will be guaranteed to return quickly.

See Also `AP.Anlr.FuncInput`, `AP.Anlr.ChALevelRdg`,
`AP.Anlr.ChALevelSettling`, `AP.Anlr.ChALevelTrig`

Example See example for `AP.Anlr.ChALevelRdg`.

AP.Anlr.ChALevelSettling

Method

Syntax `AP.Anlr.ChALevelSettling(ByVal Tolerance As Double, ByVal Floor As Double, ByVal FloorUnit As String, ByVal Points As Integer, ByVal Delay As Double, ByVal Algorithm As Integer)`

Parameters See Appendix A for Settling Algorithm and parameter name descriptions.

Description This command sets the settling parameters for the `AP.Anlr.ChALevelRdg` command.

See Also `AP.Anlr.FuncMode`, `AP.Anlr.ChALevelRdg`,
`AP.Anlr.ChALevelReady`, `AP.Anlr.ChALevelTrig`

Example See example for `AP.Anlr.ChALevelRdg`.

AP.Anlr.ChALevelTrig

Method

Syntax `AP.Anlr.ChALevelTrig`

Description Causes a restart of the reading cycle and zeros the ready count for the `AP.Anlr.ChALevelRdg` command. The reading in progress is aborted.

See Also `AP.Anlr.ChALevelRdg`, `AP.Anlr.ChALevelSettling`,
`AP.Anlr.ChALevelTrig`

Example See example for `AP.Anlr.ChALevelRdg`.

AP.Anlr.ChARange

Property

Syntax	AP.Anlr.ChARange (ByVal <i>Unit</i> As String)	
Data Type	Double	The following values are the range boundaries for the Volts unit: 160, 80, 40, 20, 10, 5, 2.5, 1.2, .600, .300, .160, .08, 04 If an arbitrary value between the range boundaries is entered the next higher range will be selected.
Parameters	Name	Description
	<i>Unit</i>	The following units are available: V, dBu, dBV
Description	This command sets the <code>AP.Anlr.ChARange</code> and returns the nominal full scale of the range in use.	
See Also	<code>AP.Anlr.ChARangeAuto</code>	
Example	See example for <code>AP.Anlr.ChAImpedance</code> .	

AP.Anlr.ChARangeAuto

Property

Syntax	AP.Anlr.ChARangeAuto	
Data Type	Boolean	
	<i>True</i>	Auto range
	<i>False</i>	Fixed range
Description	This command sets the Analyzer channel A input to Auto range or Fixed range. Care must be taken when using Fixed range that the input signal does not exceed the selected range.	
See Also	<code>AP.Anlr.ChARange</code>	
Example	See example for <code>AP.Anlr.ChAImpedance</code> .	

AP.Anlr.ChBCoupling

Property

Syntax	AP.Anlr.ChBCoupling
Data Type	Boolean

True DC coupled.
False Not DC coupled.

Description This command sets channel B Input Coupling to DC. This enables System Two to DC couple the input to the A/D converter for improved CMRR at low frequencies and increased low frequency measurement capability. By DC coupling the Analog Analyzer and DSP Audio Analyzer inputs DC Volts can be measured.

See Also AP.Anlr.ChACoupling

Example See example for AP.Anlr.ChACoupling.

AP.Anlr.ChBFreqRdg

Property

Syntax `AP.Anlr.ChBFreqRdg(ByVal Unit As String)`

Data Type Double

Description This command returns a settled reading for the channel B Frequency meter and zeros the ready count.

Parameters

Part	Description
<i>Unit</i>	The following units are available: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM.

See Also AP.Anlr.ChAFreqReady, AP.Anlr.FreqATrig, AP.Anlr.ChAFreqSettling

Example

```
Sub Main
  AP.Application.NewTest      'Reset panels
  AP.Gen.Output = 1
  AP.Anlr.ChBInput = 1
  AP.Anlr.FuncInput = 1
  AP.Anlr.ChBFreqSettling(.5, .0002, "Hz", 3, .03, 1)
  AP.Anlr.ChBFreqTrig      'Trigger new reading.
  Do
    Ready = AP.Anlr.ChBFreqReady 'Get status.
  Loop Until Ready > 0
```

```

    Reading1 = AP.Anlr.ChBFreqRdg("Hz") 'Get settled _
        reading.
    Debug.Print "Frequency B = ";Format(Reading1, _
        "#.0000");" Hz"
End Sub

```

Example Output Frequency B = 999.6856 Hz

AP.Anlr.ChBFreqReady

Property

Syntax AP.Anlr.ChBFreqReady

Data Type Integer

0 Reading not ready.
>0 Reading ready.

Description This command returns the Frequency B settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the AP.Anlr.ChBFreqRdg or AP.Anlr.FreqBTrig commands will zero the ready count.

If the reading is found to be ready, a call to the AP.Anlr.ChBFreqRdg command will be guaranteed to return quickly.

See Also AP.Anlr.FuncInput, AP.Anlr.ChBFreqRdg, AP.Anlr.ChBFreqSettling, AP.Anlr.ChBFreqTrig

Example See example for AP.Anlr.ChBFreqRdg.

AP.Anlr.ChBFreqSettling

Method

Syntax AP.Anlr.ChBFreqSettling(*ByVal Tolerance As Double, ByVal Floor As Double, ByVal FloorUnit As String, ByVal Points As Integer, ByVal Delay As Double, ByVal Algorithm As Integer*)

Parameters	See Appendix A for Settling Algorithm and parameter name descriptions.
Description	This command sets the settling parameters for the AP.Anlr.ChBFreqRdg command.
See Also	AP.Anlr.FuncInput, AP.Anlr.ChBFreqRdg, AP.Anlr.FreqBTrig, AP.Anlr.FreqBReady
Example	See example for AP.Anlr.ChBFreqRdg.

AP.Anlr.ChBFreqTrig

Method

Syntax	AP.Anlr.ChBFreqTrig
Description	Causes a restart of the reading cycle and zeros the ready count for the AP.Anlr.ChBFreqRdg command. The reading in progress is aborted.
See Also	AP.Anlr.FuncInput, AP.Anlr.ChBFreqRdg, AP.Anlr.ChBFreqReady, AP.Anlr.ChBFreqSettling
Example	See example for AP.Anlr.ChBFreqRdg.

AP.Anlr.ChBImpedance

Property

Syntax	AP.Anlr.ChBImpedance						
Data Type	Integer						
	<table> <tr> <td>0</td> <td>300 ohms</td> </tr> <tr> <td>1</td> <td>600 ohms</td> </tr> <tr> <td>2</td> <td>100k ohms</td> </tr> </table>	0	300 ohms	1	600 ohms	2	100k ohms
0	300 ohms						
1	600 ohms						
2	100k ohms						
Description	This command selects one of the available termination impedances for the Analyzer channel B Input.						
See Also	AP.Anlr.ChAImpedance						
Example	Sub Main						

```

AP.Application.NewTest 'Reset panels
AP.Gen.Impedance = 2 'Set generator output Z to _
    600 ohms.
AP.Gen.ChAAmpl("dBm") = 0
AP.Gen.Output = 1
AP.Anlr.ChBRangeAuto = 0 'Set input ranging to fixed.
AP.Anlr.ChBRange("V") = 2.5 'Set input range to _
    2.5 Volts.
AP.Anlr.ChBInput = 0 'Set anlr input to INPUT(XLR).
AP.Anlr.ChBImpedance = 1 'Set Cha A input Z to _
    600 ohms.
AP.Anlr.FuncInput = 1 'Set Function Meter Cha to B.
AP.Anlr.FuncSettling(1, .000002, "V", 4, .05, 1)
AP.Anlr.FuncTrig 'Trigger new reading.
Do
    Ready = AP.Anlr.FuncReady 'Get status.
Loop Until Ready > 0
Reading1 = AP.Anlr.FuncRdg("dBm")'Get settled _
    reading.
Debug.Print "Channel B Amplitude = ";Format _
    (Reading1, "#.0000");" dBm"
Anlr.ChBRangeAuto = 1 'Set input ranging to auto.
End Sub

```

Example Output Channel B Amplitude = -103.7187 dBm

AP.Anlr.ChBInput

Property

Syntax	AP.Anlr.ChBInput
Data Type	Integer
	0 XLR-Bal
	1 BNC-Unbal
	2 GenMon
Description	This command selects the Analog Analyzer channel B Input.
See Also	AP.Anlr.ChAInput
Example	See example for AP.Anlr.ChBImpedance.

AP.Anlr.ChBLevelRdg

Property**Syntax** `AP.Anlr.ChBLevelRdg(ByVal Unit As String)`**Data Type** Double**Parameters**

Name	Description
<i>Unit</i>	The following units are available for System Two: V, dBu, dBV, dBr A, dBr B, dBg A, dBg B, dBm, W.

Description

This command returns the channel B Level meter settled reading.

See Also

AP.Anlr.FuncMode, AP.Anlr.FuncInput,
 AP.Anlr.ChBLevelReady, AP.Anlr.ChBLevelSettling,
 AP.Anlr.ChBLevelTrig

Example

```
Sub Main
  AP.Application.NewTest
  AP.Gen.Output = 1
  AP.Anlr.ChBInput = 1
  AP.Anlr.ChBLevelSettling(1, .000025, "V", 3, .03, 1)
  AP.Anlr.ChBLevelTrig          'Trigger new reading.
  Do
    Ready = AP.Anlr.ChBLevelReady    'Get status.
    Loop Until Ready > 0
    Reading1 = AP.Anlr.ChBLevelRdg("V") 'Get settled _
    reading.
    Debug.Print "Level B Amplitude = ";Format _
      (Reading1, "#.0000");" V"
  End Sub
```

Example Output Level B Amplitude = 0.9973 V

AP.Anlr.ChBLevelReady

Property**Syntax** `AP.Anlr.ChBLevelReady`**Data Type** Integer

0	Reading not ready.
>0	Reading ready.

Description	This command returns the Level B settled reading ready count. Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the <code>AP.Anlr.ChBLevelRdg</code> or <code>AP.Anlr.ChBLevelTrig</code> commands will zero the ready count. If the reading is found to be ready, a call to the <code>AP.Anlr.ChBLevelRdg</code> command will be guaranteed to return quickly.
See Also	<code>AP.Anlr.FuncInput</code>
Example	See example for <code>AP.Anlr.ChBLevelRdg</code> .

AP.Anlr.ChBLevelSettling

Method

Syntax	<code>AP.Anlr.ChBLevelSettling</code> (ByVal <i>Tolerance</i> As Double, ByVal <i>Floor</i> As Double, ByVal <i>FloorUnit</i> As String, ByVal <i>Points</i> As Integer, ByVal <i>Delay</i> As Double, ByVal <i>Algorithm</i> As Integer)
Parameters	See Appendix A for Settling Algorithm and parameter name descriptions.
Description	This command sets the settling parameters for the <code>AP.Anlr.ChBLevelRdg</code> command.
See Also	<code>AP.Anlr.FuncMode</code> , <code>AP.Anlr.ChBLevelRdg</code> , <code>AP.Anlr.ChBLevelReady</code> , <code>AP.Anlr.ChBLevelTrig</code>
Example	See example for <code>AP.Anlr.ChBLevelRdg</code> .

AP.Anlr.ChBLevelTrig

Method

Syntax	<code>AP.Anlr.ChBLevelTrig</code>
---------------	-----------------------------------

Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP.Anlr.ChBLevelRdg</code> command. The reading in progress is aborted.
See Also	<code>AP.Anlr.FuncInput</code>
Example	See example for <code>AP.Anlr.ChBLevelRdg</code> .

AP.Anlr.ChBRange

Property

Syntax	<code>AP.Anlr.ChBRange(ByVal Unit As String)</code>					
Data Type	Double	The following values are the range boundaries for the Volts unit: 160, 80, 40, 20, 10, 5, 2.5, 1.2, .600, .300, .160, .08, .04 If an arbitrary value between the range boundaries is entered the next higher range will be selected.				
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>Unit</i></td> <td>The following units are available: V, dBu, dBV</td> </tr> </tbody> </table>	Name	Description	<i>Unit</i>	The following units are available: V, dBu, dBV	
Name	Description					
<i>Unit</i>	The following units are available: V, dBu, dBV					
Description	This command sets the <code>AP.Anlr.ChBRange</code> and returns the nominal full scale of the range in use.					
See Also	<code>AP.Anlr.ChBRangeAuto</code>					
Example	See example for <code>AP.Anlr.ChBImpedance</code> .					

AP.Anlr.ChBRangeAuto

Property

Syntax	<code>AP.Anlr.ChBRangeAuto</code>	
Data Type	Boolean	
	<i>True</i>	Auto range
	<i>False</i>	Fixed range
Description	This command sets the Analyzer channel B input to Auto range or Fixed range. Care must be taken when using Fixed range that the input signal does not exceed the selected range.	
See Also	<code>AP.Anlr.ChBRange</code>	

Example See example for AP.Anlr.ChBImpedance.

AP.Anlr.FuncBPBRFreq

Property

Syntax `AP.Anlr.FuncBPBRFreq(ByVal Unit As String)`

Data Type Double Any frequency value between 10 Hz to 204 kHz.

Parameters	Name	Description
	<i>Unit</i>	The following units are available: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM.

Description This command sets the bandpass/bandreject filter frequency.

See Also AP.Anlr.FuncBPBRTuning, AP.Anlr.FuncMode

Example

```

Sub Main
    AP.Application.NewTest          'Reset panels
    AP.Gen.Output = 1
    Frequency = AP.Gen.Freq("Hz")
    AP.Anlr.FuncMode = 1          'Set Anlr mode to Bandpass
    AP.Anlr.ChAInput = 2
    AP.Anlr.FuncBPBRTuning = 4 'Set Tuning to Fixed.
    For Harmonic = 2 To 5 Step 1
        AP.Anlr.FuncBPBRFreq("Hz") = Frequency * Harmonic
        AP.Anlr.FuncSettling(1, .000002, "V", 4, .05, 1)
        AP.Anlr.FuncTrig
        Do
            Ready = AP.Anlr.FuncReady
        Loop Until Ready > 0
        Reading1 = AP.Anlr.FuncRdg("dBV")
        Debug.Print "Harmonic Amplitude at ";Frequency * _
            Harmonic;" Hz = ";Format(Reading1, "#.0000");" _
            dBV"
        Next Harmonic
    End Sub

```

Example Output

```

Harmonic Amplitude at 2000 Hz = -33.0766 dBV
Harmonic Amplitude at 3000 Hz = -43.0576 dBV
Harmonic Amplitude at 4000 Hz = -48.9417 dBV
Harmonic Amplitude at 5000 Hz = -53.2414 dBV

```

Comment This example Macro sets the Analog Generator to 1kHz and sweeps the Bandpass filter through the 2nd to 5th harmonics. A settled reading is taken at each harmonic frequency and displayed on the Debug Immediate Tab.

AP.Anlr.FuncBPBRTuning

Property

Syntax `AP.Anlr.FuncBPBRTuning`

Data Type Integer

0	Counter tuned
1	Sweep track
2	Analog Generator track
3	Digital Generator track
4	Fixed frequency

Description This command sets the Bandpass Bandreject filter tuning source.

See Also `AP.Anlr.FuncBPBRFreq`

Example See example for `AP.Anlr.FuncBPBRFreq`.

AP.Anlr.FuncDetector

Property

Syntax `AP.Anlr.FuncDetector`

Data Type Integer

0	RMS
1	Average
2	Peak
3	Qpeak
4	Peak-Equivalent-Sine

Description This command selects the Detector Type for Function meter.

See Also `AP.Anlr.FuncInput`, `AP.Anlr.RdgRate`,
`AP.Anlr.FuncMode`, `AP.Anlr.FuncRange`,
`AP.Anlr.FuncRangeAuto`

Example See example for `AP.Anlr.FuncInput`.

AP.Anlr.FuncFilter

Property

Syntax `AP.Anlr.FuncFilter`

Data Type Integer

This command has two methods of controlling the selection of the Analog Analyzer Function Meter Filter.

The first method is to use the zero based list that selects the items in the Filter drop down list box from top to bottom. The disadvantage to this method is that if the number of filters located in the System Two hardware is different from another system then the list may be different. This may cause problems in selecting the correct filter with any macro using this method.

The following list shows how the Analog Analyzer Function Meter Filter drop down list box changes depending on what and how many filters are installed. Note that in configuration #1 the CCIR selections and the A-Weighting filter is available and in configuration #2 only the A-Weighting filter is provided. Also notice that the integer value required to select the A-Weighting filter is different.

	Configuration #1	Configuration #2
<i>0</i>	None	None
<i>1</i>	CCIR 468-3	A-Weighting
<i>2</i>	CCIR-2k	Slot #1
<i>3</i>	A-Weighting	Slot #2
<i>4</i>	Slot #1	Slot #3
<i>5</i>	Slot #2	Slot #4
<i>6</i>	Slot #3	Slot #5
<i>7</i>	Slot #4	Slot #6
<i>8</i>	Slot #5	Slot #7

9 Slot #6
10 Slot #7

The second method is to use ID numbers to select the appropriate filter. In this approach if the filter is available in the System Two the software will automatically find and activate the filter. This approach allows macros to be compatible from one System Two to another regardless of the filter configuration.

Refer to Appendix E Analog Filter ID List to obtain filter identification numbers.

Note: The return form of this command (`Var = AP.Anlr.FuncFilter`) will always return the zero based list number for the selected filter.

Description This command selects one or none of the available weighting filters. The weighting filters are optional filters that plug into the analyzer (internally). Any other setting attempted results in no weighting being selected (weighting 0).

See Also `AP.Anlr.FuncFilterHP`, `AP.Anlr.FuncFilterLP`

Example

```
Sub Main
  AP.Application.NewTest      'Reset panels
  AP.Anlr.ChAInput = 2
  AP.Sweep.Data1.Id = 5906   'Set Sweep Data 1 to _
                             "Anlr.Ampl"
  AP.Anlr.FuncFilter = 12017 "A" Weighting filter
  AP.Sweep.Start
End Sub
```

Comment The example program produces a graph that displays the frequency response for the "A" Weighting filter.

AP.Anlr.FuncFilterHP

Property

Syntax `AP.Anlr.FuncFilterHP`

Data Type Integer

0	<10 Hz
1	22 Hz
2	100 Hz
3	400 Hz

Description This command selects the value of High Pass filter in the function meter circuit.

See Also AP.Anlr.FuncFilterLowPass

Example

```

Sub Main
  AP.Application.NewTest      'Reset panels
  AP.Gen.Output = 1
  AP.Anlr.ChAInput = 1
  AP.Anlr.FuncFilterHP = 2 'Set High Pass filter to _
    100Hz.
  A = AP.Anlr.FuncFilterHP 'Return High Pass filter _
    setting value.
  AP.Sweep.Data1.Id = 5906
  AP.Sweep.Start
End Sub

```

Comment The example program produces a graph that displays the frequency response for the 100Hz High Pass filter.

AP.Anlr.FuncFilterLP

Property

Syntax AP.Anlr.FuncFilterLP

Data Type Integer

0	22 kHz
1	30 kHz
2	80 kHz
3	>500 kHz

Description This command selects the value of Low Pass filter in the function meter circuit.

See Also AP.Anlr.FuncFilterHP

Example Sub Main

```

AP.Application.NewTest      'Reset panels
AP.Gen.Output = 1
AP.Anlr.ChAInput = 1
AP.Anlr.FuncFilterLP = 1  'Set Low Pass filter to _
    30kHz.
A = AP.Anlr.FuncFilterLP  'Return Low Pass filter _
    setting value.
AP.Sweep.Data1.Id = 5906
AP.Sweep.Source1.Start("Hz") = 100000
AP.Sweep.Start
End Sub

```

Comment

The example program produces a graph that displays the frequency response for the 30kHz Low Pass filter.

AP.Anlr.FuncInput**Property****Syntax**

AP.Anlr.FuncInput

Data Type

Integer

0 Channel A

1 Channel B

Description

This command selects channel A or channel B to be used for measurements with the Function meter.

See Also

AP.Anlr.RdgRate, AP.Anlr.FuncDetector,
 AP.Anlr.FuncMode, AP.Anlr.FuncRange,
 AP.Anlr.FuncRangeBAuto

Example

```

Sub Main
AP.Application.NewTest      'Reset panels
AP.Gen.ChAAmpl("Vrms") = .01
AP.Anlr.ChAInput = 2
AP.Anlr.FuncMode = 0 'Set Function Meter mode to _
    amplitude.
AP.Anlr.FuncInput = 0      'Set Function Meter _
    channel to A.
AP.Anlr.FuncRangeAuto = 0 'Set Function Meter _
    range to fixed.

```

```

AP.Anlr.FuncRange("X/Y") = 4 'Set Function Meter _
    range to 4.0 X/Y.
AP.Anlr.FuncDetector = 1 'Set Function Meter _
    Average.
AP.Anlr.RdgRate = 1 'Set reading rate to _
    4/Sec.
AP.Anlr.FuncSettling(1, .000002, "V", 4, .05, 1)
AP.Anlr.FuncTrig 'Trigger new reading.
Do
    Ready = AP.Anlr.FuncReady 'Get status.
Loop Until Ready > 0
Reading1 = AP.Anlr.FuncRdg("dBV")'Get settled reading.
Debug.Print "Channel A Averaged Amplitude = _
    ";Format(Reading1, "#.0000");" dBV"
End Sub

```

Example Output Channel A Averaged Amplitude = -40.0410 dBV

AP.Anlr.FuncMode

Property

Syntax	AP.Anlr.FuncMode
Data Type	Integer
	0 Amplitude
	1 Bandpass
	2 Bandreject
	3 THD+N Amplitude
	4 THD+N Ratio
	5 SMPTE
	6 CCIF
	7 DIM
	8 Wow & Flutter
	9 2-Ch. Ratio
	10 Crosstalk
	11 DFD

Description This command selects the analysis mode of the Analyzer Function meter.

The measurement is taken from the selected channel, using the selected mode, using the unit specified by that mode.

If a reading is not ready when this function is called, it will wait for a reading to become available. Any particular reading will be returned only once.

See Also `AP.Anlr.FuncInput`, `AP.Anlr.FuncReady`,
`AP.Anlr.FuncSettling`, `AP.Anlr.FuncTrig`,
`AP.Anlr.RdgRate`, `AP.Anlr.FuncRange`,
`AP.Anlr.FuncRangeAuto`

Example See example for `AP.Anlr.FuncInput`.

AP.Anlr.FuncRange

Property

Syntax `AP.Anlr.FuncRange(ByVal Unit As String)`

Data Type Double

Parameters

Name	Description
<i>Unit</i>	The following units are available: X/Y, dB.

Description

This command sets the Analyzer Function meter Range.

The following table shows the gains (X/Y) and (dB) available and to what full-scale ranges they correspond for the various measurement modes of the analyzer:

X/Y	dB	Ampl(S2)	THD & DIM	CCIF/SMPTE
1	0	40 mV	100%	25%
4	12.041	10 mV	25%	6%
16	24.082	2.5 mV	6%	1.6%
64	36.124	600 μ V	1.6%	0.4%
256	48.165	150 μ V	0.4%	0.1%
1024	60.206	*40 μ V	0.1%	0.025%

* a gain of 1024 is valid only for Bandpass, Bandreject, THD+N, and Crosstalk measurements.

Note that for the amplitude ranges, the `AP.Anlr.ChARange` and `AP.Anlr.ChBRange` must be set to the 40 mV range before these ranges are valid. Likewise, the gain here should be set to 1 or Auto (See command : `AP.Anlr.FuncRangeAuto`) if the input range is set to anything other than 40 mV. While the Function meter ranges may be set independently of the input range settings, specified operation cannot be guaranteed if these cautions are not observed.

This range must be reprogrammed if the measurement mode of the Analyzer Function meter is changed (See `AP.Anlr.FuncMode`). Otherwise, the resulting range is not determinate.

A common use of this command is to set the Function meter Range by obtaining the gain reading while in auto range and then set the gain to the determined range. This keeps the Function meter Range from changing during an acquisition.

See Also

`AP.Anlr.FuncInput`, `AP.Anlr.RdgRate`,
`AP.Anlr.WFDDetector`, `AP.Anlr.FuncMode`,
`AP.Anlr.FuncRangeAuto`

Example

See example for `AP.Anlr.FuncInput`.

AP.Anlr.FuncRangeAuto**Property****Syntax**

`AP.Anlr.FuncRangeAuto`

Data Type

Boolean

True Auto range.
False Fixed range.

Description

This command sets the Function meter to Auto or Fixed Range.

See Also

`AP.Anlr.FuncInput`, `AP.Anlr.RdgRate`,
`AP.Anlr.FuncMode`, `AP.Anlr.FuncRange`

Example

See example for `AP.Anlr.FuncInput`.

AP.Anlr.FuncRdg

Property

Syntax `AP.Anlr.FuncRdg (ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
<i>Unit</i>	The following units V, dBu, dBV, dBr A, dBr B, dBg A, dBg B, dBm, W are available for the following Function meter Modes: Amplitude, Bandpass, Bandreject, THD+N Amplitude, (2-Channel Amplitude, and 2-Channel Band Pass Amplitude. The following units (% , dB, PPM, X/Y) are available for the following Function meter Modes: THD+N Ratio, SMPTE, CCIF, DIM, Wow & Flutter, 2-Channel Ratio, and Crosstalk.

Description This command returns a reading from the Function meter and zeros the ready count.

See Also `AP.Anlr.FuncInput`, `AP.Anlr.FuncMode`, `AP.Anlr.FuncReady`, `AP.Anlr.FuncSettling`, `AP.Anlr.FuncTrig`

Example

```

Sub Main
    AP.Application.NewTest          'Reset panels
    AP.Gen.Freq("Hz") = 3150
    AP.Gen.ChAAmpl("dBu") = 0.0
    AP.Anlr.ChAInput = 2          'GenMon input
    AP.Gen.Output = 1
    AP.Anlr.FuncMode = 8          'Select W&F mode
    AP.Anlr.FuncInput = 0          'Select Channel A input
    AP.Anlr.WFDetector = 1          'Set W&F detector to JIS
    AP.Anlr.WFFilter = 1 'Set W&F Filter to UnWeighted
    AP.Anlr.FuncSettling(5, .0002, "%", 3, .05, 1)
    AP.Anlr.FuncTrig          'Trigger new reading
    Do
        Ready = AP.Anlr.FuncReady 'Get status
    Loop Until Ready > 0
    Reading1 = AP.Anlr.FuncRdg("%") 'Get settled reading
    Debug.Print "Wow & Flutter = ";Format(Reading1, _
        "#.00000");" %"
End Sub

```

Example Output Wow & Flutter = .05305 %

AP.Anlr.FuncReady

Property

Syntax `AP.Anlr.FuncReady`

Data Type

Integer

`0` Reading not ready.

`>0` Reading ready.

Description This command returns the Function meter settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.Anlr.FuncRdg` or `AP.Anlr.FuncTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.Anlr.FuncRdg` command will be guaranteed to return quickly.

See Also

`AP.Anlr.FuncInput`, `AP.Anlr.FuncRdg`,
`AP.Anlr.FuncSettling`, `AP.Anlr.FuncTrig`

Example

See example for `AP.Anlr.FuncRdg`.

AP.Anlr.FuncSettling

Method

Syntax `AP.Anlr.FuncSettling`(ByVal *Tolerance* As Double,
ByVal *Floor* As Double, ByVal *FloorUnit* As String,
ByVal *Points* As Integer, ByVal *Delay* As Double,
ByVal *Algorithm* As Integer)

Parameters

See Appendix A for Settling Algorithm and parameter name descriptions.

Description

This command sets the settling parameters for the `AP.Anlr.FuncRdg` command.

See Also `AP.Anlr.FuncInput`, `AP.Anlr.FuncRdg`,
`AP.Anlr.FuncReady`, `AP.Anlr.FuncTrig`

Example See example for `AP.Anlr.FuncRdg`.

AP.Anlr.FuncTrig

Method

Syntax `AP.Anlr.FuncTrig`

Description Causes a restart of the reading cycle and zeros the ready count for the `AP.Anlr.FuncRdg` command. The reading in progress is aborted.

See Also `AP.Anlr.FuncInput`, `AP.Anlr.FuncRdg`,
`AP.Anlr.FuncReady`, `AP.Anlr.FuncSettling`

Example See example for `AP.Anlr.FuncRdg`.

AP.Anlr.PhaseMode

Property

Syntax `AP.Anlr.PhaseMode`

Data Type Integer

0	Auto
1	-180 +180 deg
2	0 +360 deg
3	-90 +270 deg

Description This function sets the Analog Analyzer Phase measurement range.

Example See example for `AP.Anlr.PhaseRdg`.

AP.Anlr.PhaseRdg

Property

Syntax `AP.Anlr.PhaseRdg(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	The following units are available: deg.
Description	This command returns the Analog Analyzer Phase meter settled reading.	
Example	<pre> Sub Main AP.Application.NewTest 'Reset panels AP.Gen.ChBInvert = 1 AP.Gen.Output = 1 AP.Anlr.ChAInput = 2 AP.Anlr.ChBInput = 2 AP.Anlr.PhaseMode = 0 AP.Anlr.PhaseSettling(0, .5, "deg", 3, .03, 1) AP.Anlr.PhaseTrig 'Trigger new reading. Do Ready = AP.Anlr.PhaseReady 'Get status. Loop Until Ready > 0 Reading1 = AP.Anlr.PhaseRdg("deg")'Get settled reading. Debug.Print "Phase A to B = ";Format(Reading1, _ "#.0000");" deg" End Sub </pre>	

Example Output Phase A to B = 179.9375 deg

AP.Anlr.PhaseReady

Property

Syntax	AP.Anlr.PhaseReady	
Data Type	Integer	
	0	Reading not ready.
	>0	Reading ready.

Description This command returns the Phase settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the AP.Anlr.PhaseRdg or AP.Anlr.PhaseTrig commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.Anlr.PhaseRdg` command will be guaranteed to return quickly.

See Also `AP.Anlr.PhaseMode`, `AP.Anlr.PhaseRdg`,
`AP.Anlr.PhaseSettling`, `AP.Anlr.PhaseTrig`

Example See example for `AP.Anlr.PhaseRdg`.

AP.Anlr.PhaseSettling

Method

Syntax `AP.Anlr.PhaseSettling`(ByVal *Tolerance* As Double,
ByVal *Floor* As Double, ByVal *FloorUnit* As String,
ByVal *Points* As Integer, ByVal *Delay* As Double,
ByVal *Algorithm* As Integer)

Parameters See Appendix A for Settling Algorithm and parameter name descriptions.

Description This command sets the settling parameters for the `AP.Anlr.PhaseRdg` command. Note that this command doesn't require a tolerance setting as in all other settling commands. Enter a 0 (Zero) as shown above for the first parameter as a place holder for the tolerance setting.

See Also `AP.Anlr.PhaseMode`, `AP.Anlr.PhaseRdg`,
`AP.Anlr.PhaseReady`, `AP.Anlr.PhaseTrig`

Example See example for `AP.Anlr.PhaseRdg`.

AP.Anlr.PhaseTrig

Method

Syntax `AP.Anlr.PhaseTrig`

Description Causes a restart of the reading cycle and zeros the ready count for the `AP.Anlr.PhaseRdg` command. The reading in progress is aborted.

See Also `AP.Anlr.PhaseMode`, `AP.Anlr.PhaseRdg`,
`AP.Anlr.PhaseReady`, `AP.Anlr.PhaseSettling`

Example See example for `AP.Anlr.PhaseRdg`.

AP.Anlr.RdgRate**Property****Syntax** `AP.Anlr.RdgRate`**Data Type**

Integer

<code>0</code>	Auto reading rate. The reading rate is automatically selected based on the measured frequency.
<code>1</code>	4/Sec fixed rate.
<code>2</code>	8/Sec fixed rate.
<code>3</code>	16/Sec fixed rate.
<code>4</code>	32/Sec fixed rate.
<code>5</code>	64/Sec fixed rate.
<code>6</code>	128/Sec fixed rate.
<code>7</code>	Auto-Fast.
<code>8</code>	Auto-Precise.

Description

This command sets the detector averaging time for the RMS and Average detectors.

These functions have no effect on the Peak and Qpeak detectors.

There is an inherent relationship between the detector averaging time and the lowest frequency component of the measured signal. The combinations of detector time constant and reading rate will affect both low frequency accuracy and digit stability.

For most applications, detector time constants should be ganged with reading rate, where the slowest time constant (range 1) is used for 4 readings/second, and the fastest (range 4) for 30 readings/sec.

See Also

`AP.Anlr.FuncInput`, `AP.Anlr.FuncDetector`,
`AP.Anlr.FuncMode`, `AP.Anlr.FuncRange`,
`AP.Anlr.FuncRangeAuto`

Example

See example for `AP.Anlr.FuncInput`.

AP.Anlr.RefChAdBr**Property****Syntax** `AP.Anlr.RefChAdBr (ByVal Unit As String)`**Data Type** Double

Reference value.

Parameters

Name	Description
<i>Unit</i>	The following units are available: V, dBV, dBu

Description

This command sets the zero dB value of Analog Analyzer dBr A reference.

Example

```

Sub Main
  AP.Application.NewTest           'Reset panels
  AP.Gen.ChAAmpl("Vrms") = 1
  AP.Gen.ChBTrackA = 1
  AP.Gen.Output = 1
  AP.Anlr.ChAInput = 2
  AP.Anlr.ChBInput = 2
  AP.Anlr.RefChAdBr("V") = 1
  AP.Anlr.RefChBdBr("V") = 1
  ReferenceA = AP.Anlr.RefChAdBr("V")
  ReferenceB = AP.Anlr.RefChBdBr("V")
  AP.Anlr.ChALevelSettling(1, .000002, "V", 4, .05, 1)
  AP.Anlr.ChBLevelSettling(1, .000002, "V", 4, .05, 1)
  AP.Anlr.ChALevelTrig 'Trigger new reading.
  AP.Anlr.ChBLevelTrig 'Trigger new reading.
  Do
    ReadyA = AP.Anlr.ChALevelReady 'Get status.
    ReadyB = AP.Anlr.ChBLevelReady 'Get status.
  Loop Until (ReadyA > 0 And ReadyB > 0)
  ReadingA = AP.Anlr.ChALevelRdg("dBr A")'Get settled _
    reading.
  ReadingB = AP.Anlr.ChBLevelRdg("dBr B")'Get settled _
    reading.
  Debug.Print "Channel A Gain = ";Format(ReadingA, _
    "#.0000");" dBr relative to";ReferenceA;" Volts"
  Debug.Print "Channel B Gain = ";Format(ReadingB, _
    "#.0000");" dBr relative to";ReferenceB;" Volts"
End Sub

```

Example Output

```

Channel A Gain = -.0296 dBr relative to 1 Volts
Channel B Gain = -.0073 dBr relative to 1 Volts

```

AP.Anlr.RefChBdB**Property**

Syntax	<code>AP.Anlr.RefdB(ByVal Unit As String)</code>	
Data Type	Double	Reference value.
Description	This command sets the zero dB value of the Analog Analyzer dBr B reference.	
Parameters	Name	Description
	<i>Unit</i>	The following units are available: V, dBV, dBu
Example	See example for <code>AP.Anlr.RefChAdBr</code> .	

AP.Anlr.RefdBm**Property**

Syntax	<code>AP.Anlr.RefdBm(ByVal Unit As String)</code>	
Data Type	Double	Reference value.
Parameters	Name	Discription
	<i>Unit</i>	The following units are available: Ohms.
Description	This command sets the Analog Analyzer dBm impedance value. This value of circuit impedance is used as the "R" value in the equation V^2/R to compute power from the measured voltage (V), followed by decibel conversion.	
See Also	<code>AP.Anlr.ChAImpedance</code> , <code>AP.Anlr.ChBImpedance</code> .	

Example

```

Sub Main
    AP.Application.NewTest          'Reset panels
    AP.Gen.ChBOutput = 0 'Set generator output B to OFF
    AP.Gen.Impedance = 2 'Set generator output Z to _
        600 Ohms
    AP.Gen.RefdBm("Ohms") = 600 'Set dBm reference to _
        600 Ohms
    AP.Gen.ChAAmpl("dBm") = 0
    AP.Anlr.ChAInput = 0
    AP.Gen.Output = 1
    AP.Anlr.ChAImpedance = 1 'Set Cha A input Z to _
        600 ohms

```

```

AP.Anlr.RefdBm("Ohms") = 600 'Set dBm reference To _
    600 Ohms
Reference = AP.Anlr.RefdBm("Ohms")
AP.Anlr.FuncSettling(1, .000002, "V", 4, .05, 1)
AP.Anlr.FuncTrig          'Trigger new reading.
Do
    Ready = AP.Anlr.FuncReady 'Get status
Loop Until Ready > 0
Reading1 = AP.Anlr.FuncRdg("dBm") 'Get settled reading
Debug.Print "Channel A Amplitude = ";Format _
    (Reading1, "#.0000");" dBm (";Reference;" Ohms)"
End Sub

```

Example Output Channel A Amplitude = -.0404 dBm (600 Ohms)

Comment This example requires that a XLR cable be connected between the Analog Generator channel A output and the Analog Analyzer channel A input.

AP.Anlr.RefdBrAuto

Method

Syntax **AP.Anlr.RefdBrAuto**

Result Boolean

True dBm reference set.
False dBm reference NOT set.

Description This command sets the Analyzer dBm Reference field(s).

The following logic is used to determine which meter reading is written into the reference field for System One:

- 1 If the Function meter has dBm units selected the Function meter is used to determine the dBm Reference value.
- 2 If the Level meter has dBm units selected the Level meter is used to determine the dBm Reference value.
- 3 If the Function meter mode `AP.Anlr.FuncMode` is set to Amplitude (0), Bandpass (1), or Bandreject (2) mode the Function meter is used to determine the dBm Reference value.

- 4 If the Function meter mode `AP.Anlr.FuncMode` is set to THD+N (3 or 4) mode with the units set to Volts, dBm, dBu, dBV, or dBr the Function meter is used to determine the dBr Reference value.
- 5 Otherwise the Level meter is used to determine the dBr Reference value.

The following logic is used to determine which meter reading is written into which reference field for System Two:

If the Function meter units selected on the Analog Analyzer panel are not either dBrA or dBrB, then the Channel A Level meter reading is written into the dBrA Reference field and the Channel B Level meter reading is written into the dBrB Reference field.

If the Function meter units are either dBrA or dBrB and the corresponding Level meter is not set to a dBr unit, the Function meter measurement is written into the corresponding dBr Reference field and the other dBr Reference field takes its value from the Level meter on the corresponding channel.

Example

```
Sub Main
  AP.File.OpenTest ("SNR.AT2") 'Open signal-to-noise _
    test.
  AP.Anlr.RefdBrAuto          'Set dBr reference.
  Return = AP.Anlr.RefdBrAuto
  If Return = True Then Debug.Print "Reference Set"
  AP.Gen.Output = 0            'Turn generator OFF.
  AP.Sweep.Start              'Start single point sweep.
End Sub
```

Example Output Reference Set

Comment This example performs a single point signal to noise measurement. The measurement result is displayed in the Data Editor. The text "Reference Set" is output to the DeBug Immediate Tab of the Macro editor.

AP.Anlr.RefFreq**Property****Syntax** `AP.Anlr.RefFreq(ByVal Unit As String)`**Data Type** Double Set reference frequency value.

Parameters	Name	Description
	<i>Unit</i>	The following units are available: Hz only.

Description This command sets the value of Analog Analyzer Frequency reference. This reference value is used by all Analog Analyzer relative frequency units (F/R, dHz, %Hz, cent, octs, decs, d%, dPPM).

Example

```

Sub Main
    AP.Application.NewTest          'Reset panels
    AP.Gen.Output = 1
    AP.Anlr.ChAInput = 1
    AP.Anlr.RefFreq("Hz") = 1000
    Ref = AP.Anlr.RefFreq("Hz")
    AP.Anlr.ChAFreqSettling(.5, .0002, "Hz", 3, .03, 1)
    AP.Anlr.ChAFreqTrig          'Trigger new reading.
    Do
        Ready = AP.Anlr.ChAFreqReady 'Get status.
    Loop Until Ready > 0
    Reading1 = AP.Anlr.ChAFreqRdg("dHz") 'Get settled _
        reading.
    Debug.Print "Frequency delta relative to";Ref; _
        "Hz = ";Format(Reading1, "#.0000");" dHz"
End Sub

```

Example Output Frequency delta relative to 1000Hz = -0.3173 dHz**AP.Anlr.RefFreqAuto****Method****Syntax** `AP.Anlr.RefFreqAuto`**Result** Boolean*True* Frequency reference set.*False* Frequency reference NOT set.

Description This command sets the Analyzer Frequency Reference field to the current frequency reading.

Example

```
Sub Main
  AP.Application.NewTest      'Reset panels
  AP.Anlr.FuncInput = 0
  AP.Gen.Output = 1
  AP.Anlr.ChAInput = 1
  AP.Anlr.RefFreqAuto      'Set frequency reference.
  Return = AP.Anlr.RefFreqAuto 'Return reference _
    frequency.
  If Return = True Then Debug.Print "Reference Set"
  AP.Gen.Freq("Hz") = 2000
  AP.Anlr.ChAFreqTrig
  Do
    Ready = AP.Anlr.ChAFreqReady
  Loop Until Ready > 0
  A = AP.Anlr.ChAFreqRdg("dHz")
  Debug.Print "Frequency change = "; Format(A, "#.0000")
End Sub
```

Example Output Reference Set
Frequency change = 999.6768

AP.Anlr.RefWatts

Property

Syntax **AP.Anlr.RefWatts**(ByVal *Unit* As String)

Data Type Double Set Watts reference Impedance value.

Parameters	Name	Description
	<i>Unit</i>	The following units are available: Ohms only.

Description This command sets the value of Analog Analyzer Watts reference impedance. The known external load impedance should be entered, from which the software computes power from the measured voltage and the equation $VU! ^ X/O!A!2/R$ where *R* is the reference impedance.

Example

```
Sub Main
  AP.Application.NewTest      'Reset panels
  AP.Gen.Output = 1
```

```

AP.Gen.RefWatts("Ohms") = 8
AP.Gen.ChAAmpl("W") = .1
AP.Anlr.ChAInput = 2
AP.Anlr.RefWatts("Ohms") = 8
AP.Anlr.FuncSettling(1, .000002, "V", 4, .05, 1)
AP.Anlr.FuncTrig           'Trigger new reading.
Do
    Ready = AP.Anlr.FuncReady 'Get status.
Loop Until Ready > 0
Reading1 = AP.Anlr.FuncRdg("W") 'Get settled reading.
Debug.Print "Output Power = ";Format(Reading1, _
    "#.0000");" Watts"
End Sub

```

Example Output Output Power = .0997 Watts

AP.Anlr.WFDetector

Property

Syntax	AP.Anlr.WFDetector
Data Type	Integer
	0 NAB-RMS
	1 JIS
	2 IEC-PK
Description	The Function meter mode must be set to Wow & Flutter.
See Also	AP.Anlr.FuncFilterWF, AP.Anlr.FuncMode
Example	<pre> Sub Main AP.Application.NewTest 'Reset panels AP.Gen.Freq("Hz") = 3150 AP.Gen.Ampl("dBu") = 0.0 AP.Gen.Output = 1 AP.Anlr.ChAInput = 1 AP.Anlr.FuncMode = 8 AP.Anlr.FuncInput = 0 AP.Anlr.WFDetector = 1 'Set W&F detector to JIS. AP.Anlr.WFFilter = 1 'Set W&F Filter to UnWeighted. AP.Anlr.FuncSettling(5, .0002, "%", 3, .05, 1) </pre>

```

AP.Anlr.FuncTrig           'Trigger new reading.
Do
    Ready = AP.Anlr.FuncReady      'Get status.
Loop Until Ready > 0
Reading1 = AP.Anlr.FuncRdg("%") 'Get settled reading.
Debug.Print "Wow & Flutter = ";Format(Reading1, _
    "#.0000");" %"
End Sub

```

Example Output Wow & Flutter = .0500 %

AP.Anlr.WFFilter

Property

Syntax **AP.Anlr.WFFilter**

Data Type Integer

0	Weighted
1	UnWeighted
2	Weighted-High Band
3	UnWeighted-High Band
4	Wide-High Band
5	Scrape-High Band

Description This command sets the Analog Analyzer Wow & Flutter Filter weighting.

See Also AP.Anlr.WFDetector, AP.Anlr.FuncMode

Example See example for AP.Anlr.WFDetector.

User Notes

User Notes

Application

AP.Application.AppDir

Method

Syntax `AP.Application.AppDir`

Result String

Description This command returns the application directory. When installing the software the default application directory is "C:\Program Files\Audio Precision\APWIN200\" for Windows 95, 98, and NT. Utility programs are located in the application directory.

Example

```
Declare Function GetShortPathName Lib "kernel32" _
    Alias "GetShortPathNameA" _
    (ByVal lpLongPath As String, _
    ByVal lpShortPath As String, _
    ByVal nSizeShortPath As Long) As Long
```

```
Sub main()
    Dim LongPath As String
    Dim ShortPath As String
    ShortPath = String(255, vbNullChar)

    LongPath = AP.Application.AppDir
    ReturnLength = GetShortPathName(LongPath, _
        ShortPath, Len(ShortPath))
    ShortPath = Left(ShortPath, ReturnLength)
    Debug.Print "Long Path = ";LongPath
    Debug.Print "Short Path = ";ShortPath
End Sub
```

Example Output Long Path = C:\PROGRAM FILES\AUDIO PRECISION\APWIN200\
Short Path = C:\PROGRA~1\AUDIOP~1\APWIN200\

AP.Application.ClearCurrentError

Method

Syntax `AP.Application.ClearCurrentError`

Description This command when executed clears the current error.

Note: In APWIN Basic, "Dim WithEvents" is allowed in any module. In Visual Basic, "Dim WithEvents" is only allowed in Class modules.

See Appendix D Extensions Error Codes for Error String numbers and descriptions.

Example

```
Sub Main
    AP.Gen.ChAAMPL("Vrms") = 111.9    'Cause an error _
    and see what happens.
End Sub
Sub APEvent_OnError(Code As Long)
    Debug.Print "Got number " & Code & " " & _
        AP.Application.GetCurrentErrorString

    ' If you are going to handle the error, then call
    ' AP.Application.ClearCurrentError before you exit
    ' this subroutine to stop APWIN from displaying the
    ' error,

AP.Application.ClearCurrentError

    ' It is also preferable to call
    ' AP.Application.ClearCurrentError before you
    ' make any other calls into APWIN in case these
    ' calls also generate an unexpected error
End Sub
```

AP.Application.CopyPanelToClipboard

Method

Syntax **AP.Application.CopyPanelToClipboard**

Description This command copies the graphic image of the panel that has focus to the Clipboard.

Example

```
Sub Main
    AP.Application.NewTest    'Create Graph with data
    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2
    AP.Anlr.FuncFilterHP = 3
```

```

AP.Anlr.FuncFilterLP = 0
AP.Sweep.Data1.Id = 5906
AP.Sweep.Source1.Start("Hz") = 50000.0
AP.Sweep.Start
AP.Graph.OptimizeLeft
AP.Application.CopyPanelToClipboard

Dim MSWord As Object
Set MSWord = CreateObject("Word.Basic") 'Start Word
With MSWord
    .AppShow                'Display MS Word
    .FileOpen Name:= CurDir & "\GENERIC.DOC"
    .EditFind "Place Graph Here" 'Search for string
    .EditPaste 'Paste Graph into Word
    Wait 10
    .FileCloseAll 2 'Close all open files
    .AppClose 'Close MS Word
End With
End Sub

```

AP.Application.DisplayDataOnTestOpen

Property

Syntax	AP.Application.DisplayDataOnTestOpen
Data Type	Boolean
	<i>True</i> Display data on test open.
	<i>False</i> Don't display data on test open.
Description	This command specifies whether the measurement data saved in a test file is displayed when the file is loaded.
Example	<pre> Sub Main AP.Application.DisplayDataOnTestOpen = 0 AP.File.OpenTest "SAMPLE1.AT2C" 'Define strings to be used in the following prompt. String1\$ = "Test Loaded and data NOT displayed." AP.Prompt.Text = String1\$ AP.Prompt.FontSize = 10 'Set prompt font size to 8 _ point. </pre>

```

AP.Prompt.Position -1,-1,290,120      'Set prompt _
    location and size.
AP.Prompt.ShowWithContinue      'Display prompt with _
    Continue Macro button displayed.
Stop 'Stop Macro until Continue Macro _
    button is pressed.

AP.Application.DisplayDataOnTestOpen = 1
AP.File.OpenTest "SAMPLE1.AT2C"
'Define strings to be used in the following prompt.
String1$ = "Test loaded and data displayed."
AP.Prompt.Text = String1$
AP.Prompt.FontSize = 10            'Set prompt _
    font size to 8 point.
AP.Prompt.Position -1,-1,290,100     'Set prompt _
    location and size.
AP.Prompt.ShowWithContinue      'Display prompt with _
    Continue Macro button displayed.
Stop 'Stop Macro until Continue Macro _
    button is pressed.
End Sub

```

AP.Application.DoReadings

Method

Syntax AP.Application.DoReadings

Result None

Description This command forces a reading cycle to take place. The reading cycle allows the reading commands (commands ending in Rdg such as AP.Anlr.FuncRdg) to make and return a measurement. Under normal conditions when a dialog is displayed the automatic readings cycle is disabled and readings will not return correctly. Use this comand to force a reading cycle to take place while a dialog is displayed.

Example

```

Sub Main
    AP.Application.NewTest
    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2
    AP.Gen.ChAAmpl("Vrms") = 0.5

```

```

Begin Dialog UserDialog 270,105,"Readings
Example",.Handler 'GRID:10,7,1,1
    PushButton 40,7,190,21,"Make Reading",.PushButton1
    TextBox 40,35,190,21,.TextBox1
    CancelButton 40,63,190,21
End Dialog
Dim dlg As UserDialog
Select Case Dialog (dlg)
Case 0
    End
End Select
End Sub

Private Function Handler(DlgItem$, Action%,
SuppValue&) As Boolean
    Select Case Action%
Case 1          'Dialog box initialization
Case 2          'Value changing or button pressed
    If DlgItem$ = "PushButton1" Then
        Handler = True 'Prevent button press from _
            closing the dialog box
        AP.Anlr.FuncSettling(5.000000, 1.000000e-007, _
            "V", 3, 0.050000, 1)
        AP.Anlr.FuncTrig          'Trigger new reading
    Do
        AP.Application.DoReadings 'Start reading _
            cycle
        Ready = AP.Anlr.FuncReady 'Get status.
    Loop Until Ready > 0
        Reading1 = AP.Anlr.FuncRdg("V") 'Get settled _
            reading.
        DlgText "TextBox1", "Amplitude = " & _
            Format(Reading1, "#.00000") & " V"
    End If
Case 3          'TextBox or ComboBox text changed
Case 4          'Focus changed
Case 5          'Idle
        Rem Handler = True 'Continue getting idle actions
Case 6          'Function key
    End Select

```

End Function

AP.Application.GetCurrentErrorString

Method

Syntax `AP.Application.GetCurrentErrorString`

Result String

Description This command returns the ASCII text string for the current error.

See Appendix D Extensions Error Codes for Error String numbers and descriptions.

Example See example for `AP.Application.ClearCurrentError`.

AP.Application.Input

Method

Syntax `AP.Application.Input(ByVal PortAddress As Integer)`

Parameters	Name	Description
	<i>PortAddress</i>	An I/O address value between 0 and 65535 (FFFF Hex).

Result Integer

Description The purpose of this command is to read input from an external device through a parallel port, or an I/O mapped interface card plugged into the computer.

The decimal read address of the first printer port (treated by DOS as LPT1) is 889 (379 Hex) and the second port (LPT2) is 633 (279 Hex).

The standard parallel port has four pins that can be used for Input.; pins 11, 12, 13, and 15. Each line is held high by an internal pull-up resistor and requires approximately 1mA to pull the line low, this will allow other parallel ports to drive the input.

When all of the input lines (11(), 12(), 13(), and 15()) are high the `AP.Application.Input` command will return decimal 127. The following list shows the return value for each line when it is pulled low.

Note: This command is not available for the Windows NT operating system.

Input Line	Bit Value	Data Line	Return Value
11	128	7	255
12	32	5	95
13	16	4	111
15	8	3	119

Example

```

Sub Main
  Return = AP.Application.Input(889)
  If Return = 127 Then Debug.Print "All Lines pulled _
    high."
  If Return And 128 Then      'Using And logic.
    Debug.Print "Pin 11 pulled low."
  End If
  If Return Xor 32 Then      'Using Xor logic.
    Debug.Print "Pin 12 pulled low."
  End If
  If Not Return And 16 Then  'Using Not And logic.
    Debug.Print "Pin 13 pulled low."
  End If
  If Not Return And 8 Then   'Using Not And logic.
    Debug.Print "Pin 15 pulled low."
  End If
End Sub

```

Example Output All Lines pulled high.

AP.Application.MacroDir**(OLE) Method**

Syntax **AP.Application.MacroDir**

Result String

Description This command returns the running macro source directory. This command is like the MacroDir\$ command in the Language reference section of APWIN Basic with the exception that this command can be used from an OLE client that is accessing APWIN to determine the directory from which the selected macro was loaded.

Example

```

Private Sub Form_Load()
  Dim AP As Object

```

```

    Set AP = CreateObject("APWIN.Application")
'The following lines makes the Visual Basic Current
' Directory and the APWIN Working Directory the same
' as the directory where the current APWIN macro was
' loaded from.

    ChDir AP.Application.MacroDir
    AP.Application.WorkingDir = AP.Application.MacroDir

'Your code goes here.

End Sub

```

AP.Application.Name

Method

Syntax

AP.Application.Name

Result

String ASCII characters.

Description

This command returns the APWIN Application Name Audio Precision APWIN. This text string is located in the upper left corner of the APWIN application before the test name. This string is useful when using the AppActivate command located in the Language reference section of APWIN Basic.

Example

```

Sub Main
    AppActivate AP.Application.Name 'Select the APWIN _
        window
    'The following SendKey command will now be sent to _
        the APWIN application.
    SendKeys "%WC",1            'Clear all windows on page.
    SendKeys "%PO",1            'Display Data Editor.

    'In Debug mode focus is automatically returned to
    ' the editor each time the user interacts with the
    ' controls. Therefore it is important to note that
    ' sections of code containing commands that are to
    ' be sent to other applications via the SendKeys
    ' command need to be executed without interruption.

```

```
'When debugging these areas place a breakpoints
' before and after the SendKeys commands to maintain
' the correct window/application focus.
```

```
End Sub
```

AP.Application.NewData

Method

Syntax

AP.Application.NewData

Result

Boolean

True Data removed from memory.

False Command failed to remove data from memory.

Description

This command deletes the measurements currently in memory. The command is functionally the same as selecting File, New, Data from the Menu bar.

Example

```
Sub Main
  AP.File.OpenTest "FRQ-RESP.AT2C"        'Open frequency _
                                          response test.
  AP.Application.NewData
  AP.Sweep.Start                            'Start sweep.
  AP.File.SaveDataAs "FRQ-RESP.DAT"       'Save data.

  AP.File.OpenTest "THD-FRQ.AT2C"        'Open total _
                                          harmonic distortion + noise test.
  AP.Application.NewData
  AP.Sweep.Start                            'Start sweep.
  AP.File.SaveDataAs "THD-FRQ.DAT"       'Save data.

  AP.File.OpenTest "RESIDNOI.AT2C"        'Open residual _
                                          noise test.
  AP.Application.NewData
  AP.Sweep.Start                            'Start sweep.
  AP.File.SaveDataAs "RESIDNOI.DAT"       'Save data.
End Sub
```

AP.Application.NewMacro**(OLE) Method**

Syntax	AP.Application.NewMacro	
Result	Boolean	
	<i>True</i>	New macro created.
	<i>False</i>	Command failed to create new macro.
Description	This command initializes the macro editor and is only to be used via OLE. The command is functionally the same as selecting File, New, Macro, and OK from the Menu bar.	

AP.Application.NewTest**Method**

Syntax	AP.Application.NewTest	
Result	Boolean	
	<i>True</i>	New test panel configuration restored.
	<i>False</i>	Command failed to restore new test panel configuration.
Description	This command initializes the current APWIN test to the default test condition. The command is functionally the same as selecting New Test from the Standard Toolbar or selecting File, New, Test, and OK from the Menu bar.	

Example

```

Sub Main
    AP.Application.NewTest 'Reset panels
    AP.Gen.Output = 1
    AP.Anlr.ChAInput = 1
    AP.Anlr.FuncFilterLP = 1
    A = AP.Anlr.FuncFilterLP
    AP.Sweep.Data1.Id = 5906
    AP.Sweep.Source1.Start("Hz") = 100000
    AP.Sweep.Start
End Sub

```

AP.Application.Output

Method

Syntax	AP.Application.Output (ByVal <i>PortAddress</i> As Integer, ByVal <i>DataByte</i> As Integer)	
Parameters	Name	Description
	<i>PortAddress</i>	An I/O address value between 0 and 65535 (FFFF Hex).
	<i>DataByte</i>	Any value between 0 and 255 (FF Hex).
Description	<p>The purpose of this command is to control an external device through a parallel port, or an I/O mapped interface card plugged into the computer.</p> <p>The standard parallel port has eight pins that can be used for Output.; pins 2, 3, 4, 5, 6, 7, 8, and 9. The decimal address of the first printer port (treated by DOS as LPT1) is 888 (378 Hex) and the second port (LPT2) is 632 (278 Hex).</p> <p>Note: This command is not available for the Windows NT operating system.</p>	
Example	<pre>Sub Main 'Set all LPT1 bits high. Application.Output(888, 255) End Sub</pre>	

AP.Application.Page

Property

Syntax	AP.Application.Page	
Data Type	Integer	
	1	Page #1.
	2	Page #2.
	3	Page #3.
	4	Page #4.
	5	Page #5.
Description	This command displays the selected page,	
Example	Sub Main	

```

AP.Application.NewTest 'Reset panels
AP.Application.Page = 1
Return = AP.Application.Page
Debug.Print "Page "; Return; " displayed."
Wait 1 'So the user can see the page change.
AP.Application.Page = 2
Return = AP.Application.Page
Debug.Print "Page "; Return; " displayed."
Wait 1
AP.Application.Page = 3
Return = AP.Application.Page
Debug.Print "Page "; Return; " displayed."
End Sub

```

AP.Application.PanelClose

Method

Syntax

```

AP.Application.PanelClose(ByVal PanelID As
AP.PanelConstants)

```

Parameters

Constants	Description
<i>apbPanelAnalogGenLarge</i>	Remove the Analog Generator panel from view.
<i>apbPanelAnalogGenSmall</i>	Remove the Analog Generator panel from view.
<i>apbPanelAnlrLarge</i>	Remove the Analog Analyzer panel from view.
<i>apbPanelAnlrSmall</i>	Remove the Analog Analyzer panel from view.
<i>apbPanelBarGraph?</i>	Remove the desired Bar Graph 1 through 32 from view.
<i>apbPanelDataEditor</i>	Remove the Data Editor panel from view.
<i>apbPanelDCXLarge</i>	Remove the DCX-127 panel from view.

apbPanelDCXSmall

Remove the DCX-127 panel from view.

apbPanelDiagnostic

Remove the Diagnostic panel from view.

apbPanelDigIOLarge

Remove the Digital Input / Output panel from view.

apbPanelDigIOSmall

Remove the Digital Input / Output panel from view.

apbPanelDigitalGenLarge

Remove the Digital Generator panel from view.

apbPanelDigitalGenSmall

Remove the Digital Generator panel from view.

apbPanelDIOStatusBitsLarge

Remove the Status Bits panel from view.

apbPanelDIOStatusBitsSmall

Remove the Status Bits panel from view.

apbPanelDSPSmall

Remove the DSP panel from view.

apbPanelDSPLarge

Remove the DSP panel from view.

apbPanelGraph

Remove the Graph from view.

apbPanelRefInput

Remove the Sync Reference panel from view.

apbPanelRegulation

Remove the Regulation panel from view.

apbPanelSerialInterface

Remove the Serial Interface panel from view.

apbPanelSettling

Remove the Settling panel from view.

apbPanelSpeaker

Remove the Speaker panel from view.

apbPanelSweepLarge

Remove the Sweep panel from view.

apbPanelSweepSmall

Remove the Sweep panel from view.

apbPanelSwitcher

Remove the Switcher panel from view.

Description

This command closes the selected panel,

Example

```

Sub Main
  AP.Application.NewTest
  AP.Application.PanelOpen(apbPanelAnalogGenLarge)
  AP.Application.PanelOpen(apbPanelAnlrLarge)
  AP.Application.PanelOpen(apbPanelSweep)
  AP.Gen.Output = True
  AP.Anlr.ChAInput = 2
  AP.Anlr.FuncFilterHP = 3
  AP.Sweep.Data1.Id = 5906
  AP.Application.Page = 2
  AP.Application.Page = 3
  AP.Application.PanelClose(apbPanelDigIOSmall)
  AP.Application.Page = 2
  AP.Sweep.Start
End Sub

```

AP.Application.PanelOpen**Method****Syntax**

```

AP.Application.PanelOpen(ByVal PanelID As
AP.PanelConstants)

```

Parameters**Constants** **Description***apbPanelAnalogGenLarge*

Display Large view of Analog Generator panel.

apbPanelAnalogGenSmall

Display Large view of Analog Generator panel.

apbPanelAnlrLarge

Display Large view of Analog Analyzer panel.

apbPanelAnlrSmall

Display Small view of Analog Analyzer panel.

apbPanelBarGraph?

Display desired Bar Graph panel 1 through 32.

apbPanelDataEditor

Display Data Editor panel.

apbPanelDCXLarge

Display Large view of DCX-127 panel.

apbPanelDCXSmall

Display Small view of DCX-127 panel.

apbPanelDiagnostic

Display Diagnostic panel.

apbPanelDigIOLarge

Display Large view of Digital Input / Output panel.

apbPanelDigIOSmall

Display Small view of Digital Input / Output panel.

apbPanelDigitalGenLarge

Display Large view of Digital Generator panel.

apbPanelDigitalGenSmall

Display Small view of Digital Generator panel.

apbPanelDIOStatusBitsLarge

Display Large view of Status Bits panel.

apbPanelDIOStatusBitsSmall

Display Small view of Status Bits panel.

apbPanelDSPSmall

Display Small view of DSP panel.

apbPanelDSPLarge

Display Large view of DSP panel. When None is selected with the `AP.S2CDsp.Program` command this constant will display the Small view of the DSP panel.

apbPanelGraph

Display Graph panel.

<i>apbPanelRefInput</i>	Display Sync Reference panel.
<i>apbPanelRegulation</i>	Display Regulation panel.
<i>apbPanelSerialInterface</i>	Display Serial Interface panel.
<i>apbPanelSettling</i>	Display Settling panel.
<i>apbPanelSpeaker</i>	Display Speaker panel.
<i>apbPanelSweepLarge</i>	Display Large view of Sweep panel.
<i>apbPanelSweepSmall</i>	Display Small view of Sweep panel.
<i>apbPanelSwitcher</i>	Display Switcher panel.

Description This command displays the selected panel.

Example See example for `AP.Application.PanelClose`.

AP.Application.Quit

Method

Syntax `AP.Application.Quit`

Description This command terminates APWIN and returns to Windows. If the "Prompt to Save Test when a test is closed" selection in the Utilities, Config menu is enabled the operator will be prompted to save changed files when APWIN quits.

Example

```
Sub Main
Start:
    ChDir MacroDir
    Begin Dialog UserDialog 430,105
        PushButton 20,21,380,28,"Your Code",.Field1
        PushButton 130,63,180,28,"Exit APWIN",.Field3
    End Dialog
```

```

Dim Main_Menu As UserDialog

Select Case Dialog(Main_Menu)
  Case 1
    'Incert your code here...
  Case Else
    AP.Application.Quit           'Exit APWIN
End Select
GoTo Start:
End Sub

```

AP.Application.Restore

Method

Syntax

AP.Application.Restore

Description

This command restores the hardware to the present state of the software.

This function should be used if the hardware loses power or becomes disconnected from the computer.

Example

```

Sub Main
  Start:
  Begin Dialog UserDialog 430,105,"Example Menu"
    PushButton 40,28,170,42,"Restore Hardware",.Field1
    PushButton 230,28,160,42,"Exit Macro",.Field2
  End Dialog
  Dim Main_Menu As UserDialog

  Select Case Dialog(Main_Menu)
    Case 1
      AP.Application.Restore
    Case Else
      End
  End Select
  GoTo Start:
End Sub

```

AP.Application.SetWatchDogTimer1**Method****Syntax**

```
AP.Application.SetWatchDogTimer1(ByVal Sec As Double [, Optional ByVal ThrowError As Variant])
```

Parameters	Name	Description
	<i>Sec</i>	Defines the amount of time that will elapse after starting the WatchDog Timer before the APEvent_OnWatchDogTimeout event is generated. To disable the timer at any time set the time value to (0) zero seconds.
	<i>ThrowError</i>	Optional parameter. Set this parameter to True to throw/raise an error (11021) when the defined time has elapsed. Basic's On Error mechanism can then detect the error. The default (False) condition will not throw/raise an error when the defined time has elapsed.

Description

This command sets up and starts timer number 1. When the defined time expires the APEvent_OnWatchDogTimeout event is generated. In addition an error can be thrown/raised to allow Basic's On Error mechanism to intercept the error.

Example

```
Dim Halt As Boolean
Sub Main
    Halt = False

    AP.Application.NewTest
    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2

    AP.Sweep.Source1.Steps = 30

    AP.Application.SetWatchDogTimer1(5.0,False)

    AP.Sweep.StartNoWait
    Do
    Loop While Halt = False

End Sub
Sub APEvent_OnWatchDogTimeout(ByVal Id As Long)
    If Id = 1 Then
        If AP.Sweep.IsRunning = True Then
```

```

        AP.Sweep.Stop
        Debug.Print "Sweep Stopped"
    End If
End If
End Sub

```

AP.Application.SetWatchDogTimer2

Method

Syntax

```

AP.Application.SetWatchDogTimer2(ByVal Sec As
Double [,Optional ByVal ThrowError As Variant])

```

Parameters	Name	Description
<i>Sec</i>		Defines the amount of time that will elapse after starting the WatchDog Timer before the APEvent_OnWatchDogTimeout event is generated. To disable the timer at any time set the time value to (0) zero seconds.
<i>ThrowError</i>		Optional parameter. Set this parameter to True to throw/raise an error (11021) when the defined time has elapsed. Basic's On Error mechanism can then detect the error. The default (False) condition will not throw/raise an error when the defined time has elapsed.

Description

This command sets up and starts timer number 2. When the defined time expires the APEvent_OnWatchDogTimeout event is generated. In addition an error can be thrown/raised to allow Basic's On Error mechanism to intercept the error.

Example

```

Dim Halt As Boolean
Sub Main
    Halt = False

    AP.Application.NewTest
    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2

    AP.Sweep.Source1.Steps = 30

    AP.Application.SetWatchDogTimer2(5.0,False)

```

```

    AP.Sweep.StartNoWait
    Do
    Loop While Halt = False

End Sub
Sub APEvent_OnWatchDogTimeout(ByVal Id As Long)
    If Id = 2 Then
        If AP.Sweep.IsRunning = True Then
            AP.Sweep.Stop
            Debug.Print "Sweep Stopped"
        End If
    End If
End Sub

```

AP.Application.SysType

Method

Syntax **AP.Application.SysType**

Result String

"1" APWIN running in System One mode.
 "2" APWIN running in System Two mode.
 "2C" APWIN running in System Two Cascade mode.

Description This command returns the current mode of the APWIN software.

Example

```

Sub Main
    Select Case AP.Application.SysType
        Case "1"
            AP.Prompt.Text = "APWIN configured for _
                System One hardware."
        Case "2"
            AP.Prompt.Text = "APWIN configured for _
                System Two hardware."
        Case "2C"
            AP.Prompt.Text = "APWIN configured for _
                System Two Cascade hardware."
    End Select
    AP.Prompt.ShowWithContinue
    Stop
End Sub

```

AP.Application.TestDir

Method

Syntax	AP.Application.TestDir
Result	String
Description	This command returns the path of the test (.AT1 or .AT2) that is currently loaded.
Example	<pre> Sub Main AP.Application.DisplayDataOnTestOpen = 0 AP.File.OpenTest "SAMPLE1.AT2C" 'Get current test name TestName\$ = AP.Application.TestName 'Get directory that the current test was loaded from TestDir\$ = AP.Application.TestDir 'Define strings to be used in the following prompt. String1\$ = "Test file " String2\$ = " was loaded from the " String3\$ = " directory." AP.Prompt.Text = String1\$ & TestName\$ & String2\$ & _ TestDir\$ & String3\$ AP.Prompt.FontSize = 10 'Set prompt font size to 8 _ point. AP.Prompt.Position -1,-1,290,130 'Set prompt _ location and size. AP.Prompt.ShowWithContinue 'Display prompt with _ Continue Macro button displayed. Stop 'Stop Macro until Continue Macro _ button is pressed. End Sub </pre>

AP.Application.TestName

Method

Syntax	AP.Application.TestName
Result	String
Description	This command returns the test (.AT1 or .AT2) file name of the test that is currently loaded.

Example See example for `AP.Application.TestDir`.

AP.Application.ThrowErrors

Property

Syntax `AP.Application.ThrowErrors`

Data Type Boolean

True Expose Errors and Warnings.
False Don't expose Errors and Warnings..

Description This command exposes errors and warnings generated by APWIN to the Err. object.

See Appendix D Extensions Error Codes for Error String numbers and descriptions.

Example

```
Sub Main
    ' Pick one the three On Error possibilities below
    ' On Error GoTo 0      ' Disable your error handler _
                        (default). Basic will handle _
                        the error by termination.
    On Error GoTo MyErrorHandler ' Send error _
                        conditions to "MyHandler"
    ' On Error Resume Next ' Error conditions continue _
                        execution at the next statement.
    ' Caution-This is generally very dangerous _
    ' as no error will be seen
    ' APWIN will not send errors to Basic's _
    ' "On Error" unless AP.Application. _
    ' ThrowErrors is set to True
    ' After this is set to True, APWIN will _
    ' no longer display errors, they will all _
    ' be passed to Basic
    AP.Application.ThrowErrors = True
    ' Now cause an error and see what happens.
    AP.Gen.Ampl("Vrms") = 111.9
    ' Cause another error and see what happens.
    AP.Gen.Freq("Hz") = 2.1
    MsgBox "Resumed after the offending Call"
Exit Sub ' Exit to avoid handler.
```



```

MyErrorHandler:
    ' show some debug info
    Debug.Print "Err=";Err.Number
    Debug.Print "Description=";Err.Description
    Debug.Print "Source=";Err.Source

    ' Select different actions for errors
    Select Case Err.Number      ' Evaluate error number
        Case 8504
            ' "Generator Amplitude" error.
            ' put your error handler code here
            MsgBox "Got to the Handler"
            ' If you handled the error, then resume
            Resume Next
        Case Else
            ' Handle other situations here...
            ' If we don't know about the err then or _
            ' instead you could decide that Basic should _
            ' handle this
            ' Note that if this is a called subroutine, _
            ' Basic will pass the error back to the _
            ' calling subroutine.
            ' Passing all the info:
            Err.Raise Err.Number, Err.Source, _
                Err.Description
            ' Or more simply
            Error Err
            ' Or you could exit this subroutine
            Exit Sub
    End Select
End Sub

```

AP.Application.Version

Method

Syntax **AP.Application.VisibleMacroEditor**

Result Double

Description This command returns the running APWIN Application Version number. This command can be used to check if the running version of APWIN is compatible with the running macro.

Example

```
Sub Main
  'APWIN version 1.6 required
  If AP.Application.Version <> 1.6 Then End
  AP.Application.NewTest
  AP.Application.PanelOpen(apbPanelAnalogGenLarge)
  AP.Application.PanelOpen(apbPanelAnlrLarge)
  AP.Application.PanelOpen(apbPanelSweep)
  AP.Gen.Output = True
  AP.Anlr.ChAInput = 2
  AP.Anlr.FuncFilterHP = 3
  AP.Sweep.Data1.Id = 5906
  AP.Application.Page = 2
  AP.Application.Page = 3
  AP.Application.PanelClose(apbPanelDigIOSmall)
  AP.Application.Page = 2
  AP.Sweep.Start
End Sub
```

AP.Application.Visible

Property

Syntax **AP.Application.Visible**

Data Type Boolean

True Restore APWIN to view.

False Remove APWIN from view.

Description This command when executed makes the APWIN window visible or invisible. The Macro Editor remains visible.

Example

```
Sub Main
  AP.Application.Visible = False 'Remove APWIN from view.
  AP.Application.NewTest          'Reset panels
  AP.Gen.Output = 1
  AP.Anlr.ChAInput = 2
  AP.Sweep.Start
  AP.Application.Visible = True 'Restore APWIN.
```

End Sub

AP.Application.VisibleAll

Property

Syntax **AP.Application.VisibleAll**

Data Type Boolean

True Restore to view.

False Remove from view.

Description This command enables or disables display of the Graph and Bar-Graph displays, Data Editor, and Panels when a test is loaded during Macro execution only. Use this command at the beginning of your macro to decrease overall test times.

Example

```
Private Sub Form_Load()
    Dim AP As Object
    Set AP = CreateObject("APWIN.Application")
    ' Create OLE link to APWIN.
    AP.Application.Visible = True ' Make APWIN visible

    AP.Application.VisibleAll = True
    AP.File.OpenTest "VIEW.AT2C"
    'Test loaded displaying ALL graphic panels
    AP.Application.VisibleBarGraphs = False
    'Disable display of Bar Graphs
    AP.File.OpenTest "VIEW.AT2C"
    AP.Application.VisibleDataEditor = False
    AP.File.OpenTest "VIEW.AT2C"
    'Disable display of Data Editor
    AP.Application.VisibleGraph = False
    AP.File.OpenTest "VIEW.AT2C"
    'Disable display of Graph
    AP.Application.VisiblePanels = False
    AP.File.OpenTest "VIEW.AT2C"
    'Disable display of Instrument panels
    AP.Application.Quit           'Quit APWIN
    End
End Sub
```

AP.Application.VisibleBarGraphs**Property****Syntax** `AP.Application.VisibleBarGraphs`**Data Type** Boolean*True* Restore to view.*False* Remove from view.**Description** This command enables or disables display of the Bar-Graph display, when a test is loaded during Macro execution only. Use this command at the beginning of your macro to decrease overall test times.**Example** See example for `AP.Application.VisibleAll`.**AP.Application.VisibleDataEditor****Property****Syntax** `AP.Application.VisibleDataEditor`**Data Type** Boolean*True* Restore to view.*False* Remove from view.**Description** This command enables or disables display of the Data Editor panel when a test is loaded during Macro execution only. Use this command at the beginning of your macro to decrease overall test times.**Example** See example for `AP.Application.VisibleAll`.**AP.Application.VisibleGraph****Property****Syntax** `AP.Application.VisibleGraph`**Data Type** Boolean*True* Restore to view.*False* Remove from view.

Description This command enables or disables display of the Graph display when a test is loaded during Macro execution only. Use this command at the beginning of your macro to decrease overall test times.

Example See example for `AP.Application.VisibleAll`.

AP.Application.VisibleMacroEditor

Method

Syntax `AP.Application.VisibleMacroEditor(ByVal bVisible As Boolean)`

Parameters	Name	Description
	<i>bVisible</i>	True = Restore Macro Editor to view. False = Remove Macro Editor from view.

Description This command when executed makes the APWIN Macro Editor visible or invisible. Dialogs displayed when the Macro Editor is invisible have a higher Z-order (which window is on top of another) than the APWIN window, therefore when focus is moved to the APWIN window the dialog remains displayed on top of the APWIN application. If the Macro Editor is visible then the Z-order is relative to the Macro Editor and the dialog may be covered by any other window that has focus.

Example

```
Sub Main
    AP.Application.VisibleMacroEditor(False) 'Remove _
        Macro Editor from view.
    AP.Application.NewTest                    'Reset panels
    AP.Gen.Output = 1
    AP.Anlr.ChAInput = 2
    AP.Sweep.Start
    AP.Application.VisibleMacroEditor(True) 'Restore _
        Macro Editor.
End Sub
```

AP.Application.VisiblePanels

Property

Syntax `AP.Application.VisiblePanels`

Data Type Boolean

True Restore to view.
False Remove from view.

Description This command enables or disables display of the Panels when a test is loaded during Macro execution only. Use this command at the beginning of your macro to decrease overall test times.

Example See example for `AP.Application.VisibleAll`.

AP.Application.WorkingDir

Property

Syntax `AP.Application.WorkingDir`

Data Type String

Description This command sets or returns the current working directory. This command is like the `ChDir$` command in the Language reference section of APWIN Basic with the exception that this command can be used from an OLE client to change the APWIN working directory.

Example

```
Private Sub Form_Load()
    Dim AP As Object
    Set AP = CreateObject("APWIN.Application")
    'The following line makes the APWIN Working Directory
    ' the same as the VB current directory.
    If AP.Application.AppDir <> CurDir Then
        AP.Application.WorkingDir = CurDir
    'Your code goes here.
End Sub
```

User Notes

User Notes

User Notes

Auxiliary Instrument

AP.Aux.Reading1Rdg

Property

Syntax `AP.Aux.Reading1Rdg`

Data Type `Double`

Description This command returns a settled reading for Auxiliary Reading #1 and zeros the ready count.

See Also `AP.Aux.Reading1Rdg`, `AP.Aux.Reading1Settling`,
`AP.Aux.Reading1Trig`

Example

```
' Uses the APWIN-GPIB Library GPIBLIB.APB
' A National Instruments GPIB card must be installed
' in your system to use this file.
'#Uses "gpib-lib.apb"
' See the GPIB-LIB.apb file for instructions on use.
'#Uses "APNiglob.bas"
'#Uses "APVbib32.bas"
Public iAP As Integer

Sub Main
Dim iStatus As Integer, iAddr As Integer, iAnyOneHome _
    As Integer, sResponse As String

    iAP = ildev(0, 1, NO_SAD, T3s, 1, 0) 'Open I/O to _
        this GPIB address ** Assumes Board 0, Address 1

    iStatus = illn(iAP, 1, NO_SAD, iAnyOneHome)
'Check for listener at address 1
    If iAnyOneHome Then 'Found a listener at address 1
        iStatus = ilclr(iAP) 'Device clear
        GpibWrite iAP, "*IDN?" 'Query for Instrument _
            Identificaiton
        sResponse = GpibRead(iAP, 60) 'Get response
        If Not (ibsta And EERR) Then 'If no GPIB _
            read error Then save response string
            GpibWrite iAP, "*CLS;*RST;" 'Clear status _
```

```

        registers and reset all settings
    End If
End If

result = GpibWrite(iAP, ":HEADER OFF;")
result = GpibWrite(iAP, ":SETTLE OFF;")
result = GpibWrite(iAP, ":OUTPUT ON;")

AP.File.OpenTest("AUX GPIB Example.at2")
AP.Aux.Reading1Settling(1.0, 0.0000001, 3, 0.03, 1)
AP.Aux.Setting1 = 1000.0 'Set frequency for _
    sweep to return to when done
AP.Sweep.Start
End Sub
Sub APEvent_OnSweepTrigger()
AP.Aux.Reading1Trig
While AP.Aux.Reading1Ready = 0
    result = GpibWrite(iAP, ":M1?")
    Str1 = GpibRead(iAP,80) 'Read METER M1 result
    Debug.Print "Reading = " & Str1
    AP.Aux.SetReading1(Val(Str1))
    Wait .1
Wend
    Debug.Print "Settled Reading = " & Str1
End Sub
Sub APEvent_OnSweepStep(Value As Variant, Source As _
    Long)
    Value = Format(Value, "###.####")
    Debug.Print "Setting = " & Value
    'GPIB Code to set ATS Generator Frequency to _
    "Value" variable
    result = GpibWrite(iAP, ":GFREQUENCY " & _
        Str$(Value)&"") 'Set Aux Generator Freq
End Sub

```

AP.Aux.Reading1Ready

Property

Syntax **AP.Aux.Reading1Ready**

Data Type Integer

0 Reading not ready.
 >0 Reading ready.

Description	<p>This command returns the Auxiliary Reading #1 settled reading ready count.</p> <p>Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the <code>AP.Aux.Reading1Rdg</code> or <code>AP.Aux.Reading1Trig</code> commands will zero the ready count.</p> <p>If the reading is found to be ready, a call to the <code>AP.Aux.Reading1Rdg</code> command will be guaranteed to return quickly.</p>
See Also	<code>AP.Aux.Reading1Rdg</code> , <code>AP.Aux.Reading1Settling</code> , <code>AP.Aux.Reading1Trig</code>
Example	See example for <code>AP.Aux.Reading1Rdg</code> .

AP.Aux.Reading1Settling

Method

Syntax	<code>AP.Aux.Reading1Settling(ByVal Tolerance As Double, ByVal Floor As Double, ByVal FloorUnit As String, ByVal Points As Integer, ByVal Delay As Double, ByVal Algorithm As Integer)</code>
Parameters	See Appendix A for Settling Algorithm and parameter name descriptions.
Description	This command sets the settling parameters for the <code>AP.Aux.Reading1Rdg</code> command.
See Also	<code>AP.Aux.Reading1Rdg</code> , <code>AP.Aux.Reading1ReadyAP_Aux_Reading1Ready</code> , <code>AP.Aux.Reading1Trig</code>
Example	See example for <code>AP.Aux.Reading1Rdg</code> .

AP.Aux.Reading1Trig**Method**

Syntax	<code>AP . Aux . Reading1Trig</code>
Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP . Aux . Reading1Rdg</code> command. The reading in progress is aborted.
See Also	<code>AP . Aux . Reading1Rdg</code> , <code>AP . Aux . Reading1ReadyAP_Aux_Reading1Ready</code> , <code>AP . Aux . Reading1Settling</code>
Example	See example for <code>AP . Aux . Reading1Rdg</code> .

AP.Aux.Reading2Rdg**Property**

Syntax	<code>AP . Aux . Reading2Rdg</code>
Data Type	Double
Description	This command returns a settled reading for Auxiliary Reading #2 and zeros the ready count.
See Also	<code>AP . Aux . Reading2Rdg</code> , <code>AP . Aux . Reading2Settling</code> , <code>AP . Aux . Reading2Trig</code>
Example	See example for <code>AP . Aux . Reading1Rdg</code> .

AP.Aux.Reading2Ready**Property**

Syntax	<code>AP . Aux . Reading2Ready</code>
Data Type	Integer
	<i>0</i> Reading not ready.
	<i>>0</i> Reading ready.
Description	This command returns the Auxiliary Reading #2 settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.Aux.Reading2Rdg` or `AP.Aux.Reading2Trig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.Aux.Reading2Rdg` command will be guaranteed to return quickly.

See Also `AP.Aux.Reading2Rdg`, `AP.Aux.Reading2Settling`, `AP.Aux.Reading2Trig`

Example See example for `AP.Aux.Reading1Rdg`.

AP.Aux.Reading2Settling

Method

Syntax `AP.Aux.Reading2Settling(ByVal Tolerance As Double, ByVal Floor As Double, ByVal FloorUnit As String, ByVal Points As Integer, ByVal Delay As Double, ByVal Algorithm As Integer)`

Parameters See Appendix A for Settling Algorithm and parameter name descriptions.

Description This command sets the settling parameters for the `AP.Aux.Reading2Rdg` command.

See Also `AP.Aux.Reading2Rdg`, `AP.Aux.Reading2Ready`, `AP.Aux.Reading2Trig`

Example See example for `AP.Aux.Reading1Rdg`.

AP.Aux.Reading2Trig

Method

Syntax `AP.Aux.Reading2Trig`

Description Causes a restart of the reading cycle and zeros the ready count for the `AP.Aux.Reading2Rdg` command. The reading in progress is aborted.

See Also AP.Aux.Reading2Rdg, AP.Aux.Reading2Ready, AP.Aux.Reading2Settling

Example See example for AP.Aux.Reading1Rdg.

AP.Aux.Reading3Rdg

Property

Syntax AP.Aux.Reading3Rdg

Data Type Long

Description This command returns a settled reading for Auxiliary Reading #3 and zeros the ready count.

See Also AP.Aux.Reading3Rdg, AP.Aux.Reading3Settling, AP.Aux.Reading3Trig

Example See example for AP.Aux.Reading1Rdg.

AP.Aux.Reading3Ready

Property

Syntax AP.Aux.Reading3Ready

Data Type Integer

0 Reading not ready.
>0 Reading ready.

Description This command returns the Auxiliary Reading #3 settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the AP.Aux.Reading3Rdg or AP.Aux.Reading3Trig commands will zero the ready count.

If the reading is found to be ready, a call to the AP.Aux.Reading3Rdg command will be guaranteed to return quickly.

See Also AP.Aux.Reading3Rdg, AP.Aux.Reading3Settling, AP.Aux.Reading3Trig

Example See example for AP.Aux.Reading1Rdg.

AP.Aux.Reading3Settling

Method

Syntax **AP.Aux.Reading3Settling**(ByVal *Tolerance* As Double, ByVal *Floor* As Double, ByVal *FloorUnit* As String, ByVal *Points* As Integer, ByVal *Delay* As Double, ByVal *Algorithm* As Integer)

Parameters See Appendix A for Settling Algorithm and parameter name descriptions.

Description This command sets the settling parameters for the AP.Aux.Reading3Rdg command.

See Also AP.Aux.Reading3Rdg, AP.Aux.Reading3Ready, AP.Aux.Reading3Trig

Example See example for AP.Aux.Reading1Rdg.

AP.Aux.Reading3Trig

Method

Syntax **AP.Aux.Reading3Trig**

Description Causes a restart of the reading cycle and zeros the ready count for the AP.Aux.Reading3Rdg command. The reading in progress is aborted.

See Also AP.Aux.Reading3Rdg, AP.Aux.Reading3Ready, AP.Aux.Reading3Settling

Example See example for AP.Aux.Reading1Rdg.

AP.Aux.Reading4Rdg**Property**

Syntax	<code>AP.Aux.Reading4Rdg</code>
Data Type	Long
Description	This command returns a settled reading for Auxiliary Reading #4 and zeros the ready count.
See Also	<code>AP.Aux.Reading4Rdg</code> , <code>AP.Aux.Reading4Settling</code> , <code>AP.Aux.Reading4Trig</code>
Example	See example for <code>AP.Aux.Reading1Rdg</code> .

AP.Aux.Reading4Ready**Property**

Syntax	<code>AP.Aux.Reading4Ready</code>				
Data Type	Integer				
	<table> <tr> <td><code>0</code></td> <td>Reading not ready.</td> </tr> <tr> <td><code>>0</code></td> <td>Reading ready.</td> </tr> </table>	<code>0</code>	Reading not ready.	<code>>0</code>	Reading ready.
<code>0</code>	Reading not ready.				
<code>>0</code>	Reading ready.				
Description	<p>This command returns the Auxiliary Reading #4 settled reading ready count.</p> <p>Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the <code>AP.Aux.Reading4Rdg</code> or <code>AP.Aux.Reading4Trig</code> commands will zero the ready count.</p> <p>If the reading is found to be ready, a call to the <code>AP.Aux.Reading4Rdg</code> command will be guaranteed to return quickly.</p>				
See Also	<code>AP.Aux.Reading4Rdg</code> , <code>AP.Aux.Reading4Settling</code> , <code>AP.Aux.Reading4Trig</code>				
Example	See example for <code>AP.Aux.Reading1Rdg</code> .				

AP.Aux.Reading4Settling

Method

Syntax	AP.Aux.Reading4Settling (ByVal <i>Tolerance</i> As Double, ByVal <i>Floor</i> As Double, ByVal <i>FloorUnit</i> As String, ByVal <i>Points</i> As Integer, ByVal <i>Delay</i> As Double, ByVal <i>Algorithm</i> As Integer)
Parameters	See Appendix A for Settling Algorithm and parameter name descriptions.
Description	This command sets the settling parameters for the AP.Aux.Reading4Rdg command.
See Also	AP.Aux.Reading4Rdg, AP.Aux.Reading4Ready, AP.Aux.Reading4Trig
Example	See example for AP.Aux.Reading1Rdg.

AP.Aux.Reading4Trig

Method

Syntax	AP.Aux.Reading4Trig
Description	Causes a restart of the reading cycle and zeros the ready count for the AP.Aux.Reading4Rdg command. The reading in progress is aborted.
See Also	AP.Aux.Reading4Rdg, AP.Aux.Reading4Ready, AP.Aux.Reading4Settling
Example	See example for AP.Aux.Reading1Rdg.

AP.Aux.SetReading1

Method

Syntax **AP.Aux.SetReading1**(ByVal *Value* as Double)

Parameters	Name	Description
<i>Value</i>		Any Double Data Type value.

Description This command sets the value used by the Aux instrument Reading 1 parameter.

Example See example for `AP.Aux.Reading1Rdg`.

AP.Aux.SetReading2

Method

Syntax `AP.Aux.SetReading2(ByVal Value as Double)`

Parameters	Name	Description
<i>Value</i>		Any Double Data Type value.

Description This command sets the value used by the Aux instrument Reading 2 parameter.

Example See example for `AP.Aux.Reading1Rdg`.

AP.Aux.SetReading3

Method

Syntax `AP.Aux.SetReading3(ByVal Value as Long)`

Parameters	Name	Description
<i>Value</i>		Any Long Data Type value.

Description This command sets the value used by the Aux instrument Reading 3 parameter.

Example See example for `AP.Aux.Reading1Rdg`.

AP.Aux.SetReading4

Method

Syntax `AP.Aux.SetReading4(ByVal Value as Long)`

Parameters	Name	Description
<i>Value</i>		Any Long Data Type value.

Description This command sets the value used by the Aux instrument Reading 4 parameter.

Example See example for `AP.Aux.Reading1Rdg`.

AP.Aux.Setting1

Property

Syntax	<code>AP.Aux.Setting1</code>
Data Type	Double
Description	This command sets the value used by the Aux instrument Setting 1 parameter.
Example	See example for <code>AP.Aux.Reading1Rdg.</code>

AP.Aux.Setting2

Property

Syntax	<code>AP.Aux.Setting2</code>
Data Type	Double
Description	This command sets the value used by the Aux instrument Setting 2 parameter.
Example	See example for <code>AP.Aux.Reading1Rdg.</code>

AP.Aux.Setting3

Property

Syntax	<code>AP.Aux.Setting3</code>
Data Type	Long
Description	This command sets the value used by the Aux instrument Setting 3 parameter.
Example	See example for <code>AP.Aux.Reading1Rdg.</code>

AP.Aux.Setting4

Property

Syntax	<code>AP.Aux.Setting4</code>
Data Type	Long

Description This command sets the value used by the Aux instrument Setting 4 parameter.

Example See example for `AP.Aux.Reading1Rdg`.

User Notes

User Notes

User Notes

User Notes

Bar Graph

AP.BarGraph.AxisAutoScale

Property

Syntax `AP.BarGraph.AxisAutoScale(ByVal BarId As Integer)`

Data Type Boolean

True Auto scale Bar Graph.

False Disable auto scale.

Parameters

Name	Description
<i>BarId</i>	Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.

Description

Set the selected Bar Graph Axis to Auto Scale.

Example

```
Sub Main
    AP.Application.NewTest          'Setup Code to make _
        something to test
    AP.Gen.Output = True
    AP.Gen.ChAFreq("Hz") = 3000.0
    AP.Anlr.ChAInput = 2
    AP.Anlr.FuncMode = 1
    AP.Anlr.FuncBPBRTuning = 4
    AP.Anlr.FuncBPBRFreq("Hz") = 3000.0 'End Setup code

    With AP.BarGraph
        GenFreqBar = .New          'Create New Bargraph
        .Id(GenFreqBar) = 5051     'Configure Bargraph _
            to control Generator Frequency
        .AxisLogLin(GenFreqBar) = 1          'Linear axis
        .AxisRight(GenFreqBar,"Hz") = 3500.0 'Right value
        .AxisLeft(GenFreqBar,"Hz") = 2500.0 'Left value
        .AxisIncrement(GenFreqBar,"Hz") = 10.0 'Step size

        AnlrFuncRdg = .New          'Create New Bargraph
        .Id(AnlrFuncRdg) = 5907     'Configure Bargraph _
            to display Function meter readings
        .DigitsOnly(AnlrFuncRdg) = False 'Display Digits _
            and Bar on the Bargraph
    End With
End Sub
```

```

.AxisLogLin(AnlrFuncRdg) = 1      'Linear axis
.AxisLeft(AnlrFuncRdg,"V") = 0.8  'Left value
.AxisRight(AnlrFuncRdg,"V") = 1.2 'Right value
.AxisAutoScale(AnlrFuncRdg) = True 'Autoscale _
    Readings
.TargetLower(AnlrFuncRdg,"V") = 0.95 'Target _
    Lower value
.TargetUpper(AnlrFuncRdg,"V") = 1.05 'Target _
    Upper value
.TargetRange(AnlrFuncRdg) = True     'Display _
    Target area

.Reset(GenFreqBar)    'Reset #1 Min/Max readings
.Reset(AnlrFuncRdg)  'Reset #2 Min/Max readings

String1$ = "Adjust Generator Frequency using _
    Bargraph #" & GenFreqBar
String2$ = " for Maximum Amplitude on _
    Bargraph #" & AnlrFuncRdg & "."
AP.Prompt.Text = String1$ & String2$
AP.Prompt.Position(0,0,1150,120)
AP.Prompt.ShowWithContinue
Stop

GenMaxSet = .Max(GenFreqBar)  'Create _
    readings prompt
GenMinSet = .Min(GenFreqBar)
AnlrMaxRdg = .Max(AnlrFuncRdg)
AnlrMinRdg = .Min(AnlrFuncRdg)
End With

MaxSet$ = "Maximum Frequency = " _
    & Left(Str$(GenMaxSet),6) & " Hz" & Chr(13)
MinSet$ = "Minimum Frequency = " _
    & Left(Str$(GenMinSet),6) & " Hz" & Chr(13) _
    & Chr$(13)
MaxRdg$ = "Maximum Voltage = " _
    & Left(Str$(AnlrMaxRdg),6) & " V" & Chr(13)
MinRdg$ = "Minimum Voltage = " _
    & Left(Str$(AnlrMinRdg),6) & " V" & Chr(13) _
    & Chr$(13)

```

```

CurSet$ = "Current Frequency Setting = " _
          & Left(Str$(Gen.Freq("Hz")),6) & " Hz" & Chr(13)
CurRdg$ = "Current Amplitude Reading = " _
          & Left(Str$(Anlr.FuncRdg("V")),6) & " V"

AP.Prompt.Text = MaxSet$ & MinSet$ & MaxRdg$ _
                & MinRdg$ & CurSet$ & CurRdg$
AP.Prompt.Position(0,0,550,350)
AP.Prompt.ShowWithContinue
Stop
End Sub

```

AP.BarGraph.AxisIncrement

Property

Syntax `AP.BarGraph.AxisIncrement` (ByVal *BarId* As Integer, ByVal *Unit* As String)

Data Type Double

Parameters	Name	Description
	<i>BarId</i>	Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.
	<i>Unit</i>	Refer to the setting or reading defined by the <code>AP.BarGraph.Id</code> command to determine the appropriate unit selections.

Description Set the selected Bar Graph increment decrement size. When the Bar Graph is configured to control a setting (for example the generator frequency) the arrow keys can be used to increment or decrement the frequency by the increment value.

Example See example for `AP.BarGraph.AxisAutoScale`.

AP.BarGraph.AxisLeft

Property

Syntax `AP.BarGraph.AxisLeft` (ByVal *BarId* As Integer, ByVal *Unit* As String)

Data Type Double

Parameters	Name	Description
	<i>BarId</i>	Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.
	<i>Unit</i>	Refer to the setting or reading defined by the <code>AP.BarGraph.Id</code> command to determine the appropriate unit selections.
Description	This command defines the value on the left side of the Bar Graph.	
See Also	<code>AP.BarGraph.AxisRight</code> , <code>AP.BarGraph.AxisAutoScale</code>	
Example	See example for <code>AP.BarGraph.AxisAutoScale</code> .	

AP.BarGraph.AxisLogLin

Property

Syntax `AP.BarGraph.AxisLogLin(ByVal BarId As Integer)`

Data Type Integer

0 Logarithmic axis.
1 Linear axis.

Parameters	Name	Description
	<i>BarId</i>	Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.

Description This command determines the Bar Graph axis data scaling type.

Example See example for `AP.BarGraph.AxisAutoScale`.

AP.BarGraph.AxisRight

Property

Syntax `AP.BarGraph.AxisRight(ByVal BarId As Integer)`

Data Type Double

Parameters	Name	Description
	<i>BarId</i>	Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.

unit\$ Refer to the setting or reading defined by the `AP.BarGraph.Id` command to determine the appropriate unit selections.

Description This command defines the value on the right side of the Bar Graph.

See Also `AP.BarGraph.AxisLeft`, `AP.BarGraph.AxisAutoScale`

Example See example for `AP.BarGraph.AxisAutoScale`.

AP.BarGraph.Comment

Property

Syntax `AP.BarGraph.Comment (ByVal BarId As Integer)`

Data Type String ASCII characters.

Parameters	Name	Description
------------	------	-------------

<i>BarId</i>		Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.
--------------	--	--

Description This command transfers the ASCII characters to or from the comment section in the BarGraph panel to a string variable.

See Also `AP.BarGraph.CommentShow`

Example

```
Sub Main
    AP.Application.NewTest
    AP.Gen.Output = True
    With AP.Anlr
        .ChAInput = 2
        .FuncMode = 1
        .FuncBPBRTuning = 4
        .FuncBPBRFreq("Hz") = 3000.0
    End With
    With AP.BarGraph
        BarID1 = .New(5051) 'Setup Settings Bar Graph
        .AxisLeft(BarID1,"Hz") = 2500.0
        .AxisRight(BarID1,"Hz") = 3500.0
        .AxisIncrement(BarID1,"Hz") = 1.0
        .Title(BarID1) = "Bar Graph 1: Analog Generator _
            Frequency"
        BarID2 = .New(5907) 'Setup Readings Bar Graph
```

```

        .AxisLeft(BarID2,"V") = 0.50
        .AxisRight(BarID2,"V") = 1.50
        .CommentShow(BarID2) = True
        .Title(BarID2) = "Analog Analyzer Bandpass _
            Amplitude"
        .Comment(BarID2) = "Adjust Bar Graph #1 for _
            maximum amplitude reading."
    End With
    With AP.Prompt
        .FontSize = 8           'Set font size to 8 point.
        .Position(290,244,225,120) 'Set location and size.
        .Text = Chr$(10) & "Press this button to _
            proceed." 'Set string to display in prompt.
        .ShowWithContinue 'Display prompt with _
            Continue button.
    Stop 'Stop macro.
    End With
    Debug.Print "Filter peek = " & AP.Gen.Freq("Hz") & _
        " Hz"
End Sub

```

AP.BarGraph.CommentShow

Property

Syntax	AP.BarGraph.CommentShow
Data Type	Boolean
	<i>True</i> Display Comment section.
	<i>False</i> Remove Comment section from view.
Description	This command displays or removes from view the comment section in the Graph panel
See Also	AP.BarGraph.Comment
Example	See example for AP.BarGraph.Comment.

AP.BarGraph.DigitsOnly

Property**Syntax** `AP.BarGraph.DigitsOnly(ByVal BarId As Integer)`**Data Type** Boolean*True* Display digits only.*False* Display digits and Bar Graph.**Parameters****Name** **Description***BarId* Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.**Description**

This command displays only the digits (numeric characters) or the digits and the bar on the BarGraph.

Example

See example for `AP.BarGraph.AxisAutoScale`.

AP.BarGraph.Id

Property**Syntax** `AP.BarGraph.Id(ByVal BarId As Integer)`**Data Type** Long Instrument Parameter ID#.**Parameters****Name** **Description***BarId* Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.**Description**

This command is used to select the instrument parameter, which will return readings or control settings for the selected Bar Graph.

Refer to Appendix B to obtain instrument parameter identification numbers.

Example

See example for `AP.BarGraph.AxisAutoScale`.

AP.BarGraph.Max

Property

Syntax `AP.BarGraph.Max`(ByVal *BarId* As Integer, ByVal *Unit* As String)

Parameters	Name	Description
	<i>BarId</i>	Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.
	<i>Unit</i>	Refer to the setting or reading defined by the <code>AP.BarGraph.Id</code> command to determine the appropriate unit selections.

Result Double

Description This command returns the maximum measured value obtained during the time since the last reset for the selected Bar Graph

See Also `AP.BarGraph.Reset`, `AP.BarGraph.Min`

Example See example for `AP.BarGraph.AxisAutoScale`.

AP.BarGraph.Min

Property

Syntax `AP.BarGraph.Min`(ByVal *BarId* As Integer, ByVal *Unit* As String)

Parameters	Name	Description
	<i>BarId</i>	Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.
	<i>Unit</i>	Refer to the setting or reading defined by the <code>AP.BarGraph.Id</code> command to determine the appropriate unit selections.

Result Double

Description This command returns the maximum measured value obtained during the time since the last reset for the selected Bar Graph

See Also `AP.BarGraph.Reset`, `AP.BarGraph.Max`

Example See example for `AP.BarGraph.AxisAutoScale`.

AP.BarGraph.New

Method

Syntax	AP.BarGraph.New [(Optional ByVal <i>ConId</i> As Variant)]	
Parameters	Name	Description
	<i>ConId</i>	Instrument identification number. Refer to Appendix B to obtain instrument parameter identification numbers.
Result	Integer	
	<i>1-32</i>	Identification number of Bar Graph created.
Description	This command creates a new Bar Graph and returns the identification number.	
Example	See example for <code>AP.BarGraph.AxisAutoScale</code> .	

AP.BarGraph.Reset

Method

Syntax	AP.BarGraph.Reset (ByVal <i>BarId</i> As Integer)	
Parameters	Name	Description
	<i>BarId</i>	Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.
Description	This command resets the selected Bar Graph. The reset action sets the Min and Max. values to the current reading and as additional readings are taken the Min and Max. readings track the deviations	
See Also	<code>AP.BarGraph.Max</code> , <code>AP.BarGraph.Min</code>	
Example	See example for <code>AP.BarGraph.AxisAutoScale</code> .	

AP.BarGraph.TargetLower

Property

Syntax `AP.BarGraph.TargetLower (ByVal BarId As Integer, ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>BarId</i>	Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.
	<i>Unit</i>	Refer to the setting or reading defined by the <code>AP.BarGraph.Id</code> command to determine the appropriate unit selections.

Description This command defines the target value for the left side of the Bar Graph.

See Also `AP.BarGraph.TargetUpper`, `AP.BarGraph.TargetRange`

Example See example for `AP.BarGraph.AxisAutoScale`.

AP.BarGraph.TargetRange

Property

Syntax `AP.BarGraph.TargetRange (ByVal BarId As Integer)`

Data Type Boolean

<i>True</i>	Target area displayed.
<i>False</i>	Target area not displayed.

Parameters	Name	Description
	<i>BarId</i>	Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.

Description This command turns the selected Bar Graph Target Range ON or OFF.

See Also `AP.BarGraph.TargetLower`, `AP.BarGraph.TargetUpper`

Example See example for `AP.BarGraph.AxisAutoScale`.

AP.BarGraph.TargetUpper

Property

Syntax `AP.BarGraph.TargetUpper (ByVal BarId As Integer, ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>BarId</i>	Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.
	<i>Unit</i>	Refer to the setting or reading defined by the <code>AP.BarGraph.Id</code> command to determine the appropriate unit selections.

Description This command defines the value on the right side of the Bar Graph.

See Also `AP.BarGraph.TargetLower`, `AP.BarGraph.TargetRange`

Example See example for `AP.BarGraph.AxisAutoScale`.

AP.BarGraph.Title

Property

Syntax `AP.BarGraph.Title (ByVal BarId As Integer)`

Data Type String ASCII characters.

Parameters	Name	Description
	<i>BarId</i>	Bar Graph identification number (1-32). The identification number is located on the Bar Graph title bar.

Description This command transfers the ASCII characters to or from the title bar in the BarGraph panel to a string variable.

Example See example for `AP.BarGraph.Comment`.

User Notes

User Notes

User Notes

Status Bits

AP.Bits.ChAAudioModeRdg

Property

Syntax `AP.Bits.ChAAudioModeRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters

Part	Description
<i>String</i>	Optional string containing status bit information.

Result

Value	Description
0	Audio Mode
1	Data Mode

Description

This command returns the Status Bits channel A Audio Mode from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also

`AP.Bits.ChAStatusXferToString`

Example

```
Sub Main
    AP.Application.NewTest 'Reset panels
    AP.S2Dio.InFormat = 3
    Wait .3 'Wait for reading to update
    Debug.Print "Audio Mode = " & AP.Bits.ChAAudioModeRdg
End Sub
```

Example Output Audio Mode = 0

AP.Bits.ChAAuxBitsRdg

Property

Syntax `AP.Bits.ChAAuxBitsRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters

Part	Description
<i>String</i>	Optional string containing status bit information.

Result

Value	Description
0	20-bit not defined
1	24-bit not defined
2	20-bit single
3	Reserved

Description

This command returns the Status Bits channel A Auxiliary Bits from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also

`AP.Bits.ChAStatusXferToString`

Example

```
Sub Main
    Dim String_Array(3)
    String_Array(0) = "20-bit not defined"
    String_Array(1) = "24-bit main audio"
    String_Array(2) = "20-bit single"
    String_Array(3) = "Reserved"

    AP.Application.NewTest 'Reset panels
    AP.Bits.Mode = 1      'Professional Mode
    AP.S2Dio.InFormat = 3
    Wait .5 'Wait for reading to update
    Debug.Print "Auxiliary Bits Reading = " & _
        String_Array(AP.Bits.ChAAuxBitsRdg)
End Sub
```

Example Output Auxiliary Bits Reading = 20-bit not defined

AP.Bits.ChACategoryRdg**Property**

Syntax `AP.Bits.ChACategoryRdg([Optional ByVal String As Variant])`

Data Type Integer

Part	Description
<i>String</i>	Optional string containing status bit information.

Value	Description
0	General
1	CD Player
2	PCM Adaptor
3	DAT Recorder
4	Digital Broadcast
5	Musical Instrument

Description This command returns the Status Bits channel A Category code from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

Example

```
Sub Main
    Dim String_Array(5)
    String_Array(0) = "General"
    String_Array(1) = "CD Player"
    String_Array(2) = "PCM Adaptor"
    String_Array(3) = "DAT Recorder"
    String_Array(4) = "Digital Broadcast"
    String_Array(5) = "Musical Instrument"

    AP.Application.NewTest 'Reset panels
    AP.S2Dio.InFormat = 3
    Wait .3 'Wait for reading to update
    Debug.Print "Category Reading = " & _
```

```

        'String_Array(AP.Bits.ChACategoryRdg)
    End Sub

```

Example Output Category Reading = General

AP.Bits.ChAChModeRdg

Property

Syntax `AP.Bits.ChAChModeRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters

Part	Description
<i>String</i>	Optional string containing status bit information.

Result

Value	Description
0	Not Indicated
1	2-channel
2	Single-channel
3	Primary/Sec
4	Stereo
5	Reserved-1
6	Reserved-2
7	Vector to byte 3
8	Mono Double Rate
9	Left Double Rate
10	Right Double Rate

Description This command returns the Status Bits channel A Channel Mode from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

Example

```

Sub Main
    Dim String_Array(7)

```

```
String_Array(0) = "Not Indicated"
String_Array(1) = "2-channel"
String_Array(2) = "Single-channel"
String_Array(3) = "Primary/Sec"
String_Array(4) = "Stereo"
String_Array(5) = "Reserved-1"
String_Array(6) = "Reserved-2"
String_Array(7) = "Vector to byte 3"
String_Array(8) = " Mono Double Rate "
String_Array(9) = " Left Double Rate "
String_Array(10) = " Right Double Rate "
```

```
AP.Application.NewTest 'Reset panels
AP.Bits.Mode = 1      'Professional Mode
AP.S2Dio.InFormat = 3
Wait .3 'Wait for reading to update
Debug.Print "Channel Mode Reading = " & _
    String_Array(AP.Bits.ChAChModeRdg)
End Sub
```

Example Output Channel Mode Reading = Not Indicated

AP.Bits.ChAChNumRdg

Property

Syntax `AP.Bits.ChAChNumRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters

Part	Description
<i>String</i>	Optional string containing status bit information.

Result

Value	Description
0	Don't Care
1	A (Left)
2	B (Right)
3	C
4	D

5	E
6	F
7	G
8	H
9	I
10	J
11	K
12	L
13	M
14	N
15	O

Description

This command returns the Status Bits channel A Channel Number from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also

`AP.Bits.ChAStatusXferToString`

Example

```
Sub Main
  Dim String_Array(15)
  String_Array(0) = "Don't Care"
  String_Array(1) = "A (Left)"
  String_Array(2) = "B (Right)"
  String_Array(3) = "C"
  String_Array(4) = "D"
  String_Array(5) = "E"
  String_Array(6) = "F":
  String_Array(7) = "G"
  String_Array(8) = "H"
  String_Array(9) = "I"
  String_Array(10) = "J"
  String_Array(11) = "K"
  String_Array(12) = "L"
  String_Array(13) = "M"
  String_Array(14) = "N"
  String_Array(15) = "O"

  AP.Application.NewTest 'Reset panels
```

```

AP.Bits.Mode = 0      'Consumer Mode
AP.S2Dio.InFormat = 3
Wait .5 'Wait for reading to update
Debug.Print "Channel Number Reading = " & _
    String_Array(AP.Bits.ChAChNumRdg)
End Sub

```

Example Output Channel Number Reading = Don't Care

AP.Bits.ChAClockAccuracyRdg

Property

Syntax `AP.Bits.ChAClockAccuracyRdg([Optional ByVal
String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	Level 1
	1	Level 2
	2	Level 3
	3	Reserved

Description This command returns the Status Bits channel A Clock Accuracy from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

Example

```

Sub Main
    Dim String_Array(3)
    String_Array(0) = "Level 1"
    String_Array(1) = "Level 2"
    String_Array(2) = "Level 3"
    String_Array(3) = "Reserved"

```

```

AP.Application.NewTest 'Reset panels
AP.Bits.Mode = 0      'Consumer Mode
AP.S2Dio.InFormat = 3
Wait .5 'Wait for reading to update
Debug.Print "Clock Accuracy Reading = " & _
    String_Array(AP.Bits.ChAClockAccuracyRdg)
End Sub

```

Example Output Clock Accuracy Reading = Level 2

AP.Bits.ChACopyrightRdg

Property

Syntax `AP.Bits.ChACopyrightRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	Copyright
	1	Non-Copyright

Description This command returns the Status Bits channel A Copyright status from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

Example

```

Sub Main
    Dim String_Array(1)
    String_Array(0) = "Copyright"
    String_Array(1) = "Non-Copyright"

```

```

AP.Application.NewTest 'Reset panels
AP.Bits.Mode = 0      'Consumer Mode

```

```

AP.S2Dio.InFormat = 3
Wait .5 'Wait for reading to update
Debug.Print "Copyright Reading = " & _
    String_Array(AP.Bits.ChACopyrightRdg) & _
    " protected."
End Sub

```

Example Output Copyright Reading = Copyright protected.

AP.Bits.ChACrcRdg

Property

Syntax `AP.Bits.ChACrcRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters

Part	Description
<i>String</i>	Optional string containing status bit information.

Result

Value	Description
0	Invalid
1	Valid

Description

This command returns the Status Bits channel A CRC state from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also

`AP.Bits.ChAStatusXferToString`

Example

```

Sub Main
    Dim String_Array(1)
    String_Array(0) = "Clear"
    String_Array(1) = "Set"
    System = AP.App.SysType
    AP.App.NewTest
    AP.Bits.Mode = 1 'Professional Mode

```



```

AP.S2Dio.InFormat = 3
Wait .5 'Wait for reading to update
    Debug.Print "Crc Valid Reading = " & _
        String_Array(AP.Bits.ChACrcRdg)
End Sub

```

AP.Bits.ChADestinationRdg

Property

Syntax `AP.Bits.ChADestinationRdg([Optional ByVal String As Variant])`

Data Type String

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Description This command returns the Status Bits channel A Destination Code from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

```

Example
Sub Main
    AP.Application.NewTest 'Reset panels
    AP.Bits.Mode = 1      'Professional Mode
    AP.S2Dio.InFormat = 3
    AP.Bits.Pro.Destination = "ABCD"
    Wait .5 'Wait for reading to update
    Debug.Print "Destination Reading = " & _
        AP.Bits.ChADestinationRdg
End Sub

```

Example Output Destination Reading = ABCD

AP.Bits.ChAEmphRdg**Property**

Syntax `AP.Bits.ChAEmphRdg([Optional ByVal String As Variant])`

Data Type Integer

Part	Description
<i>String</i>	Optional string containing status bit information.

Result

Value	Description
The following list is for Consumer Mode.	
0	No Pre-emph
1	50/15S
The following list is for Professional Mode.	
0	Not Indicated
1	None
2	50/15 uS
3	CCITT J.17

Description This command returns the Status Bits channel A Emphasis setting from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`, `AP.Bits.ChAModeRdg`

Example

```
Sub Main
    Dim String_Array_Cons(1)
    Dim String_Array_Pro(3)
    String_Array_Cons(0) = "No Pre-emph"
    String_Array_Cons(1) = "50/15S"

    String_Array_Pro(0) = "Not Indicated"
    String_Array_Pro(1) = "None"
```

```

String_Array_Pro(2) = "50/15S"
String_Array_Pro(3) = "CCITT J.17"

AP.Application.NewTest 'Reset panels
With AP.Bits
    .XmitChannel = 0
    .Mode = 0      'Consumer Mode
    .Cons.Emphasis = 1
    .XmitChannel = 1
    .Mode = 1      'Professional Mode
    .Pro.Emphasis = 3
End With

AP.S2Dio.InFormat = 3
Wait 1 'Wait for reading to update
With AP.Bits
    If .ChAModeRdg = 0 Then
        Debug.Print "Ch A Consumer Emphasis Reading = " _
            & String_Array_Cons(.ChAEmphRdg)
    Else
        Debug.Print "Ch A Professional Emphasis _
            Reading = " & String_Array_Pro(.ChAEmphRdg)
    End If
End With
End Sub

```

Example Output Ch A Consumer Emphasis Reading = 50/15S
 Ch B Professional Emphasis Reading = CCITT J.17

AP.Bits.ChAFlag0_5Rdg

Property

Syntax `AP.Bits.ChAFlag0_5Rdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters

Part	Description
<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	Clear
	1	Set
Description	This command returns the Status Bits channel A Flag 0-5 state from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the <code>AP.Bits.ChAStatusXferToString</code> command.	
See Also	<code>AP.Bits.ChAStatusXferToString</code>	
Example	<pre> Sub Main Dim String_Array(1) String_Array(0) = "Cleared" String_Array(1) = "Set" AP.Application.NewTest 'Reset panels AP.Bits.Mode = 1 'Professional Mode AP.S2Dio.InFormat = 3 AP.Bits.Pro.Flag14_17 = 1 'Set flags 14-17 AP.Bits.Pro.Flag18_21 = 1 'Set Flags 18-21 Wait .5 'Wait for reading to update Debug.Print "Reliability Flags 0-5 Reading = " & _ String_Array(AP.Bits.ChAFlag0_5Rdg) Debug.Print "Reliability Flags 6-13 Reading = " & _ String_Array(AP.Bits.ChAFlag6_13Rdg) Debug.Print "Reliability Flags 14-17 Reading = " & _ String_Array(AP.Bits.ChAFlag14_17Rdg) Debug.Print "Reliability Flags 18-21 Reading = " & _ String_Array(AP.Bits.ChAFlag18_21Rdg) End Sub </pre>	
Example Output	<pre> Reliability Flags 0-5 Reading = Cleared Reliability Flags 6-13 Reading = Cleared Reliability Flags 14-17 Reading = Set Reliability Flags 18-21 Reading = Set </pre>	

AP.Bits.ChAFlag6_13Rdg**Property**

Syntax `AP.Bits.ChAFlag6_13Rdg([Optional ByVal String As Variant])`

Data Type Integer

Part	Description
<i>String</i>	Optional string containing status bit information.

Value	Description
0	Clear
1	Set

Description This command returns the Status Bits channel A Flag 6-13 state from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

Example See example for `AP.Bits.ChAFlag0_5Rdg`.

AP.Bits.ChAFlag14_17Rdg**Property**

Syntax `AP.Bits.ChAFlag14_17Rdg([Optional ByVal String As Variant])`

Data Type Integer

Part	Description
<i>String</i>	Optional string containing status bit information.

Value	Description
0	Clear
1	Set

Description This command returns the Status Bits channel A Flag 14-17 state from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

Example See example for `AP.Bits.ChAFlag0_5Rdg`.

AP.Bits.ChAFlag18_21Rdg

Property

Syntax `AP.Bits.ChAFlag18_21Rdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	Clear
	1	Set

Description This command returns the Status Bits channel A Flag 18-21 state from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

Example See example for `AP.Bits.ChAFlag0_5Rdg`.

AP.Bits.ChAFreqModeRdg**Property**

Syntax `AP.Bits.ChAFreqModeRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters

Part	Description
<i>String</i>	Optional string containing status bit information.

Result

Value	Description
0	Unlocked
1	Locked

Description

This command returns the Status Bits channel A Frequency Mode from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also

`AP.Bits.ChAStatusXferToString`

Example

```
Sub Main
    Dim String_Array(3)
    String_Array(0) = "Unlocked"
    String_Array(1) = "Locked"

    AP.Application.NewTest 'Reset panels
    AP.Bits.Mode = 1      'Professional Mode
    AP.S2Dio.InFormat = 3
    Wait 1 'Wait for reading to update
    Debug.Print "Frequency Mode Reading = " & _
        String_Array(AP.Bits.ChAFreqModeRdg)
End Sub
```

Example Output Frequency Mode Reading = Locked

AP.Bits.ChALocalAddressRdg Property

Syntax `AP.Bits.ChALocalAddressRdg([Optional ByVal String As Variant])`

Data Type Long

Part	Description
<i>String</i>	Optional string containing status bit information.

Description This command returns the Status Bits channel A Local Address code from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

Example

```
Sub Main
    AP.Application.NewTest
    AP.Bits.Mode = 1      'Professional Mode
    AP.S2Dio.InFormat = 3
    AP.Bits.XmitChannel = 0
    AP.Bits.Pro.LocalAddress = False
    AP.Bits.Pro.LocalAddress = 1234      'Set Ch A
    AP.Bits.XmitChannel = 1
    AP.Bits.Pro.LocalAddress = 5678      'Set Ch B
    Wait .5 'Wait for reading to update
    Debug.Print "Ch A Origin Reading = " & _
        AP.Bits.ChALocalAddressRdg
    Debug.Print "Ch B Local Address Reading = " & _
        AP.Bits.ChBLocalAddressRdg
End Sub
```

AP.Bits.ChAModeRdg Property

Syntax `AP.Bits.ChAModeRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.
Result	Value	Description
	0	Consumer
	1	Professional
Description	This command returns the Status Bits channel A Mode from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the <code>AP.Bits.ChAStatusXferToString</code> command.	

See Also `AP.Bits.ChAStatusXferToString`

Example

```
Sub Main
    AP.Application.NewTest 'Reset panels
    AP.S2Dio.InFormat = 3
    AP.Bits.Mode = 0
    For LoopNum = 1 To 5 Step 1
        Debug.Print "Mode = " & AP.Bits.ChAModeRdg()
        If LoopNum = 3 Then
            AP.Bits.Mode = 1
            Wait .5
        End If
    Next LoopNum
End Sub
```

Example Output

```
Mode = 0
Mode = 0
Mode = 0
Mode = 1
Mode = 1
```

AP.Bits.ChAOriginRdg

Property

Syntax `AP.Bits.ChAOriginRdg([Optional ByVal String As Variant])`

Data Type String

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Description This command returns the Status Bits channel A Origin Code from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

Example

Sub Main

```

AP.Application.NewTest 'Reset panels
AP.Bits.Mode = 1      'Professional Mode
AP.S2Dio.InFormat = 3
AP.Bits.XmitChannel = 0
AP.Bits.Pro.Origin = "ABCD"      'Set Ch A
AP.Bits.XmitChannel = 1
AP.Bits.Pro.Origin = "1234"     'Set Ch B
Wait .5 'Wait for reading to update
Debug.Print "Ch A Origin Reading = " & _
    AP.Bits.ChAOriginRdg
Debug.Print "Ch B Origin Reading = " & _
    AP.Bits.ChBOriginRdg

```

End Sub

Example Output Ch A Origin Reading = ABCD
Ch B Origin Reading = 1234

AP.Bits.ChARefSignalRdg

Property

Syntax `AP.Bits.ChARefSignalRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	Not a ref. Signal
	1	Grade 1
	2	Grade 2
	3	Reserved

Description This command returns the Status Bits channel A Reference Signal setting from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

Example

```
Sub Main
    Dim String_Array(3)
    String_Array(0) = "Not a ref. Signal"
    String_Array(1) = "Grade 1"
    String_Array(2) = "Grade 2"
    String_Array(3) = "Reserved"

    AP.Application.NewTest 'Reset panels
    AP.Bits.Mode = 1      'Professional Mode
    AP.S2Dio.InFormat = 3
    Wait .5 'Wait for reading to update
    Debug.Print "Reference Signal Reading = " & _
        String_Array(AP.Bits.ChARefSignalRdg)
End Sub
```

Example Output Reference Signal Reading = Not a ref. Signal

AP.Bits.ChASampleFreqRdg

Property

Syntax `AP.Bits.ChASampleFreqRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result**Value** **Description**

The following list is for Consumer Mode.

0	48 kHz
1	44.1 kHz
2	32 kHz

The following list is for Professional Mode.

0	Not Indicated
1	48 kHz
2	44.1 kHz
3	32 kHz

Description

This command returns the Status Bits channel A Sample Frequency from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also

`AP.Bits.ChAStatusXferToString`

Example

```
Sub Main
  Dim String_Array_Cons(2)
  Dim String_Array_Pro(3)
  String_Array_Cons(0) = "48 kHz"
  String_Array_Cons(1) = "44.1 kHz"
  String_Array_Cons(2) = "32 kHz"

  String_Array_Pro(0) = "Not Indicated"
  String_Array_Pro(1) = "48 kHz"
  String_Array_Pro(2) = "44.1 kHz"
  String_Array_Pro(3) = "32 kHz"

  AP.Application.NewTest 'Reset panels

  With AP.Bits
    .XmitChannel = 0
    .Mode = 0           'Consumer Mode
```

```

        .Cons.SampleFreq = 0
        .XmitChannel = 1
        .Mode = 1           'Professional Mode
        .Pro.SampleFreq = 2
    End With

    AP.S2Dio.InFormat = 3
    Wait 1 'Wait for reading to update
    With AP.Bits
        If .ChAModeRdg = 0 Then
            Debug.Print "Ch A Consumer Frequency Reading _
                = " & String_Array_Cons(.ChASampleFreqRdg)
        Else
            Debug.Print "Ch A Professional Frequency _
                Reading = " & String_Array_Pro _
                (.ChASampleFreqRdg)
        End If

        If .ChBModeRdg = 0 Then
            Debug.Print "Ch B Consumer Frequency _
                Reading = " & String_Array_Cons _
                (.ChBSampleFreqRdg)
        End If
    End With
End Sub

```

Example Output Ch A Consumer Frequency Reading = 48 kHz
 Ch B Professional Frequency Reading = 44.1 kHz

AP.Bits.ChASourceNumRdg

Property

Syntax `AP.Bits.ChASourceNumRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	Don't Care
	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9
	10	10
	11	11
	12	12
	13	13
	14	14
	15	15
	16	16

Description This command returns the Status Bits channel A Source Number from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

Example

```

Sub Main
  Dim String_Array(0)
  String_Array(0)= "Don't Care"

  AP.Application.NewTest 'Reset panels
  AP.Bits.Mode = 0      'Consumer Mode
  AP.S2Dio.InFormat = 3
  AP.Bits.Cons.SourceNum = 5
  Wait .5 'Wait for reading to update

```

```

If AP.Bits.ChASourceNumRdg = 0 Then
    Debug.Print "Source Number Reading = " & _
        String_Array(AP.Bits.ChASourceNumRdg)
Else
    Debug.Print "Source Number Reading = " & _
        AP.Bits.ChASourceNumRdg
End If
End Sub

```

Example Output Source Number Reading = 5

AP.Bits.ChAStatusXferToString

Method

Syntax AP.Bits.ChAStatusXferToString

Result String

Description This command transfers the contents of the channel A Status Bits into a string. This enables the programmer to extract all of the status information from a single measurement.

See Also AP.Bits.ChAXmitData

Example

```

Sub Main
    With AP.Bits
        Channel_A_Status = .ChAStatusXferToString
        Mode = .ChAModeRdg(Channel_A_Status)
        Debug.Print "Mode = " & Mode

        If Mode = 0 Then 'Consumer
            Debug.Print "Audio Mode = " & _
                .ChAAudioModeRdg(Channel_A_Status)
            Debug.Print "Copyright = " & _
                .ChACopyrightRdg(Channel_A_Status)
            Debug.Print "Emphasis = " & _
                .ChAEmphRdg(Channel_A_Status)
            Debug.Print "Channel Mode = " & _
                .ChAChModeRdg(Channel_A_Status)
            Debug.Print "Category Code = " & _
                .ChACategoryRdg(Channel_A_Status)
        End If
    End With
End Sub

```

```

Debug.Print "Source Number = " & _
    .ChASourceNumRdg(Channel_A_Status)
Debug.Print "Channel Number = " & _
    .ChAChNumRdg(Channel_A_Status)
Debug.Print "Sample Frequency = " & _
    .ChASampleFreqRdg(Channel_A_Status)
Debug.Print "Clock Accuracy = " & _
    .ChAClockAccuracyRdg(Channel_A_Status)
Else 'Professional
Debug.Print "Audio Mode = " & _
    .ChAAudioModeRdg(Channel_A_Status)
Debug.Print "Emphasis = " & _
    .ChAEmphRdg(Channel_A_Status)
Debug.Print "Frequency Mode = " & _
    .ChAFreqModeRdg(Channel_A_Status)
Debug.Print "Sample Frequency = " & _
    .ChASampleFreqRdg(Channel_A_Status)
Debug.Print "Channel Mode = " & _
    .ChAChModeRdg(Channel_A_Status)
Debug.Print "User Bits = " & _
    .ChAUserBitsRdg(Channel_A_Status)
Debug.Print "Aux Bits = " & _
    .ChAAuxBitsRdg(Channel_A_Status)
Debug.Print "Word Length = " & _
    .ChAWordLengthRdg(Channel_A_Status)
Debug.Print "Ref Signal = " & _
    .ChARefSignalRdg(Channel_A_Status)
Debug.Print "Origin Code = " & _
    .ChAOriginRdg(Channel_A_Status)
Debug.Print "Destination Code = " & _
    .ChADestinationRdg(Channel_A_Status)
Debug.Print "Local Address = " & _
    .ChALocalAddressRdg(Channel_A_Status)
Debug.Print "Time Of Day = " & _
    .ChATimeOfDayRdg(Channel_A_Status)
Debug.Print "Flag 0-5 = " & _
    .ChAFlag0_5Rdg(Channel_A_Status)
Debug.Print "Flag 6-13 = " & _
    .ChAFlag6_13Rdg(Channel_A_Status)
Debug.Print "Flag 14-17 = " & _
    .ChAFlag14_17Rdg(Channel_A_Status)

```



```

        Debug.Print "Flag 18-21 = " & _
            .ChAFlag18_21Rdg(Channel_A_Status)
        Debug.Print "Crc Valid = " & _
            .ChACrcRdg(Channel_A_Status)
    End If
End With
End Sub

```

AP.Bits.ChATimeOfDayRdg

Property

Syntax `AP.Bits.ChATimeOfDayRdg([Optional ByVal String As Variant])`

Data Type Long

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Description This command returns the Status Bits channel A Time Of Day code from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

Example

```

Sub Main
    AP.Application.NewTest 'Reset panels
    AP.Bits.Mode = 1      'Professional Mode
    AP.Bits.Pro.LocalAddressAuto = 0
    AP.Bits.Pro.TimeOfDay = 123456789
    AP.S2Dio.InFormat = 3
    Wait .5 'Wait for reading to update
    Debug.Print "Ch A Time Of Day Reading = " & _
        AP.Bits.ChATimeOfDayRdg
    Debug.Print "Ch B Time Of Day Reading = " & _
        AP.Bits.ChBTimeOfDayRdg
End Sub

```

Example Output

```

Ch A Time Of Day Reading = 123456789
Ch B Time Of Day Reading = 123456789

```

AP.Bits.ChAUserBitsRdg**Property**

Syntax `AP.Bits.ChAUserBitsRdg([Optional ByVal String As Variant])`

Data Type Integer

Part	Description
<i>String</i>	Optional string containing status bit information.

Value	Description
0	None
1	192-bit block
2	Reserved
3	User defined

Description This command returns the Status Bits channel A User Bits from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also `AP.Bits.ChAStatusXferToString`

Example

```
Sub Main
    Dim String_Array(3)
    String_Array(0) = "None"
    String_Array(1) = "192-bit block"
    String_Array(2) = "Reserved"
    String_Array(3) = "User defined"

    AP.Application.NewTest 'Reset panels
    AP.Bits.Mode = 1      'Professional Mode
    AP.Bits.Pro.UserBits = 1
    AP.S2Dio.InFormat = 3
    Wait .5 'Wait for reading to update
    Debug.Print "User Bits Reading = " & _
        String_Array(AP.Bits.ChAUserBitsRdg)
End Sub
```

Example Output User Bits Reading = 192-bit block

AP.Bits.ChAWordLengthRdg**Property**

Syntax `AP.Bits.ChAWordLengthRdg([Optional ByVal String
As Variant])`

Data Type Integer

Parameters

Part	Description
<i>String</i>	Optional string containing status bit information.

Result

Value	Description
0	Not Indicated
1	20 bits
2	19 bits
3	18 bits
4	17 bits
5	16 bits

Description

This command returns the Status Bits channel A Word Length from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChAStatusXferToString` command.

See Also

`AP.Bits.ChAStatusXferToString`

Example

```
Sub Main
    Dim String_Array(5)
    String_Array(0) = "Not Indicated"
    String_Array(1) = "20 bits"
    String_Array(2) = "19 bits"
    String_Array(3) = "18 bits"
    String_Array(4) = "17 bits"
    String_Array(5) = "16 bits"

    AP.Application.NewTest 'Reset panels
    AP.Bits.Mode = 1      'Professional Mode
    AP.S2Dio.InFormat = 3
    Wait .5 'Wait for reading to update
```

```

        Debug.Print "Word Length Reading = " & _
            String_Array(AP.Bits.ChAWordLengthRdg)
    End Sub

```

Example Output Word Length Reading = Not Indicated

AP.Bits.ChAXmitStatus

Property

Syntax	<code>AP.Bits.ChAXmitStatus</code>
Data Type	String String containing status bit information.
Description	This command transmits the status bits data contained in the string for channel A.
See Also	<code>AP.Bits.ChAStatusXferToString</code>
Example	<pre> Sub Main With AP.Bits 'Get current Channel A&B status Channel_A_Status = .ChAStatusXferToString Channel_B_Status = .ChBStatusXferToString 'Your code goes here 'Restore Channel A&B status .ChAXmitStatus = Channel_A_Status .ChBXmitStatus = Channel_B_Status End With End Sub </pre>

AP.Bits.ChBAudioModeRdg

Property

Syntax	<code>AP.Bits.ChBAudioModeRdg([Optional ByVal <i>String</i> As Variant])</code>	
Data Type	Integer	
Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	1	Normal
	0	Non Audio
Description	This command returns the Status Bits channel B Audio Mode from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the <code>AP.Bits.ChBStatusXferToString</code> command.	
See Also	<code>AP.Bits.ChBStatusXferToString</code>	
Example	See example for <code>AP.Bits.ChAAudioModeRdg</code> .	

AP.Bits.ChBAuxBitsRdg

Property

Syntax	<code>AP.Bits.ChBAuxBitsRdg([Optional ByVal <i>String</i> As Variant])</code>	
Data Type	Integer	
Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.
Result	Value	Description
	0	20-bit not defined
	1	24-bit not defined
	2	20-bit single
	3	Reserved
Description	This command returns the Status Bits channel B Auxiliary Bits from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the <code>AP.Bits.ChBStatusXferToString</code> command.	
See Also	<code>AP.Bits.ChBStatusXferToString</code>	

Example See example for `AP.Bits.ChAAuxBitsRdg`.

AP.Bits.ChBCategoryRdg

Property

Syntax `AP.Bits.ChBCategoryRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters

Part	Description
<i>String</i>	Optional string containing status bit information.

Result

Value	Description
0	General
1	CD Player
2	PCM Adaptor
3	DAT Recorder
4	Digital Broadcast
5	Musical Instrument

Description

This command returns the Status Bits channel B Category code from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChACategoryRdg`.

AP.Bits.ChBChModeRdg

Property

Syntax `AP.Bits.ChBChModeRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	Not Indicated
	1	2-channel
	2	Single-channel
	3	Primary/Sec
	4	Stereo
	5	Reserved-1
	6	Reserved-2
	7	Vector to byte 3
	8	Mono Double Rate
	9	Left Double Rate
	10	Right Double Rate

Description This command returns the Status Bits channel B Channel Mode from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChAChModeRdg`.

AP.Bits.ChBChNumRdg

Property

Syntax	<code>AP.Bits.ChBChNumRdg([Optional ByVal <i>String</i> As Variant])</code>				
Data Type	Integer				
Parameters	<table border="1"> <thead> <tr> <th>Part</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>String</i></td> <td>Optional string containing status bit information.</td> </tr> </tbody> </table>	Part	Description	<i>String</i>	Optional string containing status bit information.
Part	Description				
<i>String</i>	Optional string containing status bit information.				

Result	Value	Description
	0	Don't Care
	1	A (Left)
	2	B (Right)
	3	C
	4	D
	5	E
	6	F
	7	G
	8	H
	9	I
	10	J
	11	K
	12	L
	13	M
	14	N
	15	O

Description This command returns the Status Bits channel B Channel Number from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChAChNumRdg`.

AP.Bits.ChBClockAccuracyRdg

Property

Syntax `AP.Bits.ChBClockAccuracyRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	Level 1
	1	Level 2
	2	Level 3
	3	Reserved

Description This command returns the Status Bits channel B Clock Accuracy setting from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChAClockAccuracyRdg`.

AP.Bits.ChBCopyrightRdg

Property

Syntax	<code>AP.Bits.ChBCopyrightRdg([Optional ByVal <i>String</i> As Variant])</code>	
Data Type	Integer	
Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.
Result	Value	Description
	1	Copyright
	0	Non-Copyright
Description	This command returns the Status Bits channel B Copyright state from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the <code>AP.Bits.ChBStatusXferToString</code> command.	
See Also	<code>AP.Bits.ChBStatusXferToString</code>	
Example	See example for <code>AP.Bits.ChACopyrightRdg</code> .	

AP.Bits.ChBCrcRdg**Property**

Syntax `AP.Bits.ChBCrcRdg([Optional ByVal String As Variant])`

Data Type Integer

Part	Description
<i>String</i>	Optional string containing status bit information.

Value	Description
0	Invalid
1	Valid

Description This command returns the Status Bits channel B CRC state from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChACrcRdg`.

AP.Bits.ChBDestinationRdg**Property**

Syntax `AP.Bits.ChBDestinationRdg([Optional ByVal String As Variant])`

Data Type String

Part	Description
<i>String</i>	Optional string containing status bit information.

Description This command returns the Status Bits channel B Destination code from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChADestinationRdg`.

AP.Bits.ChBEmphRdg

Property

Syntax `AP.Bits.ChBEmphRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters

Part	Description
<i>String</i>	Optional string containing status bit information.

Result

Value	Description
-------	-------------

The following list is for Consumer Mode.

0	No Pre-emph
1	50/15S

The following list is for Professional Mode.

0	Not Indicated
1	None
2	50/15 uS
3	CCITT J.17

Description This command returns the Status Bits channel B Emphasis setting from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChAEmphRdg`.

AP.Bits.ChBFlag0_5Rdg

Property

Syntax `AP.Bits.ChBFlag0_5Rdg([Optional ByVal String As Variant])`

Data Type Integer

Part	Description
<i>String</i>	Optional string containing status bit information.

Value	Description
0	Clear
1	Set

Description This command returns the Status Bits channel B Flag 0-5 state from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChAFlag0_5Rdg`.

AP.Bits.ChBFlag6_13Rdg

Property

Syntax `AP.Bits.ChBFlag6_13Rdg([Optional ByVal String As Variant])`

Data Type Integer

Part	Description
<i>String</i>	Optional string containing status bit information.

Value	Description
0	Clear
1	Set

Description	This command returns the Status Bits channel B Flag 6-13 state from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the <code>AP.Bits.ChBStatusXferToString</code> command.
See Also	<code>AP.Bits.ChBStatusXferToString</code>
Example	See example for <code>AP.Bits.ChAFlag0_5Rdg</code> .

AP.Bits.ChBFlag14_17Rdg

Property

Syntax `AP.Bits.ChBFlag14_17Rdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	Clear
	1	Set

Description This command returns the Status Bits channel B Flag 14-17 state from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChAFlag0_5Rdg`.

AP.Bits.ChBFlag18_21Rdg

Property

Syntax `AP.Bits.ChBFlag18_21Rdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	Clear
	1	Set

Description This command returns the Status Bits channel B Flag 18-21 state from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChAFlag0_5Rdg`.

AP.Bits.ChBFreqModeRdg

Property

Syntax `AP.Bits.ChBFreqModeRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	Unlocked
	1	Locked

Description This command returns the Status Bits channel B Frequency Mode from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChAFreqModeRdg`.

AP.Bits.ChBLocalAddressRdg

Property

Syntax `AP.Bits.ChBLocalAddressRdg([Optional ByVal String As Variant])`

Data Type Long

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Description This command returns the Status Bits channel B Local Address code from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChALocalAddressRdg`.

AP.Bits.ChBModeRdg

Property

Syntax `AP.Bits.ChBModeRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	Consumer
	1	Professional

Description	This command returns the Status Bits channel B Mode from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the <code>AP.Bits.ChBStatusXferToString</code> command.
See Also	<code>AP.Bits.ChBStatusXferToString</code>
Example	See example for <code>AP.Bits.ChAModeRdg</code> .

AP.Bits.ChBOriginRdg

Property

Syntax	<code>AP.Bits.ChBOriginRdg([Optional ByVal <i>String</i> As Variant])</code>
Data Type	String

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Description	This command returns the Status Bits channel B Origin code from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the <code>AP.Bits.ChBStatusXferToString</code> command.
See Also	<code>AP.Bits.ChBStatusXferToString</code>
Example	See example for <code>AP.Bits.ChAOriginRdg</code> .

AP.Bits.ChBRefSignalRdg

Property

Syntax	<code>AP.Bits.ChBRefSignalRdg([Optional ByVal <i>String</i> As Variant])</code>	
Data Type	Integer	
Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	Nor a ref. Signal
	1	Grade 1
	2	Grade 2
	3	Reserved
Description	This command returns the Status Bits channel B Reference Signal setting from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the <code>AP.Bits.ChBStatusXferToString</code> command.	
See Also	<code>AP.Bits.ChBStatusXferToString</code>	
Example	See example for <code>AP.Bits.ChARefSignalRdg</code> .	

AP.Bits.ChBSampleFreqRdg

Property

Syntax `AP.Bits.ChBSampleFreqRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters	Part	Description
	<i>String</i>	Optional string containing status bit information.

Result	Value	Description
	0	48 kHz
	1	44.1 kHz
	2	32 kHz

Description This command returns the Status Bits channel B Sample Frequency from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChASampleFreqRdg`.

AP.Bits.ChBSourceNumRdg

Property

Syntax `AP.Bits.ChBSourceNumRdg([Optional ByVal String As Variant])`

Data Type Integer

Parameters

Part	Description
<i>String</i>	Optional string containing status bit information.

Result

Value	Description
0	Don't Care
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16

Description

This command returns the Status Bits channel B Source Number from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChASourceNumRdg`.

AP.Bits.ChBStatusXferToString

Method

Syntax `AP.Bits.ChBStatusXferToString`

Result String

Description This command transfers the contents of the channel B Status to an string.

See Also `AP.Bits.ChBXmitData`

Example See example for `AP.Bits.ChAStatusXferToString`.

AP.Bits.ChBTimeOfDayRdg

Property

Syntax `AP.Bits.ChBTimeOfDayRdg([Optional ByVal String As Variant])`

Data Type Long

Part	Description
<i>String</i>	Optional string containing status bit information.

Description This command returns the Status Bits channel B Time Of Day code from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChATimeOfDayRdg`.

AP.Bits.ChBUserBitsRdg**Property**

Syntax `AP.Bits.ChBUserBitsRdg([Optional ByVal String As Variant])`

Data Type Integer

Part	Description
<i>String</i>	Optional string containing status bit information.

Value	Description
0	None
1	192-bit block
2	Reserved
3	User defined

Description This command returns the Status Bits channel B User Bits from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChAUserBitsRdg`.

AP.Bits.ChBWordLengthRdg**Property**

Syntax `AP.Bits.ChBWordLengthRdg([Optional ByVal String As Variant])`

Data Type Integer

Part	Description
<i>String</i>	Optional string containing status bit information.

Value	Description
0	Not Indicated

1	20 bits
2	19 bits
3	18 bits
4	17 bits
5	16 bits

Description This command returns the Status Bits channel B Word Length setting from an optional string or from the AES/EBU data stream. When the optional string parameter is included the command uses the designated string as the source for the reading. The string is obtained by using the `AP.Bits.ChBStatusXferToString` command.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChAWordLengthRdg`.

AP.Bits.ChBXmitStatus

Property

Syntax `AP.Bits.ChBXmitStatus`

Data Type String String containing status bit information.

Description This command transmits the status bits data contained in the string for channel B.

See Also `AP.Bits.ChBStatusXferToString`

Example See example for `AP.Bits.ChAXmitStatus`.

AP.Bits.Cons.AudioMode

Property

Syntax `AP.Bits.Cons.AudioMode`

Data Type Integer

0	Audio Mode
1	Data Mode

Description This command sets the Mode parameter encoded in the Consumer Status Bits.

Example See example for AP.Bits.Cons.Category.

AP.Bits.Cons.Category

Property

Syntax AP.Bits.Cons.Category

Data Type Integer

0	General
1	CD Player
2	PCM Adaptor
3	DAT Recorder
4	Digital Broadcast
5	Musical Instrument

Description This command sets the Category Code parameter (channel status bit C) encoded in the Consumer Status Bits.

System One digital I/O units always sends the same status bits on Channels A and B.

Example

```
Sub Main
  'other setup code ...
  AP.Bits.XmitChannel = 2      'channels A & B
  AP.Bits.Mode = 0           'consumer
  AP.Bits.Cons.AudioMode = 1  'data mode
  AP.Bits.Cons.CopyRight = 1  'non-copyright
  AP.Bits.Cons.Emphasis = 1   '50/15 uS
  AP.Bits.Cons.Channels = 0   '2 channel
  AP.Bits.Cons.Category = 1   'CD player
  AP.Bits.Cons.SourceNum = 1  'source 1
  AP.Bits.Cons.ChNum = 1     'A (left)
  AP.Bits.Cons.SampleFreq = 0 '48 kHz
  AP.Bits.Cons.ClockAccuracy = 0 'level 1
  'rest of program ...
End Sub
```

AP.Bits.Cons.Channels**Property**

Syntax	<code>AP.Bits.Cons.Channels</code>				
Data Type	Integer				
	<table> <tr> <td><i>0</i></td> <td>2 Channel</td> </tr> <tr> <td><i>1</i></td> <td>4 Channel</td> </tr> </table>	<i>0</i>	2 Channel	<i>1</i>	4 Channel
<i>0</i>	2 Channel				
<i>1</i>	4 Channel				
Description	This command sets the Channel Mode parameter encoded in the Consumer Status Bits.				
Example	See example for <code>AP.Bits.Cons.Category</code> .				

AP.Bits.Cons.ChNum**Property**

Syntax	<code>AP.Bits.Cons.ChNum</code>																																
Data Type	Integer																																
	<table> <tr> <td><i>0</i></td> <td>Don't Care</td> </tr> <tr> <td><i>1</i></td> <td>A (Left)</td> </tr> <tr> <td><i>2</i></td> <td>B (Right)</td> </tr> <tr> <td><i>3</i></td> <td>C</td> </tr> <tr> <td><i>4</i></td> <td>D</td> </tr> <tr> <td><i>5</i></td> <td>E</td> </tr> <tr> <td><i>6</i></td> <td>F</td> </tr> <tr> <td><i>7</i></td> <td>G</td> </tr> <tr> <td><i>8</i></td> <td>H</td> </tr> <tr> <td><i>9</i></td> <td>I</td> </tr> <tr> <td><i>10</i></td> <td>J</td> </tr> <tr> <td><i>11</i></td> <td>K</td> </tr> <tr> <td><i>12</i></td> <td>L</td> </tr> <tr> <td><i>13</i></td> <td>M</td> </tr> <tr> <td><i>14</i></td> <td>N</td> </tr> <tr> <td><i>15</i></td> <td>O</td> </tr> </table>	<i>0</i>	Don't Care	<i>1</i>	A (Left)	<i>2</i>	B (Right)	<i>3</i>	C	<i>4</i>	D	<i>5</i>	E	<i>6</i>	F	<i>7</i>	G	<i>8</i>	H	<i>9</i>	I	<i>10</i>	J	<i>11</i>	K	<i>12</i>	L	<i>13</i>	M	<i>14</i>	N	<i>15</i>	O
<i>0</i>	Don't Care																																
<i>1</i>	A (Left)																																
<i>2</i>	B (Right)																																
<i>3</i>	C																																
<i>4</i>	D																																
<i>5</i>	E																																
<i>6</i>	F																																
<i>7</i>	G																																
<i>8</i>	H																																
<i>9</i>	I																																
<i>10</i>	J																																
<i>11</i>	K																																
<i>12</i>	L																																
<i>13</i>	M																																
<i>14</i>	N																																
<i>15</i>	O																																
Description	This command sets the Source Number parameter encoded in the Consumer Status Bits.																																

See Also `AP.Bits.XmitChannel`

Example See example for `AP.Bits.Cons.Category`.

AP.Bits.Cons.ClockAccuracy

Property

Syntax `AP.Bits.Cons.ClockAccuracy`

Data Type Integer

<i>0</i>	Level 1
<i>1</i>	Level 2
<i>2</i>	Level 3
<i>3</i>	Reserved

Description This command sets the Clock Accuracy parameter encoded in the Consumer Status Bits.

Example See example for `AP.Bits.Cons.Category`.

AP.Bits.Cons.Copyright

Property

Syntax `AP.Bits.Cons.Copyright`

Data Type Integer

<i>1</i>	Copyright
<i>0</i>	Non-Copyright

Description This command sets the Copyright parameter encoded in the Consumer Status Bits.

Example See example for `AP.Bits.Cons.Category`.

AP.Bits.Cons.Emphasis

Property

Syntax `AP.Bits.Cons.Emphasis`

Data Type Integer

0	No Pre-emph
1	50/15S

Description This command sets the Emphasis parameter encoded in the Consumer Status Bits.

Example See example for AP.Bits.Cons.Category.

AP.Bits.Cons.SampleFreq

Property

Syntax AP.Bits.Cons.SampleFreq

Data Type Integer

0	48 kHz
1	44.1 kHz
2	32 kHz

Description This command sets the Frequency parameter encoded in the Consumer Status Bits.

Example See example for AP.Bits.Cons.Category.

AP.Bits.Cons.SourceNum

Property

Syntax AP.Bits.Cons.SourceNum

Data Type Integer

0	Don't Care
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

10	10
11	11
12	12
13	13
14	14
15	15
16	16

Description This command sets the Source Number parameter encoded in the Consumer Status Bits.

Example See example for AP.Bits.Cons.Category.

AP.Bits.Mode

Property

Syntax AP.Bits.Mode

Data Type Integer

0	Consumer
1	Professional

Description This command sets the Transmit Mode.

Example See example for AP.Bits.Cons.Category.

AP.Bits.Pro.AudioMode

Property

Syntax AP.Bits.Pro.AudioMode

Data Type Integer

0	Normal
1	Non Audio

Description This command sets the Audio Mode parameter encoded in the Professional Status Bits.

Example See example for AP.Bits.Pro.LocalAddress.

AP.Bits.Pro.AuxBits**Property**

Syntax	AP.Bits.Pro.AuxBits	
Data Type	Integer	
	0	20-bit not defined
	1	24-bit main audio
	2	20-bit single
	3	Reserved
Description	This command sets the Aux Bits parameter encoded in the Professional Status Bits.	
Example	See example for <code>AP.Bits.Pro.LocalAddress</code> .	

AP.Bits.Pro.ChMode**Property**

Syntax	AP.Bits.Pro.ChMode	
Data Type	Integer	
	0	Not Indicated
	1	2-channel
	2	Single-channel
	3	Primary/Sec
	4	Stereo
	5	Reserved-1
	6	Reserved-2
	7	Vector to byte 3
	8	Mono Double Rate
	9	Left Double Rate
	10	Right Double Rate
Description	This command sets the Channel Mode parameter encoded in the Professional Status Bits.	
Example	See example for <code>AP.Bits.Pro.LocalAddress</code> .	

AP.Bits.Pro.CrcEnable

Property**Syntax** `AP.Bits.Pro.CrcEnable`**Result** Boolean

<i>True</i>	Set
<i>False</i>	Clear

Description This command sets or clears the CRC parameter encoded in the Professional Status Bits.

The AES3 standard defines byte 23 as a CRC byte to assist the receiver in detecting errors in the preceding 23 bytes (0-22) of each channel status block.

Example See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.Destination

Property**Syntax** `AP.Bits.Pro.Destination`**Data Type** String**Description** This command sets a four-character alphanumeric (ASCII) code to be transmitted.**Example** See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.Emphasis

Property**Syntax** `AP.Bits.Pro.Emphasis`**Result** Integer

0	Not Indicated
1	None
2	50/15 uS
3	CCITT J.17

Description This command sets the Emphasis parameter encoded in the Professional Status Bits.

Example See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.Flag0_5

Property

Syntax `AP.Bits.Pro.Flag0_5`

Result Boolean

<i>True</i>	Set
<i>False</i>	Clear

Description This command sets or clears the Reliability Flag for bytes 0-5.

This flag is to be set if useful information is not being transmitted in the corresponding status bytes.

Note that the Reliability Flags are not indications of the quality of the signal, but are simply a way for the transmitting device to tell the receiving device whether or not the information received in each group of six status bytes is valid.

Example See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.Flag6_13

Property

Syntax `AP.Bits.Pro.Flag6_13`

Result Boolean

<i>True</i>	Set
<i>False</i>	Clear

Description This command sets or clears the Reliability Flag for bytes 6-13.

This flag is to be set if useful information is not being transmitted in the corresponding status bytes.

Note that the Reliability Flags are not indications of the quality of the signal, but are simply a way for the transmitting device to tell the

receiving device whether or not the information received in each group of eight status bytes is valid.

Example See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.Flag14_17

Property

Syntax `AP.Bits.Pro.Flag14_17`

Result Boolean

True Set
False Clear

Description This command sets or clears the Reliability Flag for bytes 14-17.

This flag is to be set if useful information is not being transmitted in the corresponding status bytes.

Note that the Reliability Flags are not indications of the quality of the signal, but are simply a way for the transmitting device to tell the receiving device whether or not the information received in each group of four status bytes is valid.

Example See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.Flag18_21

Property

Syntax `AP.Bits.Pro.Flag18_21`

Result Boolean

True Set
False Clear

Description This command sets or clears the Reliability Flag for bytes 18-21.

This flag is to be set if useful information is not being transmitted in the corresponding status bytes.

Note that the Reliability Flags are not indications of the quality of the signal, but are simply a way for the transmitting device to tell the receiving device whether or not the information received in each group of four status bytes is valid.

Example See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.FreqMode

Property

Syntax `AP.Bits.Pro.FreqMode`

Data Type Integer

0	Unlocked
1	Locked

Description This command sets the Frequency Mode parameter encoded in the Professional Status Bits.

Example See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.LocalAddress

Property

Syntax `AP.Bits.Pro.LocalAddress`

Data Type Long

Description This command sets the Local Address parameter encoded in the Professional Status Bits bytes 14-17.

The Local Address is a timer function defined in the Professional standard only.

See Also `AP.Bits.Pro.AddressAuto`

Example

```
Sub Main
    'other setup code ...
    AP.Bits.XmitChannel = 1      'transmit chan B
    AP.Bits.Mode = 1           'professional
    AP.Bits.Pro.AudioMode = 1  'non-audio
    AP.Bits.Pro.Emphasis = 2   '50/15 uS
```

```

AP.Bits.Pro.FreqMode = 0      'unlocked
AP.Bits.Pro.SampleFreq = 1   '48 kHz
AP.Bits.Pro.ChMode = 4       'stereo
AP.Bits.Pro.UserBits = 3     'user defined
AP.Bits.Pro.AuxBits = 1      '24-bit main audio
AP.Bits.Pro.WordLength = 1   '24 bits
AP.Bits.Pro.RefSignal = 2    'grade 2
AP.Bits.Pro.Origin = "SYS2"  'source SYS2
AP.Bits.Pro.Destination = "TEST" 'target TEST
AP.Bits.Pro.LocalAddressAuto = 0 'auto address off
AP.Bits.Pro.LocalAddress = 123456 'set address = _
    123456
AP.Bits.Pro.TimeOfDay = 1234 'set TOD = 1234 samples
AP.Bits.Pro.Flag0_5 = True   'unreliable
AP.Bits.Pro.Flag6_13 = True  'unreliable
AP.Bits.Pro.Flag14_17 = False 'reliable
AP.Bits.Pro.Flag18_21 = True 'unreliable
AP.Bits.Pro.CrcEnable = True 'valid
'Rest of program
End Sub

```

AP.Bits.Pro.LocalAddressAuto

Property

Syntax `AP.Bits.Pro.LocalAddressAuto`

Data Type Boolean

True Enabled
False Disabled

Description This command enables or disables automatic selection of the Local Address and Time Of Day values.

If the Local Address Auto box via this command is enabled, both the Local Address value transmitted (bytes 14-17) and the Time of Day value (bytes 18-21) are the count, in samples, of the elapsed time since the Professional format of status bytes was selected or the Auto box was checked (whichever was later). If the Auto box is not checked, an entry field is displayed to the right of the Auto box. A number may be entered into this field via the `AP.Bits.Pro.LocalAddress`

command and the number will be continuously transmitted as the Local Address code in the status bytes.

See Also `AP.Bits.Pro.LocalAddress`, `AP.Bits.Pro.TimeOfDay`

Example See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.Origin

Property

Syntax `AP.Bits.Pro.Origin`

Data Type String

Description This command sets a four-character alphanumeric (ASCII) code to be transmitted.

Example See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.RefSignal

Property

Syntax `AP.Bits.Pro.RefSignal`

Data Type Integer

<i>0</i>	Nor a ref. Signal
<i>1</i>	Grade 1
<i>2</i>	Grade 2
<i>3</i>	Reserved

Description This command sets the ReferenceSignal parameter encoded in the Professional Status Bits.

Example See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.SampleFreq

Property

Syntax `AP.Bits.Pro.SampleFreq`

Data Type Integer

0	Not Indicated
1	48 kHz
2	44.1 kHz
3	32 kHz
4	192 kHz
5	192/1.001 kHz
6	176.4 kHz
7	176.4/1.001 kHz
8	96 kHz
9	96/1.001 kHz
10	88.2 kHz
11	88.2/1.001 kHz
12	48/1.001 kHz
13	44.1/1.001 kHz
14	32/1.001 kHz
15	24 kHz
16	24/1.001 kHz
17	22.05 kHz
18	22.05/1.001 kHz

Description This command sets the Frequency parameter encoded in the Professional Status Bits.

Example See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.TimeOfDay

Property

Syntax `AP.Bits.Pro.TimeOfDay`

Data Type Long
Description This command sets the Time Of Day parameter encoded in the Professional Status Bits bytes 18-21.

See Also `AP.Bits.Pro.AddressAuto`

Example See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.UserBits**Property****Syntax** `AP.Bits.Pro.UserBits`**Data Type**

Integer

<i>0</i>	None
<i>1</i>	192-bit block
<i>2</i>	Reserved
<i>3</i>	User defined

Description

This command sets the User Bits parameter encoded in the Professional Status Bits.

Example

See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.Pro.WordLength**Property****Syntax** `AP.Bits.Pro.WordLength`**Data Type**

Integer

The following list contains the selections relevant to the `AP.Bits.Pro.AuxBits` command "20-bit not defined" selection.

<i>0</i>	Not Indicated
<i>1</i>	20 bits
<i>2</i>	19 bits
<i>3</i>	18 bits
<i>4</i>	17 bits
<i>5</i>	16 bits

The following list contains the selections relevant to the `AP.Bits.Pro.AuxBits` command "24-bit main audio" selection.

<i>0</i>	Not Indicated
<i>1</i>	24 bits
<i>2</i>	23 bits
<i>3</i>	22 bits
<i>4</i>	21 bits
<i>5</i>	20 bits

Description This command sets the Audio Word Length parameter encoded in the Professional Status Bits.

Example See example for `AP.Bits.Pro.LocalAddress`.

AP.Bits.XmitChannel

Property

Syntax `AP.Bits.XmitChannel`

Data Type Integer

<i>0</i>	A
<i>1</i>	B
<i>2</i>	A & B

Description This command sets the Transmit Channel.

Example See example for `AP.Bits.Pro.LocalAddress`.

User Notes

AP.CommA.Break**Property****AP.CommB.Break****Property****Syntax** **AP.CommA.Break****Data Type** Boolean*True* Sets the break signal.*False* Clears the break signal.

Description This command sets or clears the break signal. Setting the break signal to True stops sending characters and places the line in a break state until the Break command is set to False.

AP.CommA.CD Holding**Property****AP.CommB.CD Holding****Property****Syntax** **AP.CommA.CD Holding****Data Type** Boolean*True* Carrier Detect line high.*False* Carrier Detect line low.

Description This command returns the state of the Carrier Detect (CD) line. The state of the Carrier Detect line indicates to the computer whether or not the modem is online.

When the Carrier Detect line is high (CD Holding = True) and the time specified by the AP.CommA.CDTimeout command has expired, the AP.CommA.CommEvent command is set to comCDTO (Carrier Detect Timeout Error), and a OnComm event is generated.

The Carrier Detect is also known as the Receive Line Signal Detect (RLSD).

See Also `AP.CommA.CDTimeout`

AP.CommA.CDTimeout

Property

AP.CommB.CDTimeout

Property

Syntax `AP.CommA.CDTimeout`

Data Type Long

Description This command sets and returns the maximum amount of time (in milliseconds) that the control waits for the Carrier Detect (CD) signal before timing out. This command indicates a timeout condition by setting the `AP.CommA.CommEvent` command to `CDTO` (Carrier Detect Timeout Error) and generating the `OnComm` event.

AP.CommA.CommEvent

Property

AP.CommB.CommEvent

Property

Syntax *Event = AP.CommA.CommEvent*

Data Type Integer

The following list contains communications errors or events.

Setting	Value	Description
<code>comBreak</code>	1001	Break signal received.
<code>comCTSTO</code>	1002	Clear To Send Timeout. The Clear To Send line was low for the number of milliseconds specified by the <code>AP.CommA.CTSTimeout</code> command while trying to send a character.
<code>comDSRTO</code>	1003	Data Set Ready Timeout. The Data Set Ready line was low for the number of milliseconds specified by the <code>AP.CommA.DSRTimeout</code> command while trying to send a character.
<code>comFrame</code>	1004	Framing Error. The hardware detected a framing error.

<i>comOverrun</i>	1006	Port Overrun. A character was not read from the hardware before the next character arrived and was lost.
<i>comCDTO</i>	1007	Carrier Detect Timeout. The Carrier Detect line was low for the number of milliseconds specified by the AP.CommA.CDTimeout command while trying to send a character.
<i>comRxOver</i>	1008	Receive Buffer Overflow. The receive buffer is full.
<i>comRxParity</i>	1009	Parity Error. Parity error detected.
<i>comTxFull</i>	1010	Transmit Buffer Full. The transmit buffer was full while trying to queue a character.

Communications events include the following settings.

Setting	Value	Description
<i>comEvSend</i>	1	There are fewer than SThreshold number of characters in the transmit buffer.
<i>comEvReceive</i>	2	Received RThreshold number of characters. This event is generated continuously until you use the Input property to remove the data from the receive buffer.
<i>comEvCTS</i>	3	Change in Clear To Send line.
<i>comEvDSR</i>	4	Change in Data Set Ready line. This event is only fired when DSR changes from 1 to 0.
<i>comEvDC</i>	5	Change in Carrier Detect line.
<i>comEvRing</i>	6	Ring detected. Some UARTs (universal asynchronous receiver-transmitters) may not support this event.
<i>comEvEOF</i>	End Of File (ASCII character 26)	character received.

Description Returns the most recent communication event or error.

AP.CommA.CommId

Property

AP.CommB.CommId

Property

Syntax AP . CommA . CommIDCommId

Data Type Integer

Description This command returns a handle that identifies the communications device.

AP.CommA.CommPort

Property

AP.CommB.CommPort

Property

Syntax `AP.CommA.CommPort`

Data Type Integer

Description This command sets and returns the communications port number.

The communications control generates error 68 (Device unavailable) if the port does not exist.

Warning You must set AP.CommA.CommPort before opening the port.

Example

```
Sub Main
  If AP.CommA.PortOpen = True Then 'Close Port if Open
    AP.CommA.PortOpen = False
  End If

  'Port Setup
  AP.CommA.CommPort = 2           'Select Comm Port
  AP.CommA.Settings = "9600,N,8,1" 'Set Comm Port _
    settings baud rate etc.
  AP.CommA.OutBufferSize = 10 'Set Output buffer size
  AP.CommA.InBufferSize = 10 'Set Input buffer size

  'Output to Comm Port 2
  AP.CommA.PortOpen = True       'Open Comm Port 2
  AP.CommA.Output = "1234567890" 'Send data

  'Input from Comm Port 2
  Character$ = AP.CommA.Input 'Get data sent to Comm 2
  Debug.Print Character$      'Print Input to _
    Immediate Window

  AP.CommA.PortOpen = False     'Close Comm Port
End Sub
```

AP.CommA.CTSHolding**Property****AP.CommB.CTSHolding****Property****Syntax** `AP.CommA.CTSHolding`**Data Type** Boolean*True* Clear To Send line high.*False* Clear To Send line low.**Description** This command returns the state of the of the Clear To Send (CTS) line. The state of the Clear To Send line indicates to the computer whether or not the transmission can proceed.

When the Clear To Send line is low (CTSHolding = False) and the time specified by the AP.CommA.CTSTimeout command has expired, the AP.CommA.CommEvent command is set to comCTSTO (Clear To Send Timeout) and a OnComm event is generated.

The Clear To Send line is used in RTS/CTS (Request To Send/Clear To Send) hardware handshaking. The AP.CommA.CTSHolding command provides a way to manually determine the state of the Clear To Send line.

See Also `AP.CommA.Handshaking`**AP.CommA.CTSTimeout****Property****AP.CommB.CTSTimeout****Property****Syntax** `AP.CommA.CTSTimeout`**Data Type** Long**Description** This command sets and returns the maximum amount of time (in milliseconds) that the control waits for the Clear To Send (CTS) signal before timing out. This command indicates a timeout condition by setting the AP.CommA.CommEvent command to CTSTO (Clear To Send Timeout Error) and generating the OnComm event.

AP.CommA.DSRHolding**Property****AP.CommB.DSRHolding****Property****Syntax** `AP.CommA.DSRHolding`**Data Type** Boolean*True* Data Set Ready line high.*False* Data Set Ready line low.**Description** This command returns the state of the of the Data Set Ready (DSR) line. The state of the Data Set Ready line indicates to the computer whether or not the hardware is ready to proceed.**AP.CommA.DSRTimeout****Property****AP.CommB.DSRTimeout****Property****Syntax** `AP.CommA.DSRTimeout`**Data Type** Long**Description** This command sets and returns the maximum amount of time (in milliseconds) that the control waits for the Data Set Ready (DSR) signal before timing out. This command indicates a timeout condition by setting the AP.CommA.CommEvent command to DSRT0 (Data Set Ready Timeout Error) and generating the OnComm event.**AP.CommA.DTREnable****Property****AP.CommB.DTREnable****Property****Syntax** `AP.CommA.DTREnable`**Data Type** Boolean*True* Enable the Data Terminal Ready (line high) when port opened and (line Low) when the port is closed.*False* (Default) Disable the Data Terminal Ready (line always low).

Description This command determines whether to enable the Data Terminal Ready (DTR) line during communications. Typically, the Data Terminal Ready signal is sent by a computer to its modem to indicate that the computer is ready to accept incoming data.

Setting the Data Terminal Ready line to low in most cases hangs up the telephone.

AP.CommA.Handshaking

Property

AP.CommB.Handshaking

Property

Syntax `AP.CommA.Handshaking`

Data Type	Long	Valid protocols are listed in the following table.
	0	(Default) No handshaking.
	1	XON/XOFF handshaking.
	2	RTS/CTS (Request To Send/Clear To Send) handshaking.
	3	Both Request To Send and XON/XOFF handshaking.

Description This command sets and returns the state of the hardware handshaking.

Handshaking refers to the internal communications protocol by which data is transferred from the hardware port to the receive buffer. When a character of data arrives at the serial port, the communications device has to move it into the receive buffer so that your program can read it. If there is no receive buffer and your program is expected to read every character directly from the hardware, you will probably lose data because the characters can arrive very quickly.

A handshaking protocol insures that data is not lost due to a buffer overrun, in which case data arrives at the port too quickly for the communications device to move the data into the receive buffer.

AP.CommA.InBufferCount**Property****AP.CommB.InBufferCount****Property****Syntax** `AP.CommA.InBufferCount`**Data Type** Integer**Description** This command returns the number of characters in the receive buffer.**AP.CommA.InBufferSize****Property****AP.CommB.InBufferSize****Property****Syntax** `AP.CommA.InBufferSize`**Data Type** Integer**Description** This command sets and returns the size of the receive buffer in bytes.
The default receive buffer size is 1024.**Example** See example for `AP.CommA.CommPort`.**AP.CommA.Input****Property****AP.CommB.Input****Property****Syntax** `AP.CommA.Input`**Result** String**Description** This command returns and removes a string of characters from the receive buffer. The `AP.CommA.InputLen` command defines the number of characters that are read by the `AP.CommA.Input` command.**Example** See example for `AP.CommA.CommPort`.

AP.CommA.InputLen**Property****AP.CommB.InputLen****Property****Syntax****AP.CommA.InputLen****Data Type**

Integer

Description

This command sets and returns the number of characters the AP.CommA.Input command reads from the receive buffer.

Setting the AP.CommA.InputLen command to 0 causes the AP.CommA.Input command to read the entire contents of the receive buffer.

If InputLen characters are not available in the receive buffer, the AP.CommA.Input command returns a zero-length string (""). The AP.CommA.InBufferCount command can also be checked to determine if the required number of characters are present before using the AP.CommA.Input command.

AP.CommA.Interval**Property****AP.CommB.Interval****Property****Syntax****AP.CommA.Interval****Data Type**

Long

Description

This command sets the interval (milliseconds) for polling the hardware for data under the Windows 3.0 operating system.

AP.CommA.NullDiscard**Property****AP.CommB.NullDiscard****Property****Syntax****AP.CommA.NullDiscard****Data Type**

Boolean

<i>True</i>	Null characters are not transferred from the port to the receive buffer.
<i>False</i>	(Default) Null characters are transferred from the port to the receive buffer.

Description This command determines whether null characters are allowed into the receive buffer.

A null character is defined as ASCII character 0, Chr\$(0).

AP.CommA.OutBufferCount

Property

AP.CommB.OutBufferCount

Property

Syntax `AP.CommA.OutBufferCount`

Data Type Integer

Description This command returns the number of characters in the transmit buffer. The transmit buffer can be cleared by setting the AP.CommA.OutBufferCount command to 0.

AP.CommA.OutBufferSize

Property

AP.CommB.OutBufferSize

Property

Syntax `AP.CommA.OutBufferSize`

Data Type Integer

Description This command sets and returns the size, in characters, of the transmit buffer. The default transmit buffer size is 512 bytes.

Example See example for `AP.CommA.CommPort`.

AP.CommA.Output**Property****AP.CommB.Output****Property**

Syntax	<code>AP.CommA.Output</code>
Data Type	Variant
Description	This command sends a string of characters to the transmit buffer.
Example	See example for <code>AP.CommA.CommPort</code> .

AP.CommA.ParityReplace**Property****AP.CommB.ParityReplace****Property**

Syntax	<code>AP.CommA.ParityReplace</code>
Data Type	String
Description	<p>This command sets and returns the character that replaces an invalid character in the data if a parity error occurs.</p> <p>The parity bit refers to a bit that is transmitted along with a specified number of data bits to provide error checking. When you use a parity bit, the communications control adds up all the bits that are set (having a value of 1) in the data and tests the sum as being odd or even (according to the parity setting used when the port was opened).</p> <p>By default, the control uses a question mark (?) character for replacing invalid characters. Setting <code>ParityReplace</code> to an empty string ("") disables replacement of the character where the parity error occurs.</p>

AP.CommA.PortOpen**Property****AP.CommB.PortOpen****Property**

Syntax	<code>AP.CommA.PortOpen</code>
Data Type	Boolean Port is opened.

False Port is closed or closes the port and clears the receive and transmit buffers.

Description This command sets and returns the state of the communications port. If either the AP.CommA.DTREnable or the AP.CommA.RTSEnable commands are set to True before the port is opened, the state of each command is set to False when the port is closed. Otherwise, the DTR and RTS lines remain in their previous state.

Example See example for AP.CommA.CommPort.

AP.CommA.RThreshold

Property

AP.CommB.RThreshold

Property

Syntax AP.CommA.RThreshold

Data Type Integer

Description This command sets and returns the number of characters to receive before the communications control sets the CommEvent command to comEvReceive and generates the OnComm event.

By setting the AP.CommA.RThreshold command to 0 (the default) generation of the OnComm event is disabled when characters are received.

By setting AP.CommA.RThreshold command to 1, each time a character is placed in the receive buffer an OnComm event is generated.

AP.CommA.RTSEnable

Property

AP.CommB.RTSEnable

Property

Syntax AP.CommA.RTSEnable

Data Type Boolean

<i>True</i>	Enables the Request To Send line (line set high when port open and low when port closed).
<i>False</i>	The default condition, disables the Request To Send line.

Description

This command determines the state of the Request To Send line.

The Request To Send line is used in RTS/CTS hardware handshaking.

AP.CommA.Settings**Property****AP.CommB.Settings****Property**

Syntax `AP.CommA.Settings`

Data Type String

The following table lists the valid baud rates.

Setting	Description
<i>110</i>	
<i>300</i>	
<i>600</i>	
<i>1200</i>	
<i>2400</i>	
<i>9600</i>	(Default)
<i>14400</i>	
<i>19200</i>	

The following table describes the valid parity values.

Setting	Description
<i>E</i>	Even
<i>M</i>	Mark
<i>N</i>	None (Default)
<i>O</i>	Odd
<i>S</i>	Space

The following table lists the valid data bit values.

Setting	Description
4	
5	
6	
7	
8	(default)

The following table lists the valid stop bit values.

Setting	Description
1	(Default)
1.5	
2	

Description

This command sets and returns the baud rate, parity, data bit, and stop bit settings.

If paramString\$ is not valid when the port is opened, the communications control generates error 380 (Invalid property value).

Settings\$ consists of four parts as specified in the following format:

"B,P,D,S"

Part	Description
<i>B</i>	Baud rate
<i>P</i>	Parity
<i>D</i>	Number of data bits
<i>S</i>	Number of stop bits

The default value of Settings\$ is: "9600,N,8,1"

Example

See example for AP.CommA.CommPort.

AP.CommA.SThreshold**Property****AP.CommB.SThreshold****Property****Syntax****AP.CommA.SThreshold****Data Type**

Integer

Valid protocols are listed in the following table.

0

No handshaking.

1

XON/XOFF handshaking.

2

RTS/CTS (Request To Send/Clear To Send) handshaking.

3

Both Request To Send and XON/XOFF handshaking.

Description

This command sets and returns the minimum number of characters allowed in the transmit buffer before the communications control sets the CommEvent property to comEvSend.

Setting the AP.CommA.SThreshold command to 0 (the default) disables generating the OnComm event for data transmission events. Setting the AP.CommA.SThreshold command to 1 causes the communications control to generate the OnComm event when the transmit buffer is completely empty.

If the number of characters in the transmit buffer is less than the number specified by the AP.CommA.SThreshold command, the CommEvent property is set to comEvSend. The comEvSend event is only sent once, when the number of characters crosses the Threshold.

User Notes

User Notes

User Notes

Computes

AP.Compute.Avg.Apply

Method

Syntax `AP.Compute.Avg.Apply`

Result Boolean

True Computation performed.
False Computation NOT performed.

Description This command applies the Average computation to the selected data (1-6). All of the measurements in the selected data will be replaced with the average value of the data within the Start and Stop settings for the Compute Average function.

See Also `AP.Compute.Clear.All`, `AP.Compute.Avg.Data`, `AP.Compute.Avg.PostSweep`, `AP.Compute.Avg.Start`, `AP.Compute.Avg.Stop`,

Example

```
Sub Main
  AP.File.OpenTest "AVG.AT2C"           'Open test
  AP.Compute.Clear.All
  AP.Compute.Avg.Data(1) = True 'Use Column1 for data 1
  AP.Compute.Avg.PostSweep = False  'Disable Apply _
    after sweep
  AP.Compute.Avg.Start("Hz") = 5000
  AP.Compute.Avg.Stop("Hz") = 100
  AP.Sweep.Start
  AP.Compute.Avg.Apply           'Compute Average
End Sub
```

AP.Compute.Avg.Data

Property

Syntax `AP.Compute.Avg.Data(ByVal Source As Integer)`

Data Type Boolean

True Select specified data.

	<i>False</i>	Deselect specified data.
Parameters	Name	Description
	<i>Source</i>	1 = Data 1 measurements 2 = Data 2 measurements 3 = Data 3 measurements 4 = Data 4 measurements 5 = Data 5 measurements 6 = Data 6 measurements
Description	This command determines which data (1-6) the Average computation is to be performed on. By using this command several times to select multiple data sources, several Average computations can be performed in one operation.	
See Also	AP.Compute.Avg.Apply	
Example	See example for AP.Compute.Avg.Apply.	

AP.Compute.Avg.PostSweep

Property

Syntax AP.Compute.Avg.PostSweep

Data Type Boolean

True Enable computation to be applied after sweep.
False Disable computation after sweep.

Description This command instructs the test to perform the Average computation after a sweep is complete and sets the state of the Apply After Sweep field on the Compute Average panel.

APWIN retains the order in which the Apply After Sweep field on the Compute panels is enabled. This permits multiple computations to be performed on data from a single test. The order of the computations is also retained in the test file.

See Also AP.Compute.Avg.Apply

Example See example for AP.Compute.Avg.Apply.

AP.Compute.Avg.Start**Property****Syntax** `AP.Compute.Avg.Start(ByVal Unit As String)`**Data Type** Double

Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.

Description This command sets the Start value of the data over which the Average computation will be performed.**See Also** `AP.Compute.Avg.Stop`, `AP.Compute.Avg.Apply`**Example** See example for `AP.Compute.Avg.Apply`.**AP.Compute.Avg.Stop****Property****Syntax** `AP.Compute.Avg.Stop(ByVal Unit As String)`**Data Type** Double

Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.

Description This command sets the Stop value of the data over which the Average computation will be performed.**See Also** `AP.Compute.Avg.Start`, `AP.Compute.Avg.Apply`**Example** See example for `AP.Compute.Avg.Apply`.**AP.Compute.Center.Apply****Method****Syntax** `AP.Compute.Center.Apply`**Result** Boolean*True* Computation performed.

False Computation NOT performed.

Description This command applies the Center computation to the selected data (1-6).

See Also AP.Compute.Clear.All, AP.Compute.Center.Data, AP.Compute.Center.PostSweep, AP.Compute.Center.Start, AP.Compute.Center.Stop

Example

```
Sub Main
  Dim Status As Boolean
  Status = AP.Log.Enable           'Determine Log Status
  AP.Log.Enable = False           'Enable Log Status
  AP.File.OpenTest "CENTER.AT2C" 'Open test
  AP.Compute.Clear.All
  AP.Compute.Center.Data(1) = True 'Set Data 1
  AP.Compute.Center.PostSweep = False 'Apply after _
    sweep off
  AP.Compute.Center.Start("Hz") = 200000 'Start freq _
    200k Hz
  AP.Compute.Center.Stop("Hz") = 10 'Stop at 10 Hz
  AP.Sweep.Start
  AP.Compute.Center.Apply           'Compute Center
  AP.Log.Enable = Status           'Reset Log Status
End Sub
```

AP.Compute.Center.Data

Property

Syntax `AP.Compute.Center.Data (ByVal Source As Integer)`

Data Type Boolean

True Select specified data.
False Deselect specified data.

Parameters

Name	Description
<i>Source</i>	1 = Data 1 measurements 2 = Data 2 measurements 3 = Data 3 measurements 4 = Data 4 measurements 5 = Data 5 measurements

6 = Data 6 measurements

Description This command determines which data (1-6) the Center computation is to be performed on. By using this command several times to select multiple data sources, several Center computations can be performed in one operation.

See Also `AP.Compute.Center.Apply`

Example See example for `AP.Compute.Center.Apply`.

AP.Compute.Center.PostSweep

Property

Syntax `AP.Compute.Center.PostSweep`

Data Type Boolean

True Enable computation to be applied after sweep.
False Disable computation after sweep.

Description This command instructs the test to perform the Center computation after a sweep is complete and sets the state of the Apply After Sweep field on the ComputeCenter panel.

APWIN retains the order in which the Apply After Sweep field on the Compute panels is enabled. This permits multiple computations to be performed on data from a single test. The order of the computations is also retained in the test file.

See Also `AP.Compute.Center.Apply`

Example See example for `AP.Compute.Center.Apply`.

AP.Compute.Center.Start

Property

Syntax `AP.Compute.Center.Start(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.
Description	This command sets the Start value of the data over which the Center computation will be performed.	
See Also	AP.Compute.Center.Stop, AP.Compute.Center.Apply	
Example	See example for AP.Compute.Center.Apply.	

AP.Compute.Center.Stop

Property

Syntax	AP.Compute.Center.Stop(ByVal <i>Unit</i> As String)	
Data Type	Double	
Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.
Description	This command sets the Stop value of the data over which the Center computation will be performed.	
See Also	AP.Compute.Center.Start, AP.Compute.Center.Apply	
Example	See example for AP.Compute.Center.Apply.	

AP.Compute.Clear.All

Method

Syntax	AP.Compute.Clear.All	
Description	This command clears all computes from the current test.	
Example	See example for AP.Compute.Center.Apply.	

AP.Compute.Delta.Apply

Method

Syntax	AP.Compute.Delta.Apply	
Result	Boolean	

True Computation performed.
False Computation NOT performed.

Description This command applies the Delta computation to the selected data (1-6).

See Also AP.Compute.Clear.All, AP.Compute.Delta.Data, AP.Compute.Delta.FileName, AP.Compute.Delta.PostSweep

Example

```
Sub Main
  AP.File.OpenTest "DELTA.AT2C" 'Opens test to be run _
    with results compared to stored data file
  AP.Compute.Clear.All
  AP.Sweep.Start
  AP.Compute.Delta.FileName = "DELTA1.ADA" 'Data file _
    used in delta computation
  AP.Compute.Delta.PostSweep = False 'Disables apply _
    after sweep
  AP.Compute.Delta.Data(1,1) = True
  AP.Compute.Delta.Data(2,2) = True
  AP.Compute.Delta.Apply 'Compute Delta
End Sub
```

AP.Compute.Delta.Data

Property

Syntax **AP.Compute.Delta.Data**(ByVal *Source* As Integer, ByVal *Column* As Integer)

Data Type Boolean

True Select specified data.
False Deselect specified data.

Parameters	Name	Description
	<i>Source</i>	Number of the Sweep Data (1-6) of the data in memory.
	<i>Column</i>	Number of the Data Column (0-7) of the data specified by the AP.Compute.Delta.FileName command.

Description	This command determines which data (Data 1-6) in memory and which data (Column 0-7) as specified by the <code>AP.Compute.Delta.FileName</code> command the Delta computation is to be performed on. By using this command several times to select multiple data sources, several Delta computations can be performed in one operation.
See Also	<code>AP.Compute.Delta.Apply</code> , <code>AP.Compute.Delta.FileName</code>
Example	See example for <code>AP.Compute.Delta.Apply</code> .

AP.Compute.Delta.FileName

Property

Syntax	<code>AP.Compute.Delta.FileName</code>
Data Type	String Any valid DOS filename and extension. Enter "SweepData" for the file name to select data in memory.
Description	This command attaches a data file to be used in the Compute Delta computation. The difference between the selected column data values in the data file and the selected data in memory will be calculated and then replace the data in memory.
See Also	<code>AP.Compute.Delta.Apply</code>
Example	See example for <code>AP.Compute.Delta.Apply</code> .

AP.Compute.Delta.PostSweep

Property

Syntax	<code>AP.Compute.Delta.PostSweep</code>
Data Type	Boolean <i>True</i> Enable computation to be applied after sweep. <i>False</i> Disable computation after sweep.
Description	This command instructs the test to perform the Delta computation after a sweep is complete and sets the state of the Apply After Sweep field on the Compute Delta panel.

APWIN retains the order in which the Apply After Sweep field on the Compute panels is enabled. This permits multiple computations to be performed on data from a single test. The order of the computations is also retained in the test file.

See Also `AP.Compute.Delta.Apply`

Example See example for `AP.Compute.Delta.Apply`.

AP.Compute.Equalize.Apply

Method

Syntax `AP.Compute.Equalize.Apply`

Result Boolean

True Computation performed.

False Computation NOT performed.

Description This command applies equalization to the selected data (1-6).

See Also `AP.Compute.Clear.All`, `AP.Compute.Equalize.Data`,
`AP.Compute.Equalize.FileName`,
`AP.Compute.Equalize.PostSweep`

Example

```
Sub Main
  AP.File.OpenTest "EQ.AT2C"      'opens test to be run
    'with results compared to stored data file
  AP.Compute.Clear.All
  AP.Sweep.Start
  AP.Compute.Equalize.FileName = "EQ1.ADA" 'data
    'file used in delta computation
  AP.Compute.Equalize.PostSweep = False 'disables
    'apply after sweep
  AP.Compute.Equalize.Data(1,1) = True
  AP.Compute.Equalize.Data(2,2) = True
  AP.Compute.Equalize.Apply 'Compute Equalize
End Sub
```


AP.Compute.Equalize.Data

Property

Syntax `AP.Compute.Equalize.Data (ByVal Source As Integer, ByVal Column As Integer)`

Data Type Boolean

True Select specified data.
False Deselect specified data.

Parameters	Name	Description
	<i>Source</i>	Number of the Sweep Data (1-6) of the data in memory.
	<i>Column</i>	Number of the Data Column (0-7) of the data specified by the <code>AP.Compute.Equalize.FileName</code> command.

Description This command determines which data (Data 1-6) in memory and which data (Column 0-7) as specified by the `AP.Compute.Equalize.FileName` command the Equalization computation is to be performed on. By using this command several times to select multiple data sources, several Equalization computations can be performed in one operation.

See Also `AP.Compute.Equalize.Apply`,
`AP.Compute.Equalize.FileName`

Example See example for `AP.Compute.Equalize.Apply`.

AP.Compute.Equalize.FileName

Property

Syntax `AP.Compute.Equalize.FileName`

Data Type String Any valid DOS filename and extension. Enter "SweepData" for the file name to select data in memory.

Description This command attaches a data file (Eq) to be used in the Compute Equalize computation. The data in memory is multiplied by the data in the Eq file.

See Also `AP.Compute.Equalize.Apply`

Example See example for `AP.Compute.Equalize.Apply`.

AP.Compute.Equalize.PostSweep

Property

Syntax	<code>AP.Compute.Equalize.PostSweep</code>
Data Type	Boolean
	<i>True</i> Enable computation to be applied after sweep.
	<i>False</i> Disable computation after sweep.
Description	<p>This command instructs the test to perform equalization after a sweep is complete and sets the state of the Apply After Sweep field on the Compute Equalize panel.</p> <p>APWIN retains the order in which the Apply After Sweep field on the Compute panels is enabled. This permits multiple computations to be performed on data from a single test. The order of the computations is also retained in the test file.</p>
See Also	<code>AP.Compute.Equalize.Apply</code>
Example	See example for <code>AP.Compute.Equalize.Apply</code> .

AP.Compute.Invert.Apply

Method

Syntax	<code>AP.Compute.Invert.Apply</code>
Result	Boolean
	<i>True</i> Computation performed.
	<i>False</i> Computation NOT performed.
Description	This command applies the Invert computation to the selected data (1-6).
See Also	<code>AP.Compute.Clear.All</code> , <code>AP.Compute.Invert.Data</code> , <code>AP.Compute.Invert.Horizontal</code> , <code>AP.Compute.Invert.PostSweep</code>
Example	<pre>Sub Main AP.File.OpenTest "INVERT.AT2C" 'Open test. AP.Compute.Clear.All 'Clear Compute functions. AP.Compute.Invert.PostSweep = False 'Post Sweep Off. AP.Compute.Invert.Data(1) = True 'Data to be inverted</pre>

```

    AP.Compute.Invert.Horizontal("Hz") = 5000'Horizontal _
        Value.
    AP.Sweep.Start
    AP.Compute.Invert.Apply
End Sub

```

AP.Compute.Invert.Data

Property

Syntax `AP.Compute.Invert.Data(ByVal Source As Integer)`

Data Type Boolean

True Select specified data.
False Deselect specified data.

Parameters

Name	Description
<i>Source</i>	1 = Data 1 measurements 2 = Data 2 measurements 3 = Data 3 measurements 4 = Data 4 measurements 5 = Data 5 measurements 6 = Data 6 measurements

Description This command determines which data (1-6) the Invert computation is to be performed on. By using this command several times to select multiple data sources, several Invert computations can be performed in one operation.

See Also `AP.Compute.Invert.Apply`

Example See example for `AP.Compute.Invert.Apply`.

AP.Compute.Invert.Horizontal

Property

Syntax `AP.Compute.Invert.Horizontal(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.
Description		This command sets the horizontal value in which the data will be inverted around.
See Also		AP.Compute.Invert.Apply
Example		See example for AP.Compute.Invert.Apply.

AP.Compute.Invert.PostSweep

Property

Syntax	AP.Compute.Invert.PostSweep	
Data Type	Boolean	
	<i>True</i>	Enable computation to be applied after sweep.
	<i>False</i>	Disable computation after sweep.
Description	<p>This command instructs the test to perform the Invert computation after a sweep is complete and sets the state of the Apply After Sweep field on the Compute Invert panel.</p> <p>APWIN retains the order in which the Apply After Sweep field on the Compute panels is enabled. This permits multiple computations to be performed on data from a single test. The order of the computations is also retained in the test file.</p>	
See Also	AP.Compute.Invert.Apply	
Example	See example for AP.Compute.Invert.Apply.	

AP.Compute.Linearity.Apply

Method

Syntax	AP.Compute.Linearity.Apply	
Result	Boolean	
	<i>True</i>	Computation performed.
	<i>False</i>	Computation NOT performed.

Description This command applies the Linearity computation to the selected data (1-6). The difference

See Also `AP.Compute.Clear.All`, `AP.Compute.Linearity.Data`, `AP.Compute.Linearity.PostSweep`, `AP.Compute.Linearity.Start`, `AP.Compute.Linearity.Stop`,

Example

```
Sub Main
  AP.File.OpenTest "LINEAR.AT2C"      'Open test.
  AP.Compute.Clear.All
  AP.Compute.Linearity.PostSweep = False 'Disables _
    Apply after sweep.
  AP.Compute.Linearity.Data(1) = True 'Use column 1 _
    for data 1.
  AP.Compute.Linearity.Start("Vrms") = .5 'Start at _
    500mV.
  AP.Compute.Linearity.Stop("Vrms") = 2 'Stop at 2V.
  AP.Sweep.Start
  AP.Compute.Linearity.Apply      'Start computation.
  AP.Sweep.Data1.Bottom("V") = -.02
  AP.Sweep.Data1.Top("V") = .02
End Sub
```

AP.Compute.Linearity.Data

Property

Syntax `AP.Compute.Linearity.Data`(ByVal *Source* As Integer)

Data Type Boolean

True Select specified data.

False Deselect specified data.

Parameters	Name	Description
	<i>Source</i>	1 = Data 1 measurements
		2 = Data 2 measurements
		3 = Data 3 measurements
		4 = Data 4 measurements
		5 = Data 5 measurements
		6 = Data 6 measurements

Description	This command determines which data (1-6) the Linearity computation is to be performed on. By using this command several times to select multiple data sources, several Linearity computations can be performed in one operation.
See Also	<code>AP.Compute.Linearity.Apply</code>
Example	See example for <code>AP.Compute.Linearity.Apply</code> .

AP.Compute.Linearity.PostSweep

Property

Syntax	<code>AP.Compute.Linearity.PostSweep</code>
Data Type	Boolean
	<i>True</i> Enable computation to be applied after sweep.
	<i>False</i> Disable computation after sweep.
Description	<p>This command instructs the test to perform the Linearity computation after a sweep is complete and sets the state of the Apply After Sweep field on the Compute Linearity panel.</p> <p>APWIN retains the order in which the Apply After Sweep field on the Compute panels is enabled. This permits multiple computations to be performed on data from a single test. The order of the computations is also retained in the test file.</p>
See Also	<code>AP.Compute.Linearity.Apply</code>
Example	See example for <code>AP.Compute.Linearity.Apply</code> .

AP.Compute.Linearity.Start

Property

Syntax	<code>AP.Compute.Linearity.Start(ByVal Unit As String)</code>
Data Type	Double

Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.
Description	This command sets the Start value of the data over which the Linearity computation will be performed.	
See Also	AP.Compute.Linearity.Stop, AP.Compute.Linearity.Apply	
Example	See example for AP.Compute.Linearity.Apply.	

AP.Compute.Linearity.Stop

Property

Syntax	AP.Compute.Linearity.Stop (ByVal <i>Unit</i> As String)	
Data Type	Double	
Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.
Description	This command sets the Stop value of the data over which the Linearity computation will be performed.	
See Also	AP.Compute.Linearity.Start, AP.Compute.Linearity.Apply	
Example	See example for AP.Compute.Linearity.Apply.	

AP.Compute.Max.Apply

Method

Syntax	AP.Compute.Max.Apply	
Result	Boolean	
	<i>True</i>	Computation performed.
	<i>False</i>	Computation NOT performed.
Description	This command applies the Maximum computation to the selected data (1-6).	

See Also AP.Compute.Clear.All, AP.Compute.Max.Data, AP.Compute.Max.PostSweep, AP.Compute.Max.Start, AP.Compute.Max.Stop,

Example

```

Sub Main
    AP.File.OpenTest "MAX.AT2C"          'Open test.
    AP.Compute.Clear.All
    AP.Compute.Max.PostSweep = False 'Disables Apply _
        after Sweep.
    AP.Compute.Max.Data(1) = True 'Use column 1 for _
        data 1.
    AP.Compute.Max.Start("Hz") = 2000
    AP.Compute.Max.Stop("Hz") = 200
    AP.Sweep.Start
    AP.Compute.Max.Apply
End Sub

```

AP.Compute.Max.Data

Property

Syntax **AP.Compute.Max.Data**(ByVal *Source* As Integer)

Data Type Boolean

True Select specified data.
False Deselect specified data.

Parameters

Name	Description
<i>Source</i>	1 = Data 1 measurements 2 = Data 2 measurements 3 = Data 3 measurements 4 = Data 4 measurements 5 = Data 5 measurements 6 = Data 6 measurements

Description

This command determines which data (1-6) the Maximum computation is to be performed on. By using this command several times to select multiple data sources, several Maximum computations can be performed in one operation.

See Also AP.Compute.Max.Apply

Example See example for `AP.Compute.Max.Apply`.

AP.Compute.Max.PostSweep

Property

Syntax `AP.Compute.Max.PostSweep`

Data Type Boolean

True Enable computation to be applied after sweep.
False Disable computation after sweep.

Description This command instructs the test to perform the Maximum computation after a sweep is complete and sets the state of the Apply After Sweep field on the Compute Maximum panel.

APWIN retains the order in which the Apply After Sweep field on the Compute panels is enabled. This permits multiple computations to be performed on data from a single test. The order of the computations is also retained in the test file.

See Also `AP.Compute.Max.Apply`

Example See example for `AP.Compute.Max.Apply`.

AP.Compute.Max.Start

Property

Syntax `AP.Compute.Max.Start(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.

Description This command sets the Start value of the data over which the Maximum computation will be performed.

See Also `AP.Compute.Max.Stop`, `AP.Compute.Max.Apply`

Example See example for `AP.Compute.Max.Apply`.

AP.Compute.Max.Stop

Property

Syntax	<code>AP.Compute.Max.Stop(ByVal Unit As String)</code>	
Data Type	Double	
Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.
Description	This command sets the Stop value of the data over which the Maximum computation will be performed.	
See Also	AP.Compute.Max.Start, AP.Compute.Max.Apply	
Example	See example for AP.Compute.Max.Apply.	

AP.Compute.Min.Apply

Method

Syntax	<code>AP.Compute.Min.Apply</code>	
Result	Boolean	
	<i>True</i>	Computation performed.
	<i>False</i>	Computation NOT performed.
Description	This command applies the Minimum computation to the selected data (1-6).	
See Also	AP.Compute.Clear.All, AP.Compute.Min.Data, AP.Compute.Min.PostSweep, AP.Compute.Min.Start, AP.Compute.Min.Stop,	
Example	<pre> Sub Main AP.File.OpenTest "MIN.AT2C" 'Open test. AP.Compute.Clear.All AP.Compute.Min.PostSweep = False 'Disables apply _ after sweep. AP.Compute.Min.Data(1) = True 'Use column 1 for _ data 1. AP.Compute.Min.Start("Hz") = 10000 AP.Compute.Min.Stop("Hz") = 200 </pre>	

```

    AP.Sweep.Start
    AP.Compute.Min.Apply
End Sub

```

AP.Compute.Min.Data

Property

Syntax `AP.Compute.Min.Data (ByVal Source As Integer)`

Data Type Boolean

True Select specified data.
False Deselect specified data.

Parameters

Name	Description
<i>Source</i>	1 = Data 1 measurements 2 = Data 2 measurements 3 = Data 3 measurements 4 = Data 4 measurements 5 = Data 5 measurements 6 = Data 6 measurements

Description This command determines which data (1-6) the Minimum computation is to be performed on. By using this command several times to select multiple data sources, several Minimum computations can be performed in one operation.

See Also `AP.Compute.Min.Apply`

Example See example for `AP.Compute.Min.Apply`.

AP.Compute.Min.PostSweep

Property

Syntax `AP.Compute.Min.PostSweep`

Data Type Boolean

True Enable computation to be applied after sweep.
False Disable computation after sweep.

Description This command instructs the test to perform the Minimum computation after a sweep is complete and sets the state of the Apply After Sweep field on the Compute Minimum panel.

APWIN retains the order in which the Apply After Sweep field on the Compute panels is enabled. This permits multiple computations to be performed on data from a single test. The order of the computations is also retained in the test file.

See Also `AP.Compute.Min.Apply`

Example See example for `AP.Compute.Min.Apply`.

AP.Compute.Min.Start

Property

Syntax `AP.Compute.Min.Start(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.

Description This command sets the Start value of the data over which the Minimum computation will be performed.

See Also `AP.Compute.Min.Stop`, `AP.Compute.Min.Apply`

Example See example for `AP.Compute.Min.Apply`.

AP.Compute.Min.Stop

Property

Syntax `AP.Compute.Min.Stop(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.

Description	This command sets the Stop value of the data over which the Minimum computation will be performed.
See Also	AP.Compute.Min.Start, AP.Compute.Min.Apply
Example	See example for AP.Compute.Min.Apply.

AP.Compute.Normalize.Apply

Method

Syntax AP.Compute.Normalize.Apply

Result Boolean

True Computation performed.

False Computation NOT performed.

Description This command applies the Normalize computation to the selected data (1-6).

See Also AP.Compute.Clear.All, AP.Compute.Normalize.Data, AP.Compute.Normalize.Horizontal, AP.Compute.Normalize.PostSweep, AP.Compute.Normalize.Target

Example

```
Sub Main
  AP.File.OpenTest "NORMAL.AT2C" 'opens test to be run.
  AP.Compute.Clear.All
  AP.Compute.Normalize.PostSweep = False 'Disables _
    apply after sweep.
  AP.Compute.Normalize.Data(1) = True
  AP.Compute.Normalize.Horizontal("Hz") = 1000 'Sets _
    1kHz point to be normalized.
  AP.Compute.Normalize.Target("dBV") = 0.0 'Normalize _
    1Khz point to 0.0dBV.
  AP.Sweep.Start
  AP.Compute.Normalize.Apply
End Sub
```

AP.Compute.Normalize.Data

Property

Syntax	<code>AP.Compute.Normalize.Data(ByVal <i>Source</i> As Integer)</code>	
Data Type	Boolean	
	<i>True</i>	Select specified data.
	<i>False</i>	Deselect specified data.
Parameters	Name	Description
	<i>Source</i>	1 = Data 1 measurements 2 = Data 2 measurements 3 = Data 3 measurements 4 = Data 4 measurements 5 = Data 5 measurements 6 = Data 6 measurements
Description	This command determines which data (1-6) the Normalize computation is to be performed on. By using this command several times to select multiple data sources, several Normalize computations can be performed in one operation.	
See Also	<code>AP.Compute.Normalize.Apply</code>	
Example	See example for <code>AP.Compute.Normalize.Apply</code> .	

AP.Compute.Normalize.Horizontal

Property

Syntax	<code>AP.Compute.Normalize.Horizontal(ByVal <i>Unit</i> As String)</code>	
Data Type	Double	
Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.
Description	This command sets the horizontal value in which the data will be normalized around.	
See Also	<code>AP.Compute.Normalize.Apply</code> , <code>AP.Compute.Normalize.Target</code>	

Example See example for `AP.Compute.Normalize.Apply`.

AP.Compute.Normalize.PostSweep

Property

Syntax `AP.Compute.Normalize.PostSweep`

Data Type Boolean

True Enable computation to be applied after sweep.

False Disable computation after sweep.

Description This command instructs the test to perform the Normalize computation after a sweep is complete and sets the state of the Apply After Sweep field on the Compute Normalize panel.

APWIN retains the order in which the Apply After Sweep field on the Compute panels is enabled. This permits multiple computations to be performed on data from a single test. The order of the computations is also retained in the test file.

See Also `AP.Compute.Normalize.Apply`

Example See example for `AP.Compute.Normalize.Apply`.

AP.Compute.Normalize.Target

Property

Syntax `AP.Compute.Normalize.Target (ByVal Unit As String)`

Data Type Double

Parameters

Name	Description
<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.

Description This command sets the vertical value in which the data will be Normalized to.

See Also `AP.Compute.Normalize.Apply`,
`AP.Compute.Normalize.Horizontal`

Example See example for `AP.Compute.Normalize.Apply`.

AP.Compute.Sigma.Apply**Method****Syntax** `AP.Compute.Sigma.Apply`**Result** Boolean

True Computation performed.
False Computation NOT performed.

Description This command applies the 2-Sigma computation to the selected data (1-6).**See Also** `AP.Compute.Clear.All`, `AP.Compute.Sigma.Data`, `AP.Compute.Sigma.PostSweep`, `AP.Compute.Sigma.Start`, `AP.Compute.Sigma.Stop`,**Example**

```
Sub Main
    Dim status As Boolean
    status = AP.Log.Enable      'Get logging condition.
    AP.Log.Enable = False     'Turn logging off.
    AP.File.OpenTest "SIGMA.AT2C" 'Open test.
    AP.Compute.Clear.All
    AP.Compute.Sigma.PostSweep = False 'Disables apply _
        after sweep.
    AP.Compute.Sigma.Data(1) = True 'Set data 1 for _
        Compute Sigma.
    AP.Compute.Sigma.Start("sec") = 6
    AP.Compute.Sigma.Stop("sec") = 12
    AP.Sweep.Start
    AP.Compute.Sigma.Apply
    AP.Log.Enable = status     'Return to initial _
        logging condition.
End Sub
```

AP.Compute.Sigma.Data**Property****Syntax** `AP.Compute.Sigma.Data(ByVal Source As Integer)`**Data Type** Boolean

True Select specified data.

	<i>False</i>	Deselect specified data.
Parameters	Name	Description
	<i>Source</i>	1 = Data 1 measurements 2 = Data 2 measurements 3 = Data 3 measurements 4 = Data 4 measurements 5 = Data 5 measurements 6 = Data 6 measurements
Description	This command determines which data (1-6) the 2-Sigma computation is to be performed on. By using this command several times to select multiple data sources, several 2-Sigma computations can be performed in one operation.	
See Also	AP.Compute.Sigma.Apply	
Example	See example for AP.Compute.Sigma.Apply.	

AP.Compute.Sigma.PostSweep

Property

Syntax	AP.Compute.Sigma.PostSweep	
Data Type	Boolean	
	<i>True</i>	Enable computation to be applied after sweep.
	<i>False</i>	Disable computation after sweep.
Description	<p>This command instructs the test to perform the 2-Sigma computation after a sweep is complete and sets the state of the Apply After Sweep field on the Compute 2-Sigma panel.</p> <p>APWIN retains the order in which the Apply After Sweep field on the Compute panels is enabled. This permits multiple computations to be performed on data from a single test. The order of the computations is also retained in the test file.</p>	
See Also	AP.Compute.Sigma.Apply	
Example	See example for AP.Compute.Sigma.Apply.	

AP.Compute.Sigma.Start

Property

Syntax	<code>AP.Compute.Sigma.Start(ByVal Unit As String)</code>	
Data Type	Double	
Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.
Description	This command sets the Start value of the data over which the 2-Sigma computation will be performed.	
See Also	<code>AP.Compute.Sigma.Stop</code> , <code>AP.Compute.Sigma.Apply</code>	
Example	See example for <code>AP.Compute.Sigma.Apply</code> .	

AP.Compute.Sigma.Stop

Property

Syntax	<code>AP.Compute.Sigma.Stop(ByVal Unit As String)</code>	
Data Type	Double	
Parameters	Name	Description
	<i>Unit</i>	The desired unit has to be available to the sweep panel: Source 1 for X-Y plots and Data 2 for X-Y Data 2 On X plots.
Description	This command sets the Stop value of the data over which the 2-Sigma computation will be performed.	
See Also	<code>AP.Compute.Sigma.Start</code> , <code>AP.Compute.Sigma.Apply</code>	
Example	See example for <code>AP.Compute.Sigma.Apply</code> .	

AP.Compute.Smooth.Apply

Method

Syntax	<code>AP.Compute.Smooth.Apply</code>	
Result	Boolean	
	<i>True</i>	Computation performed.

False Computation NOT performed.

Description This command performs a running 3-point smoothing computation to the selected data (1-6).

See Also AP.Compute.Clear.All, AP.Compute.Smooth.Data,
AP.Compute.Smooth.Passes,
AP.Compute.Smooth.PostSweep

Example

```
Sub Main
  AP.File.OpenTest "SMOOTH.AT2C"      'Open test.
  AP.Compute.Clear.All
  AP.Compute.Smooth.PostSweep = False 'Disable Apply _
    after Sweep.
  AP.Compute.Smooth.Auto = False'Disable Auto Smoothing.
  AP.Compute.Smooth.Data(1) = True
  AP.Compute.Smooth.Passes = 1 'Set Smooth Passes to 1.
  AP.Sweep.Start
  AP.Compute.Smooth.Apply
End Sub
```

AP.Compute.Smooth.Auto

Property

Syntax `AP.Compute.Smooth.Auto`

Data Type Boolean
True Enable auto smoothing.
False Disable auto smoothing.

Description This command automatically determines the number of passes that the smoothing algorithm performs on the selected data based on the number of measurements in the data.

See Also AP.Compute.Smooth.Apply

Example See example for AP.Compute.Smooth.Apply.

AP.Compute.Smooth.Data

Property

Syntax `AP.Compute.Smooth.Data(ByVal Source As Integer)`

Data Type	Boolean				
	<i>True</i> Select specified data.				
	<i>False</i> Deselect specified data.				
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>Source</i></td> <td>1 = Data 1 measurements 2 = Data 2 measurements 3 = Data 3 measurements 4 = Data 4 measurements 5 = Data 5 measurements 6 = Data 6 measurements</td> </tr> </tbody> </table>	Name	Description	<i>Source</i>	1 = Data 1 measurements 2 = Data 2 measurements 3 = Data 3 measurements 4 = Data 4 measurements 5 = Data 5 measurements 6 = Data 6 measurements
Name	Description				
<i>Source</i>	1 = Data 1 measurements 2 = Data 2 measurements 3 = Data 3 measurements 4 = Data 4 measurements 5 = Data 5 measurements 6 = Data 6 measurements				
Description	This command determines which data (1-6) the Smooth computation is to be performed on. By using this command several times to select multiple data sources, several Smooth computations can be performed in one operation.				
See Also	AP.Compute.Smooth.Apply				
Example	See example for AP.Compute.Smooth.Apply.				

AP.Compute.Smooth.Passes

Property

Syntax	AP.Compute.Smooth.Passes
Data Type	Long
Description	This command sets the number of times the smoothing algorithm is applied to the selected data.
See Also	AP.Compute.Smooth.Apply, AP.Compute.Smooth.Data
Example	See example for AP.Compute.Smooth.Apply.

AP.Compute.Smooth.PostSweep

Property

Syntax	AP.Compute.Smooth.PostSweep
Data Type	Boolean

True Enable computation to be applied after sweep.
False Disable computation after sweep.

Description

This command instructs the test to perform the Smooth computation after a sweep is complete and sets the state of the Apply After Sweep field on the Compute Smooth panel.

APWIN retains the order in which the Apply After Sweep field on the Compute panels is enabled. This permits multiple computations to be performed on data from a single test. The order of the computations is also retained in the test file.

See Also

AP.Compute.Smooth.Apply

Example

See example for AP.Compute.Smooth.Apply.

User Notes

User Notes

AP.Data.AddRowToEnd**Method****Syntax** `AP.Data.AddRowToEnd(ByVal Id As Integer)`

Parameters	Name	Description
	<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to AP.Data.Id command for additional information.

Result Integer**Description** This command adds an additional row to the end of data and returns the number of the row added.

Example

```

Sub Main
  AP.Application.NewTest
  AP.Sweep.Data2.Id = 5905 'Add Phase Data to Sweep

  StartValue = 20
  StopValue = 20000
  Frequencies = 31

  Counter = StartValue
  Do
    intAddRowToEnd = AP.Data.AddRowToEnd(0) 'Add row
    AP.Data.Value(0,0,intAddRowToEnd,"Hz") = Counter
    AP.Data.Value(0,1,intAddRowToEnd,"V") = 1.0
    AP.Data.Value(0,2,intAddRowToEnd,"deg") = 0.0
    Counter = Counter * 1.25893           'Log spacing
  Loop Until Counter > StopValue

  intAddRowToEnd = AP.Data.AddRowToEnd(0)
  AP.Data.Value(0,0,intAddRowToEnd,"Hz") = StopValue
  AP.Data.Value(0,1,intAddRowToEnd,"V") = 1.0
  AP.Data.Value(0,2,intAddRowToEnd,"deg") = 0.0
  AP.Data.UpdateDisplay(0)
End Sub

```


AP.Data.CollimitError**Method**

Syntax `AP.Data.CollimitError(ByVal Id As Integer, ByVal Column As Integer)`

Parameters	Name	Description
	<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to AP.Data.Id command for additional information.
	<i>Column</i>	Number of the Data Column (0-7).

Result Integer

Description This command returns a positive value if any measurement exceeds the upper or lower limit values for the specified column of data. A zero is returned if no errors occur. The returned value defines the number of measurements that exceed a limit.

See Also AP.Data.LimitError

Example

```

Sub Main
  AP.File.OpenTest "S1-FREQ.AT2"
  AP.Sweep.Start
  Errors = AP.Data.CollimitError(0,1)
  If Errors > 0 Then
    ErrorsUpper = AP.Data.ColUpperLimitError(0,1)
    ErrorsLower = AP.Data.ColLowerLimitError(0,1)
    String1$ = "This test Failed. " & Str(Errors) _
      & " Errors."
    String2$ = Str(ErrorsUpper)&" Upper Limit Errors."
    String3$ = Str(ErrorsLower)&" Lower Limit Errors."
    Prompt.Text = String1$ & Chr(13) & String2$ & _
      Chr(13) & String3$
    AP.Prompt.FontSize = 18
    AP.Prompt.Position -1,-1,425,175
    AP.Prompt.ShowWithContinue
    Stop
  ElseIf Errors = 0 Then
    AP.Prompt.Text = "This test Passed."
    AP.Prompt.FontSize = 18
    AP.Prompt.Position -1,-1,290,100
    AP.Prompt.ShowWithContinue
  End If
End Sub

```

```

        Stop
    End If
End Sub

```

AP.Data.ColLowerLimitError

Method

Syntax `AP.Data.ColLowerLimitError(ByVal Id As Integer, ByVal Column As Integer)`

Parameters	Name	Description
	<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to <code>AP.Data.Id</code> command for additional information.
	<i>Column</i>	Number of the Data Column (0-7).

Result Integer

Description This command returns the number of lower limit errors for the selected data. A zero is returned if no errors occur.

Example See example for `AP.Data.ColLimitError`.

AP.Data.ColName

Method

Syntax `AP.Data.ColName(ByVal Id As Integer, ByVal Column As Integer)`

Parameters	Name	Description
	<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to <code>AP.Data.Id</code> command for additional information.
	<i>Column</i>	Number of the Data Column (0-7).

Result String

Description This command returns the string as shown on the sweep panel for the selected data. The string defines the meter that is returning measurements.

Example

```

Sub Main
    AP.Application.NewTest 'Reset panels

```

```

AP.Sweep.Start
ColumnName$ = AP.Data.ColName(0,1)
Debug.Print "String definition For Data 1: = " & _
    ColumnName$
End Sub

```

Comment This macro puts up a prompt displaying the contents of the Data 1 control on the Sweep Panel.

Example Output String definition For Data 1: = .Anlr.Level A

AP.Data.ColNumOf

Method

Syntax `AP.Data.ColNumOf(ByVal Id As Integer)`

Parameters	Name	Description
	<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to <code>AP.Data.Id</code> command for additional information.

Result Integer

Description This command returns the number of columns of data.

Example See example for `AP.Data.ColSize`.

AP.Data.ColSize

Method

Syntax `AP.Data.ColSize(ByVal Id As Integer, ByVal Column As Integer)`

Parameters	Name	Description
	<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to <code>AP.Data.Id</code> command for additional information.
	<i>Column</i>	Number of the Data Column (0-7).

Result Long

Description This command returns the number of rows in the specified column.

Example

```

Sub Main
  AP.Application.NewTest 'Reset panels
  AP.Gen.Output = True
  AP.Anlr.ChAInput = 2
  AP.Sweep.Start
  NumColumns = AP.Data.ColNumOf(0)
  For Column = 1 To (NumColumns - 1) Step 1
    Size = AP.Data.ColSize(0, Column)
    Debug.Print "Number of measurements for Column _
      ";Column;" ="; Size
  Next Column
End Sub

```

Example Output Number of measurements for Column 1 = 31

AP.Data.ColUnit**Property****Syntax**

```
AP.Data.ColUnit(ByVal Id As Integer, ByVal Column
As Integer)
```

Parameters

Name	Description
<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to AP.Data.Id command for additional information.
<i>Column</i>	Number of the Data Column (0-7).

Result

String

Description

This command returns the unit string for the data in the selected column. APWIN computes all other relevant units for display.

Example

```

Sub Main
  AP.Application.NewTest      'Reset panels
  AP.Gen.Output = True       'Turn Generator output ON
  AP.Anlr.ChAInput = 2      'Set Analyzer input to _
    GENMON
  AP.Sweep.SinglePoint = 1  'Set sweep for single _
    measurement
  AP.Sweep.Start             'Run sweep
  Unit = AP.Data.ColUnit(0, 1)
  Debug.Print "The Unit for Data 1 is " & Unit

```

End Sub

AP.Data.ColUpperLimitError

Method

Syntax `AP.Data.ColUpperLimitError(ByVal Id As Integer, ByVal Column As Integer)`

Parameters	Name	Description
	<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to <code>AP.Data.Id</code> command for additional information.
	<i>Column</i>	Number of the Data Column (0-7).

Result Integer

Description This command returns the number of upper limit errors for the selected data column.

Example See example for `AP.Data.ColLimitError`.

AP.Data.DeleteRow

Method

Syntax `AP.Data.DeleteRow(ByVal Id As Integer, ByVal RowNum As Integer)`

Parameters	Name	Description
	<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to <code>AP.Data.Id</code> command for additional information.
	<i>RowNum</i>	Number of row to delete.

Result Boolean

True Row deleted.
False Row not deleted.

Description This command deletes the designated row.
 Note that the row numbering begins with zero.

Example

```
Sub Main()
    Dim FreqData() As Double
```

```

DataSize = AP.Data.ColSize(0,0)
ReDim FreqData(DataSize)
FreqData = AP.Data.XferToArray(0,0,"Hz")

For Count1 = 0 To DataSize - 1
  LastDup = 0
  For Count2 = Count1 + 1 To DataSize - 1
    If FreqData(Count1) = FreqData(Count2) _
      And Count1 <> Count2 Then
      FreqData(Count2) = 0
      LastDup = Count2
      Data.Value(0,0,Count2,"Hz") = 0
    End If
  Next Count2
  If LastDup <> 0 Then Count1 = LastDup
Next Count1

Duplicates = 0
For Count1 = DataSize - 1 To 0 Step -1
  If AP.Data.Value(0,0,Count1,"Hz") = 0 Then
    AP.Data.DeleteRow (0, count1)
    Duplicates = Duplicates + 1
  End If
Next Count1
If Duplicates > 0 Then
  AP.Prompt.Text = Str$(Duplicates) & " Duplicate _
    frequency(s) removed."
  AP.Prompt.Show
  Wait 2
  AP.Prompt.Hide.
End If
End Sub

```

AP.Data.Id

① Method

Syntax **AP.Data.Id**(ByVal *FileName* As String)

Parameters	Name	Description
	<i>FileName</i>	Long Path and File Names permitted up to 128 characters. The file must be an APWIN limit, sweep, or data (.adl, .ads, .ada) file attached to the current test.
Result	Integer	
Description		This command returns an ID# that identifies the file specified in the command argument. The ID# can be use as the <i>id%</i> argument in all of the Data commands to specify which data to act upon. Use an Id# of zero (0) to access sweep data.
See Also		AP.Data.ColLimitError, AP.Data.ColLowerLimitError, AP.Data.ColName, AP.Data.ColNumOf, AP.Data.ColSize, AP.Data.ColUpperLimitError, AP.Data.LimitError, AP.Data.LowerLimitError, AP.Data.OptimizeDisplay, AP.Data.UpdateDisplay, AP.Data.UpperLimitError, AP.Data.Value, AP.Data.XferToArray
Example		<pre> Sub Main Dim Limitarray As Variant Dim Tablearray As Variant File.OpenTest "ID.AT2C" 'Open test LimitId = AP.Data.Id("LIMIT.ADL") TableId = AP.Data.Id("TABLE.ADS") LimitArray = AP.Data.XferToArray(LimitId, 1, "V") TableArray = AP.Data.XferToArray(TableId, 1, "V") Debug.Print "Limit ID # = ";LimitId Debug.Print "Table ID # = ";TableId End Sub </pre>
Example Output		<pre> Limit ID # = 100 Table ID # = 101 </pre>

AP.Data.InsertRowAfter

Method

Syntax `AP.Data.InsertRowAfter`(ByVal *Id* As Integer, ByVal *RowNum* As Integer)

Parameters	Name	Description
	<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to <code>AP.Data.Id</code> command for additional information.
	<i>RowNum</i>	Number of row to insert after.
Result	Boolean	
	<i>True</i>	Row inserted.
	<i>False</i>	Row not inserted.
Description	This command inserts an additional row before the designated row. Note that the row numbering begins with zero.	
Example	See example for <code>AP.Data.InsertRowBefore</code> .	

AP.Data.InsertRowBefore Method

Syntax	<code>AP.Data.InsertRowBefore(ByVal Id As Integer, ByVal RowNum As Integer)</code>	
Parameters	Name	Description
	<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to <code>AP.Data.Id</code> command for additional information.
	<i>RowNum</i>	Number of row to insert before.
Result	Boolean	
	<i>True</i>	Row deleted.
	<i>False</i>	Row not deleted.
Description	This command inserts an additional row after the designated row. Note that the row numbering begins with zero.	
Example	See example for <code>AP.Data.InsertRowAfter</code> .	

AP.Data.LimitError Method

Syntax `AP.Data.LimitError(ByVal Id As Integer)`

Parameters**Name****Description***Id*

Data identification number. Use an Id# of zero (0) to access sweep data. Refer to AP.Data.Id command for additional information.

Result

Integer

Description

This command returns the number of measurements that exceed a limit.

Example

Sub Main

```
AP.File.OpenTest "S2-FREQ.AT2C"
```

```
AP.Sweep.Start
```

```
Errors = AP.Data.LimitError(0)
```

```
If Errors > 0 Then
```

```
    ErrorsUpper = AP.Data.ColUpperLimitError(0,1)
```

```
    ErrorsLower = AP.Data.ColLowerLimitError(0,1)
```

```
    String1$ = "This test Failed. " & Str(Errors) & _  
              " Errors. "
```

```
    String2$ = Str(ErrorsUpper)&" Upper Limit Errors."
```

```
    String3$ = Str(ErrorsLower)&" Lower Limit Errors."
```

```
    AP.Prompt.Text = String1$ & Chr(13) & String2$ & _  
                  Chr(13) & String3$
```

```
    AP.Prompt.FontSize = 18
```

```
    AP.Prompt.Position -1,-1,425,175
```

```
    AP.Prompt.ShowWithContinue
```

```
    Stop
```

```
ElseIf Errors = 0 Then
```

```
    AP.Prompt.Text = "This test Passed."
```

```
    AP.Prompt.FontSize = 18
```

```
    AP.Prompt.Position -1,-1,290,100
```

```
    AP.Prompt.ShowWithContinue
```

```
    Stop
```

```
End If
```

```
End Sub
```

AP.Data.LowerLimitError**Method**

Syntax	<code>AP.Data.LowerLimitError (ByVal Id As Integer)</code>				
Parameters	<table border="1"> <thead> <tr> <th style="text-align: left;">Name</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td><i>Id</i></td> <td>Data identification number. Use an Id# of zero (0) to access sweep data. Refer to AP.Data.Id command for additional information.</td> </tr> </tbody> </table>	Name	Description	<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to AP.Data.Id command for additional information.
Name	Description				
<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to AP.Data.Id command for additional information.				
Result	Integer				
Description	This command returns a positive value if any measurement is less than the lower limit values. A zero is returned if no errors occur. The returned value defines the number of measurements that are less than the limit.				
Example	<pre> Sub Main AP.File.OpenTest "S2-FREQ.AT2C" AP.Sweep.Start Flag = AP.Data.LowerLimitError(0) If Flag > 0 Then AP.Prompt.Text = "This test Failed." AP.Prompt.FontSize = 18 AP.Prompt.Position -1,-1,290,100 AP.Prompt.ShowWithContinue Stop ElseIf Flag = 0 Then AP.Prompt.Text = "This test Passed." AP.Prompt.FontSize = 18 AP.Prompt.Position -1,-1,290,100 AP.Prompt.ShowWithContinue Stop End If End Sub </pre>				

AP.Data.UpdateDisplay**Method****Syntax****AP.Data.UpdateDisplay**(ByVal *Id* As Integer)**Parameters**

Name	Description
<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data only.

Description

This command updates the data displayed in the table and graph displays.

Example

```

Sub Main
    AP.Application.NewTest 'Reset panels
    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2
    AP.Anlr.FuncFilterLP = 1
    AP.Sweep.Data1.Id = 5906
    AP.Sweep.Data2.Id = 5906
    AP.Sweep.Source1.Start("Hz") = 100000
    AP.Sweep.Source1.Steps = 4
    AP.Sweep.Start
    Size = Data.ColSize (0, 1)
    For Reading = 0 To (Size - 1) Step 1 'Read readings.
        Debug.Print "Acquired Reading";Reading;" = " _
            ;Format(AP.Data.Value(0, 1, Reading), _
                "#.0000");" V" 'Return reading.
    Next Reading
    For Reading = 0 To (Size - 1) Step 1
        Measurement = AP.Data.Value(0, 1, Reading)
        AP.Data.Value(0, 2, Reading) = (Measurement * _
            1.20) 'Increase level by 20%.
    Next Reading
    For Reading = 0 To (Size - 1) Step 1
        Debug.Print "Limit";Reading;" = " _
            ;Format(AP.Data.Value(0, 2, Reading), _
                "#.0000");" V" 'Return reading.
    Next Reading
    AP.Data.UpdateDisplay(0)
    AP.File.SaveDataAs "UPPER.ADL"
End Sub

```

Example Output Acquired Reading 0 = .0219 V

```

Acquired Reading 1 = .9874 V
Acquired Reading 2 = .9937 V
Acquired Reading 3 = .9933 V
Acquired Reading 4 = .9950 V
Limit 0 = .0262 V
Limit 1 = 1.1848 V
Limit 2 = 1.1925 V
Limit 3 = 1.1920 V
Limit 4 = 1.1940 V

```

AP.Data.UpperLimitError

Method

Syntax

```
AP.Data.UpperLimitError(ByVal Id As Integer)
```

Parameters

Name	Description
<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to AP.Data.Id command for additional information.

Result

Integer

Description

This command returns a positive value if any measurement exceeds the upper limit values. A zero is returned if no errors occur. The returned value defines the number of measurements that exceed the limit.

Example

```

Sub Main
  AP.File.OpenTest "S2-FREQ.AT2C"
  AP.Sweep.Start
  Flag = AP.Data.UpperLimitError(0)
  If Flag > 0 Then
    AP.Prompt.Text = "This test Failed."
    AP.Prompt.FontSize = 18
    AP.Prompt.Position -1,-1,290,100
    AP.Prompt.ShowWithContinue
    Stop
  ElseIf Flag = 0 Then
    AP.Prompt.Text = "This test Passed."
    AP.Prompt.FontSize = 18
    AP.Prompt.Position -1,-1,290,100
    AP.Prompt.ShowWithContinue
  End If
End Sub

```

```

        Stop
    End If
End Sub

```

AP.Data.Value

Property

Syntax

```

AP.Data.Value(ByVal Id As Integer, ByVal Column
As Integer, ByVal Index As Long, ByVal Unit As
String)

```

Data Type

Double

Parameters

Name	Description
<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to AP.Data.Id command for additional information.
<i>Column</i>	Number of the Data Column (0-7).
<i>Index</i>	This value defines which row a measurement is returned from. A column may have any number of rows. Use the AP.Data.ColSize command to determine the number of rows in a column. Note that the row numbering begins with zero.
<i>Unit</i>	Refer to the setting or reading defined by the <i>column%</i> parameter to determine the appropriate unit selections.

Description

This command returns the specified reading from sweep data.

See Also

AP.Data.ColSize, AP.Data.Id

Example

```

Sub Main
    AP.Application.NewTest 'Reset panels
    AP.Gen.Output = True 'Turn Generator output ON
    AP.Anlr.ChAInput = 2 'Set Analyzer input to GENMON
    AP.Sweep.SinglePoint = 1 'Set sweep for single _
        measurement
    AP.Sweep.Start 'Run sweep
    Reading1 = AP.Data.Value(0, 1, 0, "V") 'Get Reading
    Debug.Print "Reading = ";Format(Reading1, _
        "#.0000"); "V"
End Sub

```

Example Output Reading = .9850 V

AP.Data.XferToArray

Property

Syntax `AP.Data.XferToArray`(ByVal *Id* As Integer, ByVal *Column* As Integer, ByVal *Unit* As String)

Data Type Variant

Parameters

Name	Description
<i>Id</i>	Data identification number. Use an Id# of zero (0) to access sweep data. Refer to AP.Data.Id command for additional information.
<i>Column</i>	Number of the Data Column (0-7).
<i>Unit</i>	Refer to the setting or reading defined by the <i>column%</i> parameter to determine the appropriate unit selections.

Description This command transfers the contents of a column (Sweep Source 1-2 or Data 1-6) to an array.

See Also AP.Data.Id

Example

```
Sub Main
    Dim A As Variant
    AP.Application.NewTest 'Reset panels
    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2
    AP.Anlr.FuncFilterLP = 1
    AP.Sweep.Data1.Id = 5906
    AP.Sweep.Source1.Start("Hz") = 100000
    AP.Sweep.Source1.Steps = 4
    AP.Application.NewData 'Clear data loaded with test.
    AP.Sweep.Start
    Size = AP.Data.ColSize (0, 1)
                                'Transfer data to array.
    A = AP.Data.XferToArray(0, 1, "V")
    For Counter = 0 To (Size - 1) Step 1
        Debug.Print "Reading";Counter;" = "; _
            Format(A(Counter), "#.0000");" V"
    Next Counter
```

End Sub

Example Output Reading 0 = .0223 V
Reading 1 = .9889 V
Reading 2 = .9953 V
Reading 3 = .9945 V
Reading4 = .9931 V

Comment The values in the example output are taken from the array and then displayed.

User Notes

User Notes

DCX-127**AP.DCX.Ch1DcLevel****Property****Syntax** `AP.DCX.Ch1DcLevel (ByVal Unit As String)`**Data Type** Double -10.5 to 10.5 Volts**Parameters**

Part	Description
<i>Unit</i>	The following units are available Vdc.

Description

This command sets the voltage at the DCX's channel 1 DC output.

Example

```
Sub Main
    AP.File.OpenTest "DCX1.AT2C"      'Opens test.
    AP.DCX.Ch1DcOutput = True
    AP.DCX.Ch1DcLevel("Vdc") = 1.5
End Sub
```

Comment

This macro turns on the DCX's Channel 1 DC output and sets it to 1.5 volts.

AP.DCX.Ch1DcOutput**Property****Syntax** `AP.DCX.Ch1DcOutput`**Data Type** Boolean*True* Connects the output to the front panel.*False* Disconnects the output from the front panel.**Description**

This command sets DC Volts output 1 to On or Off.

ExampleSee example for `AP.DCX.Ch1DcLevel`.

AP.DCX.Ch2DcLevel**Property**

Syntax	AP.DCX.Ch2DcLevel (ByVal <i>Unit</i> As String)	
Data Type	Double	-10.5 to 10.5 Volts
Parameters	Part	Description
	<i>Unit</i>	The following units are available Vdc.
Description	This command sets the voltage at the DCX's channel 2 DC output.	
Example	<pre>Sub Main AP.File.OpenTest "DCX1.AT2C" 'Open test. AP.DCX.Ch2DcOutput = True AP.DCX.Ch2DcLevel("Vdc") = 1.5 End Sub</pre>	
Comment	This macro turns on the DCX's Channel 2 DC output and sets it to 1.5 volts.	

AP.DCX.Ch2DcOutput**Property**

Syntax	AP.DCX.Ch2DcOutput	
Data Type	Boolean	
	<i>True</i>	Connects the output to the front panel.
	<i>False</i>	Disconnects the output from the front panel.
Description	This command sets DC Volts output 2 to On or Off.	
Example	See example for AP.DCX.Ch2DcLevel.	

AP.DCX.DigInFormat**Property**

Syntax	AP.DCX.DigInFormat	
Data Type	Integer	
	<i>0</i>	2's Complement
	<i>1</i>	BCD

Description This command sets the format of the digital input.

The digital ports are 21 bits plus a sign bit.

The normal format is two's complement. This format combines the bits into a 22 bit word that follows normal two's complement conventions (-1 is represented as 3FFFFFF hex).

The BCD (Binary coded decimal) format is a signed magnitude representation (-1 is represented as 200001 hex, -10 is 200010 hex, etc.). As is normal in the BCD format, each decimal digit is represented by 4 bits.

Example

```

Sub Main
  AP.File.OpenTest "DCX1.AT2C" 'Open test.
  AP.DCX.DigOutFormat = 1      'Sets the digital _
    output format to BCD.
  AP.DCX.DigInRdgRate = 1      'Selects input strobe _
    rate of 4/sec.
  AP.DCX.DigOut("dec") = 100   'Sets the digital _
    output to 100 dec.
  AP.DCX.DigInFormat = 1      'Sets format of _
    digital input to BCD.
  Reading1 = AP.DCX.DigInRdg("dec") 'Returns a _
    settled reading 100 in dec.
  AP.DCX.DigOut("h(x)") = 100   'Sets the digital _
    output to 100 dec scaled.
  AP.DCX.DigOutScale = 2      'Scales the digital _
    output by 2.
  AP.DCX.DigInSettling(.20, .1, "Dec", 4, .05, 0)
  AP.DCX.DigInTrig           'Trigger a new reading.
Do
  Ready = AP.DCX.DigInReady
Loop Until Ready > 0          'Wait until new _
  reading is ready.
  Reading2 = AP.DCX.DigInRdg("dec") 'Returns a settled _
    reading 200 in dec.
  AP.DCX.DigInScale = .5      'Scales the digital _
    input by .5.
  Reading3 = AP.DCX.DigInRdg("g(x)") 'Returns _
    a settled reading 100 in dec.

```

```

NewLine$ = Chr(13)
a$= "Reading1 "+Left(Str$(Reading1),6)+"dec"
b$= "Reading2 "+Left(Str$(Reading2),6)+"dec"
c$= "Reading3 "+Left(Str$(Reading3),6)+"dec"
AP.Prompt.Text = a$ + NewLine$ + b$ + NewLine$ + c$
AP.Prompt.ShowWithContinue
Beep
Stop
End Sub

```

AP.DCX.DigInRdg

Property

Syntax `AP.DCX.DigInRdg(ByVal Unit As String)`

Data Type Double

Parameters	Part	Description
	<i>Unit</i>	The following units are available: dec, hex, oct, and g(x).

Description This command returns a settled reading for the DCX-127 Digital In meter and zeros the ready count.

See Also `AP.DCX.DigInReady`, `AP.DCX.DigInSettling`, `AP.DCX.DigInTrig`

Example See example for `AP.DCX.DigInFormat`.

AP.DCX.DigInRdgRate

Property

Syntax `AP.DCX.DigInRdgRate`

Data Type Integer

0	External Strobe (Default).
1	4 readings per second.
2	8 readings per second.
3	16 readings per second.
4	32 readings per second.

Description This command selects an internal or external strobe for the digital input.

If 0 is selected, the External Strobe available on pin (25) of the digital input connector is used to trigger each new reading. If 1-4 is selected, an internal strobe is used at the specified rate.

Example See example for `AP.DCX.DigInFormat`.

AP.DCX.DigInReady

Property

Syntax `AP.DCX.DigInReady`

Data Type Integer

0 Reading not ready.
>0 Reading ready.

Description This command returns the DCX-127 Digital In settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.DCX.DigInRdg` or `AP.DCX.DigInTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.DCX.DigInRdg` command will be guaranteed to return quickly.

See Also `AP.DCX.DigInRdg`, `AP.DCX.DigInSettling`,
`AP.DCX.DigInTrig`

Example See example for `AP.DCX.DigInFormat`.

AP.DCX.DigInScale

Property

Syntax `AP.DCX.DigInScale`

Data Type Double

Description This command sets the DCX-127 Digital Input Scale factor.

When g(x) units are selected at the Digital In display, APWIN software computes the displayed value from the relationship

$$\text{display} = \text{measurement} * \text{Scale (g)}$$

where measurement is the decimal value of the binary data in the selected format and Scale (g) is the value entered in the Scale (g) field just below the Digital In display.

Example See example for AP.DCX.DigInFormat.

AP.DCX.DigInSettling

Method

Syntax	AP.Anlr.PhaseSettling (ByVal <i>Tolerance</i> As Double, ByVal <i>Floor</i> As Double, ByVal <i>FloorUnit</i> As String, ByVal <i>Points</i> As Integer, ByVal <i>Delay</i> As Double, ByVal <i>Algorithm</i> As Integer)
Parameters	See Appendix A for Settling Algorithm and parameter name descriptions.
Description	This command sets the settling parameters for the AP.Dcx.DigInRdg command.
See Also	AP.DCX.DigInFormat, AP.DCX.DigInRdg, AP.DCX.DigInReady, AP.DCX.DigInTrig
Example	See example for AP.DCX.DigInFormat.

AP.DCX.DigInTrig

Method

Syntax	AP.DCX.DigInTrig
Description	Causes a restart of the reading cycle and zeros the ready count for the AP.DCX.DigInRdg command. The reading in progress is aborted.
See Also	AP.DCX.DigInRdg, AP.DCX.DigInReady, AP.DCX.DigInSettling
Example	See example for AP.DCX.DigInFormat.

AP.DCX.DigOut**Property**

Syntax `AP.DCX.DigOut (ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	The following units are available dec, hex, oct, h(x).

Description This command sets the value of the DCX's digital output.

The output format is either two's complement or BCD as set by the `AP.DCX.DigOutFormat` command.

Example See example for `AP.DCX.DigInFormat`.

AP.DCX.DigOutFormat**Property**

Syntax `AP.DCX.DigOutFormat`

Data Type Integer

<i>0</i>	2's Complement
<i>1</i>	BCD

Description This command sets the format of the digital output.

The digital ports are 21 bits plus a sign bit.

The normal format is two's complement. This format combines the bits into a 22 bit word that follows normal two's complement conventions (-1 is represented as 3FFFFFF hex).

The BCD (Binary coded decimal) format is a signed magnitude representation (-1 is represented as 200001 hex, -10 is 200010 hex, etc.). As is normal in the BCD format, each decimal digit is represented by 4 bits.

Example See example for `AP.DCX.DigInFormat`.

AP.DCX.DigOutScale

Property**Syntax** `AP.DCX.DigOutScale`**Data Type** Double**Description** When h(x) units are selected at the Digital Output control field, APWIN software computes the actual transmitted value from the relationship

$$\text{output value} = \text{entry value} * \text{Scale (h)}$$

where entry value is the decimal value entered into the Digital Out numeric field and Scale (h) is the value entered in the Scale (h) field just below the Digital Out control field.

Example See example for `AP.DCX.DigInFormat`.

AP.DCX.DmmMode

Property**Syntax** `AP.DCX.DmmMode`**Data Type** Integer

<i>0</i>	Off This command disconnects the DMM from the front panel jacks. This allows the DMM to be wired to the circuit under test yet not be connected until needed. This is so that there is no possibility of the DMM input characteristics degrading the results of any other measurements being made by System One or System Two.
<i>1</i>	DC Volts
<i>2</i>	Ohms

Description This command sets the DMM measurement mode.

Example

```
Sub Main
  AP.File.OpenTest "DCX1.AT2C" 'Open test.
  AP.DCX.DmmRange = 2.0      'set DMM input to _
    2 Volt range.
  AP.DCX.DmmMode = 1        'set DMM mode to volts.
```

```

AP.DCX.DmmRdgRate = 0      'set DMM reading rate to 6 _
    readings per second.
AP.DCX.DmmSettling(1, .20, "Vdc", 3, .03, 0) 'Set _
    settling parameters.

AP.DCX.DmmTrig                'Trigger a new reading.
Do
    Ready = AP.DCX.DmmReady
Loop Until Ready > 0 'Loop until new reading is ready
Reading1 = AP.DCX.DmmRdg("Vdc") 'Returns a settled _
    reading.

NewLine$ = Chr(13)
a$= "DMM Reading "+Left(Str$(Reading1),6)+"Vdc"
AP.Prompt.Text = a$ + NewLine$
AP.Prompt.ShowWithContinue
Beep
Stop
End Sub

```

Comment

This macro sets the DMM to volts, selects 2 Volt range, sets the reading rate, sets settling, triggers a New reading, waits For the New reading, and stores it In a variable called Reading1.

AP.DCX.DmmOffset**Property**

Syntax	AP.DCX.DmmOffset
Data Type	Double
Description	<p>When f(V) (function of Volts) or f(O) (function of Ohms) units are selected for the DMM, APWIN software computes the value to display from the formula</p> $\text{display} = (\text{measurement} + \text{Offset}) * \text{Scale}$ <p>The measurement term is the value which would be displayed in Volts or Ohms units. The Offset and Scale values are the contents of the fields with those names, at the top right of the DCX panel.</p>
See Also	AP.DCX.DmmScale

Example See example for `AP.DCX.DmmMode`.

AP.DCX.DmmRange

Property

Syntax `AP.DCX.DmmRange`

Data Type Double

Description This command sets the DMM's input range and returns the nominal full scale of range in use.

The ranges for Ohms mode are:

2M, 200k, 20k, 2k, 200 Ohms

The ranges for Volts mode are:

500, 200, 20, 2.0, 0.2 Volts

A common use of this command is in fixing the input range by obtaining the range and then using that value for this command.

Example See example for `AP.DCX.DmmMode`.

AP.DCX.DmmRangeAuto

Property

Syntax `AP.DCX.DmmRangeAuto`

Data Type Boolean

True Auto range

False Fixed range

Description This command sets the DCX-127 DMM input to Auto range or fixed range. Care must be taken when using Fixed range that the input signal does not exceed the selected range.

See Also `AP.DCX.DmmRange`

Example

```
Sub Main
    AP.File.OpenTest "DCX1.AT2C" 'Open test.
    AP.DCX.DmmRangeAuto = 1 'set DMM input to auto _
```

```

    range.
AP.DCX.DmmMode = 1      'set DMM mode to volts.
AP.DCX.DmmRdgRate = 1  'set DMM reading rate to 25 _
    readings per second.
AP.DCX.DmmScale = 2
AP.DCX.DmmOffset = 1
AP.DCX.DmmSettling(1, .20, "Vdc", 3, .03, 0) 'Set _
    settling parameters.
AP.DCX.DmmTrig          'Trigger a new reading.
Do
    Ready = AP.DCX.DmmReady
Loop Until Ready > 0 'Loop until new reading is _
    ready.
Reading1 = AP.DCX.DmmRdg("f(v)") 'Returns _
    a settled reading.

NewLine$ = Chr(13)
a$= "DMM Reading "+Left(Str$(Reading1),6)+"f(V)"
AP.Prompt.Text = a$ + NewLine$
AP.Prompt.ShowWithContinue
Beep
Stop
End Sub

```

AP.DCX.DmmRdg

Property

Syntax `AP.DCX.DmmRdg (ByVal Unit As String)`

Data Type Variant

Parameters	Part	Description
	<i>Unit</i>	The following units are available VDC, V(f) for the AP.DCX.DmmMode command DCV mode and Ohms, and f(O) for the Ohms mode.

Description This command returns a settled reading for the DCX-127 Digital Multi meter(DMM) meter and zeros the ready count.

See Also AP.DCX.DmmMode, AP.DCX.DmmReady, AP.DCX.DmmSettling, AP.DCX.DmmTrig

Example See example for `AP.DCX.DmmMode`.

AP.DCX.DmmRdgRate

Property

Syntax `AP.DCX.DmmRdgRate`

Data Type Integer

0 6 readings per second.
1 25 readings per second.

Description This command sets the DMM reading rate.

Example See example for `AP.DCX.DmmMode`.

AP.DCX.DmmReady

Property

Syntax `AP.DCX.DmmReady`

Data Type Integer

0 Reading not ready.
>0 Reading ready.

Description This command returns the DCX-127 DMM settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.DCX.DmmRdg` or `AP.DCX.DmmTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.DCX.DmmRdg` command will be guaranteed to return quickly.

See Also `AP.DCX.DmmRdg`, `AP.DCX.DmmSettling`, `AP.DCX.DmmTrig`

Example See example for `AP.DCX.DmmMode`.

AP.DCX.DmmScale

Property**Syntax** `AP.DCX.DmmScale`**Data Type** Double**Description** When f(V) (function of Volts) or f(O) (function of Ohms) units are selected for the DMM, APWIN software computes the value to display from the following formula:

$$\text{display} = (\text{measurement} + \text{Offset}) * \text{Scale}$$

The measurement term is the value which would be displayed in Volts or Ohms units. The Offset and Scale values are the contents of the fields with those names, at the top right of the DCX panel.

See Also `AP.DCX.DmmOffset`**Example** See example for `AP.DCX.DmmRangeAuto`.

AP.DCX.DmmSettling

Method**Syntax** `AP.DCX.DmmSettling`(ByVal *Tolerance* As Double, ByVal *Floor* As Double, ByVal *FloorUnit* As String, ByVal *Points* As Integer, ByVal *Delay* As Double, ByVal *Algorithm* As Integer)**Parameters** See Appendix A for Settling Algorithm and parameter name descriptions.**Description** This command sets the settling parameters for the `AP.DCX.DmmRdg` command.**See Also** `AP.DCX.DmmRdg`, `AP.DCX.DmmReady`, `AP.DCX.DmmTrig`**Example** See example for `AP.DCX.DmmMode`.

AP.DCX.DmmTrig

Method**Syntax** `AP.DCX.DmmTrig`

Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP.DCX.DmmRdg</code> command. The reading in progress is aborted.
See Also	<code>AP.DCX.DmmRdg</code> , <code>AP.DCX.DmmReady</code> , <code>AP.DCX.DmmSettling</code>
Example	See example for <code>AP.DCX.DmmMode</code> .

AP.DCX.GateDelay

Property

Syntax `AP.DCX.GateDelay`

Data Type Double Valid settings are from 0.05 to 12.75 sec.

Description This command sets the delay time for the delayed sweep gate, pin #1 on the DCX-127 Program Control Output port transitions low after the defined delay.

Note: When using long delays the sweep duration must be longer than the programmed delay for pin #1 to respond.

Example

```
Sub Main
  AP.Application.NewTest
  AP.Application.PanelOpen apbPanelDCXLarge
  'Program Sweep Gate to transition low 100m sec after
    sweep start.
  AP.Dcx.GateDelay = 0.1

  AP.Gen.Output = True
  AP.Anlr.ChAInput = 2
  AP.Sweep.Start
End Sub
```

AP.DCX.PortAOutput

Property

Syntax `AP.DCX.PortAOutput(ByVal Unit As String)`

Data Type Long The number can be from 0 to 255. Larger numbers are truncated to 8 bits. This value can only be expressed as a decimal value.

Parameters	Part	Description
	<i>Unit</i>	The following units are available: dec, hex, oct.
Description	This command sets DCX-127 Port A 8-bit output value.	
Example	<pre> Sub Main AP.File.OpenTest "DCX1.AT2C" 'Open test. AP.DCX.PortAOutput("Dec") = 17 AP.DCX.PortBOutput("Hex") = 34 AP.DCX.PortCOutput("Oct") = 68 AP.DCX.PortDOutput("Oct") = 68 End Sub </pre>	

AP.DCX.PortBOutput

Property

Syntax	AP.DCX.PortBOutput (ByVal <i>Unit</i> As String)	
Data Type	Long	The number can be from 0 to 255. Larger numbers are truncated to 8 bits. This value can only be expressed as a decimal value.
Parameters	Part	Description
	<i>Unit</i>	The following units are available: dec, hex, oct.
Description	This command sets DCX-127 Port B 8-bit output value.	
Example	See example for AP.DCX.PortAOutput.	

AP.DCX.PortCOutput

Property

Syntax	AP.DCX.PortCOutput (ByVal <i>Unit</i> As String)	
Data Type	Long	The number can be from 0 to 255. Larger numbers are truncated to 8 bits. This value can only be expressed as a decimal value.
Parameters	Part	Description
	<i>Unit</i>	The following units are available: dec, hex, oct.
Description	This command sets DCX-127 Port C 8-bit output value.	

Example See example for `AP.DCX.PortAOutput`.

AP.DCX.PortDOutput

Property

Syntax `AP.DCX.PortDOutput (ByVal Unit As String)`

Data Type Long The number can be from 0 to 255. Larger numbers are truncated to 8 bits. This value can only be expressed as a decimal value.

Parameters	Part	Description
	<i>Unit</i>	The following units are available: dec, hex, oct.

Description This command sets DCX-127 Port D 8-bit output value.

Note: This port is labeled as J141 on the back of the DCX-127.

Example See example for `AP.DCX.PortAOutput`.

User Notes

User Notes

Digital Generator

AP.DGen.BurstInterval

Property

Syntax `AP.DGen.BurstInterval(ByVal Unit As String)`

Data Type Double 2 - 65535

Parameters	Name	Description
	<i>Unit</i>	Cycles or secB.

Description This command sets the number of cycles between the start of a burst and the start of the following burst. This number may be from 2 to 65535 cycles and must be greater than the number of ON cycles. If the number of cycles attempted is not greater than the ON cycles, the interval is not changed.

Note that the interval will occur immediately when this command is called if the burst is running.

See Also `AP.DGen.Wfm`, `AP.DGen.BurstLevel`,
`AP.DGen.BurstOnTime`

Example

```
Sub Main
    AP.Application.NewTest
    AP.DGen.Wfm 0,1
    AP.DGen.Output = 1
    AP.DGen.BurstInterval("Cycles") = 10
    AP.DGen.BurstOnTime("Cycles") = 5
    AP.DGen.BurstLevel("dB") = -40
    Interval = AP.DGen.BurstInterval("Cycles")
    Ontime = AP.DGen.BurstOnTime("Cycles")
    Level = AP.DGen.BurstLevel("%")
    Debug.Print "Burst Interval =" ; Interval ; " cycles."
    Debug.Print "Burst ON time =" ; Ontime ; " cycles."
    Debug.Print "Burst OFF time low level =" ; Level ; " %."
End Sub
```

Example Output Burst Interval = 10 cycles.
Burst ON time = 5 cycles.
Burst OFF time low level = 1 %.

AP.DGen.BurstLevel

Property

Syntax	AP.DGen.BurstLevel (ByVal <i>Unit</i> As String)	
Data Type	Double	Level of signal during burst off time. (0 - -80.25dB)
Parameters	Name	Description
	<i>Unit</i>	The following units are available X/Y, dB, %, PPM.
Description	This command sets the amplitude of the Digital Generator during the burst 'off' time. This is as a percentage of the 'on' amplitude and may range from 100.0 percent to .009716280 percent (-80.25 dB).	
See Also	AP.DGen.Wfm, AP.DGen.BurstInterval, AP.DGen.BurstOnTime	
Example	See example for AP.DGen.BurstInterval.	

AP.DGen.BurstOnTime

Property

Syntax	AP.DGen.BurstOnTime (ByVal <i>Unit</i> As String)	
Data Type	Double	From 1 to AP.Gen.BurstInterval - 1.
Parameters	Name	Description
	<i>Unit</i>	Cycles or secB.
Description	This command sets the number of cycles for the Digital Generator Burst On Time. This number may be from 1 to 65534 cycles and must be less than the number of interval cycles. If the number of cycles attempted is not less than the interval cycles, the ON time is not changed.	
See Also	AP.DGen.Wfm, AP.DGen.BurstInterval, AP.DGen.BurstLevel	
Example	See example for AP.DGen.BurstInterval.	

AP.DGen.ChAAmpl**Property**

Syntax `AP.DGen.ChAAmpl (ByVal Unit As String)`

Data Type Double Valid amplitude settings are 0.0 to 100 %FS.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, PPM, Bits, Vrms, Vp, Vpp, dBu, dBV, dBr

Description This command sets the Digital Generator channel A amplitude.

See Also `AP.DGen.ChBAmpl`, `AP.DGen.RefVFS`

Example

```

Sub Main
    AP.Application.NewTest 'Reset panels
    AP.DGen.Wfm 0,3       'Stereo Sine.
    With AP.DGen
        .RefVFS("V") = 2      'Set Volts/FS ref = 5 volts.
        .RefFreq("Hz") = 10e3  'Set freq ref = 10 kHz.
        .RefdBr("FFS") = 0.5  'Set dBr ref = 2.5 volts.
        .ChBTrackA = False    'Make sure Ch B Track _
                               Ch A is off.
        .ChAFreq("%Hz") = 50  'Set Ch A sinewave frequency.
        .ChBFreq("%Hz") = 75  'Set Ch B sinewave frequency.
        .ChAAmpl("dBr") = 0.0  'Set Ch A = 0.5 FFS.
        .ChBAmpl("dBr") = 0.0  'Set Ch B = 0.5 FFS.
        .ChAInvert = False     'Make sure Ch A invert is OFF.
        .ChBInvert = False     'Make sure Ch B invert is OFF.
        .ChAOutput = True      'Turn Ch A output ON.
        .ChBOutput = True      'Turn Ch A output ON.
        .OutDitherType = 0     'Triangular Dither.
        .Output = True         'Turn main output on (mute off).
    End With

    AP.S2CDsp.Program = 1      'Select DSP Audio Analyzer
    AP.S2CDio.InFormat = 3    'Generator monitor

    With AP.S2CDsp.Analyzer
        Do While (.ChALevelReady = False) Or _
            (.ChBLevelReady = False) Or _

```

```

        (.ChAFreqReady = False) Or _
        (.ChBFreqReady = False)
    Loop
    msg = "Ch A Level = " & Format(.ChALevelRdg("V"), _
        "#.00") & Chr(13)
    msg = msg & "Ch B Level = " & _
        Format(.ChBLevelRdg("V"), "#.00") & Chr(13)
    msg = msg & "Ch A Freq = " & _
        Format(.ChAFreqRdg("Hz"), "#.00") & Chr(13)
    msg = msg & "Ch B Freq = " & _
        Format(.ChBFreqRdg("Hz"), "#.00")
End With
AP.Prompt.Text = msg
AP.Prompt.ShowWithContinue
Stop
End Sub

```

AP.DGen.ChAEqAmpI

Property

Syntax `AP.DGen.ChAEqAmpI (ByVal Unit As String)`

Data Type Double Valid amplitude settings are 0.0 to 100 %FS.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, PPM, Bits, Vrms, Vp, Vpp, dBu, dBV, dB

Description This command sets the Digital Generator channel A post Eq amplitude.

See Also `AP.DGen.ChBEqAmpI`, `AP.DGen.EqCurve`

Example

```

Sub Main
    AP.Application.NewTest
    AP.Application.PanelClose apbPanelAnlrSmall
    AP.Application.PanelClose apbPanelAnalogGenSmall
    AP.Application.PanelOpen apbPanelDigitalGenLarge
    AP.DGen.ChAAmpI("dBFS") = -1.0
    AP.DGen.ChBAmpI("dBFS") = -1.0
    AP.DGen.EqCurve("75us-de.adq", 1) 'Load EQ file
    AP.DGen.Wfm 0, 6 'Select EQ Sine waveform

```

```

AP.DGen.ChAEqAmpl("dBFS") = -1.0
AP.DGen.ChBEqAmpl("dBFS") = -1.0
AP.DGen.Output = True           'Generator Output On

AP.S2CDsp.Program = 1           'Load DSP program
AP.Application.PanelOpen apbPanelDSPLarge

AP.Sweep.Data1.Id = 6014
AP.Sweep.Data1.Top("dBFS") = 0.0
AP.Sweep.Data1.Bottom("dBFS") = -25.0
AP.Sweep.Source1.Id = 5102
AP.Sweep.Stereo = True          'Stereo Sweep
AP.Sweep.Start                  'Start Sweep
End Sub

```

AP.DGen.ChAFreq

Property

Syntax	AP.DGen.ChAFreq (ByVal <i>Unit</i> As String)	
Data Type	Double	Valid frequency range for each channel is 10 - 22.5 kHz.
Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM
Description	This command sets the Digital Generator channel A frequency to be used when the Digital Generator waveform type is set to Sine Stereo.	
See Also	AP.DGen.	
Example	See example for AP.DGen.ChAAmpl.	

AP.DGen.ChAInvert

Property

Syntax	AP.DGen.ChAInvert	
Data Type	Boolean	
	<i>True</i>	Invert channel A output.

False Normal non-inverting output.

Description This command sets output A to normal polarity or inverted polarity (180 degrees out of phase).

See Also `AP.DGen.ChBInvert`

Example See example for `AP.DGen.ChAAmpl`.

AP.DGen.ChAOutput

Property

Syntax `AP.DGen.ChAOutput`

Data Type Boolean

True ON.

False OFF.

Description This command sets the Digital Generator Output A to ON or OFF. The command returns a TRUE if the output is ON and FALSE if the output is OFF.

See Also `AP.DGen.ChBOutput`

Example See example for `AP.DGen.ChAAmpl`.

AP.DGen.ChBAmpl

Property

Syntax `AP.DGen.ChBAmpl (ByVal Unit As String)`

Data Type Double Valid amplitude settings are 0.0 to 100 %FS.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, PPM, Bits, Vrms, Vp, Vpp, dBu, dBV, dBr

Description This command sets the Digital Generator channel B amplitude.

See Also `AP.DGen.ChAAmpl`

Example See example for `AP.DGen.ChAAmpl`.

AP.DGen.ChBEqAmpl

Property

Syntax `AP.Gen.ChBEqAmpl (ByVal Unit As String)`

Data Type Double Valid amplitude settings are 0.0 to 100 %FS.

Parameters

Name	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, PPM, Bits, Vrms, Vp, Vpp, dBu, dBV, dBr

Description This command sets the Digital Generator channel B post Eq amplitude.

See Also `AP.DGen.ChAEqAmpl`, `AP.DGen.EqCurve`

Example See example for `AP.DGen.ChAEqAmpl`.

AP.DGen.ChBFreq

Property

Syntax `AP.DGen.Freq (ByVal Unit As String)`

Data Type Double Valid frequency settings for the Hz unit and sine waveform are 10 - 22.5kHz for the 48kHz sample rate.

Parameters

Name	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM

Description This command sets the Digital Generator channel B frequency when the waveform type is set to Sine Stereo.

See Also `AP.DGen.`

Example See example for `AP.DGen.ChAAmpl`.

AP.DGen.ChBInvert**Property**

Syntax	AP.DGen.ChBInvert	
Data Type	Boolean	
	<i>True</i>	Invert channel B output.
	<i>False</i>	Normal non-inverting output.
Description	This command sets output B to normal polarity or inverted polarity (180 degrees out of phase with normal polarity).	
See Also	AP.DGen.ChAInvert	
Example	See example for AP.DGen.ChAAmpl.	

AP.DGen.ChBOutput**Property**

Syntax	AP.DGen.ChBOutput	
Data Type	Boolean	
	<i>True</i>	On
	<i>False</i>	Off
Description	This command sets the Digital Generator output B to ON or OFF. The command returns a TRUE if the output is ON and a FALSE if the output is OFF.	
See Also	AP.DGen.ChAOutput	
Example	See example for AP.DGen.ChAAmpl.	

AP.DGen.ChBTrackA**Property**

Syntax	AP.DGen.ChBTrackA	
Data Type	Boolean	
	<i>True</i>	ON, channel B amplitude tracks channel A amplitude.
	<i>False</i>	OFF, channel B amplitude independent of channel A.

Description	This command sets the Digital Generator channel B amplitude to the same amplitude as set for channel A.
See Also	AP.DGen.ChAAmpl , AP.DGen.ChBAmpl
Example	See example for AP.DGen.ChAAmpl.

AP.DGen.DitherType

Property

Syntax AP.DGen.DitherType

Data Type Integer

0	Triangular: probability function dither has no noise modulation effect but produces a slightly worse output signal to noise ratio since its maximum amplitude is one LSB. This is normally the preferred choice.
1	Rectangular: probability function dither provides the best signal to noise due to its one-half LSB amplitude, but suffers from modulation noise effects.
2	Shaped: is triangular probability distribution noise with a rising 6 dB/octave slope. This places most of the dither power at higher frequencies where some falls out of band of most devices and where the human hearing system is less sensitive.
3	None:

Description This command sets the Digital Generator Dither Type.

Dither amplitude is automatically set corresponding to the LSB of the value selected in the Output Resolution field or by the AP.S2Dio.OutResolution command.

Dither is random noise of one-half LSB (rectangular) or one LSB (triangular) in amplitude, added to the digital output to improve linearity, reduce distortion, and extend the dynamic range downwards below the theoretical undithered value. The amplitude at which dither is added is determined by the value entered in the Output Resolution field or by the AP.S2Dio.OutResolution command.

Example See example for AP.DGen.ChAAmpl.

AP.DGen.DualAmplRatio

Property**Syntax** `AP.DGen.DualAmplRatio(ByVal Unit As String)`**Data Type** Double Valid settings are 0.00001% to 100%

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: %, dB, PPM, X/Y

Description This command sets the Digital Generator Dual waveform amplitude ratio. The amplitude of Frequency 2 is set relative to the main frequency.**See Also** `AP.DGen.Freq`, `AP.DGen.ChAFreq`, `AP.DGen.ChBFreq`**Example**

AP.DGen.EqCurve

Method**Syntax** `AP.DGen.EqCurve(ByVal FileName As String, ByVal Column As Integer)`

Parameters	Name	Description
	<i>FileName</i>	Long Path and File Names permitted up to 128 characters. The file must be an APWIN Eq file (.adq).
	<i>Column</i>	0 = Source 1 settings. 1 = Data 1 measurements. 2 = Data 2 measurements. 3 = Data 3 measurements. 4 = Data 4 measurements. 5 = Data 5 measurements. 6 = Data 6 measurements. 7 = Source 2 settings.

Result Boolean
True File open successfull.
False File open failed.

Description This command attaches a Eq file to the Digital Generator. Values in the file will be used as multiply factors in czlculating the Digital Generator Amplitude values.

See Also `AP.DGen.ChAEqAmpl`, `AP.DGen.ChBEqAmpl`

Example See example for `AP.DGen.ChAEqAmpl`.

AP.DGen.Freq

Property

Syntax `AP.DGen.Freq(ByVal Unit As String)`

Data Type Double Valid frequency settings for the Hz unit and sine waveform are 10 - 22.5kHz for the 48kHz sample rate.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM

Description This command sets the Digital Generator main frequency for the Sine waveforms.

AP.DGen.IMCenterFreq

Property

Syntax `AP.DGen.IMCenterFreq(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM

Description This command sets the Digital Generator IMD Center Frequency. The frequency passed is rounded to the closest available value.

Set the Digital Generator waveform to an IMD CCIF before calling this command.

See Also `AP.DGen.Wfm`, `AP.DGen.IMFreq`

AP.DGen.IMFreq

Property

Syntax	<code>AP.DGen.IMFreq(ByVal Unit As String)</code>	
Data Type	Double	For a SMPTE mode waveform, this is the lower frequency tone. The following frequency range is available for SMPTE and CCIF IMD waveforms for the 48kHz sample rate: 10 Hz to 22.5kHz.
Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command. Hz
Description	This command sets the Digital Generator IMD frequency. The frequency passed is rounded to the closest available value. Set the Digital Generator waveform to IMD (SMPTE or CCIF) before calling this command.	
See Also	<code>AP.DGen.Wfm</code> , <code>AP.DGen.IMCenterFreq</code>	

AP.DGen.IMHighFreq

Property

Syntax	<code>AP.DGen.IMHighFreq(ByVal Unit As String)</code>	
Data Type	Double	
Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command. Hz
Description	This command sets the Digital Generator IMD High Frequency. The frequency passed is rounded to the closest available value. Set the Digital Generator waveform to an IMD SMPTE before calling this command.	
See Also	<code>AP.DGen.Wfm</code> , <code>AP.DGen.IMFreq</code>	

AP.DGen.Offset

Property

Syntax `AP.Gen.Offset (ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, Bits, Vrms, Vp, Vpp, dBu, dBV, dBr, dBrlnv

Description This command sets the Digital Generator Sine + Offset waveform Offset value.

See Also `AP.DGen.Wfm`

AP.DGen.Output

Property

Syntax `AP.Gen.Output`

Data Type Boolean

<i>True</i>	On
<i>False</i>	Off

Description This command sets the Digital Generator channel A and B outputs to ON or OFF if they have been individually enabled by the `AP.DGen.ChAOutput` and `AP.DGen.ChBOutput` commands.

See Also `AP.DGen.ChAOutput`, `AP.DGen.ChBOutput`

Example See example for `AP.DGen.ChAAmpl`.

AP.DGen.Phase

Property

Syntax `AP.DGen.Phase (ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: deg

Description This command sets the Digital Generator Phase value.

Set the Digital Generator waveform to Sine Var Phase before calling this command.

See Also AP.DGen.Wfm

Example

```
Sub Main
    AP.Application.NewTest
    AP.Application.PanelClose apbPanelAnalogGenSmall
    AP.Application.PanelOpen apbPanelDigitalGenSmall
    AP.DGen.Wfm 0, 2
    AP.DGen.Phase("deg") = 90.000000
    AP.DGen.Output = True
    'Send digital signal through D/A converter.
    AP.Sweep.Data1.Id = 5905
    AP.Sweep.SinglePoint = True
    AP.Sweep.Start
    Debug.Print "Channel B is " &
    Format(AP.Data.Value(0,1,0,"deg"),"###.000") & " deg
    relative to channel A"
End Sub
```

Example Output Channel B is 89.994 deg relative to channel A.

AP.DGen.RefdBr

Property

Syntax **AP.DGen.RefdBr**(ByVal *Unit* As String)

Data Type Double Amplitude value.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, Bits, V, Vp, Vpp, dBu, dBV

Description	This command sets the zero dBr value for the Digital Generator dBr unit.
See Also	AP.DGen.ChAAmpl, AP.DGen.ChBAmpl, AP.DGen.RefVFS
Example	See example for AP.DGen.ChAAmpl.

AP.DGen.RefFreq

Property

Syntax `AP.DGen.RefFreq(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command: Hz

Description This command sets the Digital Generator relative frequency reference value. This reference is used for all the Digital Generator relative frequency units (F/R, dHz, %Hz, cent, octs, decs, d%, dPPM)

See Also AP.DGen.Freq, AP.DGen.ChAFreq, AP.DGen.ChBFreq

Example See example for AP.DGen.ChAAmpl.

AP.DGen.RefVFS

Property

Syntax `AP.Gen.RefVFS(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command: V

Description This command sets the Digital Generator relative Volts Full Scale (VFS) reference value. This reference is used for all the Digital Generator relative amplitude units (Vrms, Vp, Vpp, dBu, dBV)

Example See example for AP.DGen.ChAAmpl.

AP.DGen.StepRate

Property**Syntax** `AP.DGen.StepRate`**Data Type** Double**Description** This command sets the rate at which the Digital Generator Special Walking Ones, and Walking Zeros waveform changes state. If the `AP.DGen.StepRate` command is set to 5 the Digital Generator will output five words with the same bit pattern and then change to the next bit pattern.**See Also** `AP.DGen.Wfm`

AP.DGen.Wfm

Method**Syntax** `AP.DGen.Wfm` ByVal *Primary* As Integer [, ByVal *Secondary* As Variant]**Parameters**

Name	Description
<i>Primary</i>	This parameter defines the basic waveform type.
<i>Secondary</i>	This parameter defines the basic waveform modifier.

Primary	Secondary	Description
0		Sine
	0	Normal
	1	Burst
	2	Var Phase
	3	Stereo
	4	Dual
	5	Sine + Offset
	6	EQ Sine
	6	Shaped Burst
1		Square.
2		IMD
	0	SMPTE/DIN 4:1

	1	CCIF/DFD
	2	DIM 100
	3	DIM 30
	4	DIM B
	5	SMPTE/DIN 1:1
3		Noise
	0	Pink
	1	White
	2	Burst USASI
4		Special
	0	Monotonicity
	1	J-Test
	2	Polarity
	3	Walking Ones
	4	Walking Zeros
	5	Constant Value
	6	Bittest Random
	7	Pass Thru
5		MLS
	0	Pink #1
	1	Pink #2
	2	Pink #3
	3	Pink #4
	4	White #1
	5	White #2
	6	White #3
	7	White #4
6		Arb Wfm

Description

This command sets the Digital Generator waveform. The table above shows the possible settings for the AP.DGen.WfmName command.

Example

See example for AP.DGen.WfmName.

AP.DGen.WfmName**Property****Syntax**

AP.DGen.WfmName

Data Type String Long Path and File Names permitted up to 128 characters.

Description This command loads the designated arbitrary waveform file (.AGM or .AGS) into the Digital Generator. A Mono waveform (.AGM) is loaded into both the 1 and 2 generator buffers.

Buffer 1 : This buffer is associated with the DSP channel 1.

Buffer 2 : This buffer is associated with the DSP channel 2.

Note: This command can also be used to control the Analog Generator arbitrary waveform file selection.

See Also AP.DGen.Wfm

Example

```
Sub Main
  AP.Application.NewTest
  AP.Application.PanelClose apbPanelAnalogGenSmall
  AP.Application.PanelClose apbPanelAnlrSmall
  AP.Application.PanelOpen apbPanelDigitalGenSmall
  AP.Application.PanelOpen apbPanelDSPSmall
  'Load Digital Analyzer (Multitone Audio Analyzer)
  AP.S2CDsp.Program = 4
  AP.Application.PanelOpen apbPanelDigIOSmall
  'Select Gen Mon on the Digital I/O panel to route the
  ' Digital generator directly To the Digital Analyzer.
  AP.S2Dio.InFormat = 3
  AP.DGen.Wfm 6          'Select arbitrary waveform
  AP.DGen.WfmName = AP.Application.WorkingDir & _
    "Iso31.agm"
  AP.DGen.Output = True  'Digital Generator Output ON
  'Set up Sweep panel to display test data.
  AP.Application.PanelOpen apbPanelSweepSmall
  AP.Sweep.Data1.Id = 6309
  AP.Sweep.Data1.Top("dBFS") = 0.000000

  AP.Sweep.Source1.Id = 5621
  AP.Sweep.Source1.Steps = 200
  AP.Sweep.Stereo = True

  AP.Sweep.Start          'Run Test
  AP.Graph.OptimizeLeft
End Sub
```

User Notes

User Notes

APEvent_OnAuxSetting1

Event

Syntax `APEvent_OnAuxSetting1(ByVal Value As Double)`

Parameters	Part	Description
	<i>Value</i>	Sweep source or settings bargraph control current value.

Description This event is called when a sweep source or settings bargraph control changes which in turn generates this event, for example, the Instrument parameter "Aux.Setting 1 (Double)" used as the Sweep panel source or Bargraph browser ID. Events are generated as the sweep runs or as the user manipulates the settings bargraph control.

Example

```
Public Halt As Boolean
Sub Main
    Halt = False

    AP.Application.NewTest
    AP.Gen.ChBTrackA = False
    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2
    AP.Anlr.ChBInput = 2

    ID = AP.BarGraph.New(6271) 'Aux.Setting 1 (Double)
    AP.BarGraph.AxisLeft(ID, "") = 0.0
    AP.BarGraph.AxisRight(ID, "") = 2.0
    AP.BarGraph.AxisIncrement(ID, "") = 0.1

    ID = AP.BarGraph.New(6275) 'Aux.Reading 1 (Double)
    AP.BarGraph.AxisLeft(ID, "") = 0.2
    AP.BarGraph.AxisRight(ID, "") = 4.0

    AP.Aux.Setting1 = 1.0
    AP.Application.SetWatchDogTimer1(5.0, False)

    Do
    Loop While Halt = False
```



```

End Sub
Sub APEvent_OnAuxSetting1(ByVal Value As Double)
    AP.Gen.ChAAmpl("Vrms") = Value - .2
    AP.Gen.ChBAmpl("Vrms") = Value + .2

    AP.Aux.Reading1Settling 3.0, 0.0, 1, 0.0, 0
    AP.Aux.Reading1Trig
    ReadyCount = AP.Aux.Reading1Ready
    AP.Aux.SetReading1 AP.Gen.ChAAmpl("Vrms") + _
        AP.Gen.ChBAmpl("Vrms")
    ReadyCount = AP.Aux.Reading1Ready
End Sub
Sub APEvent_OnWatchDogTimeout(ByVal Id As Long)
    If Id = 1 Then
        Halt = True
    End If
End Sub

```

APEvent_OnAuxSetting2

Event

Syntax **APEvent_OnAuxSetting2**(ByVal *Value* As Double)

Parameters	Part	Description
	<i>Value</i>	Sweep source or settings bargraph control current value.

Description This event is called when a sweep source or settings bargraph control changes which in turn generates this event, for example, the Instrument parameter "Aux.Setting 2 (Double)" used as the Sweep panel source or Bargraph browser ID. Events are generated as the sweep runs or as the user manipulates the settings bargraph control.

Example See example for APEvent_OnAuxSetting1.

APEvent_OnAuxSetting3

Event

Syntax **APEvent_OnAuxSetting3**(ByVal *Value* As Long)

Parameters	Part	Description
	<i>Value</i>	Sweep source or settings bargraph control current value.

Description This event is called when a sweep source or settings bargraph control changes which in turn generates this event, for example, the Instrument parameter "Aux.Setting 3 (Long)" used as the Sweep panel source or Bargraph browser ID. Events are generated as the sweep runs or as the user manipulates the settings bargraph control.

Example See example for APEvent_OnAuxSetting1.

APEvent_OnAuxSetting4

Event

Syntax	APEvent_OnAuxSetting4 (ByVal <i>Value</i> As Long)	
Parameters	Part	Description
	<i>Value</i>	Sweep source or settings bargraph control current value.
Description	This event is called when a sweep source or settings bargraph control changes which in turn generates this event, for example, the Instrument parameter "Aux.Setting 4 (Long)" used as the Sweep panel source or Bargraph browser ID. Events are generated as the sweep runs or as the user manipulates the settings bargraph control.	
Example	See example for APEvent_OnAuxSetting1.	

APEvent_OnDcxProgramControlInput

Event

Syntax	APEvent_OnDcxProgramControlInput (ByVal <i>Value</i> As Long)	
Parameters	Part	Description
	<i>Value</i>	The value returned is the pin number of the DCX-127 Program Control Input connector pin that is pulled low. A zero (0) is returned if more than one button is pressed at a time.
Description	This event is called when one of the DCX-127 Program Control Input pins (1) is momentarily shorted to ground (Pin 9).	
Example	Dim Halt As Boolean	

```

Sub Main
    Halt = False
    Do
        Loop While Halt = False
    End Sub
Sub APEvent_OnDcxProgramControlInput(Value As Long)
    Debug.Print "Program Control = " & Value
    If Value = 0 Then Debug.Print "More than one button _
        pressed."
    If Value = 8 Then Halt = True
End Sub

```

APEvent_OnError

Event

Syntax

```
APEvent_OnError(ByVal Value As Long)
```

Parameters

Part	Description
<i>Value</i>	Error value as defined in Appendix D Error Codes

Description

This event is called when an Error is encountered.

Example

```

Sub Main
    AP.Gen.ChAAmpl("Vrms") = 111.9    'Cause an error _
        and see what happens.
End Sub
Sub APEvent_OnError(Code As Long)
    Debug.Print "Got number " & Code & " " & _
        AP.Application.GetCurrentErrorString

    ' If you are going to handle the error, then call
    ' AP.Application.ClearCurrentError before you exit
    ' this subroutine to stop APWIN from displaying the
    ' error,

    AP.Application.ClearCurrentError

    ' It is also preferable to call
    ' AP.Application.ClearCurrentError before you
    ' make any other calls into APWIN in case these
    ' calls also generate an unexpected error

```

```
End Sub
```

APEvent_OnSweepEnd

Event

Syntax

```
APEvent_OnSweepEnd
```

Description

This event is called when the sweep has terminated and the initial source value has been restored.

Example

```
Sub Main
    AP.Sweep.SinglePoint = True
    AP.Sweep.Start
End Sub
Sub APEvent_OnSweepStart()
    Debug.Print "Sweep Start"
End Sub
Sub APEvent_OnSweepNestStart(Source As Long)
    Debug.Print "Sweep Nest Start "
End Sub
Sub APEvent_OnSweepStep(Value As Variant, Source As _
    Long)
    Debug.Print "Sweep Step = " & Value
End Sub
Sub APEvent_OnSweepTrigger()
    Debug.Print "Sweep Trigger"
End Sub
Sub APEvent_OnSweepStepEnd()
    Debug.Print "Sweep Step End"
End Sub
Sub APEvent_OnSweepNestEnd()
    Debug.Print "Sweep Nest End"
End Sub
Sub APEvent_OnSweepEnd()
    Debug.Print "Sweep End"
End Sub
```

Example Output

```
Sweep Start
Sweep Nest Start
Sweep Step = 20000
Sweep Trigger
```

```
Sweep Step End
Sweep Nest End
Sweep Step = 1000
Sweep End #
```

APEvent_OnSweepNestEnd

Event

Syntax	<code>APEvent_OnSweepNestEnd</code>
Description	This event is called after a single sweep is completed.
Example	See example for <code>APEvent_OnSweepEnd</code> .

APEvent_OnSweepNestStart

Event

Syntax	<code>APEvent_OnSweepNestStart(ByVal Source As Long)</code>	
Parameters	Part	Description
	<i>Source</i>	Sweep panel Source 1 Step value.
Description	This event is called before the first step of a sweep.	
Example	See example for <code>APEvent_OnSweepEnd</code> .	

APEvent_OnSweepReverseChannels

Event

Syntax	<code>APEvent_OnSweepReverseChannels(ByVal Reverse As Long)</code>	
Parameters	Part	Description
	<i>Reverse</i>	0 = Channels reversed. 1 = Channels Restored.
Description	This event is called when a stereo sweep is performed that requires channel switching. If the parameter is 1 then the channels are being reversed in preparation for the second half of a stereo sweep. If the parameter is 0 then the channels are being restored to their pre-sweep condition.	

Example

```

Sub Main
    AP.Sweep.Data1.Id = 5906 'Analog Analyzer Function _
        Meter
    AP.Sweep.SinglePoint = True
    AP.Sweep.Stereo = True
    AP.Sweep.Start
End Sub
Sub APEvent_OnSweepReverseChannels(ByVal Reverse As _
    Long)
    If Reverse = 0 then
        Debug.Print "Channels Restored"
    ElseIf Reverse = 1 then
        Debug.Print "Channels Reversed"
    End If
End Sub
Sub APEvent_OnSweepStart()
    Debug.Print "Sweep Start"
End Sub
Sub APEvent_OnSweepNestStart(Source As Long)
    Debug.Print "Sweep Nest Start "
End Sub
Sub APEvent_OnSweepStep(Value As Variant, Source As
Long)
    Debug.Print "Sweep Step = " & Value
End Sub
Sub APEvent_OnSweepTrigger()
    Debug.Print "Sweep Trigger"
End Sub
Sub APEvent_OnSweepStepEnd()
    Debug.Print "Sweep Step End"
End Sub
Sub APEvent_OnSweepNestEnd()
    Debug.Print "Sweep Nest End"
End Sub
Sub APEvent_OnSweepEnd()
    Debug.Print "Sweep End"
End Sub

```

Example Output

```

Sweep Start
Sweep Nest Start
Sweep Step = 20000

```

```

Sweep Trigger
Sweep Step End
Sweep Nest End
Channels Reversed
Sweep Step = 1000
Sweep Nest Start
Sweep Step = 20000
Sweep Trigger
Sweep Step End
Sweep Nest End
Channels Restored
Sweep Step = 1000
Sweep End

```

APEvent_OnSweepStart

Event

Syntax `APEvent_OnSweepStart`

Description This event is called at the start of a sweep. It prepares for the rest of the upcoming sweep by storing the initial sweep value and pre-calculating steps.

See Also `APEvent_OnSweepEnd`

Example See example for `APEvent_OnSweepEnd`.

APEvent_OnSweepStep

Event

Syntax `APEvent_OnSweepStep(ByVal Value As Variant, ByVal Source as Long)`

Parameters	Part	Description
	<i>Value</i>	Setting value.
	<i>Source</i>	Indicates Source 1 or Source 2 settings (1 or 2 only).

Description This event is called after the setting for this sweep has been done.

Example See example for `APEvent_OnSweepEnd`.

APEvent_OnSweepStepEnd

Event

Syntax `APEvent_OnSweepStepEnd`**Description** This event is called after a reading cycle has completed. The reading cycle may return up to six settled measurements.**Example** See example for `APEvent_OnSweepEnd`.

APEvent_OnSweepTrigger

Event

Syntax `APEvent_OnSweepTrigger`**Description** This event is after a new step value is sent, to trigger a new reading cycle.**Example** See example for `APEvent_OnSweepEnd`.

APEvent_OnWatchDogTimeout

Event

Syntax `APEvent_OnWatchDogTimeout(ByVal Id As Long)`**Parameters**

Part	Description
<i>Id</i>	Timer identification 1 or 2.

Description This event is called when one of the two WatchDog Timers has expired.

Example

```
Dim Halt As Boolean
Sub Main
    Halt = False
    AP.Application.NewTest    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2
    AP.Sweep.Source1.Steps = 30
    AP.Application.SetWatchDogTimer1(5.0,False)
    AP.Sweep.StartNoWait
    Do
        Loop While Halt = False
    End Sub
Sub APEvent_OnWatchDogTimeout(ByVal Id As Long)
    If Id = 1 Then
```



```
    If AP.Sweep.IsRunning = True Then
        AP.Sweep.Stop
        Debug.Print "Sweep Stopped"
    End If
End If
End Sub
```

User Notes

User Notes

AP.File.AppendData**Method****Syntax** `AP.File.AppendData(ByVal FileName As String)`

Parameters	Name	Description
	<i>FileName</i>	Long Path and File Names permitted up to 128 characters.

Result	Boolean	Description
	<i>True</i>	File Data Append successfull.
	<i>False</i>	File Data Append failed.

Description This command appends data from the designated data file into memory. This comand will only load data from a data file that has identical Sweep panel Data 1-6 and Source 1-2 instrument parameters.

See Also `AP.File.AppendTest`

AP.File.ExportASCIIData**Method****Syntax** `AP.File.ExportASCIIData(ByVal FileName As String)`

Parameters	Name	Description
	<i>FileName</i>	Long Path and File Names permitted up to 128 characters.

Result	Boolean	Description
	<i>True</i>	File export successfull.
	<i>False</i>	File export failed.

Description This command saves the measurement data in memory to a coma delimited ASCII text file.

See Also `AP.File.ImportASCIIData`

Example

```

Sub Main
'Smooth Data from ASCII Data file
  AP.Application.NewTest
'Load ASCII data file
  AP.File.ImportASCIIIData( "TEMP.ADX" )
  AP.Application.PanelOpen apbPanelGraph
  AP.Sweep.Data1.LogLin = 1
  AP.Graph.OptimizeLeft
  If AP.Sweep.Data1.Id <> 5049 Then _
    AP.Compute.Smooth.Data(1) = True
  If AP.Sweep.Data2.Id <> 5049 Then _
    AP.Compute.Smooth.Data(2) = True
  If AP.Sweep.Data3.Id <> 5049 Then _
    AP.Compute.Smooth.Data(3) = True
  If AP.Sweep.Data4.Id <> 5049 Then _
    AP.Compute.Smooth.Data(4) = True
  If AP.Sweep.Data5.Id <> 5049 Then _
    AP.Compute.Smooth.Data(5) = True
  If AP.Sweep.Data6.Id <> 5049 Then _
    AP.Compute.Smooth.Data(6) = True
  AP.Compute.Smooth.Auto = True
  AP.Compute.Smooth.Apply
'Export ASCII data file
  AP.File.ExportASCIIIData( "TEMP.ADX" )
End Sub

```

AP.File.ExportGraphic**Method****Syntax**

```

AP.File.ExportGraphic(ByVal FileName As String,
ByVal Type As Integer)

```

Parameters

Name	Description
<i>FileName</i>	Long Path and File Names permitted up to 128 characters.
<i>Type</i>	0 = Windows Meta File (.WMF). 1 = Windows Extended Meta File (.EMF).

Result

Boolean	
<i>True</i>	File export successful.
<i>False</i>	File export failed.

Description This command saves the current graph measurement data in memory to the designated file.

Example

```

Sub Main
  On Error Resume Next
  AP.Application.NewTest
  AP.Gen.Output = True
  AP.Anlr.ChAInput = 2
  AP.Anlr.FuncFilterLP = 0
  AP.Anlr.FuncFilterHP = 3
  AP.Sweep.Data1.Id = 5906
  AP.Sweep.Source1.Start("Hz") = 50000
  AP.Sweep.Start

  Kill "C:\GRAPH.EMF"

  'Export Windows Meta File
  blnExport = AP.File.ExportGraphic("C:\GRAPH.EMF", 1)
  If blnExport = False Then End

  Dim MSWord As Object
  Set MSWord = CreateObject("Word.Basic")' Start Word
  MSWord.AppShow      'Word is invisible on startup.
                      'Set to visible
  MSWord.FileOpen Name:= "C:\GENERIC.DOC"
  MSWord.EditFind "Graph"      'Search for string
  'Import Windows Meta File Graph
  MSWord.InsertPicture "C:\GRAPH.EMF"
  MSWord.FilePrint            'Print Doc from MS Word
  Wait 10
  MSWord.FileCloseAll 2      'Close all open files
  MSWord.AppClose            'Close MS Word
End Sub

```

AP.File.ImportASCIIData**Method****Syntax** `AP.File.ImportASCIIData (ByVal FileName As String)`

Parameters	Name	Description
	<i>FileName</i>	Long Path and File Names permitted up to 128 characters.

Result	Value	Description
	<i>True</i>	File import successful.
	<i>False</i>	File import failed.

Description This command loads into memory the designated ASCII data file. This command only loads files that have been exported from APWIN or conform to the APWIN ASCII data file format.**See Also** `AP.File.ExportASCIIData`**Example** See example for `AP.File.ExportASCIIData`.**AP.File.ImportDOSS1Test****Method****Syntax** `AP.File.ImportDOSS1Test (ByVal FileName As String[, Optional ByVal IgnoreAttachedFiles As Variant])`

Parameters	Name	Description				
	<i>FileName</i>	Long Path and File Names permitted up to 128 characters.				
	<i>IgnoreAttachedFiles</i>	<table border="0"> <tr> <td><i>True</i></td> <td>= Remove attached files.</td> </tr> <tr> <td><i>False</i></td> <td>= Keep attached files.</td> </tr> </table>	<i>True</i>	= Remove attached files.	<i>False</i>	= Keep attached files.
<i>True</i>	= Remove attached files.					
<i>False</i>	= Keep attached files.					

Result	Value	Description
	<i>True</i>	File import successful.
	<i>False</i>	File import failed.

Description This command translates S1.EXE test files to the APWIN System One configuration only.**See Also** `AP.Application.SysType`

AP.File.OpenData

Method**Syntax** `AP.File.OpenData(ByVal FileName As String)`**Parameters**

Name	Description
<i>FileName</i>	Long Path and File Names permitted up to 128 characters.

Result

Boolean

True File open successfull.*False* File open failed.**Description**

This command loads the designated data file.

Example

```

Sub Main
    OpenResult = AP.File.OpenTest("FRQ-RESP.AT1")
    If OpenResult = False Then Call Open_Failed
    OpenResult = AP.File.OpenData("FRQ-RESP.DAT")
    If OpenResult = False Then Call Open_Failed
    AP.Data.UpdateDisplay 0
    Wait 5
    OpenResult = AP.File.OpenTest("THD-FRQ.AT1")
    If OpenResult = False Then Call Open_Failed
    OpenResult = AP.File.OpenData("THD-FRQ.DAT")
    If OpenResult = False Then Call Open_Failed
    AP.Data.UpdateDisplay 0
    Wait 5
    OpenResult = AP.File.OpenTest("RESIDNOI.AT1")
    If OpenResult = False Then Call Open_Failed
    OpenResult = AP.File.OpenData("RESIDNOI.DAT")
    If OpenResult = False Then Call Open_Failed
    AP.Data.UpdateDisplay 0
    Wait 5
End Sub

Sub Open_Failed
    Debug.Print"File Open FAILED."
End
End Sub

```


AP.File.OpenMacro

Method**Syntax** `AP.File.OpenMacro(ByVal FileName As String)`

Parameters	Name	Description
	<i>FileName</i>	Long Path and File Names permitted up to 128 characters.

Result

Boolean

<i>True</i>	Not applicable.
<i>False</i>	File open failed.

Description This command loads the designated file into the macro editor and automatically runs the macro.**Example**

```
'Visual Basic example
Private Sub Form_Load()
    Dim AP As Object
    'Create OLE link to APWIN.
    Set AP = CreateObject("APWIN.Application")
    AP.Application.Visible = True ' Make APWIN visible

    'Place your code here

    'Run an APWIN Macro and wait for it to finish
    AP.File.OpenMacro "C:\BUSY.APB"
    While AP.Macro.IsRunning = True
    Wend

    'Change Visual Basic directory to APWIN Working _
    Directory.
    ChDir AP.Application.MacroDir

    'Place your code here

    AP.Application.Quit           'Quit APWIN
    End
End Sub
```

AP.File.OpenTest

Method

Syntax `AP.File.OpenTest(ByVal FileName As String)`

Parameters	Name	Description
	<i>FileName</i>	Long Path and File Names permitted up to 128 characters.

Result

Boolean

<i>True</i>	File open successfull.
<i>False</i>	File open failed.

Description This command loads the designated test file.**Example**

```

Sub Main
  OpenResult = AP.File.OpenTest ("FRQ-RESP.AT1")
  If OpenResult = False Then Call Open_Failed
  AP.Sweep.Start
  SaveResult = AP.File.SaveDataAs ("FRQ-RESP.DAT")
  If SaveResult = False Then Call Save_Failed

  OpenResult = AP.File.OpenTest ("THD-FRQ.AT1")
  If OpenResult = False Then Call Open_Failed
  AP.Sweep.Start
  SaveResult = File.SaveDataAs ("THD-FRQ.DAT")
  If SaveResult = False Then Call Save_Failed

  OpenResult = AP.File.OpenTest ("RESIDNOI.AT1")
  If OpenResult = False Then Call Open_Failed
  AP.Sweep.Start
  SaveResult = AP.File.SaveDataAs ("RESIDNOI.DAT")
  If SaveResult = False Then Call Save_Failed
End Sub
Sub Open_Failed
  Debug.Print "Test Open FAILED."
End
End Sub
Sub Save_Failed
  Debug.Print "Test Save FAILED."
End
End Sub

```

AP.File.OpenWfm**Method**

Syntax `AP.File.OpenWfm(ByVal FileName As String, ByVal siOption1 As Integer, ByVal siOption2 As Integer)`

Parameters

Name	Description
<i>FileName</i>	Long Path and File Names permitted up to 128 characters.
<i>siOption1</i>	This option defines the buffer that the first waveform in a waveform file is loaded into. 0 = None 1 = Acquisition buffer 1 2 = Acquisition buffer 2 3 = Transform buffer 1 4 = Transform buffer 2
<i>siOption2</i>	This option defines the buffer that the second waveform in a two-waveform file is loaded into. 0 = None 1 = Acquisition buffer 1 2 = Acquisition buffer 2 3 = Transform buffer 1 4 = Transform buffer 2

Result

Boolean	
<i>True</i>	File open successfull.
<i>False</i>	File open failed.

Description

This command loads the designated waveform file into the analyzer or generator buffers designated by Option1 and 2.

Comments

Acquisition buffer : This buffer holds waveform data that has been generated by executing an acquisition (F9). Opening a waveform file containing a previously-acquired and saved waveform and specifying the acquisition buffer as the destination permits further analysis of the waveform including FFT spectrum analysis and waveform display.

Transform buffer : The transform buffer is the sub-section of the acquisition buffer starting at the FFT start time with a length equal to the presently-set FFT length.

Buffer 1 : This buffer is associated with the DSP channel 1.

Buffer 2 : This buffer is associated with the DSP channel 2.

Recommended file extensions :

Extension	Description
.AAM	Acquired waveform, 1 channel
.AAS	Acquired waveform, 2 channels

Example

```
Sub Main
    AP.File.OpenTest ("FFTSAVE.AT2")
    OpenResult = AP.File.OpenWfm("TEMP.AAS", 1,2)
    If OpenResult = False Then Call Open_Failed
    AP.Sweep.Reprocess
End Sub

Sub Open_Failed
    Debug.Print "Waveform Open FAILED."
End Sub
```

AP.File.SaveAll

Method

Syntax `AP.File.SaveAll`

Description This command saves the current test and all macros loaded in the macro editor.

AP.File.SaveDataAs

Method

Syntax `AP.SaveDataAs`(ByVal *FileName* As String)

Parameters	Name	Description
	<i>FileName</i>	Long Path and File Names permitted up to 128 characters.
Result	Boolean	
	<i>True</i>	File save successful.
	<i>False</i>	File save failed.

Description This command saves the measurement data in memory to the designated file.

Example See example for `AP.File.OpenTest`.

AP.File.SaveTest

Method

Syntax `AP.File.SaveTest`

Result Boolean

True File save successful.
False File save failed.

Description This command saves the current test.

Example

```
Sub Main
    AP.File.OpenTest "FRQ-RESP.AT2" 'Open frequency _
        response test.
    Sweep.Start 'Start sweep.
    If AP.File.SaveTest = False Then GoTo Quit 'Save Test
    AP.File.OpenTest "THD-FRQ.AT2" 'Open total _
        harmonic distortion + noise test.
    AP.Sweep.Start 'Start sweep.
    If AP.File.SaveTest = False Then GoTo Quit 'Save Test
    AP.File.OpenTest "RESIDNOI.AT2" 'Open residual _
        noise test.
    AP.Sweep.Start 'Start sweep.
    If AP.File.SaveTest = False Then GoTo Quit 'Save Test
End
Quit:
Debug.Print "Test Save FAILED"
End Sub
```

AP.File.SaveTestAs

Method

Syntax `AP.File.SaveTestAs(ByVal FileName As String)`

Parameters	Name	Description
	<i>FileName</i>	Long Path and File Names permitted up to 128 characters.
Result	Boolean	
	<i>True</i>	File save successfull.
	<i>False</i>	File save failed.
Description	This command saves the current test as defined by the panels to the designated file. The data currently in memory as well as panel and page configuration information is also saved in the test file.	
Example	<pre> Sub Main OpenResult = AP.File.OpenTest("FRQ-RESP.AT1") If OpenResult = False Then Call Open_Failed AP.Sweep.Start SaveResult = AP.File.SaveTestAs("FRQ-RESP.AT1") If SaveResult = False Then Call Save_Failed OpenResult = AP.File.OpenTest("THD-FRQ.AT1") If OpenResult = False Then Call Open_Failed AP.Sweep.Start SaveResult = AP.File.SaveTestAs("THD-FRQ.AT1") If SaveResult = False Then Call Save_Failed OpenResult = AP.File.OpenTest("RESIDNOI.AT1") If OpenResult = False Then Call Open_Failed AP.Sweep.Start SaveResult = AP.File.SaveTestAs("RESIDNOI.AT1") If SaveResult = False Then Call Save_Failed End Sub Sub Open_Failed Debug.Print"Test Open FAILED." End End Sub Sub Save_Failed Debug.Print"Test Save FAILED." End End Sub </pre>	

AP.File.SaveWfmAs

Method

Syntax `AP.File.SaveWfmAs (ByVal FileName As String, ByVal siOption1 As Integer, ByVal siOption2 As Integer)`

Parameters	Name	Description
	<i>FileName</i>	Long Path and File Names permitted up to 128 characters.
	<i>siOption1</i>	This option determines the source of the waveform to be stored in the first section of the disk file. 1 = Acquisition buffer 1 2 = Acquisition buffer 2 3 = Transform buffer 1 4 = Transform buffer 2
	<i>siOption2</i>	This option determines the source of the waveform to be stored in the last section of the disk file. 0 = None 1 = Acquisition buffer 1 2 = Acquisition buffer 2 3 = Transform buffer 1 4 = Transform buffer 2
Result	Boolean	
	<i>True</i>	File save successful.
	<i>False</i>	File save failed.

Description This command saves waveform data contained in the buffers designated by Option #1 and #2 into the designated file. The waveform designated by Option #1 saves to the first section of the file and the Option #2 waveform to the last section of the file.

Comment Acquisition buffer : This buffer holds waveform data captured into DSP memory by an acquisition (F9). Selecting the acquisition buffer causes the complete acquired signal to be saved to a disk file for later download (via the `AP.File.OpenWfm` command) for further analysis including FFT spectrum analysis and waveform display.

Transform buffer : The transform buffer is the sub-section of the acquisition buffer starting at the FFT start time with a length equal to the presently-set FFT length. Selecting this option results in a smaller disk file since only a portion of the acquired signal is saved.

Buffer 1 : This buffer is associated to the DSP channel 1.

Buffer 2 : This buffer is associated to the DSP channel 2.

Recommended file extensions :

Extension	Description
<i>.AAM</i>	Acquired waveform, 1 channel
<i>.AAS</i>	Acquired waveform, 2 channels

Example

```
Sub Main
  AP.File.OpenTest "FFTSAVE.AT2"
  AP.Sweep.Start
  AP.File.SaveWfmAs "TEMP.AAS", 1,2
End Sub
```


User Notes

Analog Generator

AP.Gen.Ampl

Property

Syntax `AP.Gen.Ampl (ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Vrms, Vp, Vpp, dBu, dBV, dBr, dBm, W, dBrlnv

Description This command sets the Analog Generator channel A and B amplitude.

Example

```

Sub Main
    Dim reading(1 To 31)    'Dimension array.
    ndx = 1                'Array index.
    AP.Application.NewTest 'Reset panels
    AP.Gen.Output = True   'Turn output ON.
    AP.Anlr.ChAInput = 2  'Select GENMON internal _
                          connection.
    AP.Gen.Ampl("Vrms") = 5    'Set output level to 5V.
    'Sweep 20 Hz to 20 kHz in 30 linear steps.
    For NewFreq = 20 To 20e3 Step (20e3 - 20)/30
        AP.Gen.Freq ("Hz") = NewFreq
        'Measure amplitude from DUT.
        reading(ndx) = AP.Anlr.FuncRdg ("V")
        ndx = ndx + 1
    Next
End Sub

```

AP.Gen.BurstInterval

Property

Syntax `AP.Gen.BurstInterval(ByVal Unit As String)`

Data Type Double 2 - 65535

Parameters	Name	Description
	<i>Unit</i>	Cycles only.

Description This command sets the number of cycles between the start of a burst and the start of the following burst. This number may be from 2 to 65535 cycles and must be greater than the number of ON cycles. If the number of cycles attempted is not greater than the ON cycles, the interval is not changed.

Note that the interval will occur immediately when this command is called if the burst is running.

See Also `AP.Gen.Wfm`, `AP.Gen.BurstLevel`, `AP.Gen.BurstOnTime`

Example

```

Sub Main
    AP.Application.NewTest 'Reset panels
    AP.Gen.Wfm(0,1)
    AP.Gen.Output = True
    AP.Gen.BurstInterval("Cycles") = 10
    AP.Gen.BurstOnTime("Cycles") = 5
    AP.Gen.BurstLevel("dB") = -40
    Interval = AP.Gen.BurstInterval("Cycles")
    Ontime = AP.Gen.BurstOnTime("Cycles")
    Level = AP.Gen.BurstLevel("%")
    Debug.Print "Burst Interval =";Interval;" cycles."
    Debug.Print "Burst ON time =";Ontime;" cycles."
    Debug.Print "Burst OFF time low level =";Level;" %."
End Sub

```

Example Output

```

Burst Interval = 10 cycles.
Burst ON time = 5 cycles.
Burst OFF time low level = 1 %.

```

AP.Gen.BurstLevel

Property

Syntax	AP.Gen.BurstLevel (ByVal <i>Unit</i> As String)	
Data Type	Double	Level of signal during burst off time. (0 - -80.25dB)
Parameters	Name	Description
	<i>Unit</i>	The following units are available X/Y, dB, %, PPM.
Description	This command sets the amplitude of the Analog Generator during the burst 'off' time. This is as a percentage of the 'on' amplitude and may range from 100.0 percent to .009716280 percent (-80.25 dB).	
See Also	AP.Gen.Wfm, AP.Gen.BurstInterval, AP.Gen.BurstOnTime	
Example	See example for AP.Gen.BurstInterval.	

AP.Gen.BurstOnTime

Property

Syntax	AP.Gen.BurstOnTime (ByVal <i>Unit</i> As String)	
Data Type	Double	From 1 to AP.Gen.BurstInterval - 1.
Parameters	Name	Description
	<i>Unit</i>	Cycles only.
Description	This command sets the number of cycles for the Analog Generator Burst On Time. This number may be from 1 to 65534 cycles and must be less than the number of interval cycles. If the number of cycles attempted is not less than the interval cycles, the ON time is not changed.	
See Also	AP.Gen.Wfm, AP.Gen.BurstInterval, AP.Gen.BurstLevel	
Example	See example for AP.Gen.BurstInterval.	

AP.Gen.ChAAmpl**Property**

Syntax `AP.Gen.ChAAmpl (ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Vrms, Vp, Vpp, dBu, dBV, dBr, dBm, W

Description This command sets the Analog Generator channel A amplitude.

See Also `AP.Gen.ChBAmpl`

Example

```

Sub Main
    AP.Application.NewTest 'Reset panels
    AP.Gen.ChAOutput = True
    AP.Gen.ChBOutput = True
    AP.Gen.ChAAmpl("Vrms") = 1
    AP.Gen.ChBAmpl("Vrms") = 2
    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2
    AP.Anlr.ChBInput = 2
    AP.Anlr.ChALevelSettling(1, .000025, "V", 3, .03, 1)
    AP.Anlr.ChBLevelSettling(1, .000025, "V", 3, .03, 1)
    AP.Anlr.ChALevelTrig
    AP.Anlr.ChBLevelTrig
    Do
        ReadyA = AP.Anlr.ChALevelReady
        ReadyB = AP.Anlr.ChBLevelReady
    Loop Until ReadyA > 0 And ReadyB > 0
    ReadingA = AP.Anlr.ChALevelRdg("V")
    ReadingB = AP.Anlr.ChBLevelRdg("V")
    Prompt.Text = "Level A amplitude =" & Format _
        (ReadingA, "#.0000") & " V" & Chr(13) & "Level B _
        amplitude =" & Format(ReadingB, "#.0000") & " V"
    Prompt.ShowWithContinue
    Stop 'Wait of user to press continue.
End Sub

```

AP.Gen.ChAEqAmpl

Property

Syntax `AP.Gen.ChAEqAmpl (ByVal Unit As String)`

Data Type Double Valid amplitude settings are 0.0 to 100 %FS.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, PPM, Bits, Vrms, Vp, Vpp, dBu, dBV, dBr

Description This command sets the Analog Generator channel A post Eq amplitude.

See Also `AP.Gen.ChBEqAmpl`, `AP.Gen.EqCurve`

Example

```

Sub Main
    AP.Application.NewTest
    AP.Gen.EqCurve("75US-PRE.ADQ", 1) 'Load EQ file
    AP.Gen.Wfm 0, 4 'Select EQ Sine waveform
    AP.Gen.ChAEqAmpl("dBV") = -10.0
    AP.Gen.ChBEqAmpl("dBV") = -10.0
    AP.Gen.Output = True 'Generator Output On

    AP.Sweep.Data1.Id = 5903
    AP.Sweep.Source1.Id = 5051
    AP.Sweep.Data1.Top("dBV") = 12.0
    AP.Sweep.Data1.Bottom("dBV") = -12.0

    AP.Sweep.Stereo = True
    AP.Sweep.Start
End Sub

```

AP.Gen.ChAFreq

Property

Syntax `AP.Gen.ChAFreq (ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM
Description	This command sets the Analog Generator channel A frequency.	
See Also	AP.Gen.ChBFreq	
Example	<pre> Sub Main AP.Application.NewTest AP.Gen.Wfm 1, 2 'Select Stereo Sine waveform AP.Gen.ChAFreq("Hz") = 5000.0 'Set cha A frequency AP.Gen.ChBFreq("Hz") = 7000.0 'Set cha B frequency AP.Gen.ChBTrackA = False 'Set amplitude tracing OFF AP.Gen.ChAAmpl("dBV") = -0.0 'Set cha A amplitude AP.Gen.ChBAmpl("dBV") = -20.0 'Set cha B amplitude AP.Anlr.ChAInput = 2 'Select channel A Generator _ Monitor input AP.Anlr.ChBInput = 2 'Select channel B Generator _ Monitor input AP.Anlr.FuncMode = 9 'Select 2-Ch. Ratio measurement AP.Anlr.FuncInput = 1 'Measure B relative to A AP.Gen.Output = True 'Turn on generator output var = AP.Anlr.FuncRdg("dB") 'Return measurement Debug.Print "Channel B Amlitude is " & _ Format(var,"##.0000") & " dB relative to A" End Sub </pre>	

AP.Gen.ChAInvert

Property

Syntax	AP.Gen.ChAInvert	
Data Type	Boolean	
	<i>True</i>	Invert channel A output.
	<i>False</i>	Normal non-inverting output.
Description	This command sets Analog Generator channel A output to normal polarity or inverted polarity (180 degrees out of phase).	
See Also	AP.Gen.ChBInvert	

Example

```

Sub Main
  AP.Application.NewTest 'Reset panels
  AP.Gen.ChAInvert = False
  AP.Gen.ChBInvert = True
  AP.Gen.Output = True
  AP.Anlr.ChAInput = 2
  AP.Anlr.ChBInput = 2
  AP.Anlr.PhaseSettling(0, .5, "deg", 3, .03, 1)
  AP.Anlr.PhaseTrig
  Do
    Ready = AP.Anlr.PhaseReady
  Loop Until Ready > 0
  Reading1 = AP.Anlr.PhaseRdg("deg")
  Debug.Print "Phase A to B = ";Format(Reading1, _
    "#.0000");" deg"
End Sub

```

Example Output Phase A to B = 180.0110 deg

AP.Gen.ChAOutput**Property****Syntax**

AP.Gen.ChAOutput

Data Type

Boolean

True ON.
False OFF.

Description

This command sets the Analog Generator channel A output to ON or OFF.

The command returns a TRUE if the output is ON and FALSE if the output is OFF.

See Also

AP.Gen.ChBOutput

Example

See example for AP.Gen.ChAAmpl.

AP.Gen.ChBAmpl

Property

Syntax `AP.Gen.ChBAmpl (ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Vrms, Vp, Vpp, dBu, dBV, dBr, dBm, W

Description This command sets the generator channel B amplitude.

See Also `AP.Gen.ChAAmpl`

Example See example for `AP.Gen.ChAAmpl`.

AP.Gen.ChBEqAmpI

Property

Syntax `AP.Gen.ChBEqAmpI (ByVal Unit As String)`

Data Type Double Valid amplitude settings are 0.0 to 100 %FS.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, PPM, Bits, Vrms, Vp, Vpp, dBu, dBV, dBr

Description This command sets the Analog Generator channel B post Eq amplitude.

See Also `AP.Gen.ChAEqAmpI`, `AP.Gen.EqCurve`

Example See example for `AP.Gen.ChAEqAmpI`.

AP.Gen.ChBFreq

Property

Syntax `AP.Gen.ChBFreq (ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM
Description	This command sets the Analog Generator channel B frequency.	
See Also	AP.Gen.ChAFreq	

AP.Gen.ChBInvert

Property

Syntax	AP.Gen.ChBInvert	
Data Type	Boolean	
	<i>True</i>	Invert channel B output.
	<i>False</i>	Normal non-inverting output.
Description	This command sets output B to normal polarity or inverted polarity (180 degrees out of phase with channel A normal polarity).	
See Also	AP.Gen.ChAInvert	
Example	See example for AP.Gen.ChAInvert.	

AP.Gen.ChBOutput

Property

Syntax	AP.Gen.ChBOutput	
Data Type	Boolean	
	<i>True</i>	On
	<i>False</i>	Off
Description	This command sets output B to ON or OFF. The command returns a TRUE if the output is ON and a FALSE if the output is OFF.	
See Also	AP.Gen.ChAOutput	
Example	See example for AP.Gen.ChAAmpl.	

AP.Gen.ChBTrackA**Property****Syntax** **AP.Gen.ChBTrackA****Data Type** Boolean*True* ON, channel B amplitude tracks channel A amplitude.*False* OFF, channel B amplitude independent of channel A.**Description** This command sets channel "B" amplitude to the same amplitude as set for channel "A".**Example**

```

Sub Main
  AP.Application.NewTest 'Reset panels
  AP.Gen.ChAOutput = True
  AP.Gen.ChBOutput = True
  AP.Gen.ChAAmpl("Vrms") = 1
  AP.Gen.ChBTrackA = 1
  AP.Gen.Output = True
  AP.Anlr.ChAInput = 2
  AP.Anlr.ChBInput = 2
  AP.Anlr.ChALevelSettling(1, .000025, "V", 3, .03, 1)
  AP.Anlr.ChBLevelSettling(1, .000025, "V", 3, .03, 1)
  AP.Anlr.ChALevelTrig
  AP.Anlr.ChBLevelTrig
  Do
    ReadyA = AP.Anlr.ChALevelReady
    ReadyB = AP.Anlr.ChBLevelReady
  Loop Until ReadyA > 0 And ReadyB > 0
  ReadingA = AP.Anlr.ChALevelRdg("V")
  ReadingB = AP.Anlr.ChBLevelRdg("V")
  Debug.Print "Level A amplitude = "; _
    Format(ReadingA, "#.0000");" V"
  Debug.Print "Level B amplitude = "; _
    Format(ReadingB, "#.0000");" V"
End Sub

```

Example Output Level A amplitude = .9970 V
Level B amplitude = .9995 V

AP.Gen.Config**Property****Syntax** **AP.Gen.Config****Data Type** Integer

0	Bal - Float.
1	Bal - Gnd.
2	Unbal - Float.
3	Unbal - Gnd.
4	CMTST.

Description This command sets both outputs to a balanced or unbalanced configuration.

Note that the output impedance may change between balanced and unbalanced.

It is possible for this command to cause an amplitude error since the maximum allowable amplitude in the unbalanced configurations is half that for the balanced configuration.

This command sets both outputs to grounded or floating.

This command sets both outputs to a common mode test configuration.

See Also AP.Gen.Impedance**Example** Sub Main

```

AP.Application.NewTest 'Reset panels
AP.Gen.Config = 0 'Set output configuration to _
    balanced floating.
AP.Gen.Impedance = 2 'Set generator output _
    impedance to 600 ohms.
AP.Gen.Ampl("dBm") = 0
AP.Gen.Output = True
AP.Anlr.ChBRangeAuto = 0 'Set input ranging to fixed.
AP.Anlr.ChBRange("V") = 2.5 'Set input range to _
    2.5 Volts.
AP.Anlr.ChBInput = 0 'Set anlr input to INPUT(XLR).
AP.Anlr.ChBImpedance = 1 'Set Cha A input Z to _
    600 ohms.
AP.Anlr.FuncInput = 1 'Set Function Meter Cha to B.
AP.Anlr.FuncSettling(1, .000002, "V", 4, .05, 1)

```

```

AP.Anlr.FuncTrig
Do
    Ready = AP.Anlr.FuncReady
Loop Until Ready > 0
Reading1 = AP.Anlr.FuncRdg("dBm")
Debug.Print "Channel B Amplitude = ";Format _
    (Reading1, "#.0000");" dBm"
AP.Anlr.ChBRangeAuto = 1 'Set input ranging to auto.
End Sub

```

Example Output Channel B Amplitude = .0199 dBm

AP.Gen.DACBandwidth

Property

Syntax `AP.Gen.DACBandwidth`

Data Type Integer

0 30 kHz

1 50 kHz

Description This command sets the Analog Generator Digital to Analog converter Bandwidth for all (D/A) waveforms except Arb Wfm (D/A), MLS (D/A), and Special Pass Thru.

See Also `AP.Gen.Wfm`

Example

```

Sub Main
    AP.Application.NewTest
    AP.Gen.Wfm 1, 0
    AP.Gen.Freq("Hz") = 30000.0
    AP.Gen.DACBandwidth = 0 'Set DAC to 30k Bandwidth
    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2
    AP.Anlr.FuncMode = 4
    AP.Anlr.FuncSettling(5, .0002, "%", 3, .05, 1)
    AP.Anlr.FuncTrig 'Trigger new reading
Do
    Ready = AP.Anlr.FuncReady 'Get status.
Loop Until Ready > 0
Reading1 = AP.Anlr.FuncRdg("%") 'Get settled reading.

```

```

Debug.Print "DAC THD" & Chr$(43) & "N at 30k BW = _
";Format(Reading1, "#.00000");" %"
AP.Gen.DACBandwidth = 1 'Set DAC to 60k Bandwidth
AP.Anlr.FuncTrig      'Trigger new reading
Do
    Ready = AP.Anlr.FuncReady 'Get status.
Loop Until Ready > 0
Reading1 = AP.Anlr.FuncRdg("%") 'Get settled reading.
Debug.Print "DAC THD" & Chr$(43) & "N at 60k BW = _
";Format(Reading1, "#.00000");" %"
End Sub

```

Example**Output**

```

DAC THD+N at 30k BW = .00328 %
DAC THD+N at 60k BW = .00162 %

```

AP.Gen.DACSampleRate**Property****Syntax****AP.Gen.DACSampleRate****Data Type**

Integer

0	65536
1	131072
2	OSR (Output Sample Rate)
3	ISR (Input Sample Rate)

Description

This command sets the Digital to Analog converter sample rate for the Arb Wfm (D/A), and MLS (D/A) waveforms selected with the `AP.Gen.Wfm` command.

See Also`AP.Gen.Wfm`**AP.Gen.DualAmplRatio****Property****Syntax****AP.Gen.DualAmplRatio**(ByVal *Unit* As String)**Data Type**

Double Valid settings are 0.00001% to 100%

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: %, dB, PPM, X/Y
Description	This command sets the Analog Generator amplitude ratio to be used with the channel A and B waveforms for the Sine (D/A) Dual waveform selection.	
See Also	AP.Gen.ChAFreq, AP.Gen.ChBFreq	

AP.Gen.EqCurve

Method

Syntax `AP.Gen.EqCurve(ByVal FileName As String, ByVal Column As Integer)`

Data Type Boolean

Parameters	Name	Description
	<i>FileName</i>	Long Path and File Names permitted up to 128 characters. The file must be an APWIN Eq file (.adq).
	<i>Column</i>	0 = Source 1 settings. 1 = Data 1 measurements. 2 = Data 2 measurements. 3 = Data 3 measurements. 4 = Data 4 measurements. 5 = Data 5 measurements. 6 = Data 6 measurements. 7 = Source 2 settings.

Description This command attaches a Eq file to the Analog Generator. Values in the file will be used as multiply factors in calculating the Analog Generator Amplitude value.

Example See example for AP.Gen.ChAEqAmp1.

AP.Gen.Freq

Property

Syntax `AP.Gen.Freq(ByVal Unit As String)`

Data Type Double

Valid frequency settings for the Hz unit and sine waveform are 10 - 204775.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM

Description This command sets the Analog Generator frequency.

See Also AP.Gen.FreqAccuracy

Example

```

Sub Main
  AP.Application.NewTest 'Reset panels
  AP.Gen.Output = True
  AP.Gen.Freq("Hz") = 10000
  AP.Anlr.ChAInput = 2
  AP.Anlr.ChAFreqSettling(.5, .0002, "Hz", 3, .03, 1)
  AP.Anlr.ChAFreqTrig
  Do
    Ready = AP.Anlr.ChAFreqReady
  Loop Until Ready > 0
  Reading1 = AP.Anlr.ChAFreqRdg("Hz")
  Debug.Print "Fast Frequency = ";Format(Reading1, _
    "#.0000");" Hz"
  AP.Gen.FreqAccuracy = 1      'Set Frequency to _
    High Accuracy.
  AP.Anlr.ChAFreqTrig
  Do
    Ready = AP.Anlr.ChAFreqReady
  Loop Until Ready > 0
  Reading1 = AP.Anlr.ChAFreqRdg("Hz")
  Debug.Print "High Accuracy Frequency = _
    ";Format(Reading1, "#.0000");" Hz"
End Sub

```

Example Output Fast Frequency = 9998.2681 Hz
High Accuracy Frequency = 10000.0637 Hz

AP.Gen.FreqAccuracy**Property**

Syntax	AP.Gen.FreqAccuracy	
Data Type	Integer	
	0	Set frequency accuracy mode to Fast. Fast mode produces the most rapid frequency settling along with frequency accuracy and resolution suitable for nearly all audio tests.
	1	Set frequency accuracy mode to High Accuracy. High accuracy mode provides greater accuracy and resolution, but requires from 150 milliseconds (above 50Hz) to 750 milliseconds (at 10Hz) for complete settling each time the frequency is changed.
Description	<p>This command sets the frequency accuracy mode.</p> <p>Fast mode produces the most rapid frequency settling along with frequency accuracy and resolution suitable for nearly all audio tests. High accuracy mode provides greater accuracy and resolution, but requires from 150 milliseconds (above 50 Hz) to 750 milliseconds (at 10 Hz) for complete settling each time the frequency is changed.</p> <p>Note that this command does not cause an immediate frequency calibration. The calibration will be done at the next call to <code>AP.Gen.Freq</code>.</p>	
See Also	<code>AP.Gen.Freq</code>	
Example	See the example macro for <code>AP.Gen.Freq</code> .	

AP.Gen.IMCenterFreq**Property**

Syntax	AP.Gen.IMCenterFreq (ByVal <i>Unit</i> As String)	
Data Type	Double	
Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command. Hz
Description	This command sets the Analog Generator IMD Center Frequency. The frequency passed is rounded to the closest available value.	

Set the Analog Generator waveform to an IMD or IMD (D/A) CCIF before calling this command.

See Also

AP.Gen.IMFreq

Example

```
Sub Main
  AP.Application.NewTest
  AP.Gen.Wfm(2,2)
  AP.Anlr.ChAInput = 2
  AP.Gen.IMCenterFreq("Hz") = 10000
  AP.Gen.IMFreq("Hz") = 80
  AP.Gen.Ampl("dBu") = 0.0
  AP.Gen.Output = True
  AP.Anlr.FuncMode = 6      'CCIF measurement mode
  AP.Anlr.FuncInput = 0
  AP.Anlr.FuncSettling(3, .00003, "%", 3, .05, 1)
  AP.Anlr.FuncTrig
  Do
    Ready = AP.Anlr.FuncReady
    Loop Until Ready > 0
  Reading1 = AP.Anlr.FuncRdg("%")
  Debug.Print "CCIF/DFD = ";Format(Reading1, _
    "#.0000");" %"
End Sub
```

AP.Gen.IMFreq**Property****Syntax**

AP.Gen.IMFreq(ByVal *Unit* As String)

Data Type

Double

For a SMPTE mode waveform, this is the lower frequency tone. The following frequencies are available for System One:

500Hz, 250Hz, 125Hz, 100Hz, 60Hz, 50Hz, 40Hz

The following frequencies are available for System Two:

500Hz, 250Hz, 125Hz, 100Hz, 70Hz, 60Hz, 50Hz, 40Hz

For a CCIF mode waveform, this is the spacing between the two tones. The following frequencies are available:

1kHz, 500Hz, 250Hz, 200Hz, 120Hz, 100Hz, 80Hz

Parameters

Name	Description
<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command. Hz

Description

This command sets the Analog Generator IMD frequency. The frequency passed is rounded to the closest available value.

Set the generator to an IMD waveform before calling this command in order to have the proper IMD frequency selected.

For a DIM mode waveform, this command has no effect. The frequencies are determined by the DIM mode selected. (See: `AP.Gen.Wfm`)

Because of frequency limitations, the actual frequency set may not be exactly what was requested. Therefore, when setting the IM Frequency it is important to check the returned frequency, and to use that value as the actual IM Frequency setting of the generator.

See Also

`AP.Gen.Wfm`, `AP.Gen.IMCenterFreq`, `AP.Gen.IMHighFreq`

Example

```
Sub Main
  AP.Application.NewTest
  AP.Gen.Wfm(2,1)
  AP.Gen.IMHighFreq("Hz") = 7000
  AP.Gen.IMFreq("Hz") = 60
  AP.Gen.Ampl("dBu") = 0.0
  AP.Gen.Output = True
  AP.Anlr.ChAInput = 2
  AP.Anlr.FuncMode = 5
  AP.Anlr.FuncInput = 0
  AP.Anlr.FuncSettling(3, .00003, "%", 3, .05, 1)
  AP.Anlr.FuncTrig
  Do
    Ready = AP.Anlr.FuncReady
    Loop Until Ready > 0
  Reading1 = AP.Anlr.FuncRdg("%")
  Debug.Print "SMPTE 4:1 = ";Format(Reading1,
"#.0000");" %"
End Sub
```

Example Output

```
SMPTE 4:1 = .0010 %
```

AP.Gen.IMHighFreq

Property

Syntax `AP.Gen.IMHighFreq(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command. Hz

Description This command sets the Analog Generator IMD High Frequency. The frequency passed is rounded to the closest available value.
Set the Analog Generator waveform to an IMD or IMD (D/A) SMPTE before calling this command.

See Also `AP.Gen.IMFreq`

Example See example for `AP.Gen.IMFreq`.

AP.Gen.Impedance

Property

Syntax `AP.Gen.Impedance`

Data Type Integer

The following list contains the selections relevant to the `AP.Gen.Config` command for the Balanced and CMTST selections.

<i>0</i>	40
<i>1</i>	150
<i>2</i>	600

The following list contains the selections relevant to the `AP.Gen.Config` command for the Un-Balanced selections.

<i>0</i>	20
<i>1</i>	600

Description This command controls the output impedance for Balanced and Un-Balanced generator output configurations.

See Also AP.Gen.Config

Example

```
Sub Main
  AP.Application.NewTest 'Reset panels
  AP.Gen.Config = 0      'Set output configuration _
    to balanced floating.
  AP.Gen.Impedance = 2  'Set generator output _
    impedance to 600 ohms.
  AP.Gen.Ampl("dBm") = 0
  AP.Gen.Output = True
  AP.Anlr.ChBRangeAuto = 0 'Set input ranging to fixed.
  AP.Anlr.ChBRange("V") = 2.5 'Set input range to _
    2.5 Volts.
  AP.Anlr.ChBInput = 0 'Set anlr input to INPUT(XLR).
  AP.Anlr.ChBImpedance = 1      'Set Cha B input Z to _
    600 ohms.
  AP.Anlr.FuncInput = 1      'Set Function Meter Cha to B.
  AP.Anlr.FuncSettling(1, .000002, "V", 4, .05, 1)
  AP.Anlr.FuncTrig
Do
  Ready = AP.Anlr.FuncReady
Loop Until Ready > 0
Reading1 = AP.Anlr.FuncRdg("dBm")
Debug.Print "Channel B Amplitude = ";Format _
  (Reading1, "#.0000");" dBm"
  Anlr.ChBRangeAuto = 1      'Set input ranging to auto.
End Sub
```

Example Output Channel B Amplitude = .0199 dBm

AP.Gen.Output

Property

Syntax	AP.Gen.Output
Data Type	Boolean
	<i>True</i> On
	<i>False</i> Off

Description	This command sets the Analog Generator channel A and B outputs to ON or OFF if they have been individually enabled by the <code>AP.Gen.ChAOutput</code> and <code>AP.Gen.ChBOutput</code> commands.
See Also	<code>AP.Gen.ChAOutput</code> , <code>AP.Gen.ChBOutput</code>
Example	See example for <code>AP.Gen.ChAAmpl</code> .

AP.Gen.Phase

Property

Syntax `AP.Gen.Phase(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: deg

Description This command sets the Analog Generator Phase value.
Set the Analog Generator waveform to Sine (D/A) Var Phase before calling this command.

Example

```
Sub Main
    AP.Application.NewTest
    AP.Gen.Wfm 1, 1
    AP.Gen.Phase("deg") = 90.000000
    AP.Anlr.ChAInput = 2
    AP.Anlr.ChBInput = 2
    AP.Gen.Output = True
    Debug.Print "Channel B is " _
        & Format(AP.Anlr.PhaseRdg("deg"), "##.000") _
        & " deg relative to channel A."
End Sub
```

Example Output Channel B is 89.984 deg relative to channel A.

AP.Gen.RefdBm

Property

Syntax `AP.Gen.RefdBm(ByVal Unit As String)`

Data Type Double

Impedance value.

Parameters

Name	Description
<i>Unit</i>	Ohms only.

Description

This command sets the value known to be the generator load impedance for use in dBm computations. When a value of generator output amplitude is requested via the `AP.Gen.Ampl` command using the dBm unit, the software uses this dBm reference impedance value as the "R" in the V^2/R power computation and sets the generator open-circuit voltage commensurately with the voltage division ratio of the present generator source impedance and the specified load impedance in order to deliver the specified dBm value to the load.

See Also

`AP.Gen.Config`, `AP.Gen.Impedance`

Example

```
Sub Main
  AP.Application.NewTest 'Reset panels
  AP.Gen.ChBOutput = False 'Set generator output B to OFF.
  AP.Gen.Impedance = 2     'Set generator output Z to _
    600 Ohms.
  AP.Gen.RefdBm("Ohms") = 600 'Set dBm reference to _
    600 Ohms.
  AP.Gen.Ampl("dBm") = 0
  AP.Gen.Output = True
  AP.Anlr.ChAImpedance = 1     'Set Cha A input Z to _
    600 ohms.
  AP.Anlr.RefdBm("Ohms") = 600 'Set dBm reference to _
    600 Ohms.
  Reference = AP.Anlr.RefdBm("Ohms")
  AP.Anlr.FuncSettling(1, .000002, "V", 4, .05, 1)
  AP.Anlr.FuncTrig
  Do
    Ready = AP.Anlr.FuncReady
  Loop Until Ready > 0
  Reading1 = AP.Anlr.FuncRdg("dBm")
  Debug.Print "Channel A Amplitude = ";Format _
    (Reading1, "#.0000");" dBm (";Reference;" Ohms )"
End Sub
```

Example Output

```
Channel A Amplitude = .0063 dBm ( 600 Ohms )
```

AP.Gen.RefdBr**Property**

Syntax `AP.Gen.RefdBr (ByVal Unit As String)`

Data Type Double Amplitude value.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: V, dBr, dBV

Description This command sets the zero dBr value for the Analog Generator dBr units.

Example

```

Sub Main
    AP.Application.NewTest 'Reset panels
    AP.Gen.RefdBr("V") = 1
    AP.Gen.ChAAmpl("dBr") = 1
    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2
    AP.Anlr.RefChAdBr("V") = 1
    Reference = AP.Gen.RefdBr("V")
    AP.Anlr.ChALevelSettling(1, .000002, "V", 4, .05, 1)
    AP.Anlr.ChALevelTrig
    Do
        Ready = AP.Anlr.ChALevelReady
    Loop Until (Ready > 0)
    Reading1 = AP.Anlr.ChALevelRdg("dBr A")
    Debug.Print "Channel A Amplitude = ";Format _
        (Reading1, "#.0000");" dBr relative to"; _
        Reference;" Volts"
End Sub

```

Example Output Channel A Amplitude = .9679 dBr relative to 1 Volts

AP.Gen.RefdBrAuto**Method**

Syntax `AP.Gen.RefdBrAuto`

Result Boolean

True dBr reference set.

False dBr reference not set.

Description This command sets the generator dBr reference field to the current generator Amplitude setting. If the command is successful a boolean True is returned. If the command is not successful a boolean False is returned.

Example

```

Sub Main
  AP.Application.NewTest 'Reset panels
  AP.Gen.ChAAmpl("dBV") = 0
  AP.Gen.RefdBrAuto
  AP.Gen.Ampl("dbr") = 2 'Increase amplitude 2 dB.
  AP.Gen.Output = True
  AP.Anlr.ChAInput = 2
  AP.Anlr.FuncMode = 0
  AP.Anlr.FuncInput = 0
  AP.Anlr.FuncSettling(1, .000002, "V", 4, .05, 1)
  AP.Anlr.FuncTrig
  Do
    Ready = AP.Anlr.FuncReady
  Loop Until Ready > 0
  Reading1 = AP.Anlr.FuncRdg("dBV")
  Debug.Print "Channel A Amplitude =" ; Format$ _
    (Reading1, "#.000000"); " dBV"
End Sub

```

Example Output Channel A Amplitude = 1.974047 dBV

AP.Gen.RefFreq

Property

Syntax **AP.Gen.RefFreq**(ByVal *Unit* As String)

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command: Hz

Description This command sets the Analog Generator relative frequency reference value. This reference is used for all the Analog Generator relative frequency units (F/R, dHz, %Hz, cent, octs, decs, d%, dPPM)

See Also AP.Gen.Freq

Example

```

Sub Main
  AP.Application.NewTest 'Reset panels
  AP.Gen.Output = True
  AP.Gen.RefFreq("Hz") = 5000
  AP.Gen.Freq("dHz") = 5000
  AP.Anlr.ChAInput = 2
  AP.Anlr.ChAFreqSettling(.5, .0002, "Hz", 3, .03, 1)
  AP.Anlr.ChAFreqTrig
  Do
    Ready = AP.Anlr.ChAFreqReady
  Loop Until Ready > 0
  Reading1 = AP.Anlr.ChAFreqRdg("Hz")
  Debug.Print "Frequency A = ";Format(Reading1, _
    "#.0000");" Hz"
End Sub

```

Example Output Frequency A = 9996.5878 Hz

AP.Gen.RefFreqAuto

Method

Syntax AP.Gen.RefFreqAuto

Result Boolean

True Frequency reference set.
False Frequency reference not set.

Description This command sets the generator frequency reference field to the current generator frequency setting. If the command is successful a boolean True is returned. If the command is not successful a boolean False is returned.

Example

```

Sub Main
  AP.Application.NewTest 'Reset panels
  AP.Gen.Output = True
  AP.Gen.RefFreqAuto
  AP.Gen.ChAFreq("dHz") = 2000 'Increase frequency 2kHz
  AP.Anlr.ChAInput = 2
  AP.Anlr.ChAFreqSettling(.5, .0002, "Hz", 3, .03, 1)

```

```

AP.Anlr.ChAFreqTrig
Do
    Ready = AP.Anlr.ChAFreqReady
Loop Until Ready > 0
Reading1 = AP.Anlr.ChAFreqRdg("Hz")
Debug.Print "Frequency A ="; _
    Format$(Reading1,"#.000000");" Hz"
End Sub

```

Example Output Frequency A = 2998.543549 Hz

AP.Gen.RefWatts

Property

Syntax `AP.Gen.RefWatts(ByVal Unit As String)`

Data Type Double Impedance value.

Parameters	Name	Description
	<i>Unit</i>	Ohms only.

Description This command sets the value known to be the generator load impedance for use in Watts computations. When a value of generator output amplitude is requested via the `AP.Gen.Ampl` command using the Watts unit, the software uses this Watts reference impedance value as the "R" in the $VU!^2/R$ power computation and sets the generator open-circuit voltage commensurately with the voltage division ratio of the present generator source impedance and the specified load impedance in order to deliver the specified power value to the load.

Example

```

Sub Main
  AP.Application.NewTest 'Reset panels
  AP.Gen.Output = True
  AP.Gen.RefWatts ("Ohms") = 8
  AP.Gen.ChAAmpl ("W") = .1
  AP.Anlr.ChAInput = 2
  AP.Anlr.RefWatts ("Ohms") = 8
  AP.Anlr.FuncSettling(1, .000002, "V", 4, .05, 1)
  AP.Anlr.FuncTrig
  Do
    Ready = AP.Anlr.FuncReady
  Loop Until Ready > 0
  Reading1 = AP.Anlr.FuncRdg ("W")
  Debug.Print "Output Power = "; _
    Format$(Reading1, "#.0000"); " Watts"
End Sub

```

Example Output Output Power = 3.5797 Watts

AP.Gen.Wfm**Method****Syntax**

```

AP.Gen.Wfm(ByVal Primary As Integer [, ByVal
  Secondary As Variant])

```

Parameters

Name	Description
<i>Primary</i>	This parameter defines the basic waveform type.
<i>Secondary</i>	This parameter defines the basic waveform modifier.

Primary	Secondary	Description
0		Sine
	0	Normal
	1	Normal Burst
	2	Gated Burst
	3	Trig. Burst
	4	EQ Sine

1		Sine (D/A)
	0	Normal
	1	Var Phase
	2	Stereo
	3	Dual
	4	Shaped Burst
	5	EQ Sine
2		IMD
	0	SMPTE 1:1
	1	SMPTE 4:1
	2	CCIF/DFD
	3	DIM 30
	4	DIM B
	5	DIM 100
3		IMD (D/A)
	0	SMPTE/DIN 4:1
	1	CCIF/DFD
	2	SMPTE/DIN 1:1
	3	DIM 30
	4	DIM B
	5	DIM 100
4		Square
5		Noise
	0	Pink - Pseudo
	1	White - Pseudo
	2	Pink BP - Pseudo
	3	Pink - Random
	4	White - Random
	5	Pink BP - Random
	6	Burst USASI
6		Arb Wfm (D/A)
7		MLS (D/A)
	0	Pink #1
	1	Pink #2
	2	Pink #3
	3	Pink #4

	4	White #1
	5	White #2
	6	White #3
	7	White #4
8		Special (D/A)
	0	Polarity
	1	Pass Thru
9		Square (D/A)
10		Noise (D/A)

Description

This command selects the Analog Generator waveform. The table above shows the possible settings for the `AP.Gen.Wfm` command.

Example

```

Sub Main
  AP.Application.NewTest 'Reset panels
  AP.Gen.Wfm(2,2)
  AP.Gen.ChAFreq("Hz") = 13500
  AP.Gen.IMFreq("Hz")= 1000
  AP.Gen.Ampl("Vrms") = 2
  AP.Gen.Output = True
  AP.Anlr.ChAInput = 2
  AP.Anlr.FuncMode = 6
  AP.Anlr.FuncInput = 0
  AP.Anlr.FuncSettling(1, .00003, "%", 3, .05, 1)
  AP.Anlr.FuncTrig
  Do
    Ready = AP.Anlr.FuncReady
  Loop Until Ready > 0
  Reading1 = AP.Anlr.FuncRdg("dB")
  Debug.Print "CCIF = ";Format(Reading1,"#.0000");" dB"
End Sub

```

Example Output CCIF = -118.5611 dB

AP.Gen.WfmName

Property

Syntax **AP.Gen.WfmName**

Data Type String Long Path and File Names permitted up to 128 characters.

Description This command loads the designated arbitrary waveform file (.AGM or .AGS) into the Digital Generator. The file must be an APWIN waveform file (.agm or .ags).

Note: This command can also be used to control the Digital Generator arbitrary waveform file selection.

See Also AP.Gen.Wfm

AP.Graph.Comment

Property

Syntax `AP.Graph.Comment`**Data Type** `String` ASCII characters.**Description** This command transfers the ASCII characters to or from the comment section in the Graph panel to a string variable.**See Also** `AP.Graph.CommentShow`**Example** `Public Comment As String`

```

Sub Main
    AP.Application.NewTest
    AP.File.OpenTest("COMMENT.AT2C")
'Display Comment area In Graph
    AP.Graph.CommentShow = True
    Comment = "Run Test"

DisplayDialog:
    Begin Dialog UserDialog 170,84,.DlgHandler
        PushButton 20,14,130,21,"&Run Test",.PushButton1
        CancelButton 20,56,130,21
    End Dialog
    Dim dlg As UserDialog
    Select Case Dialog (dlg) 'Display User Dialog
        Case 0
            'Remove Comment area from Graph
            AP.Graph.CommentShow = False
        End
        Case 1
            AP.Graph.Comment = "Test Running"
            Wait .5
            AP.Sweep.Start
            Errors = AP.Data.LimitError(0)'Check for err
            If Errors >0 Then
                Comment = "Test FAILED"
            End If
        End Case
    End Select
End Sub

```



```

        Else
            Comment = "Test PASSED"
        End If
    End Select
    GoTo DisplayDialog
End Sub

Private Function DlgHandler(DlgItem$, Action%,
    SuppValue%) As Boolean
    Select Case Action%
        Case 1 ' Dialog box initialization
        Case 2 ' Value changing or button pressed
            DlgHandler = False 'Exit the dialog
        Case 3 ' TextBox or ComboBox text changed
        Case 4 ' Focus changed
        Case 5 ' Idle
            DlgHandler = True ' Continue getting idle actions
            AP.Graph.Comment = Comment$ 'Display comment
            Wait .5
            AP.Graph.Comment = "" 'Remove comment
            Wait .2
        End Select
    End Function

```

AP.Graph.CommentShow

Property

Syntax	AP.Graph.CommentShow
Data Type	Boolean
	<i>True</i> Display Comment section.
	<i>False</i> Remove Comment section from view.
Description	This command displays or removes from view the comment section in the Graph panel
See Also	AP.Graph.Comment
Example	See example for AP.Graph.Comment.

AP.Graph.CopyToSweepPanel

Method

Syntax**AP.Graph.CopyToSweepPanel****Description**

This command transfers the current graphic display vertical (Top/Bottom) and horizontal (Start/Stop) axis values to the Sweep panel Data 1, Data 2, and Sweep 1 settings.

Example

Sub Main

```

AP.Application.NewTest    'New Test
AP.Gen.Output = True      'Generator Output ON
AP.Anlr.ChAInput = 2     'Analyzer Ch A Input to GenMon
AP.Anlr.ChBInput = 2     'Analyzer Ch B Input to GenMon
AP.Anlr.FuncMode = 3     'Analyzer Function Meter _
                          to THD+N Ampl

AP.S2CDsp.Program = 2    'Select FFT Digital Analyzer
AP.S2CDsp.FFT.InputFormat = 1 'Select HiRes A/D _
                          @65536 Input
AP.S2CDsp.FFT.Ch1Source = 2 'Digital Analyzer Ch 1 _
                          Source to Anlr Rdg Ampl

AP.Sweep.Data1.Id = 6024 'Select Fft.Ch.1 Ampl _
                          for Data 1
AP.Sweep.Data2.Id = 6027 'Select Fft.Ch.2 Ampl _
                          for Data 2
AP.Sweep.Source1.Id = 5515 'Select Fft.FFT Freq. _
                          for Source 1
AP.Sweep.Start           'Acquire waveform
'Display data so that the vertical scaling is _
                          relative to optimized data for Data 1
AP.Graph.OptimizeLeft    'Optimize Data 1
AP.Graph.CopyToSweepPanel 'Copy Left and Right _
                          graph vertical scale information to Sweep Panel
AP.Sweep.CopyData1to2    'Copy Data 1 settings _
                          to Data 2
Wait 5
'Display data so that the vertical scaling is _
                          relative to optimized data for Data 2
AP.Graph.OptimizeRight  'Optimize Data 2

```

```

AP.Graph.CopyToSweepPanel 'Copy Left and Right _
    graph vertical scale information to Sweep Panel
AP.Sweep.CopyData2to1      'Copy Data 2 settings _
    to Data 1
Wait 5
'Display data so that the vertical scaling is _
    optimized together for Data 1 and Data 2
AP.Graph.OptimizeTogether 'Optimize Data 1 and _
    Data 2 Together
Wait 5
'Display data so that the vertical scaling is _
    optimized individually for Data 1 and Data 2
AP.Graph.OptimizeIndividually 'Optimize Data 1 _
    and Data 2 Individually
End Sub

```

AP.Graph.CursorPosition

Property

Syntax	AP.Graph.CursorPosition (ByVal <i>CursorNum</i> As Integer, ByVal <i>Unit</i> As String)						
Data Type	Double						
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>CursorNum</i></td> <td>1 = Cursor #1 2 = Cursor #2</td> </tr> <tr> <td><i>Unit</i></td> <td>Refer to the setting or reading defined by the <i>column%</i> parameter to determine the appropriate unit selections.</td> </tr> </tbody> </table>	Name	Description	<i>CursorNum</i>	1 = Cursor #1 2 = Cursor #2	<i>Unit</i>	Refer to the setting or reading defined by the <i>column%</i> parameter to determine the appropriate unit selections.
Name	Description						
<i>CursorNum</i>	1 = Cursor #1 2 = Cursor #2						
<i>Unit</i>	Refer to the setting or reading defined by the <i>column%</i> parameter to determine the appropriate unit selections.						
Description	This command returns the horizontal axis position value where the designated cursor is positioned.						
See Also	AP.Graph.CursorRow						

Example

```

Sub Main
    AP.Application.NewTest 'New Test
    AP.Gen.Output = True   'Generator Output ON
    AP.Anlr.ChAInput = 2 'Analyzer Ch A Input to GenMon
    AP.Anlr.ChBInput = 2 'Analyzer Ch B Input to GenMon

    AP.S2CDsp.Program = 2 'Select FFT Digital Analyzer

```

```

AP.S2CDsp.FFT.InputFormat = 1'Select RiRes A/D _
    @65536 Input
AP.S2CDsp.FFT.Window = 5    'Select None, Move to _
    Bin Center

AP.Sweep.Data1.Id = 6024    'Select Fft.Ch.1 Ampl _
    for Data 1
AP.Sweep.Data2.Id = 6027    'Select Fft.Ch.2 Ampl _
    for Data 2
AP.Sweep.Source1.Id = 5515 'Select Fft.FFT Freq. _
    for Source 1
AP.Sweep.Start              'Acquire waveform

AP.Graph.OptimizeTogether  'Optimize Data 1 and _
    Data 2 Together
AP.Graph.CursorsOn(True)

AP.Prompt.Text = "Select one of the Traces from _
    the Graph Legend then Position Cursor #1 on the _
    fundamental then press Continue."
AP.Prompt.FontSize = 8      'Set font size to 8 point.
AP.Prompt.Position(-1,-1,250,150) 'Set location and _
    size.
AP.Prompt.ShowWithContinue 'Display prompt with _
    Continue button.
Stop                        'Stop macro.

Debug.Print "Frequency = " & _
    Format(AP.Graph.CursorPosition(1, "Hz"), _
    "##.0000")
Debug.Print "Data Editor Row = " & _
    AP.Graph.CursorRow(1)
Debug.Print "Value = " &
    Format(AP.Graph.CursorValue(1, "V"), "##.0000")
End Sub

```

AP.Graph.CursorRow**Property****Syntax** `AP.Graph.CursorRow(ByVal CursorNum As Integer)`**Data Type** Integer**Parameters**

Name	Description
<i>CursorNum</i>	1 = Cursor #1 2 = Cursor #2

Description

This command returns the nearest row number to the position of the designated cursor. The row number can be used to extract access data in with the `AP.Data.Value` command.

See Also`AP.Graph.CursorPosition`**Example**See example for `AP.Graph.CursorPosition`.**AP.Graph.CursorsOn****Method****Syntax** `AP.Graph.CursorsOn(ByVal bOn As Boolean)`**Parameters**

Name	Description
<i>bOn</i>	True = Display cursors. False = Remove cursors from view.

Result

Boolean

True Cursors displayed.*False* Cursors not displayed. This may be because the Graph Panel is not displayed.**Description**

This command displays or removes from view the cursors on the Graph panel.

ExampleSee example for `AP.Graph.CursorPosition`.

AP.Graph.CursorValue

Property

Syntax	AP.Graph.CursorValue (ByVal <i>CursorNum</i> As Integer, ByVal <i>Unit</i> As String)	
Data Type	Double	
Parameters	Name	Description
	<i>CursorNum</i>	1 = Cursor #1 2 = Cursor #2
	<i>Unit</i>	Refer to the setting or reading defined by the <i>column%</i> parameter to determine the appropriate unit selections.
Description	This command returns the vertical axis value where the designated cursor is positioned.	
See Also	AP.Graph.CursorPosition	
Example	See example for AP.Graph.CursorPosition.	

AP.Graph.OptimizeIndividually

Method

Syntax	AP.Graph.OptimizeIndividually
Description	This command optimizes the graph to display all data.
See Also	AP.Graph.OptimizeLeft, AP.Graph.OptimizeRight, AP.Graph.OptimizeTogether
Example	See example for AP.Graph.CopyToSweepPanel.

AP.Graph.OptimizeLeft

Method

Syntax	AP.Graph.OptimizeLeft
Description	This command optimizes the graph to display the data on the Left axis (Data 1).
See Also	AP.Graph.OptimizeIndividually, AP.Graph.OptimizeRight, AP.Graph.OptimizeTogether

Example See example for `AP.Graph.CopyToSweepPanel`.

AP.Graph.OptimizeRight

Method

Syntax `AP.Graph.OptimizeRight`

Description This command optimizes the graph to display the data on the Right axis (Data 2).

See Also `AP.Graph.OptimizeIndividually`,
`AP.Graph.OptimizeLeft`, `AP.Graph.OptimizeTogether`

Example See example for `AP.Graph.CopyToSweepPanel`.

AP.Graph.OptimizeTogether

Method

Syntax `AP.Graph.OptimizeTogether`

Description This command optimizes the graph to display all data (both Data 1 and Data 2) using the same vertical axis values (Top and Bottom).

See Also `AP.Graph.OptimizeIndividually`,
`AP.Graph.OptimizeLeft`, `AP.Graph.OptimizeRight`

Example See example for `AP.Graph.CopyToSweepPanel`.

User Notes

User Notes

AP.Log.AddEntry**Method**

Syntax `AP.Log.AddEntry(ByVal Text As String)`

Parameters	Name	Discription
	<i>Text</i>	Any valid string

Description This command appends the current date and time and the defined string to the log file.

Example

```

Sub Main
  AP.Log.Enable = True           'Enable logging.
  'Set log file to "S2-22CK.ALG"
  AP.Log.FileName = "S2-22CK.ALG"
  AP.Log.ErrorMessages = True   'Log error messages.
  AP.Log.FileActivity = True    'Log File Input/Out.
  AP.Log.PassFailMessages = True 'Log P/F messages.
  AP.Log.TestName = True       'Log test name.
  AP.Log.GraphTitle = True     'Log graph title.
  AP.Log.Data = 1              'Log all data.
  AP.Log.Clear                 'Clear log file.
  AP.Log.AddEntry "Test Start."
  AP.Log.AddEntryWithoutTimeDate "Amplitude Linearity."
  AP.File.OpenTest "AMPLIN.AT2C" 'Open test.
  AP.Sweep.Start                'Start sweep.
  AP.Log.View                    'View log file.
End Sub

```

Example Output

```

01/09/96 14:27:01 Test Start.
                    Amplitude Linearity.

```

```

01/09/96 14:27:01 Open Test: AMPLIN.AT2C
PASS 01/09/96 14:27:02 Execute sweep: AMPLIN.AT2C

```

Comment The example output is from the log file created when the example macro is run.

AP.Log.AddEntryWithoutTimeDate Method

Syntax `AP.Log.AddEntryWithoutTimeDate (ByVal Text As String)`

Parameters	Name	Discription
	<i>Text</i>	Any valid string

Description This command appends the defined string to the log file.

Example See example for `AP.Log.AddEntry`.

AP.Log.Clear Method

Syntax `AP.Log.Clear`

Description This command clears the contents of the current log file.

Example See example for `AP.Log.AddEntry`.

AP.Log.Data Property

Syntax `AP.Log.Data`

Data Type	Integer
0	None
1	All
2	Failed data only

Description This command controls whether no test point values (None), all test point values (All), or only those test points which were outside limits (Failed Data Only) are written into the log file. Any values written into the log file which were outside limits will have parenthesis at the end with the (less than) or (greater than) symbol and the value of the limit which they failed.

Example See example for `AP.Log.AddEntry`.

AP.Log.Enable

Property

Syntax	<code>AP.Log.Enable</code>
Data Type	Boolean <i>True</i> Enable <i>False</i> Disable
Description	This command when enabled controls whether logging actually takes place. If disabled, no logging takes place
Example	See example for <code>AP.Log.AddEntry</code> .

AP.Log.ErrorMessages

Property

Syntax	<code>AP.Log.ErrorMessages</code>
Data Type	Boolean <i>True</i> Enable <i>False</i> Disable
Description	This command when enabled logs into the log file any APWIN or windows error messages which occur during the period that logging is enabled.
Example	See example for <code>AP.Log.AddEntry</code> .

AP.Log.FileActivity

Property

Syntax	<code>AP.Log.FileActivity</code>
Data Type	Boolean <i>True</i> Enable <i>False</i> Disable
Description	This command when enabled will enter into the log file a text message for every disk file opened or every file saved to disk. The message includes the name and full path name of the file, and the date and time at which it was opened or saved.

Example See example for `AP.Log.AddEntry`.

AP.Log.FileName

Property

Syntax `AP.Log.FileName`

Data Type String

Description This command defines the file name to use for the log file.

Example See example for `AP.Log.AddEntry`.

AP.Log.GraphTitle

Property

Syntax `AP.Log.GraphTitle`

Data Type Boolean
True Enable
False Disable

Description This command when enabled logs the Graph Title, and the Time and Date at which the test was executed, exactly as they are displayed in the title bar of the graph.

Example See example for `AP.Log.AddEntry`.

AP.Log.PassFailMessages

Property

Syntax `AP.Log.PassFailMessages`

Data Type Boolean
True Enable
False Disable

Description This command when enabled causes an error summary message to be written into the log file each time a test is run. The first word of the message will be PASS or FAIL (See example for `AP.Log.AddEntry`). Following a colon (:) the error message will include the number of measurements which were below the lower

limit, the number of measurements that were above the upper limit, and the number of timeouts which occurred. If disabled, no error message is written to the error file.

Example See example for `AP.Log.AddEntry`.

AP.Log.PrintLogFile

Method

Syntax `AP.Log.PrintLogFile`

Description This command loads the NOTEPAD application and prints the current log file and then closes the NOTEPAD application.

Example

```

Sub Main
    AP.Log.Enable = True           'Enable logging.
    AP.Log.FileName = "S2-22CK.ALG" 'Set log file to _
        "S2-22CK.ALG"
    AP.Log.ErrorMessages = True   'Log error messages.
    AP.Log.FileActivity = True    'Log File Input/Out.
    AP.Log.PassFailMessages = True 'Log Pass/Fail messa
    AP.Log.TestName = True        'Log test name.
    AP.Log.GraphTitle = True      'Log graph title.
    AP.Log.Data = 1               'Log all data.
    AP.Log.Clear                  'Clear log file.
    AP.Log.AddEntry "Amplitude Linearity."
    AP.File.OpenTest "AMPLIN.AT2C" 'Open test.
    AP.Sweep.Start                'Start sweep.
    AP.Log.PrintLogFile        'Print log file.
End Sub

```

AP.Log.TestName

Property

Syntax `AP.Log.TestName`

Data Type Boolean

<i>True</i>	Enable
<i>False</i>	Disable

Description This command when enabled logs the test name, including path name, of the test when executed.

Example See example for `AP.Log.AddEntry`.

AP.Log.View

Method

Syntax `AP.Log.View`

Description This command loads the NOTEPAD application and displays the current log file.

Example See example for `AP.Log.AddEntry`.

User Notes

User Notes

Macro

AP.Macro.IsRunning

OLE Method

Syntax **AP.Macro.IsRunning**

Result Boolean

True APWIN Macro running.
False APWIN Macro not running.

Description This command returns the status of the APWIN macro.

Example

```
Private Sub Form_Load()
    Dim AP As Object
    'Create OLE link to APWIN.
    Set AP = CreateObject("APWIN.Application")
    AP.Application.Visible = True ' Make APWIN visable

    'Place your code here

    'Run an APWIN Macro and wait for it to finish
    AP.File.OpenMacro "C:\BUSY.APB"
    While AP.Macro.IsRunning = True
    Wend

    'Change Visual Basic directory to APWIN Working _
    Directory.
    ChDir AP.Application.MacroDir

    'Place your code here

    AP.Application.Quit                    'Quit APWIN
End
End Sub
```

AP.Macro.Name**Method****Syntax** **AP.Macro.Name****Result** Boolean ASCII characters.

Description This command returns the APWIN Macro Editor Name. This text string is located in the upper left corner of the APWIN Macro/Procedure Editor before the Macro/Procedure name. This string is usefull when using the AppActivate command in the Language reference section of APWIN Basic.

See Also AP.Application.Name**Example**

```

Sub Main
  AP.Application.VisibleMacroEditor(True)  'Restore _
      Macro Editor.
  AppActivate AP.Application.Name  'Select the APWIN _
      window
  'The following SendKey command will now be sent to _
      the APWIN application.
  SendKeys "%WC",1  'Clear all windows on page.
  SendKeys "%PO",1  'Display Data Editor.
  AppActivate AP.Macro.Name  'Change focus back to _
      the Procedure/Macro editor

  'In Debug mode focus is automatically returned to
  ' the editor each time the user interacts with the
  ' controls. Therefore it is important to note that
  ' sections of code containing commands that are to
  ' be sent to other applications via the SendKeys
  ' command need to be executed without interruption.
  'When debugging these areas place a breakpoints
  ' before and after the SendKeys commands to maintain
  ' the correct window/application focus.

End Sub

```

User Notes

User Notes

AP.Print.Data**Method****Syntax** **AP.Print.Data****Result** Boolean*True* Printout of tabular data successful.*False* Printout of tabular data failed.**Description** This command prints the data displayed in the Data Editor in tabular format. The Data Editor is automatically displayed on the current page if it is not displayed on at least one of the five APWIN Pages. The printer is defined by the settings on the File, Print Setup menu.**See Also** **AP.Print.Graph****Example** Sub Main
 'Load test containing measurement data
 AP.File.OpenTest("GRAPH.AT2C")
 AP.Application.PanelOpen apbPanelDataEditor
 AP.Sweep.Start
 AP.Print.Data 'Print Data in tabular form
 End Sub

AP.Print.Graph**Method****Syntax** **AP.Print.Graph****Result** Boolean*True* Printout of graphic data successful.*False* Printout of graphic data failed.**Description** This command sends the graph as defined by the settings on the File, Page Setup menu to the selected printer as define by the File, Print Setup menu. A graph must be displayed on at least one of the five APWIN Pages to print or preview the graph.

See Also `AP.Print.LoadFromTest`

Example

```
Sub Main
    'Load test containing graph setup
    AP.File.OpenTest("GRAPH.AT2C")
    AP.Print.LoadFromTest
    AP.File.OpenTest("TEST.AT2C")
    AP.Sweep.Start
    AP.Print.Graph
End Sub
```

AP.Print.LoadFromTest

Method

Syntax `AP.Print.LoadFromTest`

Result Boolean

True Page Setup settings loaded from test file.
False Page Setup settings not loaded from test file.

Description This command loads the page setup settings from the currently loaded test. The printout settings are global and can only be changed via this command or manually via the user interacting with the Page Setup menu. This allows the system to produce graphic printouts that have a consistent format over many different tests. Loading a test doesn't automatically update the Page Setup menu and change the graphic output.

See Also `AP.Print.Graph`

Example See example macro `AP.Print.Graph`.

AP.Print.TrackGraph

Property

Syntax `AP.Print.TrackGraph`

Data Type Boolean

True Use Graph Trace settings.
False Use settings defined on Page Setup menu.

Description

This command causes the printout Color, Line Style, and Thickness to automatically track the settings used in the Graph Window legend. A graph must be displayed on at least one of the five APWIN Pages to print or preview the graph.

See Also

AP.Print.LoadFromTest

Example

```
Sub Main
  AP.Application.NewTest      'Start with New Test
  AP.Print.TrackGraph = True 'Have graph printout _
    track graph panel legend settings
  AP.Gen.Output = True       'Turn On Analog Generator
  AP.Anlr.ChAInput = 2      'Have Analog Analyzer _
    monitor Generator Channel A
  AP.Anlr.FuncFilterHP = 3  'Select 400Hz High Pass _
    Filter
  AP.Sweep.Start            'Run Sweep
  AP.Print.Graph            'Print graph
End Sub
```


User Notes

Prompt

AP.Prompt.FontSize

Property

Syntax	<code>AP.Prompt.FontSize</code>
Data Type	Double Default size = 16.
Description	This command sets the font size of the characters used in the User Prompt.
Example	<pre> Sub Main 'Set string to display in prompt. With AP.Prompt .Text = "This prompt will be shown for 5 seconds." .FontSize = 8 'Set font size to 8 point. .Position(-1,-1,190,120)'Set location and size. .Show 'Show prompt. Shown = .IsUp Wait 5 'Wait 5 seconds. .Hide 'Hide prompt. Shown = Prompt.IsUp .Text = "This prompt will be shown until the _ Continue Macro button is selected below." _ 'Set string to display in prompt. .ShowWithContinue 'Display prompt with _ Continue button. Stop 'Stop macro. End With End Sub </pre>

AP.Prompt.Hide

Method

Syntax	<code>AP.Prompt.Hide</code>
Description	This command removes the user prompt from view.
Example	See example macro <code>AP.Prompt.FontSize</code> .

AP.Prompt.IsUp**Method**

Syntax	<code>AP.Prompt.IsUp</code>
Result	Boolean
	<i>True</i> User prompt is displayed.
	<i>False</i> User prompt is NOT displayed.
Description	This command returns the current status of the user prompt.
Example	See example for <code>AP.Prompt.FontSize</code> .

AP.Prompt.Position**Method**

Syntax	<code>AP.Prompt.Position(ByVal X As Integer, ByVal Y As Integer, ByVal CX As Integer, ByVal CY As Integer)</code>	
Parameters	Name	Description
	<i>X</i>	This number value is the distance from the left edge of the monitor screen. It is measured in 1/8ths of the average character width for the dialog's font. Setting this number value to -1 centers the prompt horizontally.
	<i>Y</i>	This number value is the distance from the top edge of the monitor screen. It is measured in 1/12ths of the character height for the dialog's font. Setting this number value to -1 centers the prompt vertically.
	<i>CX</i>	This number value is the width. It is measured in 1/8ths of the average character width for the dialog's font.
	<i>CY</i>	This number value is the height. It is measured in 1/12ths of the character height for the dialog's font.
Description	This command defines the position and size of the User Prompt.	
Example	See example for <code>AP.Prompt.FontSize</code> .	

AP.Prompt.Show**Method**

Syntax	<code>AP.Prompt.Show</code>
Description	This command displays the user prompt.
See Also	(Language Reference) Stop Instruction
Example	See example for <code>AP.Prompt.FontSize</code> .

AP.Prompt.ShowWithContinue**Method**

Syntax	<code>AP.Prompt.ShowWithContinue</code>
Description	This command displays the current user prompt window with a continue button.
See Also	(Language Reference) Stop Instruction
Example	See example for <code>AP.Prompt.FontSize</code> .

AP.Prompt.ShowWithContinueAndStopSweep**Method**

Syntax	<code>AP.Prompt.ShowWithContinueAndStopSweep</code>
Description	This command displays the current user prompt window with a continue button. When the continue button is selected a sweep if running is terminated.
See Also	(Language Reference) Stop Instruction

Example	<pre> Sub Main AP.Application.NewTest 'Reset panels AP.Gen.Ampl("Vrms") = 2 AP.Gen.Output = True AP.Anlr.ChAInput = 1 AP.Anlr.FuncMode = 0 AP.Anlr.FuncInput = 0 </pre>
----------------	--

```

With AP.Prompt
    .Text = "Press the Continue button to STOP _
            the SWEEP and continue the Macro."
    .FontSize = 8           'Set font size to 8 point.
    .Position (-1,-1,180,120) 'Set location and size.
    'Display prompt with Continue and also stop the _
    current sweep.
    .ShowWithContinueAndStopSweep
    AP.Sweep.Start           'Start Sweep.
    .Text = "Press the Continue button to END the_
            Macro."
    .FontSize = 8           'Set font size to 8 point.
    .Position (-1,-1,200,100) 'Set location and size.
    .ShowWithContinue 'Display prompt with Continue.
    Stop
End With
End Sub

```

AP.Prompt.Text

Property

Syntax	AP.Prompt.Text
Data Type	String User defined string.
Description	This command defines the string to be displayed in a user prompt.
Example	See example for AP.Prompt.FontSize.

User Notes

User Notes

Regulation

AP.Reg.IsRunning

Method

Syntax `AP.Reg.IsRunning`

Result Boolean

True Regulation process running.

False Regulation process not running.

Description This command returns the status of the Regulation process.

See Also `AP.Macro.LoadFromTest`

Example

```
Sub Main
    AP.Application.NewTest
    AP.Gen.ChAAmpl("dBV") = 0.0
    AP.Application.PanelOpen apbPanelRegulation
    'Regulate Analyzer Level A to -80 dBV to within a _
      Tolerance 1.0dB
    AP.Reg.TargetID = 5903      'Regulate Analyzer Level A _
      to -80 dBV to within a Tolerance 1.0dB
    AP.Reg.TargetValue("dBV") = -80.000000
    AP.Reg.TargetToleranceMode = 1      'Tolerance Mode dB
    AP.Reg.TargetTolerance("dB") = 1.0  'Tolerance 1.0dB

    AP.Reg.SourceID = 5052      'by varying the Generator _
      A Amplitude
    AP.Reg.SourceHigh("dBV") = -70.0    'High Amplitude _
      boundary
    AP.Reg.SourceLow("dBV") = -90.0     'Low Amplitude _
      boundary
    AP.Reg.SourceOperation = 2         '-Normal Operatio
    AP.Reg.SourceStepSize("dB") = 0.1
    AP.Reg.SourceIteration = 30

    AP.Reg.StartNowait(True)      'Start the Regulation _
      process then continue.
    StartTime = Timer
```



```

Do      'Wait until Regulation process is finished _
      or Timeout has elapsed.
      Wait .1
      Debug.Print Timer
Loop While AP.Reg.IsRunning And Timer < StartTime _
      + Reg.Timeout
If AP.Reg.IsRunning = True Then 'Stop Regulation _
      process if running
      AP.Reg.StartNowait(False)
      Debug.Print "Regulation Stopped!"
End If
End Sub

```

AP.Reg.SourceHigh

Property

Syntax	AP.Reg.SourceHigh (ByVal <i>Unit</i> As String)	
Data Type	Double	Refer to the setting defined by the AP.Reg.SourceId command (ID#) to determine the appropriate range of acceptable values.
Parameters	Name	Description
	<i>Unit</i>	Refer to the setting defined by the AP.RegSourceId command (ID#) to determine the appropriate unit selections.
Description	This command sets the upper boundary for the source parameter used in the regulation process.	
See Also	AP.Reg.SourceLow	
Example	See example for AP.Reg.SourceId.	

AP.Reg.SourceId

Property

Syntax	AP.Reg.SourceId	
Data Type	Long	Instrument Parameter ID#.
Description	This command is used to select the instrument parameter which will define settings used in the regulation process.	

Refer to Appendix B to obtain instrument parameter identification numbers.

Example

```

Sub Main
  AP.File.OpenTest "REG3.AT2C"
  AP.Reg.SourceID = 5051           'Set Source To GenFreq
  AP.Reg.TargetID = 5906         'Set Source to AnlrAmpl
  AP.Reg.SourceOperation = 1     'Set Operation to +Normal
  AP.Reg.SourceStepSize("%") = 2 'Set Stepsize to 2%
  AP.Reg.SourceIteration = 30    'Set iterations to 30
  AP.Reg.TargetToleranceMode = 0 'Tolerance Mode to %
  AP.Reg.TargetValue("dBrA") = -3 'Tolerance to -3
  AP.Reg.TargetTolerance("%") = 5 'Tolerance units %
  AP.Reg.SourceHigh("Hz") = 5000 'High Bound to 5kHz
  AP.Reg.SourceLow("Hz") = 200   'Low Bound to 20 Hz
  AP.Reg.SweepEnable = False     'Don't Regulate each _
                                step in sweep.
  AP.Reg.Timeout = 2.0           'Terminate regulation _
                                process if timed out

  AP.Reg.Start                   'Start Regulation
End Sub

```

AP.Reg.Sourcelteration**Property**

Syntax	<code>AP.Reg.SourceIteration</code>
Data Type	Long
Description	This command sets the number of Source Iterations. Iterations limit the maximum number of regulation attempt steps the source will make before exiting the search and moving on.
See Also	<code>AP.Reg.SourceOperation</code>
Example	See example for <code>AP.Reg.SourceId</code> .

AP.Reg.SourceLow

Property

Syntax	<code>AP.Reg.SourceLow(ByVal Unit As String)</code>	
Data Type	Double	Refer to the setting defined by the <code>AP.Reg.SourceId</code> command (ID#) to determine the appropriate range of acceptable values.
Parameters	Name	Description
	<i>Unit</i>	Refer to the setting defined by the <code>AP.RegSourceId</code> command (ID#) to determine the appropriate unit selections.
Description	This command sets the lower boundary for the source parameter used in the regulation process.	
See Also	<code>AP.Reg.SourceHigh</code>	
Example	See example for <code>AP.Reg.SourceId</code> .	

AP.Reg.SourceOperation

Property

Syntax	<code>AP.Reg.SourceOperation</code>	
Data Type	Integer	
	0	Linear: assumes a linear relationship between the source setting and the target reading.
	1	+ Normal: assumes that an increasing source setting will cause an increasing target reading, but not necessarily linearly.
	2	- Normal: assumes that an increasing source setting will cause a decreasing target reading, but not necessarily linearly.
	3	Maximum: each cycle of regulation starts from the source low boundry value. For example the source will increase by the specified step size as long as the target value also increases. If the target value goes through a peak and starts to decrease, the direction of the source reverses and the step size is cut in half. These half-size steps continue untill the target value again starts to decrease, at which time the direction of change again reverses and the step size is again cut in half. This process will continue until the number of peak

4 crossings equal the value defined by the `AP.Reg.SourceIterations` command. Minimum: each cycle of regulation starts from the source low boundry value. For example the source will increase by the specified step size as long as the target value decreases. If the target value goes through a notch and starts to increase, the direction of the source reverses and the step size is cut in half. These half-size steps continue untill the target value again starts to increase, at which time the direction of change again reverses and the step size is again cut in half. This process will continue until the number of peak crossings equal the value defined by the `AP.Reg.SourceIterations` command.

Description This command selects the type of algorithm used to control the source parameter specified by the `AP.Reg.SourceId` command.

See Also `AP.Reg.SourceStepSize`, `AP.Reg.SourceIteration`

Example See example for `AP.Reg.SourceId`.

AP.Reg.SourceStepSize

Property

Syntax `AP.Reg.SourceStepSize(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: X/Y, dB, %

Description This command sets the Source Step Size for the + Normal, - Normal, Maximum, and Minimum algorithm selections as it begins its search at the beginning of each new regulation process.

See Also `AP.Reg.SourceOperation`, `AP.Reg.SourceIteration`

Example See example for `AP.Reg.SourceId`.

AP.Reg.Start

Method

Syntax	<code>AP.Reg.Start</code>
Description	This command initiates a regulation process.
Example	See example for <code>AP.Reg.SourceId</code> .

AP.Reg.StartNoWait

Method

Syntax	<code>AP.Reg.StartNoWait(ByVal <i>bStart</i> As Boolean)</code>	
Parameters	Name	Description
	<i>bStart</i>	True = Start regulation process. False = Stop regulation process.
Description	This command initiates a regulation process and then continues macro execution.	
Example	See example for <code>AP.Reg.SourceId</code> .	

AP.Reg.SweepEnable

Property

Syntax	<code>AP.Reg.SweepEnable</code>	
Data Type	Boolean	
	<i>True</i>	Enable regulation for each sweep step.
	<i>False</i>	Disable regulation for each sweep step.
Description	This command enables or disables regulation before each step in a sweep.	
See Also	<code>AP.Reg.Start</code>	
Example	See example for <code>AP.Reg.SourceId</code> .	

AP.Reg.TargetId

Property

Syntax	<code>AP.Reg.TargetId</code>
Data Type	Long Instrument Parameter ID#.
Description	This command is used to select the instrument parameter which will return readings used in the regulation process. Refer to Appendix B to obtain instrument parameter identification numbers.
Example	See example for <code>AP.Reg.SourceId</code> .

AP.Reg.TargetTolerance

Property

Syntax	<code>AP.Reg.TargetTolerance(ByVal Unit As String)</code>				
Data Type	Double	Refer to the setting defined by the <code>AP.Reg.TargetId</code> command (ID#) to determine the appropriate range of acceptable values.			
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>Unit</i></td> <td>Refer to the setting defined by the <code>AP.RegTargetId</code> command (ID#) to determine the appropriate unit selections for the Abs selection for the <code>AP.Reg.TargetToleranceMode</code> command. The following units are available when % is selected with the <code>AP.Reg.TargetToleranceMode</code> command: X/Y, %</td> </tr> </tbody> </table>	Name	Description	<i>Unit</i>	Refer to the setting defined by the <code>AP.RegTargetId</code> command (ID#) to determine the appropriate unit selections for the Abs selection for the <code>AP.Reg.TargetToleranceMode</code> command. The following units are available when % is selected with the <code>AP.Reg.TargetToleranceMode</code> command: X/Y, %
Name	Description				
<i>Unit</i>	Refer to the setting defined by the <code>AP.RegTargetId</code> command (ID#) to determine the appropriate unit selections for the Abs selection for the <code>AP.Reg.TargetToleranceMode</code> command. The following units are available when % is selected with the <code>AP.Reg.TargetToleranceMode</code> command: X/Y, %				
Description	This command sets the tolerance that the regulation algorithm uses to determine if the regulation process is complete.				
See Also	<code>AP.Reg.TargetToleranceMode</code> , <code>AP.Reg.TargetValue</code>				
Example	See example for <code>AP.Reg.SourceId</code> .				

AP.Reg.TargetToleranceMode

Property

Syntax	<code>AP.Reg.TargetToleranceMode</code>
Data Type	Integer
	0 %
	1 dB
	2 Abs
Description	This command selects the type of units the regulation algorithm uses to specify the Target Tolerance.
See Also	<code>AP.Reg.TargetTolerance</code> , <code>AP.Reg.TargetValue</code>
Example	See example for <code>AP.Reg.SourceId</code> .

AP.Reg.TargetValue

Property

Syntax	<code>AP.Reg.TargetValue(ByVal Unit As String)</code>	
Data Type	Double	Refer to the setting defined by the <code>AP.Reg.TargetId</code> command (ID#) to determine the appropriate range of acceptable values.
Parameters	Name	Description
	<i>Unit</i>	Refer to the setting defined by the <code>AP.RegTargetId</code> command (ID#) to determine the appropriate unit selections.
Description	This command sets the Target value that the regulation algorithm attempts to obtain.	
See Also	<code>AP.Reg.TargetTolerance</code> , <code>AP.Reg.TargetToleranceMode</code>	
Example	See example for <code>AP.Reg.SourceId</code> .	

AP.Reg.TimeOut

Property

Syntax	<code>AP.Reg.Timeout</code>
Data Type	Double
Description	This command sets the period of time allowed to complete each regulation process.
Example	See example for <code>AP.Reg.SourceId</code> .

User Notes

Digital Input/Output

AP.S2CDio.ChAPeakRdg

Property

Syntax `AP.S2CDio.ChAPeakRdg`(ByVal *Unit* As String)

Data Type Double

Parameters

Part	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, Bits.

Description

This command returns a unsettled reading for the Digital Input/Output channel A Peak Monitor meter and zeros the ready count.

See Also

`AP.S2CDio.ChAPeakReady`, `AP.S2CDio.ChAPeakTrig`

Example

```
Sub Main
    Dim rdgA As Double, rdgB As Double
    AP.Application.NewTest
    'S2Dio Peak Meter Sample code
    AP.S2CDio.InFormat = 0      'XLR balanced input
    AP.S2CDio.InImpedance = 0  'High impedance input
    AP.S2CDio.InMonitorMode = 2 'Measure absolute peak
    AP.S2CDio.ChAPeakTrig 'Trigger channel A peak meter
    AP.S2CDio.ChBPeakTrig 'Trigger channel B peak meter
    Do Until (AP.S2CDio.ChAPeakReady And _
        AP.S2CDio.ChBPeakReady)
        'perform other actions while waiting for readings
        '...
    Loop
    rdgA = AP.S2CDio.ChAPeakRdg("FFS") 'Get channel A _
        peak reading
    rdgB = AP.S2CDio.ChBPeakRdg("FFS") 'Get channel B _
        peak reading
    AP.Prompt.Text = "Ch A = " & rdgA & " FFS" & _
        Chr(13) & "Ch B = " & rdgB & " FFS"
    AP.Prompt.ShowWithContinue
    Stop
End Sub
```

AP.S2CDio.ChAPeakReady**Property****Syntax** `AP.S2CDio.ChAPeakReady`**Data Type**

Integer

`0` Reading not ready.`>0` Reading ready.**Description**

This command returns the Digital Input/Output channel A Peak Monitor meter unsettled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDio.ChAPeakRdg` or `AP.S2CDio.ChAPeakTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDio.ChAPeakRdg` command will be guaranteed to return quickly.

See Also`AP.S2CDio.ChAPeakRd`, `AP.S2CDio.ChAPeakTrig`**Example**See example for `AP.S2CDio.ChAPeakRdg`.**AP.S2CDio.ChAPeakTrig****Method****Syntax** `AP.S2CDio.ChAPeakTrig`**Description**

Causes a restart of the reading cycle and zeros the ready count for the `AP.S2CDio.ChAPeakRdg` command. The reading in progress is aborted.

See Also`AP.S2CDio.ChAPeakRdg`, `AP.S2CDio.ChAPeakReady`**Example**See example for `AP.S2CDio.ChAPeakRdg`.

AP.S2CDio.ChBPeakRdg

Property**Syntax** `AP.S2CDio.ChBPeakRdg(ByVal Unit As String)`**Data Type** Double**Parameters**

Part	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, Bits.

Description

This command returns a unsettled reading for the Digital Input/Output channel B Peak Monitor meter and zeros the ready count.

See Also`AP.S2CDio.ChBPeakReady`, `AP.S2CDio.ChBPeakTrig`**Example**

See example for `AP.S2CDio.ChAPeakRdg`.

AP.S2CDio.ChBPeakReady

Property**Syntax** `AP.S2CDio.ChBPeakReady`**Data Type**

Integer

0 Reading not ready.*>0* Reading ready.**Description**

This command returns the Digital Input/Output channel B Peak Monitor meter unsettled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDio.ChBPeakRdg` or `AP.S2CDio.ChBPeakTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDio.ChBPeakRdg` command will be guaranteed to return quickly.

See Also`AP.S2CDio.ChBPeakRdg`, `AP.S2CDio.ChBPeakTrig`**Example**

See example for `AP.S2CDio.ChAPeakRdg`.

AP.S2CDio.ChBPeakTrig

Method

Syntax	<code>AP.S2CDio.ChBPeakTrig</code>
Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP.S2CDio.ChBPeakRdg</code> comand. The reading in progress is aborted.
See Also	<code>AP.S2CDio.ChBPeakRdg</code> , <code>AP.S2CDio.ChBPeakReady</code>
Example	See example for <code>AP.S2CDio.ChAPeakRdg</code> .

AP.S2CDio.DelayRdg

Property

Syntax	<code>AP.S2CDio.DelayRdg(ByVal Unit As String)</code>	
Data Type	Double	
Parameters	Part	Description
	<code>Unit</code>	String that designates the desired unit. The following unit is valid for this command: sec.
Description	This command returns a settled reading for the Digital Input/Output Delay from Out meter and zeros the ready count.	
See Also	<code>AP.S2CDio.DelayReady</code> , <code>AP.S2CDio.DelaySettling</code> , <code>AP.S2CDio.DelayTrig</code>	
Example	<pre> Sub Main Dim rdgA As Double AP.Application.NewTest 'S2Dio Delay From Out meter sample code AP.S2CDio.DelaySettling(1.0, 100e-6, "SEC", 1, 0.0, 0) AP.S2CDio.DelayTrig 'Trigger channel A peak meter Do Until AP.S2CDio.DelayReady 'perform other actions while waiting for reading '... Loop rdgA = AP.S2CDio.DelayRdg("SEC") 'Get channel A _ peak reading AP.Prompt.Text = "Delay = " & rdgA AP.Prompt.ShowWithContinue </pre>	

```

        Stop
    End Sub

```

AP.S2CDio.DelayReady

Property

Syntax `AP.S2CDio.DelayReady`

Data Type Integer

`0` Reading not ready.

`>0` Reading ready.

Description This command returns the Digital Input/Output Delay from Out meter settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDio.DelayRdg` or `AP.S2CDio.DelayTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDio.DelayRdg` command will be guaranteed to return quickly.

See Also `AP.S2CDio.DelayRd`, `AP.S2CDio.DelaySettling`, `AP.S2CDio.DelayTrig`

Example See example for `AP.S2CDio.DelayRdg`.

AP.S2CDio.DelaySettling

Method

Syntax `AP.S2CDio.DelaySettling(ByVal Tolerance As Double, ByVal Floor As Double, ByVal FloorUnit As String, ByVal Points As Integer, ByVal Delay As Double, ByVal Algorithm As Integer)`

Parameters See Appendix A for Settling Algorithm and parameter name descriptions.

Description	This command sets the settling parameters for the AP.S2CDio.DelayRdg command.
See Also	AP.S2CDio.DelayRdg, AP.S2CDio.DelayReady, AP.S2CDio.DelayTrig
Example	See example for AP.S2CDio.DelayRdg.

AP.S2CDio.DelayTrig

Method

Syntax	AP.S2CDsp.DelayTrig
Description	Causes a restart of the reading cycle and zeros the ready count for the AP.S2CDio.DelayRdg command. The reading in progress is aborted.
See Also	AP.S2CDio.DelayRdg, AP.S2CDio.DelayReady, AP.S2CDio.DelaySettling
Example	See example for AP.S2CDio.DelayRdg.

AP.S2CDio.FlagChAInvalidRdg

Method

Syntax	AP.S2CDio.FlagChAInvalidRdg
Result	Boolean <i>True</i> Error <i>False</i> Proper operation
Description	This reading returns the state of the channel A Validity bit. The CH A reading is driven directly by the V (Validity) bit defined in the Professional and Consumer standards. One Validity bit is sent in each subframe.
See Also	AP.S2CDio.OutSendInvalid, AP.S2CDio.FlagChBInvalidRdg, AP.S2CDio.FlagInvalidRdg.
Example	Sub Main AP.Application.NewTest

```

AP.Application.PanelOpen apbPanelDigIOSmall
AP.S2CDio.InFormat = 3
If AP.S2CDio.FlagInvalidRdg = True Then _
    Debug.Print "Invalid Error"
AP.Application.PanelOpen apbPanelDigIOLarge
S2Dio.OutSendInvalid = True
Wait 1
If AP.S2CDio.FlagConfidenceRdg = True Then _
    Debug.Print "Confidence Error"
If AP.S2CDio.FlagLockRdg = True Then _
    Debug.Print "Lock Error"
If AP.S2CDio.FlagCodingRdg = True Then _
    Debug.Print "Coding Error"
If AP.S2CDio.FlagParityRdg = True Then _
    Debug.Print "Parity Error"
If AP.S2CDio.FlagChAInvalidRdg = True Then
    Debug.Print "ChA Invalid Error"
If AP.S2CDio.FlagChBInvalidRdg = True Then _
    Debug.Print "ChB Invalid Error"

End Sub

```

AP.S2CDio.FlagChBInvalidRdg

Method

Syntax `AP.S2CDio.FlagChBInvalidRdg`

Result Boolean

True Error
False Proper operation

Description This reading returns the state of the channel B Validity bit. The CH B reading is driven directly by the V (Validity) bit defined in the Professional and Consumer standards. One Validity bit is sent in each subframe.

See Also `AP.S2CDio.OutSendInvalid`,
`AP.S2CDio.FlagChAInvalidRdg`,
`AP.S2CDio.FlagInvalidRdg`.

Example See example for `AP.S2CDio.FlagChAInvalidRdg`.

AP.S2CDio.FlagCodingRdg**Method****Syntax** `AP.S2CDio.FlagCodingRdg`**Result** Boolean*True* Error*False* Proper operation

Description This reading returns the status of the biphasic coding for the input serial data stream. The Coding reading indicates a deviation from proper biphasic coding in the input serial stream (ignoring preambles). Proper biphasic signals can never remain at a logic high or logic low level for more than two consecutive Unit Intervals (UI) except in the preamble. The preamble deliberately deviates from biphasic coding in order to provide a unique frame synchronization signal, so preambles are excluded from the function of the Coding indicators.

Example See example for `AP.S2CDio.FlagChAInvalidRdg`.

AP.S2CDio.FlagConfidenceRdg**Method****Syntax** `AP.S2CDio.FlagConfidenceRdg`**Result** Boolean*True* Error*False* Proper operation

Description The Confidence reading returns *True* when the ratio between the amplitude of the three UI long pulse and the following one UI-long pulse in a preamble becomes large enough to cause an increasing probability of errors when slicing the received signal into logic high and low values. This large ratio occurs when the transmission bandwidth has been reduced to marginal or unacceptable values. Under these conditions, selection of hardware input equalization (XLR with EQ or BNC with EQ rather than XLR or BNC selections of the Input Format field) will often compensate for the cable bandwidth reduction, and provide reliable measurements.

Example See example for `AP.S2CDio.FlagChAInvalidRdg`.

AP.S2CDio.FlagInvalidRdg

Method

Syntax `AP.S2CDio.FlagInvalidRdg`

Result Boolean

True Error
False Proper operation

Description This reading uses OR logic to determine if either the channel A or channel B Validity bit is set as defined in the Professional and Consumer standards. One Validity bit is sent in each subframe.

See Also `AP.S2CDio.OutSendInvalid`,
`AP.S2CDio.FlagChAInvalidRdg`,
`AP.S2CDio.FlagChBInvalidRdg`.

Example See example for `AP.S2CDio.FlagChAInvalidRdg`.

AP.S2CDio.FlagLockRdg

Method

Syntax `AP.S2CDio.FlagLockRdg`

Result Boolean

True Error
False Proper operation

Description The Lock reading indicates when the digital input phase-locked loop is unable to lock to the incoming signal.

Example See example for `AP.S2CDio.FlagChAInvalidRdg`.

AP.S2CDio.FlagParityRdg

Method

Syntax `AP.S2CDio.FlagParityRdg`

Result Boolean

<i>True</i>	Error
<i>False</i>	Proper operation

Description The Parity reading indicates a parity error in either subframe. Correct parity is determined by comparing the P (parity) bit with the sum of the remaining 31 bits in each subframe. Any single bit error or odd number of bit errors introduced in transmission within a subframe will cause a Parity error indication, but even numbers of bit errors cannot be detected by this technique.

Example See example for `AP.S2CDio.FlagChAInvalidRdg`.

AP.S2CDio.InBitsDisplay

Property

Syntax `AP.S2CDio.InBitsDisplay`

Data Type Integer

<i>0</i>	Display Data bits.
<i>1</i>	Display Active bits.

Description This command sets the Digital Input/Output Input Bits Display mode.

AP.S2CDio.InDeEmp

Property

Syntax `AP.S2CDio.InDeEmp`

Data Type Integer

<i>0</i>	Off
<i>1</i>	50/15us 0dB
<i>2</i>	50/15us + 10dB
<i>3</i>	J17 0dB
<i>4</i>	J17 + 20dB

Description This command selects the Digital Input/Output input deemphasis. CD type (50/15 us) or CCITT J17 deemphasis may be selected as desired. Either deemphasis characteristic may be selected with either zero dB insertion loss at low frequencies (0 dB selections in each case) or with

a gain factor (+10 dB for 50/15us, +20 dB for J17) to compensate for the matching headroom allowances of the System Two Digital Generator preemphasis `AP.S2CDio.OutPreEmp` capability.

See Also `AP.S2CDio.OutPreEmp`

Example See example for `AP.S2CDio.OutFormat`.

AP.S2CDio.InFormat

Property

Syntax `AP.S2CDio.InFormat`

Data Type Integer

<i>0</i>	XLR (Bal): Front panel XLR digital input connector, balanced
<i>1</i>	BNC (unbal): Front panel BNC digital input connector, unbalanced
<i>2</i>	Optical: Front panel Toslink optical input connector
<i>3</i>	Gen Mon: Digital Generator XLR or BNC output connector
<i>4</i>	XLR w/Eq: Front panel XLR with equalization for 100 meter cable roll-off
<i>5</i>	BNC w/EQ: Front panel BNC with equalization for 100 meter cable roll-off
<i>6</i>	XLR common: Center tap of digital input transformer vs ground
<i>7</i>	Serial: Rear-panel general-purpose serial input connector
<i>8</i>	Parallel: Rear-panel parallel input connector
<i>9</i>	Dual XLR (bal)
<i>10</i>	Dual BNC (unbal)
<i>11</i>	Dual XLR w/Eq
<i>12</i>	Dual BNC w/Eq

Description This command sets the Digital Input/Output Input Format.

Example See example for `AP.S2CDio.ChAPeakRdg`.

AP.S2CDio.InImpedance

Property**Syntax** `AP.S2CDio.InImpedance`**Data Type** Integer

The following list contains the selections relevant to the `AP.S2CDio.InFormat` command XLR (Bal), XLR w/EQ, XLR Common, Dual XLR (bal), and Dual BNC w/Eq selections.

<code>0</code>	Hi Z
<code>1</code>	110 Ohms

The following list contains the selections relevant to the `AP.S1Dio.InFormat` command BNC (unbal), Gen Mon, BNC w/EQ, Dual XLR (unbal), and Dual BNC w/Eq selections.

<code>0</code>	Hi Z
<code>1</code>	75 Ohms

Description This command sets the Digital Input/Output Input Impedance based on the selection for the `AP.S2CDio.InFormat` command.**Example** See example for `AP.S2CDio.ChAPeakRdg`.

AP.S2CDio.InInput

Property**Syntax** `AP.S2CDio.InInput`**Data Type** Integer

<code>0</code>	Select channel I XLR/BNC Input.
<code>1</code>	Select channel II XLR/BNC Input.

Description This command selects the Digital Input/Output Input connector. When the Digital Input/Output Input Format is set by the `AP.S2CDio.InFormat` command to any of the XLR or BNC selections this command serves as an input switcher to select from one connector (I) or the other (II).**Example** See example for `AP.S2CDio.InJitterBW`.

AP.S2CDio.InJitterBW

Property

Syntax `AP.S2CDio.InJitterBW`

Data Type Integer

0	50Hz to 100kHz
1	120Hz to 100kHz
2	700Hz to 100kHz
3	1200Hz to 100kHz

Description This command sets the Digital Input/Output Input bandwidth of the Interface Jitter `AP.S2CDio.JitterRdg` meter.

Example

```

Sub Main
  Dim rdg As Double

  'S2Dio Jitter meter sample code
  'Set up Jitter output
  AP.S2CDio.OutJitterType = 1          'Sine
  AP.S2CDio.OutJitterAmpl("UI") = 5.0
  AP.S2CDio.OutJitterFreq("Hz") = 2e3
  'Set up Jitter input
  AP.S2CDio.InInput = 0              'Select connector I
  AP.S2CDio.InJitterMode = 0         'Unit interval mode
  AP.S2CDio.InJitterDetector = 1     'Average detector
  AP.S2CDio.InJitterBW = 0           '50Hz - 100kHz bandwidth
  AP.S2CDio.JitterSettling(5.0, 1e-6, "UI", 1, 0.0, 0)
  AP.S2CDio.JitterTrig                'Trigger channel A _
    peak meter
  Do Until AP.S2CDio.JitterReady
    'perform other actions while waiting for reading
    '...
  Loop
  rdg = AP.S2CDio.JitterRdg("UI")    'Get channel A _
    peak reading
  AP.Prompt.Text = "Jitter = " & rdg & " UI"
  AP.Prompt.ShowWithContinue
  Stop
End Sub

```

AP.S2CDio.InJitterDetector**Property**

Syntax	<code>AP.S2CDio.InJitterDetector</code>
Data Type	Integer
	0 Pk
	1 Avg
Description	This command selects the Digital Input/Output Input detector type for the Interface Jitter <code>AP.S2CDio.JitterRdg</code> meter.
Example	See example for <code>AP.S2CDio.InJitterBW</code> .

AP.S2CDio.InJitterMode**Property**

Syntax	<code>AP.S2CDio.InJitterMode</code>
Data Type	Integer
	0 UI: Unit interval
	1 Sec: Time
Description	This command selects the measurement mode for the Digital Input/Output Interface Jitter <code>AP.S2CDio.JitterRdg</code> meter.
Example	See example for <code>AP.S2CDio.InJitterBW</code> .

AP.S2CDio.InMonitorMode**Property**

Syntax	<code>AP.S2CDio.InMonitorMode</code>
Data Type	Integer
	0 Pos. Peak: causes the Level Monitors to display the most positive value during each measurement interval, which is approximately 1/4 second.
	1 Neg. Peak: causes the monitors to display the most negative value during each measurement interval (dBFS units cannot be used with the Min mode since the numbers are negative).

- 2 Abs. Peak: causes display of the absolute value of the largest positive-going or negative-going value during each measurement interval.
- 3 1/2 Pk-Pk : causes display of the value which is one-half the peak-to-peak range measured during the measurement interval.

Description This command sets the Digital Input/Output Peak Monitor

Example See example for `AP.S2CDio.ChAPeakRdg`.

AP.S2CDio.InResolution

Property

Syntax `AP.S2CDio.InResolution`

Data Type Long 8 to 24 bits.

Description This command sets the Digital Input/Output Input Resolution.

See Also `AP.S2CDio.OutResolution`

Example See example for `AP.S2CDio.OutFormat`.

AP.S2CDio.InScaleFreq

Property

Syntax `AP.S2CDio.InScaleFreq`

Data Type Integer

- 0 Output Rate: is the Digital Generator output sample rate set by the Rate field near the top of the Output section of the DIO panel.
- 1 Measured Rate: is the input signal sample rate value displayed in the Sample Rate field near the top of the Input section of the DIO panel.
- 2 Status Bits A: is the value of sample frequency encoded into the received channel A status bits.
- 3 Dio Rate Ref: is the sample rate value defined by the `AP.S2CDio.RefRate` command.

Description	This command selects a source from which the digital audio sample rate is determined. The frequency of imbedded digital audio signals must be normalized by a digital sample rate before display, whether it is displayed as a numeric frequency counter display or as a frequency component on an FFT graph.
See Also	AP.S2CDio.RefRate
Example	See example for AP.S2CDio.OutFormat.

AP.S2CDio.JitterRdg

Property

Syntax `AP.S2CDio.JitterRdg(ByVal Unit As String)`

Data Type Double

Parameters	Part	Description
	<i>Unit</i>	String that designates the desired unit. The following unit is valid when the AP.S2Dio.InJitterMode command is set to UI (Unit Interval): UI The following unit is valid when the AP.S2Dio.InJitterMode command is set to Sec (Time): sec

Description This command returns a settled reading for the Digital Input/Output Interface Jitter meter and zeros the ready count.

See Also AP.S2CDio.JitterReady, AP.S2CDio.JitterSettling, AP.S2CDio.JitterTrig, AP.S2CDio.InJitterMode

Example See example for AP.S2CDio.InJitterBW.

AP.S2CDio.JitterReady

Property

Syntax `AP.S2CDio.JitterReady`

Data Type Integer

<i>0</i>	Reading not ready.
<i>>0</i>	Reading ready.

Description	<p>This command returns the Digital Input/Output Interface Jitter meter meter settled reading ready count.</p> <p>Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the <code>AP.S2CDio.JitterRdg</code> or <code>AP.S2CDio.JitterTrig</code> commands will zero the ready count.</p> <p>If the reading is found to be ready, a call to the <code>AP.S2CDio.JitterRdg</code> command will be guaranteed to return quickly.</p>
See Also	<code>AP.S2CDio.JitterRdg</code> , <code>AP.S2CDio.JitterSettling</code> , <code>AP.S2CDio.JitterTrig</code>
Example	See example for <code>AP.S2CDio.InJitterBW</code> .

AP.S2CDio.JitterSettling

Method

Syntax	<code>AP.S2CDio.JitterSettling</code> (ByVal <i>Tolerance</i> As Double, ByVal <i>Floor</i> As Double, ByVal <i>FloorUnit</i> As String, ByVal <i>Points</i> As Integer, ByVal <i>Delay</i> As Double, ByVal <i>Algorithm</i> As Integer)
Parameters	See Appendix A for Settling Algorithm and parameter name descriptions.
Description	This command sets the settling parameters for the <code>AP.S2CDio.JitterRdg</code> command.
See Also	<code>AP.S2CDio.JitterRdg</code> , <code>AP.S2CDio.JitterReady</code> , <code>AP.S2CDio.JitterTrig</code>
Example	See example for <code>AP.S2CDio.JitterBW</code> .

AP.S2CDio.JitterTrig**Method**

Syntax	<code>AP.S2CDsp.JitterTrig</code>
Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP.S2CDio.JitterRdg</code> command. The reading in progress is aborted.
See Also	<code>AP.S2CDio.JitterRdg</code> , <code>AP.S2CDio.JitterReady</code> , <code>AP.S2CDio.JitterSettling</code>
Example	See example for <code>AP.S2CDio.InJitterBW</code> .

AP.S2CDio.OutCableSim**Property**

Syntax	<code>AP.S2CDio.OutCableSim</code>
Data Type	Boolean
	<i>True</i> Enable cable simulation.
	<i>False</i> Disable cable simulation.
Description	<p>This command enables or disables cable simulation.</p> <p>A fixed hardware filter may be switched into the path to the XLR or BNC output connectors to simulate the effect of a typical 100-meter-long cable, to test the ability of a digital device under test to function with impaired signals. This feature is not available at the optical, general purpose serial, or parallel outputs. This cable simulation filter is approximately the inverse of the input cable equalization filter selectable as XLR w/EQ, BNC w/EQ, Dual XLR w/Eq, and Dual BNC w/Eq selections in the Digital Input Format field, so the two should approximately compensate for one another when a short external cable is connected from Digital Output to Digital Input. However, there will still be some attenuation of the interface signal introduced by the cable simulation hardware. To switch the cable simulator in and out of the circuit:</p>
Example	See example for <code>AP.S2CDio.OutFormat</code> .

AP.S2CDio.OutCM

Property

Syntax	<code>AP.S2CDio.OutCM</code>	
Data Type	Boolean	
	<code>True</code>	Enable common mode output.
	<code>False</code>	Disable common mode output.
Description	This command enables or disables the Digital Input/Output Common Mode output.	
Example	See example for <code>AP.S2CDio.CMAmpl</code> .	

AP.S2CDio.OutCMAmpl

Property

Syntax	<code>AP.S2CDio.OutCMAmpl (ByVal Unit As String)</code>	
Data Type	Double	Range of values: 0.0 to 20.4
Parameters	Name	Description
	<code>Unit</code>	String that designates the desired unit. The following units are valid for this command: Vpp
Description	This command sets the Digital Input/Output Common Mode Amplitude value.	
Example	<pre> Const INPUT_VOLTAGE As Integer = 6102 Const COMMON_MODE_AMPL As Integer = 5317 Sub Main Dim reading As Double AP.Application.NewTest 'Reset panels AP.S2CDio.OutCMFreq("Hz") = 20e3 AP.S2CDio.OutCMAmpl("Vpp") = 1.0 AP.S2CDio.OutCM = True AP.S2CDio.InFormat = 6 'XLR Common AP.S2CDio.VoltageSettling(3.0, .01, "Vpp", 3, 0.0, 2) AP.Sweep.CreateGraph = True AP.Sweep.Data1.Id = INPUT_VOLTAGE AP.Sweep.Data1.Top("Vpp") = 8.0 </pre>	

```

AP.Sweep.Data1.Bottom("Vpp") = 0.0
AP.Sweep.Source1.Id = COMMON_MODE_AMPL
AP.Sweep.Source1.Start("Vpp") = 0.0
AP.Sweep.Source1.Stop("Vpp") = 20.0
AP.Sweep.Source1.Steps = 10
AP.Sweep.Start
End Sub

```

AP.S2CDio.OutCMFreq

Property

Syntax `AP.S2CDio.OutCMFreq(ByVal Unit As String)`

Data Type Double Range of values: 20.0 Hz to 40.0 kHz

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command: Hz

Description This command sets the Digital Input/Output Common Mode signal (sinewave) Frequency value.

Example See example for `AP.S2CDio.CMAmpl`.

AP.S2CDio.OutFormat

Property

Syntax `AP.S2CDio.OutFormat`

Data Type	Integer	
	0	XLR @ OSR (Output Sample Rate)
	1	BNC @ OSR
	2	Optical @ OSR
	3	Serial @ OSR
	4	Parallel @ OSR
	5	Dual XLR @ OSR
	6	Dual BNC @ OSR
	7	Dual XLR @ 2xOSR
	8	Dual BNC @ 2xOSR

Description This command sets the Digital Input/Output Output source.

Example

```

Const INTERVU_AMPL As Integer = 6053
Const INTERVU_TIME As Integer = 5612

Sub Main
  Dim rftime As Double
  AP.Application.NewTest
  With AP.S2CDio
    .OutRiseFall = True      'Variable Rise/Fall Time
    .OutRiseFallTime("sec") = 16e-9 'Set Rise/Fall time
    .OutCableSim = False    'Cable simulation OFF
    .OutFormat = 0         'Output XLR Bal signal to DUT
    .OutRate("Hz") = 48000  'Set 48 kHz sample rate
    .OutNoise = True       'Let's inject some noise
    .OutNoiseAmpl("Vpp") = 0.1 'Set Noise Ampl@ 0.1 Vpp
    .OutResolution = 20    '20-bit resolution
    .OutPreEmp = 0         'No Pre-Emphasis
    .InDeEmp = 0           'No De-Emph
    .InFormat = 0         'Input signal from DUT
    .InResolution = 20    'Same resolution as output
    .InScaleFreq = 1      'Scale Frequency by measured Rate
  End With
  AP.S2CDsp.Program = 3      'Load Intervu DSP program
  AP.S2CDsp.Intervu.AmplVsTime = 0 'Interpolate mode
  AP.S2CDsp.Intervu.TrigSource = 0

  With AP.Sweep
    .Source1.Id = INTERVU_TIME 'Sweep time
    .Source1.Start("sec") = 0.0 'start at 0.0 sec
    .Source1.Stop("sec") = 4e-6 'end at 4 nSec
    .Source1.Steps = 255 '255 points
    .Data1.Id = INTERVU_AMPL   'Measure amplitude
    .Data1.Top("V") = 3.0      'Max amplitude of +2 V
    .Data1.Bottom("V") = -3.0 'Min amplitude of -2 V
    .Append = False          'Don't append first waveform
    AP.S2CDio.OutCableSim = True 'Cable simulation ON
    .Start
    AP.S2CDio.OutCableSim = False 'Cable simulation OFF
    .Append = True           'Append any additional waveforms
    .Start                   'Get a waveform
  End With
End Sub

```

```

End With
For rftime = 50e-9 To 350e-9 Step 100e-9
    AP.S2CDio.OutRiseFallTime("sec") = rftime
    AP.Sweep.Start 'Append next sweep
Next rftime
End Sub

```

Comment Cable simulation looks just like RiseFallTime = 350 nSec

AP.S2CDio.OutInvert

Property

Syntax `AP.S2CDio.OutInvert`

Data Type Boolean

True Invert output.
False Normal non-inverting output.

Description This command sets the Digital Output to normal polarity or inverted polarity (180 degrees out of phase).

AP.S2CDio.OutJitterAmpl

Property

Syntax `AP.S2CDio.OutInvert`(ByVal *Unit* As String)

Data Type Double

Parameters

Name	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: UI, dBUI, sec.

Description This command sets the Digital Input/Output Jitter Amplitude value.

See Also `AP.S2CDio.OutJitterEqCurve`

Example See example for `AP.S2CDio.InJitterBW`.

AP.S2CDio.OutJitterEqCurve

Method

Syntax `AP.S2CDio.OutJitterEqCurve(ByVal FileName As String, ByVal Column As Integer)`

Parameters

Name	Description
------	-------------

<i>FileName</i>	Any valid DOS path and file name. The file must be an APWIN Eq file (.adq).
<i>Column</i>	0 = Source 1 settings. 1 = Data 1 measurements. 2 = Data 2 measurements. 3 = Data 3 measurements. 4 = Data 4 measurements. 5 = Data 5 measurements. 6 = Data 6 measurements. 7 = Source 2 settings.

Result

Boolean

<i>True</i>	File attachment successful.
<i>False</i>	File attachment failed.

Description

This command attaches a Eq file to the Jitter Generator. Values in the file will be used as multiply factors in calculating the Digital Input/Output Jitter Amplitude value.

See Also

AP.S2CDio.OutJitterAmpl

AP.S2CDio.OutJitterFreq

Property

Syntax `AP.S2CDio.OutJitterFreq(ByVal Unit As String)`

Data Type

Double

Parameters

Name	Description
------	-------------

<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz
-------------	---

Description

This command sets the Digital Input/Output Jitter Frequency value.

Example

See example for AP.S2CDio.InJitterBW.

AP.S2CDio.OutJitterType

Property**Syntax** `AP.S2CDio.OutJitterType`**Data Type** Integer

<i>0</i>	Off
<i>1</i>	Sinusoidal
<i>2</i>	Lowpass Random
<i>3</i>	Squarewave
<i>4</i>	Wideband Random
<i>5</i>	EQ Sine

Description This command sets the type of jitter that may be added to the digital output signal at the XLR, BNC, and optical outputs to test the ability of a digital device to reject input jitter.**Example** See example for `AP.S2CDio.InJitterBW`.

AP.S2CDio.OutNoise

Property**Syntax** `AP.S2CDio.OutNoise`**Data Type** Boolean

<i>True</i>	Enable noise output.
<i>False</i>	Disable noise output.

Description This command enables or disables the Digital Input/Output Interfering Noise output.**See Also** `AP.S2CDio.OutNoiseAmpl`**Example** See example for `AP.S2CDio.OutFormat`.

AP.S2CDio.OutNoiseAmpl

Property**Syntax** `AP.S2CDio.OutNoiseAmpl(ByVal Unit As String)`**Data Type** Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Vpp
Description		This command sets the Digital Input/Output Noise Amplitude value.
Example		See example for AP.S2CDio.OutFormat.

AP.S2CDio.OutParityError

Property

Syntax	AP.S2CDio.OutParityError	
Data Type	Boolean	
	<i>True</i>	Generate Parity Error in digital output.
	<i>False</i>	Normal output.
Description	This command sets the Digital Output Parity Bit to produce a Parity Error or to normal operation.	
See Also	AP.S2CDio.FlagParityRdg	
Example	See example for AP.S2CDio.FlagChAInvalidRdg.	

AP.S2CDio.OutPreEmp

Property

Syntax	AP.S2CDio.OutPreEmp	
Data Type	Integer	
	0	Off
	1	50/15us 0dB
	2	50/15us - 10dB
	3	J17 0dB
	4	J17 - 20dB
Description	This command selects the Digital Input/Output Output Preemphasis. CD type (50/15 us) or CCITT J17 deemphasis may be selected as desired. Either preemphasis function may be selected at normal gain or with a headroom allowance. When program material is put through a	

preemphasis function, the natural high-frequency roll-off of most music and voice signals and typical practices of headroom allowance for peaks are sufficient to assure that high-frequency signals will not clip (exceed digital full scale). However, full-scale test signals such as sinewave sweeps or multitone signals with equal amplitude at all frequencies will clip at high frequencies when preemphasis is applied. To prevent this clipping due to the high-frequency boost, two additional selections are available which automatically attenuate the signal level sufficiently to provide headroom at the highest frequencies. These headroom allowances are selected by the 50/15 us -10 dB and J17 -20 dB choices. Each will attenuate the audio signal by the specified amount, which is slightly greater than the boost at the maximum possible audio frequency for the chosen preemphasis characteristic. If desired, a matching deemphasis with gain selection is available in the Deemphasis field or via the `AP.S2CDio.InDeEmp` command of the Input section of the DIO panel to provide an overall unity gain and flat response during digital domain stimulus/response measurements.

See Also `AP.S2CDio.InDeEmp`

Example See example for `AP.S2CDio.OutFormat`.

AP.S2CDio.OutRate

Property

Syntax `AP.S2CDio.OutRate(ByVal Unit As String)`

Data Type Double The digital output sample rate of System Two may be freely set across the range from 8 kHz to 100 kHz.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, and dPPM.

Description This command sets the Digital Output Sample Rate Frequency value.

Example See example for `AP.S2CDio.OutFormat`.

AP.S2CDio.OutResolution

Property

Syntax	<code>AP.S2CDio.OutResolution</code>	
Data Type	Integer	The width or resolution of the imbedded digital audio signal may be set to any value from 8 to 24 bits.
Description	<p>This command sets the Digital Output Resolution.</p> <p>Internally, the imbedded digital audio signal is always generated at 24 bits. When any smaller value is selected in the Resolution field, the 24-bit word is rounded (not truncated) to the specified value and dither is added (unless disabled) at the proper amplitude for the value entered. Bits below the value entered in the Resolution field are set to zero. The output resolution is independent from the input resolution.</p>	
See Also	<code>AP.S2CDio.InResolution</code>	
Example	See example for <code>AP.S2CDio.OutFormat</code> .	

AP.S2CDio.OutRiseFall

Property

Syntax	<code>AP.S2CDio.OutRiseFall</code>	
Data Type	Boolean	
	<i>True</i>	Fixed Rise and fall times.
	<i>False</i>	Variable Rise and fall times.
Description	<p>This command enables or disables variable rise and fall times of the pulse train at the XLR and BNC outputs. The fixed transition time is approximately 16 nanoseconds.</p>	
See Also	<code>AP.S2CDio.OutRiseFallTime</code>	
Example	See example for <code>AP.S2CDio.OutFormat</code> .	

AP.S2CDio.OutRiseFallTime

Property

Syntax	<code>AP.S2CDio.OutRiseFallTime(ByVal Unit As String)</code>	
Data Type	Double	

The rise and fall times may be varied from 16 to 400 nanoseconds.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: sec
Description		This command sets the variable rise and fall times of the pulse train at the XLR and BNC outputs.
See Also		AP.S2CDio.OutRiseFall
Example		See example for AP.S2CDio.OutFormat.

AP.S2CDio.OutSendInvalid

Property

Syntax `AP.S2CDio.OutSendInvalid`

Data Type Boolean

True Set
False Clear validity bit.

Description This command sets or clears the validity bit.

The AES/EBU and Consumer standards define a data invalid bit for each subframe. This is the V bit of the VUCP bits (validity, user, channel status, parity). Actual usage of this bit is not totally standardized, but a common usage in digital tape recorders (for example) is to set this bit as invalid if the tape is not moving and valid if the tape is playing. System Two permits the user to simultaneously set both channel A and B validity bits as true (check the Send Invalid box) or false (Send Invalid box unchecked) in order to test whether and how digital devices respond to the bit.

AP.S2CDio.OutVoltage

Property

Syntax `AP.S2CDio.Voltage(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command: Vpp
Description		This command sets the amplitude of the serial pulse train at the XLR, BNC and optical outputs, which may be used to simulate cable attenuation.
Example		See example for <code>AP.S2CDio.VoltageRdg</code> .

AP.S2CDio.RateRdg

Property

Syntax `AP.S2CDio.RateRdg(ByVal Unit As String)`

Data Type Double

Parameters	Part	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, and dPPM.

Description This command returns a settled reading for the Digital Input/Output Sample Rate meter and zeros the ready count.

See Also `AP.S2CDio.RateReady`, `AP.S2CDio.RateSettling`, `AP.S2CDio.RateTrig`

Example

```

Sub Main
    Dim rdgA As Double

    'S2Dio Sample Rate meter sample code
    AP.S2CDio.InFormat = 0 'XLR balanced input
    AP.S2CDio.InImpedance = 1 'High impedance input
    AP.S2CDio.RateSettling(5.0, 100e-3, "Hz", 1, 0.0, 0)
    AP.S2CDio.RateTrig 'Trigger sample rate meter
    Do Until AP.S2CDio.RateReady
        'perform other actions while waiting for readings
        '...
    Loop
    rdgA = AP.S2CDio.RateRdg("Hz") 'Get sample rate
    AP.Prompt.Text = "Ch A = " & rdgA & " Vpp"

```

```

        AP.Prompt.ShowWithContinue
    Stop
End Sub

```

AP.S2CDio.RateReady

Property

Syntax `AP.S2CDio.RateReady`

Data Type Integer

0 Reading not ready.
>0 Reading ready.

Description This command returns the Digital Input/Output Sample Rate meter settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDio.RateRdg` or `AP.S2CDio.RateTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDio.RateRdg` command will be guaranteed to return quickly.

See Also `AP.S2CDio.RateRdg`, `AP.S2CDio.RateSettling`, `AP.S2CDio.RateTrig`

Example See example for `AP.S2CDio.RateRdg`.

AP.S2CDio.RateSettling

Method

Syntax `AP.S2CDio.RateSettling(ByVal Tolerance As Double, ByVal Floor As Double, ByVal FloorUnit As String, ByVal Points As Integer, ByVal Delay As Double, ByVal Algorithm As Integer)`

Parameters See Appendix A for Settling Algorithm and parameter name descriptions.

Description	This command sets the settling parameters for the <code>AP.S2CDio.RateRdg</code> command.
See Also	<code>AP.S2CDio.RateRdg</code> , <code>AP.S2CDio.RateReady</code> , <code>AP.S2CDio.RateTrig</code>
Example	See example for <code>AP.S2CDio.RateRdg</code> .

AP.S2CDio.RateTrig

Method

Syntax	<code>AP.S2CDsp.RateTrig</code>
Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP.S2CDio.RateRdg</code> command. The reading in progress is aborted.
See Also	<code>AP.S2CDio.RateRdg</code> , <code>AP.S2CDio.RateReady</code> , <code>AP.S2CDio.RateSettling</code>
Example	See example for <code>AP.S2CDio.RateRdg</code> .

AP.S2CDio.RefRate

Property

Syntax	<code>AP.S2CDio.RefRate(ByVal Unit As String)</code>	
Data Type	Double	Set Reference Sample Rate value.
Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz
Description	This command sets the Digital Input/Output Sample Rate reference.	
See Also	<code>AP.S2CDsp.InScaleFreq</code> , <code>AP.S2CDsp.OutRate</code> , <code>AP.S2CDsp.RateRdg</code>	

AP.S2CDio.VoltageRdg

Property

Syntax	<code>AP.S2CDio.VoltageRdg(ByVal Unit As String)</code>
Data Type	Double

Parameters

Part	Description
<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command: Vpp

Description

This command returns a settled reading for the Digital Input/Output Voltage meter and zeros the ready count.

See Also

AP.S2CDio.VoltageReady,
AP.S2CDio.VoltageSettling, AP.S2CDio.VoltageTrig

Example

```
Sub Main
  Dim rdgA As Double

  'S2Dio Voltage out sample code
  AP.S2CDio.OutFormat = 0
  AP.S2CDio.InFormat = 0      'XLR balanced input
  AP.S2CDio.VoltageSettling(5.0, .001, "Vpp", 1, 0.0, 0)
  AP.S2CDio.OutVoltage("Vpp") = 1.0 'Set output voltage
  AP.S2CDio.VoltageTrig      'Trigger voltage meter
  Do Until AP.S2CDio.VoltageReady
    'perform other actions while waiting for readings
    '...
  Loop
  rdgA = AP.S2CDio.VoltageRdg("Vpp") 'Get reading
  AP.Prompt.Text = "Ch A = " & rdgA & " Vpp"
  AP.Prompt.ShowWithContinue
  Stop
End Sub
```

AP.S2CDio.VoltageReady

Property

Syntax `AP.S2CDio.VoltageReady`**Data Type** Integer`0` Reading not ready.`>0` Reading ready.**Description** This command returns the Digital Input/Output Voltage meter settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDio.VoltageRdg` or `AP.S2CDio.VoltageTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDio.VoltageRdg` command will be guaranteed to return quickly.

See Also `AP.S2CDio.VoltageRdg`, `AP.S2CDio.VoltageSettling`, `AP.S2CDio.VoltageTrig`**Example** See example for `AP.S2CDio.VoltageRdg`.

AP.S2CDio.VoltageSettling

Method

Syntax `AP.S2CDio.VoltageSettling(ByVal Tolerance As Double, ByVal Floor As Double, ByVal FloorUnit As String, ByVal Points As Integer, ByVal Delay As Double, ByVal Algorithm As Integer)`**Parameters** See Appendix A for Settling Algorithm and parameter name descriptions.**Description** This command sets the settling parameters for the `AP.S2CDio.VoltageRdg` command.**See Also** `AP.S2CDio.VoltageRdg`, `AP.S2CDio.VoltageReady`, `AP.S2CDio.VoltageTrig`

Example See example for `AP.S2CDio.VoltageRdg`.

AP.S2CDio.VoltageTrig

Method

Syntax `AP.S2CDsp.VoltageTrig`

Description Causes a restart of the reading cycle and zeros the ready count for the `AP.S2CDio.DelayRdg` command. The reading in progress is aborted.

See Also `AP.S2CDio.VoltageRdg`, `AP.S2CDio.VoltageReady`, `AP.S2CDio.VoltageSettling`

Example See example for `AP.S2CDio.VoltageRdg`.

User Notes

User Notes

DSP Audio Analyzer

AP.S2CDsp.Analyzer.ChACoupling

Property

Syntax	<code>AP.S2CDsp.Analyzer.ChACoupling</code>						
Data Type	Integer						
	<table> <tr> <td><i>0</i></td> <td>AC Coupled</td> </tr> <tr> <td><i>1</i></td> <td>DC Coupled</td> </tr> <tr> <td><i>2</i></td> <td>DC only: This enables System Two Cascade to DC couple the input to the Digital Domain Audio Analyzer. The channel A level meter can then be used to measure DC.</td> </tr> </table>	<i>0</i>	AC Coupled	<i>1</i>	DC Coupled	<i>2</i>	DC only: This enables System Two Cascade to DC couple the input to the Digital Domain Audio Analyzer. The channel A level meter can then be used to measure DC.
<i>0</i>	AC Coupled						
<i>1</i>	DC Coupled						
<i>2</i>	DC only: This enables System Two Cascade to DC couple the input to the Digital Domain Audio Analyzer. The channel A level meter can then be used to measure DC.						
Description	This command sets the Digital Domain Audio Analyzer channel A Input Coupling.						
See Also	<code>AP.S2CDsp.Analyzer.ChBCoupling</code>						
Example	See example for <code>AP.S2CDsp.Analyzer.ChAFreqRdg</code> .						

AP.S2CDsp.Analyzer.ChAFreqRdg

Property

Syntax	<code>AP.S2CDsp.Analyzer.ChAFreqRdg</code> (ByVal <i>Unit</i> As String)				
Data Type	Double				
Parameters	<table> <thead> <tr> <th>Part</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>Unit</i></td> <td>The following units are available: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM.</td> </tr> </tbody> </table>	Part	Description	<i>Unit</i>	The following units are available: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM.
Part	Description				
<i>Unit</i>	The following units are available: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM.				
Description	This command returns a settled reading for the Digital Domain Audio Analyzer channel A Frequency meter and zeros the ready count.				
See Also	<code>AP.S2CDsp.Analyzer.ChAFreqReady</code> , <code>AP.S2CDsp.Analyzer.ChAFreqSettling</code> , <code>AP.S2CDsp.Analyzer.ChAFreqTrig</code>				

Example

```

Sub Main
  AP.Application.NewTest
  AP.Gen.Output = True
  AP.Anlr.ChAInput = 2
  AP.Application.PanelOpen apbPanelDSPSmall
  AP.S2CDsp.Program = 1           'Load DSP Audio Analyzer
  AP.Application.PanelOpen apbPanelDSPLarge
  With AP.S2CDsp.Analyzer
    .ChACoupling = 0             'Input AC coupled
    .InputFormat = 1            'HiRes A/D @65k input
    .ChARangeAuto = False       'Fixed Range input
    .ChARange("FFS") = 1.000000 'In Range Full Scale

    .ChALevelSettling 1.000000, 0.000001, "V", 3, _
      0.030000, 1
    .ChAFreqSettling 0.500000, 0.010000, "Hz", 3, _
      0.030000, 1
  Wait .5
    .ChALevelTrig           'Trigger new Level reading
    .ChAFreqTrig           'Trigger new Frequency reading
  Do                       'Wait for new readings
    Loop Until .ChALevelReady And .ChAFreqReady
  var1 = .ChALevelRdg("V")
  var2 = .ChAFreqRdg("Hz")
End With
Text1$= "Channel A Level " & Str$(Format(var1, _
  "##.000")) & "V"
Text2$= "Channel A Frequency " & Str$(Format(var2, _
  "##.000")) & "Hz"
AP.Prompt.Text = Text1$ & Chr(13) & Text2$ 'Text _
  String and New Line
AP.Prompt.ShowWithContinue           'Display Prompt
Stop
End Sub

```

AP.S2CDsp.Analyzer.ChAFreqReady**Property****Syntax** `AP.S2CDsp.Analyzer.ChAFreqReady`**Data Type** Integer`0` Reading not ready.`>0` Reading ready.**Description** This command returns the Digital Domain Audio Analyzer channel A Frequency meter settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.Analyzer.ChAFreqRdg` or `AP.S2CDsp.Analyzer.ChAFreqTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.Analyzer.ChAFreqRdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.Analyzer.ChAFreqRdg`,
`AP.S2CDsp.Analyzer.ChAFreqSettling`,
`AP.S2CDsp.Analyzer.ChAFreqTrig`**Example** See example for `AP.S2CDsp.Analyzer.ChAFreqRdg`.**AP.S2CDsp.Analyzer.ChAFreqSettling****Method****Syntax** `AP.S2CDsp.Analyzer.ChAFreqSettling(ByVal Tolerance As Double, ByVal Floor As Double, ByVal FloorUnit As String, ByVal Points As Integer, ByVal Delay As Double, ByVal Algorithm As Integer)`**Parameters** See Appendix A for Settling Algorithm and parameter name descriptions.**Description** This command sets the settling parameters for the `AP.S2CDsp.Analyzer.ChAFreqRdg` command.

See Also `AP.S2CDsp.Analyzer.ChAFreqRdg`,
`AP.S2CDsp.Analyzer.ChAFreqReady`,
`AP.S2CDsp.Analyzer.ChAFreqTrig`

Example See example for `AP.S2CDsp.Analyzer.ChAFreqRdg`.

AP.S2CDsp.Analyzer.ChAFreqTrig

Method

Syntax `AP.S2CDsp.Analyzer.ChAFreqTrig`

Description Causes a restart of the reading cycle and zeros the ready count for the `AP.S2CDsp.Analyzer.ChAFreqRdg` command. The reading in progress is aborted.

See Also `AP.S2CDsp.Analyzer.ChAFreqRdg`,
`AP.S2CDsp.Analyzer.ChAFreqReady`,
`AP.S2CDsp.Analyzer.ChAFreqSettling`

Example See example for `AP.S2CDsp.Analyzer.ChAFreqRdg`.

AP.S2CDsp.Analyzer.ChALevelRdg

Property

Syntax `AP.S2CDsp.Analyzer.ChALevelRdg`(ByVal *Unit* As String)

Data Type Double

Description This command returns a settled reading for the Digital Domain Audio Analyzer channel A level meter and zeros the ready count.

Parameters

Name	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for the "Digital @ ISR" selection of the <code>AP.S2CDSP.Analyzer.InputFormat</code> command: FFS, %FS, dBFS, Bits, V, Vp, Vpp, dBu, dBV, dBr 1, dBr 2.

The following units are valid for the " A/D ?? " selections of the `AP.S2CDSP.Analyzer.InputFormat` command:
V, dBu, dBV, dBr A, dBr B, dBg A, dBg B, dBm, and W.

See Also `AP.S2CDsp.Analyzer.ChALevelReady`,
`AP.S2CDsp.Analyzer.ChALevelSettling`,
`AP.S2CDsp.Analyzer.ChALevelTrig`

Example See example for `AP.S2CDsp.Analyzer.ChAFreqRdg`.

AP.S2CDsp.Analyzer.ChALevelReady

Property

Syntax `AP.S2CDsp.Analyzer.ChALevelReady`

Data Type Integer

0 Reading not ready.
>0 Reading ready.

Description This command returns the Digital Domain Audio Analyzer channel A Level Monitor meter settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.Analyzer.ChALevelRdg` or `AP.S2CDsp.Analyzer.ChALevelTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.Analyzer.ChALevelRdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.Analyzer.ChALevelRdg`,
`AP.S2CDsp.Analyzer.ChALevelSettling`,
`AP.S2CDsp.Analyzer.ChALevelTrig`

Example See example for `AP.S2CDsp.Analyzer.ChAFreqRdg`.

AP.S2CDsp.Analyzer.ChALevelSettling**Method**

Syntax	<code>AP.S2CDsp.Analyzer.ChALevelSettling</code> (ByVal <i>Tolerance</i> As Double, ByVal <i>Floor</i> As Double, ByVal <i>FloorUnit</i> As String, ByVal <i>Points</i> As Integer, ByVal <i>Delay</i> As Double, ByVal <i>Algorithm</i> As Integer)
Parameters	See Appendix A for Settling Algorithm and parameter name descriptions.
Description	This command sets the settling parameters for the <code>AP.S2CDsp.Analyzer.ChALevelRdg</code> command.
See Also	<code>AP.S2CDsp.Analyzer.ChALevelRdg</code> , <code>AP.S2CDsp.Analyzer.ChALevelReady</code> , <code>AP.S2CDsp.Analyzer.ChALevelTrig</code>
Example	See example for <code>AP.S2CDsp.Analyzer.ChAFreqRdg</code> .

AP.S2CDsp.Analyzer.ChALevelTrig**Method**

Syntax	<code>AP.S2CDsp.Analyzer.ChALevelTrig</code>
Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP.S2CDsp.Analyzer.ChALevelRdg</code> command. The reading in progress is aborted.
See Also	<code>AP.S2CDsp.Analyzer.ChALevelRdg</code> , <code>AP.S2CDsp.Analyzer.ChALevelSettling</code> , <code>AP.S2CDsp.Analyzer.ChALevelTrig</code>
Example	See example for <code>AP.S2CDsp.Analyzer.ChAFreqRdg</code> .

AP.S2CDsp.Analyzer.ChARange**Property**

Syntax	<code>AP.S2CDsp.Analyzer.ChARange</code> (ByVal <i>Unit</i> As String)
Data Type	Double The following values are the range boundaries for the dBFS unit: 0.000, -6.021, -12.041, -18.062, -24.82, -30.103, -36.124,

-42.144, -48.165, -54.185, -60.206, -66.227, -72.247, -78.268, -84.288, -90.309, -186.639.

If an arbitrary value between the range boundaries is entered the next higher range will be selected.

Parameters	Name	Description
	<i>Unit</i>	The following units are available: FFS, %FS, dBFS, dBr 1.
Description		This command sets the Digital Domain Audio Analyzer Input Range and returns the nominal full scale of the range in use.
See Also		AP.S2CDsp.Analyzer.ChARangeAuto
Example		See example for AP.S2CDsp.Analyzer.ChAFreqRdg.

AP.S2CDsp.Analyzer.ChARangeAuto

Property

Syntax AP.S2CDsp.Analyzer.ChARangeAuto

Data Type Boolean

<i>True</i>	Auto range
<i>False</i>	Fixed range

Description This command sets the Digital Domain Audio Analyzer channel A input to Auto range or Fixed range. Care must be taken when using Fixed range that the input signal does not exceed the selected range.

See Also AP.S2CDsp.Analyzer.ChARange

Example See example for AP.S2CDsp.Analyzer.ChAFreqRdg.

AP.S2CDsp.Analyzer.ChBCoupling

Property

Syntax AP.S2CDsp.Analyzer.ChBCoupling

Data Type Integer

0	AC Coupled
1	DC Coupled

2 DC only: This enables System Two to DC couple the input to the Digital Domain Audio Analyzer. The channel B level meter can then be used to measure DC.

Description This command sets the Digital Domain Audio Analyzer channel B Input Coupling.

See Also `AP.S2CDsp.Analyzer.ChACoupling`

Example See example for `AP.S2CDsp.Analyzer.ChBFreqRdg`.

AP.S2CDsp.Analyzer.ChBFreqRdg

Property

Syntax `AP.S2CDsp.Analyzer.ChBFreqRdg(ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
<i>Unit</i>	The following units are available: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM.

Description This command returns a settled reading for the Digital Domain Audio Analyzer channel B Frequency meter and zeros the ready count.

See Also `AP.S2CDsp.Analyzer.ChBFreqReady`,
`AP.S2CDsp.Analyzer.FreqBTrig`,
`AP.S2CDsp.Analyzer.ChBFreqSettling`

Example

```
Sub Main
    AP.Application.NewTest
    AP.Gen.Output = True
    AP.Anlr.ChBInput = 2
    AP.Application.PanelOpen apbPanelDSPSmall
    AP.S2CDsp.Program = 1      'Load DSP Audio Analyzer
    AP.Application.PanelOpen apbPanelDSPLarge
    With AP.S2CDsp.Analyzer
        .ChBCoupling = 0      'Input AC coupled
        .InputFormat = 1      'HiRes A/D @65k input
        .ChBRangeAuto = False 'Fixed Range input
        .ChBRange("FFS") = 1.000000 'In Range Full Scale
    End With
End Sub
```

```

        .ChBLevelSettling 1.000000, 0.000001, "V", 3, _
            0.030000, 1
        .ChBFreqSettling 0.500000, 0.010000, "Hz", 3, _
            0.030000, 1
    Wait .5
    .ChBLevelTrig      'Trigger new Level reading
    .ChBFreqTrig      'Trigger new Frequency reading
    Do                 'Wait for new readings
        Loop Until .ChBLevelReady And .ChBFreqReady
    var1 = .ChBLevelRdg("V")
    var2 = .ChBFreqRdg("Hz")
End With
Text1$= "Channel B Level " & Str$(Format(var1, _
    "##.000")) & "V"
Text2$= "Channel B Frequency " & Str$(Format(var2, _
    "##.000")) & "Hz"
AP.Prompt.Text = Text1$ & Chr(13) & Text2$ 'Text _
    String and New Line
AP.Prompt.ShowWithContinue      'Display Prompt
Stop
End Sub

```

AP.S2CDsp.Analyzer.ChBFreqReady

Property

Syntax `AP.S2CDsp.Analyzer.ChBFreqReady`

Data Type Integer

0 Reading not ready.

>0 Reading ready.

Description This command returns the Digital Domain Audio Analyzer Frequency B settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.Analyzer.ChBFreqRdg` or `AP.S2CDsp.Analyzer.FreqBTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.Analyzer.ChBFreqRdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.Analyzer.ChBFreqRdg`,
`AP.S2CDsp.Analyzer.ChBFreqSettling`,
`AP.S2CDsp.Analyzer.ChBFreqTrig`

Example See example for `AP.S2CDsp.Analyzer.ChBFreqRdg`.

AP.S2CDsp.Analyzer.ChBFreqSettling

Method

Syntax `AP.S2CDsp.Analyzer.ChBFreqSettling`(ByVal *Tolerance* As Double, ByVal *Floor* As Double, ByVal *FloorUnit* As String, ByVal *Points* As Integer, ByVal *Delay* As Double, ByVal *Algorithm* As Integer)

Parameters See Appendix A for Settling Algorithm and parameter name descriptions.

Description This command sets the settling parameters for the `AP.S2CDsp.Analyzer.ChBFreqRdg` command.

See Also `AP.S2CDsp.Analyzer.ChBFreqRdg`,
`AP.S2CDsp.Analyzer.FreqBTrig`,
`AP.S2CDsp.Analyzer.FreqBReady`

Example See example for `AP.S2CDsp.Analyzer.ChBFreqRdg`.

AP.S2CDsp.Analyzer.ChBFreqTrig

Method

Syntax `AP.S2CDsp.Analyzer.ChBFreqTrig`

Description Causes a restart of the reading cycle and zeros the ready count for the `AP.S2CDsp.Analyzer.ChBFreqRdg` command. The reading in progress is aborted.

See Also `AP.S2CDsp.Analyzer.ChBFreqRdg`,
`AP.S2CDsp.Analyzer.ChBFreqReady`,
`AP.S2CDsp.Analyzer.ChBFreqSettling`

Example See example for `AP.S2CDsp.Analyzer.ChBFreqRdg`.

AP.S2CDsp.Analyzer.ChBLevelRdg

Property

Syntax `AP.S2CDsp.Analyzer.ChBLevelRdg(ByVal Unit As String)`

Data Type Double

Parameters

Name	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for the "Digital @ ISR" selection of the <code>AP.S2CDSP.Analyzer.InputFormat</code> command: FFS, %FS, dBFS, Bits, V, Vp, Vpp, dBu, dBV, dB _r 1, dB _r 2. The following units are valid for the "A/D ???" selections of the <code>AP.S2CDSP.Analyzer.InputFormat</code> command: V, dBu, dBV, dB _r A, dB _r B, dB _g A, dB _g B, dB _m , and W.

Description This command returns a settled reading for the Digital Domain Audio Analyzer channel B Level meter and zeros the ready count.

See Also `AP.S2CDsp.Analyzer.ChBLevelReady`,
`AP.S2CDsp.Analyzer.ChBLevelSettling`,
`AP.S2CDsp.Analyzer.ChBLevelTrig`

Example See example for `AP.S2CDsp.Analyzer.ChBFreqRdg`.

AP.S2CDsp.Analyzer.ChBLevelReady

Property

Syntax `AP.S2CDsp.Analyzer.ChBLevelReady`

Data Type Integer

<i>0</i>	Reading not ready.
<i>>0</i>	Reading ready.

Description This command returns the Digital Domain Audio Analyzer channel B Level Monitor meter settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.Analyzer.ChBLevelRdg` or `AP.S2CDsp.Analyzer.ChBLevelTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.Analyzer.ChBLevelRdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.Analyzer.ChBLevelRdg`,
`AP.S2CDsp.Analyzer.ChBLevelSettling`,
`AP.S2CDsp.Analyzer.ChBLevelTrig`,

Example See example for `AP.S2CDsp.Analyzer.ChBFreqRdg`.

AP.S2CDsp.Analyzer.ChBLevelSettling

Method

Syntax `AP.S2CDsp.Analyzer.ChBLevelSettling`(ByVal *Tolerance* As Double, ByVal *Floor* As Double, ByVal *FloorUnit* As String, ByVal *Points* As Integer, ByVal *Delay* As Double, ByVal *Algorithm* As Integer)

Parameters See Appendix A for Settling Algorithm and parameter name descriptions.

Description This command sets the settling parameters for the `AP.S2CDsp.Analyzer.ChBLevelRdg` command.

See Also `AP.S2CDsp.Analyzer.ChBLevelRdg`,
`AP.S2CDsp.Analyzer.ChBLevelReady`,
`AP.S2CDsp.Analyzer.ChBLevelTrig`

Example See example for `AP.S2CDsp.Analyzer.ChBFreqRdg`.

AP.S2CDsp.Analyzer.ChBLevelTrig

Method

Syntax `AP.S2CDsp.Analyzer.ChBLevelTrig`

Description Causes a restart of the reading cycle and zeros the ready count for the `AP.S2CDsp.Analyzer.ChBLevelRdg` command. The reading in progress is aborted.

See Also `AP.S2CDsp.Analyzer.ChBLevelRdg`,
`AP.S2CDsp.Analyzer.ChBLevelReady`,
`AP.S2CDsp.Analyzer.ChBLevelSettling`

Example See example for `AP.S2CDsp.Analyzer.ChBFreqRdg`.

AP.S2CDsp.Analyzer.ChBRange

Property

Syntax `AP.S2CDsp.Analyzer.ChBRange(ByVal Unit As String)`

Data Type Double

The following values are the range boundaries for the dBFS unit: 0.000, -6.021, -12.041, -18.062, -24.82, -30.103, -36.124, -42.144, -48.165, -54.185, -60.206, -66.227, -72.247, -78.268, -84.288, -90.309, -186.639.

If an arbitrary value between the range boundaries is entered the next higher range will be selected.

Parameters

Name

Description

Unit

The following units are available: FFS, %FS, dBFS, dBr 2.

Description

This command sets the Digital Domain Audio Analyzer Input Range and returns the nominal full scale of the range in use.

See Also `AP.S2CDsp.Analyzer.ChBRangeAuto`

Example See example for `AP.S2CDsp.Analyzer.ChBFreqRdg`.

AP.S2CDsp.Analyzer.ChBRangeAuto

Property

Syntax `AP.S2CDsp.Analyzer.ChBRangeAuto`

Data Type Boolean

True Auto range

False Fixed range

Description This command sets the Digital Domain Audio Analyzer channel B input to Auto range or Fixed range. Care must be taken when using Fixed range that the input signal does not exceed the selected range.

See Also `AP.S2CDsp.Analyzer.ChBRange`

Example See example for `AP.S2CDsp.Analyzer.ChBFreqRdg`.

AP.S2CDsp.Analyzer.FuncBPBRFreq

Property

Syntax `AP.S2CDsp.Analyzer.FuncBPBRFreq(ByVal Unit As String)`

Data Type Double

Parameters

Name	Description
<i>Unit</i>	The following units are available: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM.

Description

This command sets the Digital Domain Audio Analyzer BandPass/BandReject filter to the frequency value passed.

The DSP-implemented Bandpass filter affects only the main (Reading) meter, not the Level Monitor or Frequency reading. It is a highly-selective filter of about 1/13 octave bandwidth (Q=19, 3 dB bandwidth about 5.2% of center frequency). The bandpass filter is tunable across the audio spectrum from 0.04% to 42% of the sample rate (20 Hz to 20 kHz at a 48 kHz sample rate). It is used in Bandpass and Crosstalk functions.

The Bandreject (notch) function of the filter is used in the two THD+N functions. It is tunable from 0.04% to 42% of the sample rate (20 Hz to 20 kHz at a 48 kHz rate).

See Also `AP.S2CDsp.Analyzer.FuncBPBRTuning`,
`AP.S2CDsp.Analyzer.FuncMode`

Example

```
Sub Main
    AP.Application.NewTest
    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2
    AP.Application.PanelOpen apbPanelDSPSmall
```

```

AP.S2CDsp.Program = 1      'Load DSP Audio Analyzer
AP.Application.PanelOpen apbPanelDSPLarge
With AP.S2CDsp.Analyzer
    .InputFormat = 1      'HiRes A/D @65k input
    .FuncMode = 5        'Mode Bandpass
    .RdgRate = 4         'Set Reading Rate 32/Sec
    .FuncDetector = 0    'Set Detector to RMS
    .FuncBPBRTuning = 4    'Fixed Tuning at 1kHz
    .FuncBPBRFreq("Hz") = 999.999046

    .FuncChASettling 3.000000, 0.000000, "V", 3, _
        0.100000, 1
    Wait .5
    .FuncChATrig          'Trigger new reading
    Do                    'Wait for new reading
        Loop Until .FuncChAReady
    var1 = .FuncChARdg("V") 'Return reading
End With
Text$= "Bandpass Amplitude " & Str$(Format(var1, _
    "##.000")) & "V"
AP.Prompt.Text = Text$ & Chr(13) 'Text and New Line
AP.Prompt.ShowWithContinue      'Display Prompt
Stop
End Sub

```

AP.S2CDsp.Analyzer.FuncBPBRTuning

Property

Syntax

AP.S2CDsp.Analyzer.FuncBPBRTuning

Data Type

Integer

0

Counter Tuned: the frequency value measured by the ANALYZER Frequency counter is the filter steering source. This function would be selected when making THD+N or Crosstalk measurements from an external signal such as reproduction of a Compact Disc or digital audio tape or reception of a digital signal from a distant source.

1

Sweep Track: the filter tracks the frequency of whichever generator is selected in the Source 1 or Source 2 fields of the Sweep panel.

- 2 AGen Track: the digital bandpass-bandreject filter tracks the frequency of the Analog Generator, This mode is useful for testing A/D converters driven from System Two's analog output.
- 3 DGen Track: the filter will automatically track the frequency of the Digital Generator. This mode would normally be used when sweeping digital input-digital output devices with stimulus coming from System Two's Digital Generator.
- 4 Fixed: the filter will be fixed at the frequency entered in the BP/BR Filter Freq field just below unless the filter is being deliberately varied as part of a sweep test. To sweep the filter frequency during a test, select BP/BR Filter Freq as the Source 1 or Source 2 parameter on the Sweep panel. Fixed tuning mode must be selected in order to use the BP/BR Filter Freq parameter as a Source value.

Description This command sets the Digital Domain Audio Analyzer Bandpass Bandreject filter Tuning source.

See Also `AP.S2CDsp.Analyzer.FuncBPBRFreq`

Example See example for `AP.S2CDsp.Analyzer.FuncBPBRFreq`.

AP.S2CDsp.Analyzer.FuncChARange

Property

Syntax `AP.S2CDsp.Analyzer.FuncRange(ByVal Unit As String)`

Data Type Double

The following values are the range boundaries for the dBFS unit: 0.000, -6.021, -12.041, -18.062, -24.82, -30.103, -36.124, -42.144, -48.165, -54.185, -60.206, -66.227, -72.247, -78.268, -84.288, -90.309, -186.639.

If an arbitrary value between the range boundaries is entered the next higher range will be selected.

Parameters	Name	Description
	<i>Unit</i>	The following units are available: FFS, %FS, dBFS, dBr 1, dBr 2.

Description This command sets the Digital Domain Audio Analyzer channel A Function meter Range.

See Also `AP.S2CDsp.Analyzer.RdgRate`,
`AP.S2CDsp.Analyzer.FuncMode`,
`AP.S2CDsp.Analyzer.FuncChARangeAuto`

Example See example for `AP.S2CDsp.Analyzer.FuncDetector`.

AP.S2CDsp.Analyzer.FuncChARangeAuto

Property

Syntax `AP.S2CDsp.Analyzer.FuncChARangeAuto`

Data Type Boolean

True Auto range.
False Fixed range.

Description This command sets the Digital Domain Audio Analyzer channel A Function meter to Auto or Fixed Range.

See Also `AP.S2CDsp.Analyzer.RdgRate`,
`AP.S2CDsp.Analyzer.FuncMode`,
`AP.S2CDsp.Analyzer.FuncChARange`

Example See example for `AP.S2CDsp.Analyzer.FuncDetector`.

AP.S2CDsp.Analyzer.FuncChARdg

Property

Syntax `AP.S2CDsp.Analyzer.FuncChARdg`(ByVal *Unit* As String)

Data Type Double

Parameters

Part	Description
<i>Unit</i>	String that designates the desired unit.

The following units are valid for the "Digital" selection of the `AP.S2DSP.Analyzer.InputFormat` command for the Amplitude, THD+N Ampl, and Bandpass Function meter Modes: FFS, %FS, dBFS, Bits, V, Vp, Vpp, dBu, dBV, dBr 1, dBr 2.

The following units are valid for the "Low BW (1x) A/D" selection of the `AP.S2DSP.Analyzer.InputFormat` command for the Amplitude, THD+N Ampl, and Bandpass Function meter Modes: V, dBu, dBV, dBr A, dBr B, dBg A, dBg B, dBm, W.

The following units (% , dB, PPM, X/Y) are available for the following Function meter Modes: 2-Ch Ratio, Crosstalk, THD+N Ratio, SMPTE IMD.

The (deg) units is available for the Function meter Phase Mode.

Description	This command returns a settled reading from the Digital Domain Audio Analyzer channel A Function meter and zeros the ready count.
See Also	<code>AP.S2CDsp.Analyzer.FuncMode</code> , <code>AP.S2CDsp.Analyzer.FuncChAReady</code> , <code>AP.S2CDsp.Analyzer.FuncChASettling</code> , <code>AP.S2CDsp.Analyzer.FuncChATrig</code>
Example	See example for <code>AP.S2CDsp.Analyzer.FuncDetector</code> .

AP.S2CDsp.Analyzer.FuncChAReady

Property

Syntax `AP.S2CDsp.Analyzer.FuncChAReady`

Data Type Integer

0 Reading not ready.
>0 Reading ready.

Description This command returns the Digital Domain Audio Analyzer channel A Function meter settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.Analyzer.FuncChARdgr`

AP.S2CDsp.Analyzer.FuncChATrig commands will zero the ready count.

If the reading is found to be ready, a call to the AP.S2CDsp.Analyzer.FuncChARdg command will be guaranteed to return quickly.

See Also AP.S2CDsp.Analyzer.FuncChARdg,
AP.S2CDsp.Analyzer.FuncChASettling,
AP.S2CDsp.Analyzer.FuncChATrig

Example See example for AP.S2CDsp.Analyzer.FuncDetector.

AP.S2CDsp.Analyzer.FuncChASettling

Method

Syntax **AP.S2CDsp.Analyzer.FuncChASettling**(ByVal *Tolerance* As Double, ByVal *Floor* As Double, ByVal *FloorUnit* As String, ByVal *Points* As Integer, ByVal *Delay* As Double, ByVal *Algorithm* As Integer)

Parameters See Appendix A for Settling Algorithm and parameter name descriptions.

Description This command sets the settling parameters for the AP.S2CDsp.Analyzer.FuncChARdg command.

See Also AP.S2CDsp.Analyzer.FuncChARdg,
AP.S2CDsp.Analyzer.FuncChAReady,
AP.S2CDsp.Analyzer.FuncChATrig

Example See example for AP.S2CDsp.Analyzer.FuncDetector.

AP.S2CDsp.Analyzer.FuncChATrig

Method

Syntax **AP.S2CDsp.Analyzer.FuncChATrig**

Description Causes a restart of the reading cycle and zeros the ready count for the AP.S2CDsp.Analyzer.FuncChARdg command. The reading in progress is aborted.

See Also `AP.S2CDsp.Analyzer.FuncChARdg`,
`AP.S2CDsp.Analyzer.FuncChAReady`,
`AP.S2CDsp.Analyzer.FuncChASettling`

Example See example for `AP.S2CDsp.Analyzer.FuncDetector`.

AP.S2CDsp.Analyzer.FuncChBRange Property

Syntax `AP.S2CDsp.Analyzer.FuncChBRange(ByVal Unit As String)`

Data Type Double

The following values are the range boundaries for the dBFS unit: 0.000, -6.021, -12.041, -18.062, -24.82, -30.103, -36.124, -42.144, -48.165, -54.185, -60.206, -66.227, -72.247, -78.268, -84.288, -90.309, -186.639.

If an arbitrary value between the range boundaries is entered the next higher range will be selected.

Parameters	Name	Description
	<i>Unit</i>	The following units are available: FFS, %FS, dBFS, dBr 1, dBr 2.

Description This command sets the Digital Domain Audio Analyzer channel B Function meter Range.

See Also `AP.S2CDsp.Analyzer.RdgRate`,
`AP.S2CDsp.Analyzer.FuncMode`,
`AP.S2CDsp.Analyzer.FuncChBRangeAuto`

Example See example for `AP.S2CDsp.Analyzer.FuncDetector`.

AP.S2CDsp.Analyzer.FuncChBRangeAuto Property

Syntax `AP.S2CDsp.Analyzer.FuncChBRangeAuto`

Data Type Boolean

<i>True</i>	Auto range.
<i>False</i>	Fixed range.

Description This command sets the Digital Domain Audio Analyzer channel B Function meter to Auto or Fixed Range.

See Also `AP.S2CDsp.Analyzer.RdgRate`,
`AP.S2CDsp.Analyzer.FuncMode`,
`AP.S2CDsp.Analyzer.FuncChBRange`

Example See example for `AP.S2CDsp.Analyzer.FuncDetector`.

AP.S2CDsp.Analyzer.FuncChBRdg

Property

Syntax `AP.S2CDsp.Analyzer.FuncChBRdg(ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
<i>Unit</i>	String that designates the desired unit. <p>The following units are valid for the "Digital" selection of the <code>AP.S2DSP.Analyzer.InputFormat</code> command for the Amplitude, THD+N Ampl, and Bandpass Function meter Modes: FFS, %FS, dBFS, Bits, V, Vp, Vpp, dBu, dBV, dBr 1, dBr 2.</p> <p>The following units are valid for the "Low BW (1x) A/D" selection of the <code>AP.S2DSP.Analyzer.InputFormat</code> command for the Amplitude, THD+N Ampl, and Bandpass Function meter Modes: V, dBu, dBV, dBr A, dBr B, dBg A, dBg B, dBm, W.</p> <p>The following units (% , dB, PPM, X/Y) are available for the following Function meter Modes: 2-Ch Ratio, Crosstalk, THD+N Ratio, SMPTE IMD.</p> <p>The (deg) units is available for the Function meter Phase Mode.</p>

Description This command returns a settled reading from the Digital Domain Audio Analyzer channel B Function meter and zeros the ready count.

See Also `AP.S2CDsp.Analyzer.FuncMode`,
`AP.S2CDsp.Analyzer.FuncChBReady`,
`AP.S2CDsp.Analyzer.FuncChBSettling`,
`AP.S2CDsp.Analyzer.FuncChBTrig`

Example See example for `AP.S2CDsp.Analyzer.FuncDetector`.

AP.S2CDsp.Analyzer.FuncChBReady

Property

Syntax `AP.S2CDsp.Analyzer.FuncChBReady`

Data Type Integer

0 Reading not ready.
 >0 Reading ready.

Description This command returns the Digital Domain Audio Analyzer Function meter settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.Analyzer.FuncChBRdg` or `AP.S2CDsp.Analyzer.FuncChBTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.Analyzer.FuncChBRdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.Analyzer.FuncChBRdg`,
`AP.S2CDsp.Analyzer.FuncChBSettling`,
`AP.S2CDsp.Analyzer.FuncChBTrig`

Example See example for `AP.S2CDsp.Analyzer.FuncDetector`.

AP.S2CDsp.Analyzer.FuncChBSettling**Method**

Syntax	AP.S2CDsp.Analyzer.FuncChBSettling (ByVal <i>Tolerance</i> As Double, ByVal <i>Floor</i> As Double, ByVal <i>FloorUnit</i> As String, ByVal <i>Points</i> As Integer, ByVal <i>Delay</i> As Double, ByVal <i>Algorithm</i> As Integer)
Parameters	See Appendix A for Settling Algorithm and parameter name descriptions.
Description	This command sets the settling parameters for the AP.S2CDsp.Analyzer.FuncChBRdg command.
See Also	AP.S2CDsp.Analyzer.FuncChBRdg, AP.S2CDsp.Analyzer.FuncChBReady, AP.S2CDsp.Analyzer.FuncChBTrig
Example	See example for AP.S2CDsp.Analyzer.FuncDetector.

AP.S2CDsp.Analyzer.FuncChBTrig**Method**

Syntax	AP.S2CDsp.Analyzer.FuncChBTrig
Description	Causes a restart of the reading cycle and zeros the ready count for the AP.S2CDsp.Analyzer.FuncChBRdg command. The reading in progress is aborted.
See Also	AP.S2CDsp.Analyzer.FuncChBRdg, AP.S2CDsp.Analyzer.FuncChBReady, AP.S2CDsp.Analyzer.FuncChBSettling
Example	See example for AP.S2CDsp.Analyzer.FuncDetector.

AP.S2CDsp.Analyzer.FuncDetector**Property**

Syntax	AP.S2CDsp.Analyzer.FuncDetector
Data Type	Integer
	0 RMS: Level and Function meters use RMS detector.

- 1 Fast RMS: Level and Function meters use Fast RMS detector.
- 2 Qpeak: Level meters use RMS detection and Function meters use Qpeak detector.

Description This command selects the Digital Domain Audio Analyzer Detector type for the Level and Function meters.

See Also AP.S2CDsp.Analyzer.RdgRate,
AP.S2CDsp.Analyzer.FuncMode,
AP.S2CDsp.Analyzer.FuncChBRange,
AP.S2CDsp.Analyzer.FuncChBRangeAuto

Example

```
Sub Main
  AP.Application.NewTest
  AP.Gen.Output = True
  AP.Anlr.ChAInput = 2
  AP.Application.PanelOpen apbPanelDSPSmall
  AP.S2CDsp.Program = 1      'Load DSP Audio Analyzer
  AP.Application.PanelOpen apbPanelDSPLarge
  With AP.S2CDsp.Analyzer
    .InputFormat = 1        'HiRes A/D @65k input
    .FuncMode = 3          'THD+N measurement Mode
    .FuncRangeAuto = True  'Auto Range
    ' .ChARange("dBFS") = 0.000000 'Use with Fixed Range
    .RdgRate = 0           'Auto Reading Rate
    .FuncDetector = 0      'RMS Detector
    .FilterHP = 0         'Set HP Filter to <10Hz
    .FuncFilterLP = 1     'Set LP Filter to 20kHz
    .FuncFilter = 0       'No Auxiliary Filter

    .FuncChASettling 3.000000, 0.000010, "%", 3, _
      0.100000, 1
    Wait .5
    .FuncChATrig          'Trigger new Function _
      meter reading
    Do                    'Wait for new readings
      Loop Until .FuncChAReady
    var = .FuncChARDg("%") 'Get Reading
  End With
```

```

Text$= "THD+N = " & Str$(Format(var, "##.00000")) _
      & "%"
AP.Prompt.Text = Text$ & Chr(13) 'Text and New Line
AP.Prompt.ShowWithContinue 'Display Prompt
Stop
End Sub

```

AP.S2CDsp.Analyzer.FuncFilter

Property

Syntax `AP.S2CDsp.Analyzer.FuncFilter`

Data Type Integer

The following list contains the selections relevant to the `AP.S2CDsp.Analyzer.FuncMode` command Amplitude, THD+N Abs, THD+N Ratio, and 2-Ch Ratio selections.

0	None
1	"A" Weighting
2	CCIR Weighting
3	F Weighting
4	CCITT Weighting
5	C-Message Weighting
6	Harmonic Weighting (THD Only)

The following list contains the selections relevant to the `AP.S2CDsp.Analyzer.FuncMode` command Bandpass selection.

0	Narrow
1	Narrow, Freq x2
2	Narrow, Freq x3
3	Narrow, Freq x4
4	Narrow, Freq x5

The following list contains the selections relevant to the `AP.S2CDsp.Analyzer.FuncMode` command Crosstalk selection.

0	Narrow
---	--------

Description	This command selects the Digital Domain Audio Analyzer Weighting Filter.
See Also	AP.S2CDsp.Analyzer.FuncFilterHP, AP.S2CDsp.Analyzer.FuncFilterLP
Example	See example for AP.S2CDsp.Analyzer.FuncDetector.

AP.S2CDsp.Analyzer.FuncFilterHP

Property

Syntax `AP.S2CDsp.Analyzer.FuncFilterHP`

Data Type Integer

0	<10 Hz
1	22 Hz
2	100 Hz
3	400 Hz

Description This command selects the Digital Domain Audio Analyzer High Pass filter used with the function meters.

See Also AP.S2CDsp.Analyzer.FuncFilterLowPass

Example See example for AP.S2CDsp.Analyzer.FuncDetector.

AP.S2CDsp.Analyzer.FuncFilterLP

Property

Syntax `AP.S2CDsp.Analyzer.FuncFilterLP`

Data Type Integer

0	Fs/2
1	20 kHz LP
2	15 kHz LP

Description This command selects the Digital Domain Audio Analyzer Low Pass filter used with the function meters.

See Also AP.S2CDsp.Analyzer.FuncFilterHP

Example See example for `AP.S2CDsp.Analyzer.FuncDetector`.

AP.S2CDsp.Analyzer.FuncMode

Property

Syntax `AP.S2CDsp.Analyzer.FuncMode`

Data Type Integer

- 0* Amplitude: Amplitude mode connects the Function Reading meter to the channel selected by the `AP.S2CDsp.Analyzer.FuncInput` command. Amplitude mode measurements can differ from the Level meter measurements due to two factors: Amplitude mode measurements are affected by the high-pass (`AP.S2CDsp.Analyzer.FilterHP`), low-pass (`AP.S2CDsp.Analyzer.FilterLP`), and weighting filter (`AP.S2CDsp.Analyzer.FuncFilter`) commands, while Level meter readings are unfiltered. Amplitude measurements may be made with the quasi-peak or one of the RMS detectors, while the Level meters always use the same type of RMS detector selected with the `AP.S2CDsp.Analyzer.FuncDetector` command.
- 1* 2-Channel Ratio: 2-Ch Ratio mode displays in the Function Reading meter display the amplitude ratio between the selected channel and the alternate channel. Both Level meters continue to display the absolute level on each channel. 2-Ch Ratio function is useful while adjusting stereo channel amplitudes to match or for measuring gain or loss when the analyzer inputs are connected at the input and output of a device.
- 2* Crosstalk: Crosstalk mode is identical to the 2-Ch Ratio mode except that the tunable bandpass filter is also engaged in the main (Function Reading) meter before the measurement. Crosstalk mode will thus provide more accurate measurements of low-amplitude signals in the presence of noise, since most wide-band noise will be rejected by the filter. The filter must be tuned to the frequency of the signal on the driven channel.

- 3 THD+N Ratio: The THD+N mode uses a DSP-implemented bandreject (notch) filter to remove the fundamental sinewave signal so that the detector may measure the remaining harmonic distortion products and noise. The THD+N Ratio mode expresses the distortion product and noise amplitudes relative to the amplitude of the unfiltered signal measured by the Level meter. Units of % and dB (below fundamental) are commonly used in THD+N Ratio function. THD+N Ratio is used much more commonly than THD+N Amplitude, but in an amplitude sweep THD+N Ratio appears to show increasing distortion and noise with decreasing signal amplitude because the distortion and noise is stated as a ratio to the decreasing signal. THD+N Amplitude may be more useful for amplitude sweeps.
- The bandreject filter center frequency may be fixed or may track one of several other parameters.
- 4 THD+N Ampl: The THD+N Amplitude mode uses a DSP-implemented bandreject (notch) filter to remove the fundamental sinewave signal so that the detector may measure the remaining harmonic distortion products and noise. The THD+N Ampl (amplitude) mode expresses amplitude of the remaining distortion products and noise in absolute units (FFS, %FS, dBFS, bits with digital signals; Volts, dBV, dBu, etc. with analog signals), independent of the amplitude of the fundamental signal. THD+N Ampl mode is particularly useful when performing amplitude sweeps of audio devices, since it helps make clear that the noise component is (typically) a constant amplitude unrelated to the signal amplitude. THD+N Ratio in an amplitude sweep obscures this fact, since the measured distortion and noise appears to increase with decreasing signal amplitude because it is being stated as a ratio to the decreasing signal. The bandreject filter center frequency may be fixed or may track one of several other parameters.
- 5 Bandpass: Bandpass mode is a selective voltmeter (wave analyzer) implemented by DSP techniques. It includes a narrow bandpass filter of about 1/13 octave ($Q=19$, 3 dB bandwidth about 5.2% of center frequency). The bandpass filter center frequency may be fixed or may track one of

several other parameters. The filter may be tuned to the steering source fundamental frequency or to the 2nd, 3rd, 4th, or 5th harmonic of the tuning source. This harmonic tracking ability permits swept measurements of individual harmonic distortion, limited to a maximum value of 42% of the sample rate:

6 SMPTE/DIN: The SMPTE/DIN mode of DSP Audio Analyzer is designed to function with two-tone intermodulation distortion test signals complying with the SMPTE or DIN standard, or similar to these. These signals normally consist of a low-frequency tone between 40 Hz and 500 Hz combined with a high-frequency tone above 2 kHz. Typical SMPTE tone combinations are 60 Hz and 7 kHz, while 250 Hz and 8 kHz are often used with the DIN standard. The amplitude ratio of the LF tone to the HF tone is commonly 4:1 but the tones are sometimes at equal amplitudes. The analyzer function measures the amplitude of the sidebands near the high-frequency tone and expresses their amplitude as a ratio to the HF tone amplitude.

7 Phase: The Phase mode of DSP Audio Analyzer measures the phase of the B channel signal relative to the A channel signal (B-A), expressing the result in degrees. This corresponds to the phase measurement technique of the hardware analog analyzer, and to the Sine Variable Phase waveform of both the analog and digital generators. The Channel B display field is gray when Phase function is selected.

Description This command selects the analysis mode of the Digital Domain Audio Analyzer Function meters.

See Also `AP.S2CDsp.Analyzer.RdgRate`,
`AP.S2CDsp.Analyzer.FuncBPBRTuning`

Example See example for `AP.S2CDsp.Analyzer.FuncDetector`.

AP.S2CDsp.Analyzer.FuncPhaseMode **Property**

Syntax `AP.S2CDsp.Analyzer.FuncPhaseMode`

Data Type	Integer								
	<table> <tr> <td>0</td> <td>Auto</td> </tr> <tr> <td>1</td> <td>-180 +180 deg</td> </tr> <tr> <td>2</td> <td>0 +360 deg</td> </tr> <tr> <td>3</td> <td>-90 +270 deg</td> </tr> </table>	0	Auto	1	-180 +180 deg	2	0 +360 deg	3	-90 +270 deg
0	Auto								
1	-180 +180 deg								
2	0 +360 deg								
3	-90 +270 deg								
Description	This function sets the Digital Domain Audio Analyzer Function meter Phase measurement range.								

AP.S2CDsp.Analyzer.InputFormat

Property

Syntax	<code>AP.S2CDsp.Analyzer.InputFormat</code>												
Data Type	Integer												
	<table> <tr> <td>0</td> <td>Digital @ ISR:</td> </tr> <tr> <td>1</td> <td>HiRes A/D @65536</td> </tr> <tr> <td>2</td> <td>HiBW A/D @131072</td> </tr> <tr> <td>3</td> <td>HiBW A/D @262144</td> </tr> <tr> <td>4</td> <td>HiRes A/D @OSR</td> </tr> <tr> <td>5</td> <td>HiBW A/D @2xOSR</td> </tr> </table>	0	Digital @ ISR:	1	HiRes A/D @65536	2	HiBW A/D @131072	3	HiBW A/D @262144	4	HiRes A/D @OSR	5	HiBW A/D @2xOSR
0	Digital @ ISR:												
1	HiRes A/D @65536												
2	HiBW A/D @131072												
3	HiBW A/D @262144												
4	HiRes A/D @OSR												
5	HiBW A/D @2xOSR												
Description	This command sets the Digital Domain Audio Analyzer Input Format.												
Example	See example for <code>AP.S2CDsp.Analyzer.FuncDetector</code> .												

AP.S2CDsp.Analyzer.RdgRate

Property

Syntax	<code>AP.S2CDsp.Analyzer.RdgRate</code>				
Data Type	Integer				
	<table> <tr> <td>0</td> <td>Auto: this selection manages selection of the reading rate as a function of the frequency being measured and the instrument function so as to provide rapid testing speeds along with sufficient integration for accuracy at the present test frequency.</td> </tr> <tr> <td>1</td> <td>4/Sec</td> </tr> </table>	0	Auto: this selection manages selection of the reading rate as a function of the frequency being measured and the instrument function so as to provide rapid testing speeds along with sufficient integration for accuracy at the present test frequency.	1	4/Sec
0	Auto: this selection manages selection of the reading rate as a function of the frequency being measured and the instrument function so as to provide rapid testing speeds along with sufficient integration for accuracy at the present test frequency.				
1	4/Sec				

2	8/Sec
3	16/Se
4	32/Sec
5	64/Sec
6	128/Sec

Description This command sets the Digital Domain Audio Analyzer Reading update Rate (integration period) for all of the meters in this DSP program.

See Also `AP.S2CDsp.Analyzer.FuncDetector`,
`AP.S2CDsp.Analyzer.FuncMode`

Example See example for `AP.S2CDsp.Analyzer.FuncDetector`.

User Notes

Digital Data Analyzer

AP.S2CDsp.BitTest.ChADataRdg

Property

Syntax `AP.S2CDsp.BitTest.ChADataRdg(ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: dec.

Description

This command returns a unsettled reading for the Digital Data Analyzer channel A Data meter and zeros the ready count.

See Also

AP.S2CDsp.BitTest.ChADataReady,
AP.S2CDsp.BitTest.ChADataTrig

Example

```
Sub Main
    AP.Application.NewTest
    AP.S2CDsp.Program = 6
    AP.DGen.Wfm 4, 5
    AP.DGen.ChAAmpl("dBFS") = -3
    AP.DGen.OutDitherType = 3
    AP.DGen.Output = True
    AP.S2CDio.InFormat = 3
    AP.S2CDsp.BitTest.ChADataTrig 'Trigger a new reading
    Do
        Ready = AP.S2CDsp.BitTest.ChADataReady
    Loop Until Ready > 0           'Wait for new reading
    Reading1 = AP.S2CDsp.BitTest.ChADataRdg ("dec")
                                   'Get new reading

    NewLine$ = Chr(13)
    a$= "Ch A Data "+Left(Str$(Reading1),8)+" dec"
    AP.Prompt.Text = a$ + NewLine$
    AP.Prompt.ShowWithContinue
    Beep
    Stop
End Sub
```

AP.S2CDsp.Bittest.ChADataReady**Property****Syntax** `AP.S2CDsp.Bittest.ChADataReady`**Data Type** Integer`0` Reading not ready.`>0` Reading ready.**Description** This command returns the Digital Data Analyzer channel A Data meter unsettled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.Bittest.ChADataRdg` or `AP.S2CDsp.Bittest.ChADataTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S1DSP.Bittest.ChADataRdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.Bittest.ChADataRd`,
`AP.S2CDsp.Bittest.ChADataTrig`**Example Output** See example for `AP.S2CDsp.Bittest.ChADataRdg`.**AP.S2CDsp.Bittest.ChADataTrig****Method****Syntax** `AP.S2CDsp.Bittest.ChADataTrig`**Description** Causes a restart of the reading cycle and zeros the ready count for the `AP.S2CDsp.Bittest.ChADataRdg` command. The reading in progress is aborted.**See Also** `AP.S2CDsp.Bittest.ChADataRdg`,
`AP.S2CDsp.Bittest.ChADataReady`**Example Output** See example for `AP.S2CDsp.Bittest.ChADataRdg`.

AP.S2CDsp.BitTest.ChAErrRdg**Property**

Syntax `AP.S2CDsp.BitTest.ChAErrRdg(ByVal Unit As String)`

Data Type Double

Part	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: dec.

Description This command returns a unsettled reading for the Digital Data Analyzer channel A Errors meter and zeros the ready count.

See Also `AP.S2CDsp.BitTest.ChAErrReady`,
`AP.S2CDsp.BitTest.ChAErrTrig`

Example

```

Sub Main
    AP.Application.NewTest
    AP.S2CDsp.Program = 6
    AP.DGen.Wfm 4, 5
    AP.DGen.ChAAmpl("dBFS") = -3
    AP.DGen.OutDitherType = 0
    AP.DGen.Output = True
    AP.S2CDio.InFormat = 3
    With AP.S2CDsp.BitTest
        .DisplayError = 0           'Error display normal
        .RdgRate = 3                'Reading rate to 16/second
        .Wfm = 4 'Set waveform analysis pattern to constant
        .FreezeOnError = False     'Don't freeze data on error
        .ChAErrTrig                'Trigger a new reading
    End With
    Do
        Ready = .ChAErrReady
    Loop Until Ready > 0          'Wait for new reading
    Reading1 = .ChAErrRdg("dec") 'Get new reading
    NewLine$ = Chr(13)
    a$= "Ch A Errors "+Left(Str$(Reading1),8)+" dec"
    AP.Prompt.Text = a$ + NewLine$
    AP.Prompt.ShowWithContinue
    Beep
    Stop
End Sub

```


AP.S2CDsp.Bittest.ChAErrReady**Property****Syntax** `AP.S2CDsp.Bittest.ChAErrReady`**Data Type** Integer`0` Reading not ready.`>0` Reading ready.**Description** This command returns the Digital Data Analyzer channel A Errors meter unsettled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.Bittest.ChAErrRdg` or `AP.S2CDsp.Bittest.ChAErrTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S1DSP.Bittest.ChAErrRdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.Bittest.ChAErrRdg`,
`AP.S2CDsp.Bittest.ChAErrTrig`**Example** See example for `AP.S2CDsp.Bittest.ChAErrRdg`.**AP.S2CDsp.Bittest.ChAErrTrig****Method****Syntax** `AP.S2CDsp.Bittest.ChAErrTrig`**Description** Causes a restart of the reading cycle and zeros the ready count for the `AP.S1DSP.Bittest.ChAErrRdg` command. The reading in progress is aborted.**See Also** `AP.S2CDsp.Bittest.ChAErrRdg`,
`AP.S2CDsp.Bittest.ChAErrReady`**Example** See example for `AP.S2CDsp.Bittest.ChAErrRdg`.

AP.S2CDsp.BitTest.ChBDataRdg**Property**

Syntax `AP.S2CDsp.BitTest.ChBDataRdg` (ByVal *Unit* As String)

Data Type Double

Parameters

Part	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: dec.

Description

This command returns a unsettled reading for the Digital Data Analyzer channel B Data meter and zeros the ready count.

See Also

AP.S2CDsp.BitTest.ChBDataReady,
AP.S2CDsp.BitTest.ChBDataTrig

Example

```
Sub Main
    AP.Application.NewTest
    AP.S2CDsp.Program = 6
    AP.DGen.Wfm 4, 5
    AP.DGen.ChAAmpl("dBFS") = -3
    AP.DGen.OutDitherType = 3
    AP.DGen.Output = True
    AP.S2CDio.InFormat = 3
    AP.S2CDsp.BitTest.ChBDataTrig 'Trigger a new reading
    Do
        Ready = AP.S2CDsp.BitTest.ChBDataReady
    Loop Until Ready > 0           'Wait for new reading
    Reading1 = AP.S2CDsp.BitTest.ChBDataRdg("dec")
                                     'Get new reading

    NewLine$ = Chr(13)
    a$ = "Ch B Data "+Left(Str$(Reading1),8)+" dec"
    AP.Prompt.Text = a$ + NewLine$
    AP.Prompt.ShowWithContinue
    Beep
    Stop
End Sub
```

AP.S2CDsp.Bittest.ChBDataReady**Property**

Syntax	<code>AP.S2CDsp.Bittest.ChBDataReady</code>
Data Type	Integer
	0 Reading not ready.
	>0 Reading ready.
Description	<p>This command returns the Digital Data Analyzer channel B Data meter unsettled reading ready count.</p> <p>Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the <code>AP.S2CDsp.Bittest.ChBDataRdg</code> or <code>AP.S2CDsp.Bittest.ChBDataTrig</code> commands will zero the ready count.</p> <p>If the reading is found to be ready, a call to the <code>AP.S2CDsp.Bittest.ChBDataRdg</code> command will be guaranteed to return quickly.</p>
See Also	<code>AP.S2CDsp.Bittest.ChBDataRdg</code> , <code>AP.S2CDsp.Bittest.ChBDataTrig</code>
Example	See example for <code>AP.S2CDsp.Bittest.ChBDataRdg</code> .

AP.S2CDsp.Bittest.ChBDataTrig**Method**

Syntax	<code>AP.S2CDsp.Bittest.ChBDataTrig</code>
Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP.S2CDsp.Bittest.ChBDataRdg</code> comand. The reading in progress is aborted.
See Also	<code>AP.S2CDsp.Bittest.ChBDataRdg</code> , <code>AP.S2CDsp.Bittest.ChBDataReady</code>
Example	See example for <code>AP.S2CDsp.Bittest.ChBDataRdg</code> .

AP.S2CDsp.BitTest.ChBErrRdg**Property**

Syntax `AP.S2CDsp.BitTest.ChBErrRdg(ByVal Unit As String)`

Data Type Double

Parameters	Part	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: dec.

Description This command returns a unsettled reading for the Digital Data Analyzer channel B Errors meter and zeros the ready count.

See Also `AP.S2CDsp.BitTest.ChBErrReady`,
`AP.S2CDsp.BitTest.ChBErrTrig`

Example

```

Sub Main
    AP.Application.NewTest
    AP.S2CDsp.Program = 6
    AP.DGen.Wfm 4, 5
    AP.DGen.ChAAmpl("dBFS") = -3
    AP.DGen.OutDitherType = 0
    AP.DGen.Output = True
    AP.S2CDio.InFormat = 3
    AP.S2CDsp.BitTest.ChBErrTrig 'Trigger a new reading
    Do
        Ready = AP.S2CDsp.BitTest.ChBErrReady
    Loop Until Ready > 0 'Wait for new reading
    Reading1 = AP.S2CDsp.BitTest.ChBErrRdg("dec")
        'Get new reading

    NewLine$ = Chr(13)
    a$= "Ch B Errors "+Left(Str$(Reading1),8)+" dec"
    AP.Prompt.Text = a$ + NewLine$
    AP.Prompt.ShowWithContinue
    Beep
    Stop
End Sub

```

AP.S2CDsp.Bittest.ChBErrReady**Property****Syntax** `AP.S2CDsp.Bittest.ChBErrReady`**Data Type** Integer`0` Reading not ready.`>0` Reading ready.**Description** This command returns the Digital Data Analyzer channel B Errors meter unsettled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.Bittest.ChBErrRdg` or `AP.S2CDsp.Bittest.ChBErrTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.Bittest.ChBErrRdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.Bittest.ChBErrRdg`,
`AP.S2CDsp.Bittest.ChBErrTrig`**Example** See example for `AP.S2CDsp.Bittest.ChBErrRdg`.**AP.S2CDsp.Bittest.ChBErrTrig****Method****Syntax** `AP.S2CDsp.Bittest.ChBErrTTrig`**Description** Causes a restart of the reading cycle and zeros the ready count for the `AP.S2CDsp.Bittest.ChBErrRdg` command. The reading in progress is aborted.**See Also** `AP.S2CDsp.Bittest.ChBErrTRdg`,
`AP.S2CDsp.Bittest.ChBErrTReady`**Example** See example for `AP.S2CDsp.Bittest.ChBErrRdg`.

AP.S2CDsp.Bittest.DisplayError**Property****Syntax** `AP.S2CDsp.Bittest.DisplayError`**Data Type** Integer

0	Normal
1	Maximum.
2	Totalize.

Description This command sets the mode for the Digital Data Analyzer channel A and B Error displays.

Received data is also measured to determine if it matches the data transmitted. Only the number of bits selected in the Resolution field `AP.S2CDio.Resolution` of the Digital I/O panel will be analyzed. This comparison is done with algorithms which are insensitive to delay between the send and receive sections. The number of errors in the received data per measurement interval are counted for each channel. The `AP.S2CDsp.Bittest.DisplayError` command selects the type of analysis to be performed. In the Normal mode, the number of errors detected during the last measurement interval are displayed directly in the Ch 1 and Ch 2 Errors fields of the panel. If Error Display is selected as Maximum, the maximum error count during any measurement interval will be held in the display. A running total of all errors may be accumulated by using the Totalize mode of the Error Display field.

See Also `AP.S2CDsp.Bittest.RdgRate`**Example** See example for `AP.S2CDsp.Bittest.ChAErrRdg`.

AP.S2CDsp.Bittest.FreezeOnError**Property****Syntax** `AP.S2CDsp.Bittest.FreezeOnError`**Data Type** Boolean*True* Hold first error reading..*False* Continue updating data readings.**Description** This command sets or clears the Freeze Data on Error field on the Digital Data Analyzer.

If the `AP.S2CDsp.Bittest.FreezeOnError` command is set to (True), the Data fields will continue to display the value which was received when the first error occurred. If

`AP.S2CDsp.Bittest.FreezeOnError` is set to (False), the Data fields will continue updating, independent of any errors detected.

See Also `AP.S2CDsp.Bittest.RdgRate`**Example** See example for `AP.S2CDsp.Bittest.ChAErrRdg`.**AP.S2CDsp.Bittest.RdgRate****Property****Syntax** `AP.S2CDsp.Bittest.RdgRate`**Data Type** Integer*0* Auto reading rate. The reading rate is automatically selected based on the measured frequency.*1* 4/ second update rate.*2* 8/ second update rate.*3* 16/ second update rate.**Description** This command sets the rate a which the Data (and Errors) readings are updated.**Example** See example for `AP.S2CDsp.Bittest.ChAErrRdg`.

AP.S2CDsp.BitTest.Wfm**Property**

Syntax `AP.S2CDsp.BitTest.Wfm`

Data Type Integer

0	Constant
1	Random.
2	Walking-1.
3	Walking-0.
4	Sine.

Description This command selects the Digital Data Analyzer Waveform pattern to analyze.

Example See example for AP.S2CDsp.BitTest.ChAErrRdg.

User Notes

Multitone Audio Analyzer

AP.S2CDsp.FastTest.Ch1Rdg

Property

Syntax `AP.S2CDsp.FastTest.Ch1Rdg(ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, Bits.

Description

This command returns a unsettled reading for the Multitone Audio Analyzer channel 1 Peak Monitor meter and zeros the ready count.

See Also

`AP.S2CDsp.FastTest.Ch1Ready`,
`AP.S2CDsp.FastTest.Ch1Trig`

Example

```

Sub Main
  AP.File.OpenTest "FASTTSTC.AT2C" 'Opens test
  Wait 1
  With AP.S2CDsp.FastTest
    .Ch1Trig                'Trigger a new reading
  Do
    Ready1 = .Ch1Ready
    Loop Until Ready1 > 0  'Wait for reading
    Reading1 = .Ch1Rdg("dBFS") 'Get reading
  End With
  NewLine$ = Chr(13)
  a$= "Ch1 Peak Mon " & Left(Str$(Reading1),6) & "dBFS"
  AP.Prompt.Text = a$ & NewLine$ & b$ + NewLine
  AP.Prompt.ShowWithContinue
  Beep
  Stop
End Sub

```

AP.S2CDsp.FastTest.Ch1Ready**Property****Syntax** `AP.S2CDsp.FastTest.Ch1Ready`**Data Type** Integer`0` Reading not ready.`>0` Reading ready.**Description** This command returns the Multitone Audio Analyzer channel 1 Peak Monitor meter unsettled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.FastTest.Ch1Rdg` or `AP.S2CDsp.FastTest.Ch1Trig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.FastTest.Ch1Rdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.FastTest.Ch1Rd`,
`AP.S2CDsp.FastTest.Ch1Trig`**Example** See example for `AP.S2CDsp.FastTest.Ch1Rdg`.**AP.S2CDsp.FastTest.Ch1Source****Property****Syntax** `AP.S2CDsp.FastTest.Ch1Source`**Data Type** Integer

The following list contains the selections relevant to the `AP.S2CDsp.FastTest.InputFormat` command Digital input selection.

`0` A`1` B`2` None

The following list contains the selections relevant to the `AP.S2CDsp.FastTest.InputFormat` command A/D input selections.

0	Anlr-A
1	Anlr-B
2	Anlr Reading Ampl
3	Anlr Reading Ratio
4	Ch. A Generator
5	Ch. B Generator
6	Jitter Signal (UI)
7	None
8	Jitter Signal (sec)

Description This command sets the Multitone Audio Analyzer Channel 1 Input.

Example See example for `AP.S2CDsp.FastTest.FFTLength`.

AP.S2CDsp.FastTest.Ch1Trig

Method

Syntax `AP.S2CDsp.FastTest.Ch1Trig`

Description Causes a restart of the reading cycle and zeros the ready count for the `AP.S2CDSP.FastTest.Ch1Rdg` command. The reading in progress is aborted.

See Also `AP.S2CDsp.FastTest.Ch1Rdg`,
`AP.S2CDsp.FastTest.Ch1Ready`

Example See example for `AP.S2CDsp.FastTest.Ch1Rdg`.

AP.S2CDsp.FastTest.Ch2Rdg

Property

Syntax `AP.S2CDsp.FastTest.Ch2Rdg(ByVal Unit As String)`

Data Type Double

Parameters	Part	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, Bits.
Description		This command returns a unsettled reading for the Multitone Audio Analyzer channel 2 Peak Monitor meter and zeros the ready count.
See Also		AP.S2CDsp.FastTest.Ch2Ready, AP.S2CDsp.FastTest.Ch2Trig
Example		<pre> Sub Main AP.File.OpenTest "FASTTSTC.AT2C" 'Open test Wait 1 With AP.S2CDsp.FastTest .Ch2Trig 'Trigger a new reading Do Ready2 = .Ch2Ready Loop Until Ready2 > 0 'Wait for new reading Reading2 = .Ch2Rdg("dBFS") 'Get new reading End With NewLine\$ = Chr(13) a\$= "Ch2 Peak Mon " & Left(Str\$(Reading2),6) & "dBFS" AP.Prompt.Text = a\$ & NewLine\$ & b\$ + NewLine AP.Prompt.ShowWithContinue Beep Stop End Sub </pre>

AP.S2CDsp.FastTest.Ch2Ready

Property

Syntax AP.S2CDsp.FastTest.Ch2Ready

Data Type Integer

0 Reading not ready.
>0 Reading ready.

Description This command returns the Multitone Audio Analyzer channel 2 Peak Monitor meter unsettled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT

zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.FastTest.Ch2Rdg` or `AP.S2CDsp.FastTest.Ch2Trig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.FastTest.Ch2Rdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.FastTest.Ch2Rdg`,
`AP.S2CDsp.FastTest.Ch2Trig`

Example See example for `AP.S2CDsp.FastTest.Ch2Rdg`.

AP.S2CDsp.FastTest.Ch2Source

Property

Syntax `AP.S2CDsp.FastTest.Ch2Source`

Data Type Integer

The following list contains the selections relevant to the `AP.S2CDsp.FastTest.InputFormat` command Digital input selection.

0	A
1	B
2	None

The following list contains the selections relevant to the `AP.S2CDsp.FastTest.InputFormat` command A/D input selections.

0	Anlr-A
1	Anlr-B
2	Anlr Reading Ampl
3	Anlr Reading Ratio
4	Ch. A Generator
5	Ch. B Generator
6	Jitter Signal (UI)

7	None
8	Jitter Signal (sec)

Description This command sets the Multitone Audio Analyzer Channel 2 Input.

Example See example for `AP.S2CDsp.FastTest.FFTLength`.

AP.S2CDsp.FastTest.Ch2Trig

Method

Syntax `AP.S2CDsp.FastTest.Ch2Trig`

Description Causes a restart of the reading cycle and zeros the ready count for the `AP.S2CDsp.FastTest.Ch2Rdg` comand. The reading in progress is aborted.

See Also `AP.S2CDsp.FastTest.Ch2Rdg`,
`AP.S2CDsp.FastTest.Ch2Ready`

Example See example for `AP.S2CDsp.FastTest.Ch2Rdg`.

AP.S2CDsp.FastTest.FFTLength

Property

Syntax `AP.S2CDsp.FastTest.FFTLength`

Data Type Integer

0	Auto: the Auto selection will automatically set the acquisition buffer and transform length to be exactly twice the length of any generator waveform loaded into the Digital Generator buffer. This condition is necessary for the Noise function of FASTTEST to work.
1	512
2	1024
3	2048
4	4096
5	8192
6	16384
7	32768

Description

This command sets the Multitone Audio Analyzer FFT Length.

The FFT Length field value of the FASTTEST program controls the record length used as input to the FFT process when either F9/Go or AP.Sweep.Start is initiated to acquire and transform, or the F6 or Ap.Sweep.Retransform function key or Sweep Transform Data without Acquire menu command is used to re-transform any portion of a record previously acquired. Longer transform lengths produce greater frequency resolution in the resulting FFT, but require longer times to acquire and transform the signal.

Example

```

Sub Main
  AP.File.OpenTest "FasttstB.AT2C" 'Open test
  With AP.S2CDsp.FastTest
    .InputFormat = 1 'Set input to HiRes A/D @OSR
    .Ch1Source = 0 'Set Source to Anlr-A
    .Ch2Source = 1 'Set Source to Anlr-B
    .Mode = 0 'Set Measurement to Spectrum
    .FreqRes("%") = 1 'Set Freq Res to 1%
    .FFTLength = 6 'Set FFT length to 16384
    .Processing = 0 'Set Processing to Synchronous
    .TrigSource = 0 'Set Triggering to DGEN
    .TrigDelay("sec") = 0 'Set Trig Delay to 0
    .PhaseDisplay = 0 'Set Ch 2 Phase Display to _
      Independent

  AP.Sweep.Start
  'Attach sweep file
  AP.Sweep.Source1.Table("FASTTST.ADS", 0)
  .Mode = 1 'Set Measurement to Response
  AP.Sweep.Reprocess
  .Mode = 2 'Set Measurement to Distortion
  AP.Sweep.Reprocess
  .Mode = 3 'Set Measurement to Noise
  AP.Sweep.Reprocess
  .Mode = 4 'Set Measurement to Masking Curve
  AP.Sweep.Reprocess
  End With
End Sub

```


AP.S2CDsp.FastTest.FreqRes Property

Syntax `AP.S2CDsp.FastTest.FreqRes (ByVal Unit As String)`

Data Type Double Valid amplitude settings are from +/- 0.0 to 13.0 %.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command: %

Description This command sets the Multitone Audio Analyzer Frequency Resolution.

The Frequency Resolution field is a numeric entry field with % units. The user may enter values up to 13% which are used in Response and Distortion Measurement functions.

In Response function, the amplitudes of all FFT bins within plus and minus the Frequency Resolution value of each sweep table value are combined in RSS (root-sum-square) fashion and furnished to the computer as the integrated amplitude of the bins within that range. The purpose of this function is to provide accurate frequency response measurements of devices with wow and flutter. Wow and flutter spreads the energy from a single tone across a narrow spectral band.

In Distortion function, the amplitudes of all FFT bins within plus and minus the Frequency Resolution value of each sweep table value are excluded from the RSS computation of energy falling between tones. Distortion function defines all signals other than the fundamental tones as distortion and noise. Entering a non-zero value of Frequency Resolution causes flutter sidebands to not be included in the distortion measurement.

Example See example for `AP.S2CDsp.FastTest.FFTLength`.

AP.S2CDsp.FastTest.InputFormat Property

Syntax `AP.S2CDsp.FastTest.InputFormat`

Data Type Integer

0 Digital @ ISR:

- 1 HiRes A/D @65536
- 2 HiBW A/D @131072
- 3 HiBW A/D @262144
- 4 HiRes A/D @OSR
- 5 HiBW A/D @2xOSR

Description This command sets the Multitone Audio Analyzer Input Format.

Example See example for `AP.S2CDsp.FastTest.FFTLength`.

AP.S2CDsp.FastTest.Mode

Property

Syntax `AP.S2CDsp.FastTest.Mode`

Data Type Integer

- 0 Spectrum: Provides a normal FFT spectrum display with no processing except for peak picking. The Spectrum selection is typically used without a sweep table (.ADS file), and with a relative large number of Steps at Source 1 of the Sweep panel to provide good frequency resolution. Typical Steps values are from 250 to 500. If the transform length results in more FFT bins between the Start-Stop frequency span than are being plotted, peak-picking takes place. With peak-picking, the DSP searches all FFT bins between the previous plotted point and the point presently being plotted and sends the highest bin amplitude in that range as the amplitude of the new point to be sure that no signals are missed.
- 1 Response: is always used with a sweep table (.ADS file) listing the exact frequencies of the sinewaves in the multitone signal to be used for frequency response measurements. The DSP returns to the computer for plotting only the amplitudes of the FFT bins containing those exact frequencies, resulting in a frequency response graph.

If the value in the Frequency Resolution field is greater than zero, the DSP performs an RSS (root-sum-square) integration of all the bin amplitudes within plus or minus the Frequency

2

Resolution value around each sweep table frequency and sends the integrated sum value to the computer to be plotted. This mode is intended for frequency response measurements on devices such as analog tape recorders which introduce frequency modulation (flutter) to signals. Flutter spreads each tone's energy across a small region of the spectrum. This reduces the amplitude of the fundamental tone, since the total energy in the fundamental and all sidebands remains constant during frequency modulation. The RSS summation combines this spread energy back into a single value, much as the human hearing system responds to signals with small amounts of FM.

Distortion: excludes the amplitudes of the FFT bins known (from the generator waveform) to contain fundamental signals. All other bin amplitudes are summed (RSS) between each adjacent pair of frequencies requested from the DSP by the computer. It is not necessary to use a sweep table (.ADS file) listing the fundamental frequencies of the sinewaves in the multitone signal being used. Distortion and noise can thus be summed across spans determined by the Sweep panel Start, Stop, Log/Lin, and number of Steps, or the spans can be determined by a sweep table. If it is desired to sum the noise and distortion into critical bands, a sweep table can be used which defines the edges of the human hearing system critical bands. The resulting distortion and noise curve is normally compared to the composite masking curve generated in Masking function.

If the value in the Frequency Resolution field is greater than zero, the DSP also excludes all the bin amplitudes within plus or minus the Frequency Resolution value around each sweep table frequency before sending the integrated sum value to the computer to be plotted. This mode is intended for distortion measurements on devices such as analog tape recorders which introduce frequency modulation (flutter) to signals. Flutter spreads each tone's energy across a small region of the spectrum. If these close-in sidebands which fall outside the bin containing the fundamental are not to be measured as distortion, they must be excluded, much as the

- 3 human hearing system masks low amplitude signals nearby in frequency to a stronger signal.
- Noise: this selection may be used with a sweep table (.ADS file) listing the fundamental frequencies of the multitone signal in use, but need not be. Noise mode depends on the FASTTEST Transform length being set to the value twice the length of the waveform file which generates the multitone signal. The analyzer frequency resolution is thus twice the resolution of the generated signal. The result is that every alternate analyzer FFT bin falls between bins at which the generated signal could contain fundamentals or bins into which harmonic or intermodulation distortion products due to the generated signal fundamental signals could fall (assuming that the device under test does not shift fundamental frequencies or produce frequency modulation). The amplitudes of these alternate empty bins consists of noise generated in the device under test, largely unaffected by fundamental signals or distortion. If the same sweep table is used in Noise mode that is used for response and distortion measurements, the resulting graph will be a spectrum analysis of noise in the presence of test signal. If a two-point sweep is made with Start at 20 Hz and Stop at 20 kHz, for example, the plotted value at 20 kHz represents the RSS integration of all empty bins across the audio band.
- 4 Masking: this selection generates a composite masking curve for the particular multitone signal in use. The shape of the curves is based on a model published by psychacoustician Brian Moore in the Proceedings of the AES 12th International Conference, June 1993, pp 22-23. The shape of the curves varies with frequency. The center frequency of each section of the composite masking curve is located at the fundamental frequencies present in the waveform file downloaded to the generator buffer. The reference amplitude at each frequency is determined by the measured amplitude at each fundamental frequency. The masking curve is normally used by saving it as a limit (.ADL) file, then comparing a Distortion function curve (usually with critical band spacing) to that limit curve.

5

Crosstalk: depends upon the multitone test signal having one or more unique tone frequencies on each stereo channel, in addition to any number of tones which are common to both channels. Crosstalk function determines which generator frequencies are unique to a channel and measures the amplitude of the corresponding FFT frequency bin on the opposite channel. Unique frequencies are typically created in multitone signals at frequencies above 500 Hz, where the generator resolution is less limiting and where a bin occupied for crosstalk measurement purposes represents a small portion of the total bins for measurement of total integrated noise and distortion across that portion of the spectrum. In order to measure crosstalk in both directions (from A to B and from B to A), it is common to insert unique tones at pairs of nearby frequencies on each channel. For example, if monaural signals (tones on both channels) exist at about 500 Hz and 640 Hz, a crosstalk-measurement tone might be inserted at 560 Hz on Channel A and at 575 Hz on Channel B. Crosstalk is commonly used with a sweep table corresponding to the approximate frequencies where the pairs of crosstalk frequencies have been inserted. At each frequency in the sweep table, the DSP will report the amplitude of the crosstalk-containing bin nearest the requested frequency. The FASTTEST Channel 1 curve will show measurements of crosstalk into that frequency from Channel 2, and vice-versa. If the stereo channels have been mistakenly reversed, the crosstalk measurements will show the levels of the tones in the channel on which they were transmitted. This makes it easy to automatically determine cases of swapped channels by setting an upper limit file for each channel.

Description

This command sets the Multitone Audio Analyzer measurement mode. The `AP.S2CDsp.FastTest.Mode` command controls the type of post-processing done to FFT results before they are sent to the computer for display and possible limits comparison.

Example

See example for `AP.S2CDsp.FastTest.FFTLength`.

AP.S2CDsp.FastTest.PhaseDisplay

Property

Syntax `AP.S2CDsp.FastTest.PhaseDisplay`

Data Type Integer

0 Independent

1 Interchannel

Description This command sets the Multitone Audio Analyzer Phase Display mode selection.

The FFT of FASTTEST computes both magnitude and phase arrays as a function of frequency. The phase of coherent signals, such as multitone signals, may be plotted for either or both channels by selecting FASTTEST as the instrument and Ch 1 Phase or Ch 2 Phase as the parameter in the Data browser of the Sweep panel. A sweep table (.ADS file) listing the fundamental signals would be used in this mode. When the channel 2 Phase Display is selected as Independent, the Ch 1 and Ch 2 Phase parameters each show the absolute phase of the fundamental tones.

It is also possible to plot the interchannel phase difference of stereo signals with FASTTEST. Selecting Interchannel causes the DSP to compute the phase difference between the Ch 1 and Ch 2 Phase signals at each sweep table value and report that computed value to the computer as the Ch 2 Phase parameter. The Ch 1 Phase parameter is unaffected by the Interchannel setting and plots absolute phase of the channel 1 signal.

Example See example for `AP.S2CDsp.FastTest.FFTLength`.

AP.S2CDsp.FastTest.Processing

Property

Syntax `AP.S2CDsp.FastTest.Processing`

Data Type Integer

0 Synchronous: Normal operation of FASTTEST involves acquisition of a multitone signal which was generated from a multitone waveform file by System Two's Digital Generator.

The multitone waveform files furnished with APWIN and System Two are created so as to be synchronous with one or another of the analyzer acquisition buffer lengths available in FASTTEST. Every sinewave in the generated signal goes through an exact integer number of cycles in the generator buffer and in the analyzer transform buffer. No windowing function is required and maximum theoretical FFT selectivity is achieved with full dynamic range available in bins adjacent to a bin containing a full-scale signal.

1

Freq Corrected: A key feature of FASTTEST is its ability to compare the tone frequencies in an acquired multitone waveform with the digital reference copy of the transmitted or pre-recorded waveform presently in the Digital Generator buffers. If this comparison shows that the tone frequencies have been shifted up or down due to the signal originating from a device with a different clock frequency from the analyzer or due to analog tape player speed errors, FASTTEST corrects all the tone frequencies to the reference signal values. This re-creates the original synchronous relationship so that no window function is required before the FFT, and maximum theoretical FFT selectivity is obtained. The maximum frequency difference which can be corrected is +/-3%. FASTTEST is normally operated with Frequency Error Correction enabled when analyzing signals generated by another Audio Precision instrument or previously recorded and now being reproduced. This mode of operation is selected by the Freq Corrected selection in the Processing field.

2

Windowed: If for some reason it is desired to measure remotely-generated or pre-recorded signals without use of the Frequency Error Correction feature, it will normally be necessary to use the Hann window function to obtain useful results. The Windowed selection of the Processing field enables the Hann window.

Description This command sets the Multitone Audio Analyzer processing.

Example See example for `AP.S2CDsp.FastTest.FFTLength`.

AP.S2CDsp.FastTest.TrigDelay

Property

Syntax `AP.S2CDsp.FastTest.TrigDelay(ByVal Unit As String)`

Data Type Double Values up to 1.365 seconds may be entered.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command: sec.

Description This command sets the Multitone Audio Analyzer trigger delay.

When testing audio transmission paths which include audio processors (compressors, limiters, etc.), it may be desirable to make measurements after the processors have stabilized following any change of level resulting between the multitone burst and the preceding program material. The Trigger Delay filed controls the interval between initial recognition of the incoming multitone signal and capture of the portion of signal which will finally be analyzed for response, distortion, noise, etc. Use of any non-zero Trigger Delay requires that the duration of multitone burst transmitted be increased by the same amount over normal minimum burst length.

Example See example for `AP.S2CDsp.FastTest.FFTLength`.

AP.S2CDsp.FastTest.TrigSource

Property

Syntax `AP.S2CDsp.FastTest.TrigSource`

Data Type	Integer	
	0	DGen: this selection functions only on Dual Domain units (SYS-2300 series). If the Digital Generator is generating a signal from a waveform file, a Digital Generator trigger is issued each time the first sample from the file is generated.
	1	Tight: See description below.
	2	Normal: See description below.
	3	Loose: See description below.
	4	External: this selection is operational only with SYS-2300 series (Dual Domain) units. It is the signal connected to pin 3 of the 15-pin D-sub connector on the rear of the DSP module.

- If pin 3 is high (or open circuit, in which case it is pulled high by an internal pull-up resistor), triggering occurs at the next digital sample. Pulling pin 3 low from an external device holds off triggering, with acquisition being triggered on the next sample after pin 3 is pulled high.
- 5 Off: this selection produces untriggered or free-running operation. Acquisition and processing begins as soon as the F9 key, Go, or `AP . Sweep . Start` command is initiated. The Off selection is the recommended triggering mode when System Two and FASTTEST are testing devices by simultaneously driving their input and measuring their output as opposed to capturing a pre-recorded or remotely-originated multitone signal.
- 6 AGen:

Description

This command sets the Multitone Audio Analyzer Triggering.

1. Using the tone frequencies represented in the Digital Generator buffer as a reference, FASTTEST looks at the received signal to see if the amplitude at each of a majority of those frequencies is within an acceptable relative amplitude range of the corresponding component of the reference signal. This criterion allows FASTTEST to ignore simple single-tone test signals, relatively-simple program material such as may be produced by a solo musical instrument, and conditions of silence.
2. Across all sections of the spectrum between tones in the reference signal, FASTTEST looks at the received signal to assure that its amplitude does not exceed a threshold of acceptability. This criterion allows FASTTEST to ignore complex voice and music program material which tends to have energy spread across much of the spectrum.

To permit user control of the triggering criteria, the allowable deviation from reference signal amplitude at generator tone frequencies (1 above) and the amount that energy at all other frequencies must be attenuated (2 above) are settable at three values. The Tight, Normal, and Loose selections each represents a different trade-off between the chance of false response on non-multitone signals versus the possibility of not triggering on legitimate multitone signals from a device with large amounts of noise and distortion and/or large deviations from flat

frequency response. Select Tight for the minimum chance of false triggering. This may be necessary when using very short generator waveform files (less than 2048 samples) since the consequent poorer frequency resolution makes it more difficult to discriminate between multitone signals and program material. Use Loose if FASTTEST will not otherwise trigger on highly distorted or noisy signals or signals passed through narrow-band or otherwise non-flat devices.

Example

See example for `AP.S2CDsp.FastTest.FFTLength`.

User Notes

FFT Spectrum Analyzer

AP.S2CDsp.FFT.AcquireLength

Property

Syntax `AP.S2CDsp.FFT.AcquireLength`

Data Type Integer

<i>0</i>	Track FFT
<i>1</i>	800
<i>2</i>	1.5k
<i>3</i>	2.5k
<i>4</i>	5k
<i>5</i>	10k
<i>6</i>	19k
<i>7</i>	24k
<i>8</i>	36k
<i>9</i>	72k
<i>10</i>	144k
<i>11</i>	256k

Description This command sets the FFT Spectrum Analyzer acquire buffer record length.

See Also `AP.S2CDsp.FFT.TransformLength`

Example See example for `AP.S2CDsp.FFT.Averages`.

AP.S2CDsp.FFT.Averages

Property

Syntax `AP.S2CDsp.FFT.Averages`

Data Type Integer

<i>0</i>	1
<i>1</i>	2
<i>2</i>	4
<i>3</i>	8

4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096

Description

This command sets the FFT Spectrum Analyzer number of FFT averages.

FFT has the ability to average a number of successive acquisitions and spectrum analyses of a signal and display the averaged result. Since noise is random in amplitude and phase, averaging a succession of noise measurements results in a degree of cancellation and the averaged result will have less variance than the initial acquisition. Coherent signals, however, are the same at each acquisition and thus are not affected by averaging. Thus, spectral averaging will reduce the maximum peak excursions of the noise baseline in a typical signal spectrum while not affecting continuous signals, making it easier to detect and measure low amplitude signals and distortion products. Averaging over many seconds or minutes of program material such as music or voice may also be useful in order to determine the long-term average amplitude versus frequency distribution.

See Also

AP.S2CDsp.FFT.AveragesType

Example

```
Sub Main
  AP.File.OpenTest "FFTTEST1.AT2" 'Open test
  With AP.S2CDsp.FFT
    .InputFormat = 1 'HiRes A/D @65k input
    .Ch1Source = 0 'Analyzer A input
    .Ch2Source = 1 'Analyzer B input
    .AverageType = 0 'Averaging to Power _
      (Spectrum Only)
    .Averages = 4 'Number of Averages to 16
    .AcquireLength = 5 'Acquisition Length 10k
    .TransformLength = 5 'FFT Length 8192
    .StartTime("sec") = 0 'Set Start Time to 0 sec
```

```

        .SubtractDC = 1      'Subtract Average waveform level
        .WfmDisplay = 0     'Waveform Display to Interpolate
        .Window = 0        'Window to Blackman-Harris
        .TrigDelay("sec") = 0.000000 'Trigger Delay
        .TrigSource = 0     'Trigger Source Free Run
        .TrigSensitivity("dBFS") = -59.999594
        .TrigPolarity = 0  'Trigger Slope Positive
    End With

    AP.Sweep.Start          'Perform FFT
End Sub

```

AP.S2CDsp.FFT.AveragesType

Property

Syntax

AP.S2CDsp.FFT.AveragesType

Data Type

Integer

The following list contains the selections relevant to the `AP.S2CDsp.FFT.Window` command Blackman-Harris, Hann, Flat-Top, Equiripple, and None Window selections.

0	Power (spectrum only)
1	Sync, re-align
2	Sync

The following list contains the selections relevant to the `AP.S2CDsp.FFT.Window` command None, move to bin center Window selection.

0	Power (spectrum only)
1	Sync, re-align, move center first
2	Sync, re-align, average first
3	Sync, move center first
4	Sync, average first

Description

This command sets the type of Averaging the FFT Spectrum Analyzer uses when producing Time and Frequency domain measurements.

This command enables or disables computation of the average value of all samples in the acquisition buffer and subtraction that computed value from the value of each sample before an FFT transform or processing the values according to the Wave Display field and sending the results to the computer for display. The effect of the Subtract Average Value function is thus very similar having used AC coupling before acquiring the signal, as long as no signal peaks exceeded digital full scale. Use of the Subtract Average Value function may be valuable when examining low-level signals which contain a significant amount of DC offset, particularly in time domain (oscilloscope) presentations where the DC offset might otherwise cause the signal to be off-screen at the selected vertical span.

See Also `AP.S2CDsp.FFT.Averages`

Example See example for `AP.S2CDsp.FFT.Averages`.

AP.S2CDsp.FFT.Ch1Rdg

Property

Syntax `AP.S2CDsp.FFT.Ch1Rdg(ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, Bits.

Description This command returns a unsettled reading for the FFT Spectrum Analyzer channel 1 Peak Monitor meter and zeros the ready count.

See Also `AP.S2CDsp.FFT.Ch1Ready`, `AP.S2CDsp.FFT.Ch1Trig`

Example

```
Sub Main
    AP.File.OpenTest "FFTTEST2.AT2"      'Open test
    AP.S2CDsp.FFT.Ch1Source = 1'Set Ch 1 Source to Anlr-B
    Wait 1
    AP.S2CDsp.FFT.Ch1Trig      'Trigger a new reading
    Do
        Ready = AP.S2CDsp.FFT.Ch1Ready
    Loop Until Ready > 0      'Wait for new reading
    Reading1 = AP.S2CDsp.FFT.Ch1Rdg("FFS")'Get new reading
```

```

NewLine$ = Chr(13)
a$= "Ch 1 Source "+Left(Str$(Reading1),6)+"FFS"
AP.Prompt.Text = a$ + NewLine$ + b$ + NewLine
AP.Prompt.ShowWithContinue
Beep
Stop
End Sub

```

AP.S2CDsp.FFT.Ch1Ready

Property

Syntax `AP.S2CDsp.FFT.Ch1Ready`

Data Type Integer

0 Reading not ready.

>0 Reading ready.

Description This command returns the FFT Spectrum Analyzer channel 1 Peak Monitor meter unsettled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.FFT.Ch1Rdg` or `AP.S2CDsp.FFT.Ch1Trig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.FFT.Ch1Rdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.FFT.Ch1Rdg`, `AP.S2CDsp.FFT.Ch1Trig`

Example See example for `AP.S2CDsp.FFT.Ch1Rdg`.

AP.S2CDsp.FFT.Ch1Source**Property****Syntax** `AP.S2CDsp.FFT.Ch1Source`**Data Type** Integer

The following list contains the selections relevant to the `AP.S2CDsp.FFT.InputFormat` command A/D Input selection.

0	Anlr-A
1	Anlr-B
2	Anlr Reading Ampl
3	Anlr Reading Ratio
4	Ch. A Generator
5	Ch. B Generator
6	Jitter Signal
7	None

The following list contains the selections relevant to the `AP.S2CDsp.FFT.InputFormat` command Digital Input selection.

0	A
1	B
2	None

Description This command sets the FFT Spectrum Analyzer Channel 1 Input.**Example** See example for `AP.S2CDsp.FFT.Ch1Rdg`.**AP.S2CDsp.FFT.Ch1Trig****Method****Syntax** `AP.S2CDsp.FFT.Ch1Trig`**Description** Causes a restart of the reading cycle and zeros the ready count for the `AP.S2CDsp.FFTGen.Ch1Rdg` command. The reading in progress is aborted.**See Also** `AP.S2CDsp.FFT.Ch1Rdg`, `AP.S2CDsp.FFT.Ch1Ready`**Example** See example for `AP.S2CDsp.FFT.Ch1Rdg`.

AP.S2CDsp.FFT.Ch2Rdg**Property**

Syntax `AP.S2CDsp.FFT.Ch2Rdg (ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, Bits.

Description

This command returns a unsettled reading for the FFT Spectrum Analyzer channel 2 Peak Monitor meter and zeros the ready count.

See Also

`AP.S2CDsp.FFT.Ch2Ready`, `AP.S2CDsp.FFT.Ch2Trig`

Example

```
Sub Main
    AP.File.OpenTest "FFTTEST2.AT2"      'Open test
    AP.S2CDsp.FFT.Ch2Source = 0 'Set Ch 2 Source to Anlr-A
    Wait 1
    AP.S2CDsp.FFT.Ch2Trig      'Trigger a new reading
    Do
        Ready = AP.S2CDsp.FFT.Ch2Ready
    Loop Until Ready > 0      'Wait for a new reading
    Reading1 = AP.S2CDsp.FFT.Ch2Rdg ("FFS") 'Get a new
        reading
    NewLine$ = Chr(13)
    a$= "Ch 2 Source "+Left(Str$(Reading1),6)+"FFS"
    AP.Prompt.Text = a$ + NewLine$ + b$ + NewLine
    AP.Prompt.ShowWithContinue
    Beep
    Stop
End Sub
```

AP.S2CDsp.FFT.Ch2Ready**Property**

Syntax `AP.S2CDsp.FFT.Ch2Ready`

Data Type Integer

0 Reading not ready.

>0 Reading ready.

- Description** This command returns the FFT Spectrum Analyzer channel 2 Peak Monitor meter unsettled reading ready count.
- Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.FFT.Ch2Rdg` or `AP.S2CDsp.FFT.Ch2Trig` commands will zero the ready count.
- If the reading is found to be ready, a call to the `AP.S2CDsp.FFT.Ch2Rdg` command will be guaranteed to return quickly.
- See Also** `AP.S2CDsp.FFT.Ch2Rdg`, `AP.S2CDsp.FFT.Ch2Trig`
- Example** See example for `AP.S2CDsp.FFT.Ch2Rdg`.

AP.S2CDsp.FFT.Ch2Source

Property

Syntax `AP.S2CDsp.FFT.Ch2Source`

Data Type Integer

The following list contains the selections relevant to the `AP.S2CDsp.FFT.InputFormat` command A/D input selection.

0	Anlr-A
1	Anlr-B
2	Anlr Reading Ampl
3	Anlr Reading Ratio
4	Ch. A Generator
5	Ch. B Generator
6	Jitter Signal
7	None

The following list contains the selections relevant to the `AP.S2CDsp.FFT.InputFormat` command Digital input selection.

0	A
1	B

2 None

Description This command sets the FFT Spectrum Analyzer Channel 2 Input.

Example See example for AP . S2CDsp . FFT . Ch2Rdg.

AP.S2CDsp.FFT.Ch2Trig

Method

Syntax AP . S2CDsp . FFT . Ch2Trig

Description Causes a restart of the reading cycle and zeros the ready count for the AP . S2CDsp . FFT . Ch2Rdg comand. The reading in progress is aborted.

See Also AP . S2CDsp . FFT . Ch2Rdg , AP . S2CDsp . FFT . Ch2Ready

Example See example for AP . S2CDsp . FFT . Ch2Rdg.

AP.S2CDsp.FFT.InputFormat

Property

Syntax AP . S2CDsp . FFT . InputFormat

Data Type Integer

0	Digital @ ISR:
1	HiRes A/D @65536
2	HiBW A/D @131072
3	HiBW A/D @262144
4	HiRes A/D @OSR
5	HiBW A/D @2xOSR

Description This command sets the FFT Spectrum Analyzer Input Format.

Example See example for AP . S2CDsp . FFT . Averages.

AP.S2CDsp.FFT.StartTime

Property

Syntax AP . S2CDsp . FFT . StartTime (ByVal Unit As String)

Data Type Double The acceptable range of numbers depends upon the sample Rate set on the Digital I/O panel, since the acquisition buffer is a fixed length in samples. At a 48 kHz sample rate, for example, the Start Time field will accept numbers between plus and minus 341 milliseconds.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: sec.

Description This command sets the FFT Spectrum Analyzer Start Time.

FFT permits the user to select any point in the acquired signal record as the beginning of the portion to be transformed. The FFT transform is then computed for the contiguous section of samples starting at that sample and continuing for the number of samples chosen in the Length field. FFT thus permits selective spectrum analysis of different sections of a complex signal such as program material or special test signals such as sinewave bursts.

If the original signal acquisition (F9) was made with a negative value in the Pre-trigger Time field, negative values up to and including that same value may be used as FFT Start Time values to permit spectrum analysis of the pre-trigger section of the acquired record.

Example See example for AP.S2CDsp.FFT.Averages.

AP.S2CDsp.FFT.SubtractDC

Property

Syntax AP.S2CDsp.FFT.SubtractDC

Data Type Integer

0	DC Coupled
1	Subtract Average
2	Subtract 1/2pk-pk

Description This command sets the FFT Spectrum Analyzer DC offset processing mode.

Example See example for AP.S2CDsp.FFT.Averages.

AP.S2CDsp.FFT.TransformLength**Property****Syntax** `AP.S2CDsp.FFT.TransformLength`**Data Type** Integer

0	256
1	512
2	1024
3	2048
4	4096
5	8192
6	16384
7	32768

Description This command sets the FFT Spectrum Analyzer FFT length.**Example** See example for `AP.S2CDsp.FFT.Averages`.**AP.S2CDsp.FFT.TrigDelay****Property****Syntax** `AP.S2CDsp.FFT.TrigDelay(ByVal Unit As String)`**Data Type** Double**Parameters**

Name	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: sec.

Description This command sets the FFT Spectrum Analyzer Trigger Delay time.

FFT has the ability to fill the acquisition buffer with signal samples starting at a user-defined time before the trigger occurs, then continuing until the buffer is full. This permits analysis of signal conditions both before and after the triggering event. A negative value entered in the Trigger Delay field determines how much time (and how many samples) prior to the trigger event are retained. The Pre-Trigger Time field is visible only on the large form of the Digital Analyzer panel. The total length of signal acquired will be as set in FFT Transform Length, with the remainder of the acquisition buffer filled after the trigger. For example, with maximum memory the length of

the acquisition buffer for each channel is 341 milliseconds at a 48 kHz rate. If the Pre-Trigger Time value is -50 milliseconds, for example, then 291 additional milliseconds of signal following the trigger will also be acquired to fill the entire 341 ms buffer.

Pre-trigger data is acquired in this fashion: when the F9 key is pressed or Go is clicked, FFT and the DSP module immediately begin acquiring data samples, even though no trigger event may have yet occurred. If the acquisition buffer should completely fill before a trigger event occurs, data continues to be acquired in a FIFO (first in first out) basis with the oldest data being dropped as new data is added. When the trigger event occurs, FFT effectively creates a marker at that location (time zero) and another marker at the pre-trigger time before time zero and continues acquiring until every location up to the pre-trigger marker is filled. Any portion from the pre-trigger time through time zero to the end of the record may then be displayed in oscilloscope fashion or transformed and viewed as a spectrum analysis.

Example

See example for `AP.S2CDsp.FFT.Averages`.

AP.S2CDsp.FFT.TrigLevel**Property**

Syntax `AP.S2CDsp.FFT.TrigLevel(ByVal Unit As String)`

Data Type Double The acceptable range of numbers is between plus and minus 1 for the FFS unit.

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, FS, and dBFS.

Description This command sets the FFT Spectrum Analyzer Trigger Level. This control determines the signal level that must be obtained before a trigger event can occur. Once the Trigger event occurs the next and following samples are placed into the acquisition buffer until full.

See Also `AP.S2CDsp.FFT.Ch2Rdg`, `AP.S2CDsp.FFT.Ch2Ready`

Example See example for `AP.S2CDsp.FFT.Averages`.

AP.S2CDsp.FFT.TrigPolarity

Property

Syntax	<code>AP.S2CDsp.FFT.TrigPolarity</code>	
Data Type	Integer	
	0	Positive: time zero will be the first positive-going zero crossing of the trigger signal selected in the Trigger Source field.
	1	Negative: time zero will be the first negative-going zero crossing of the selected trigger signal.
Description	This command sets the FFT Spectrum Analyzer trigger polarity.	
Example	See example for <code>AP.S2CDsp.FFT.Averages</code> .	

AP.S2CDsp.FFT.TrigSensitivity

Property

Syntax	<code>AP.S2CDsp.FFT.TrigSensitivity(ByVal Unit As String)</code>					
Data Type	Double	The acceptable range of numbers is between plus and minus 1 for the FFS unit.				
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>Unit</i></td> <td>String that designates the desired unit. The following units are valid for this command: FFS, FS, and dBFS.</td> </tr> </tbody> </table>	Name	Description	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, FS, and dBFS.	
Name	Description					
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, FS, and dBFS.					
Description	This command sets the FFT Spectrum Analyzer Trigger Sensitivity. This control determines the signal level that must be obtained before a zero crossing trigger event can occur.					
Example	See example for <code>AP.S2CDsp.FFT.Averages</code> .					

AP.S2CDsp.FFT.TrigSource

Property

Syntax	<code>AP.S2CDsp.FFT.TrigSource</code>
Data Type	Integer

- 0 Free Run: signal acquisition begins immediately after F9 or Go is initiated, regardless of signal amplitude. This is the typical operating mode with steady-state test signals.
- 1 Ch. 1 Auto:
- 2 Ch. 1 Fixed:
- 3 Ch. 2 Auto:
- 4 Ch. 2 Fixed:
- 5 External: The External selection refers to pin 3 of the 15-pin D-sub connector on the rear of the DSP module. This source is operational only with the SYS-2300 series Dual Domain units. If pin 3 is high (or open circuit, in which case it is pulled high by an internal pull-up resistor), triggering occurs at the next digital sample. Pulling pin 3 low from an external device holds off triggering, with acquisition being triggered on the next sample after pin 3 is pulled high. This External selection is unaffected by the Slope buttons.
- 6 Digital Gen: The Digital Generator selection functions only on Dual Domain units (SYS-2300 series). If the Digital Generator is generating any of the waveforms selectable in the Waveform field, a Digital Generator trigger occurs at each zero crossing of the waveform, positive-going or negative-going as selected by the Slope buttons. If the Digital Generator is generating a signal from a waveform file, a Digital Generator trigger occurs as the first sample is read from the waveform file.
- 7 Analog Gen: The Analog Generator Sync selection is the same signal as at the Generator Aux Signals Sync Output BNC on the front panel of System Two. This signal is a squarewave at the Analog Generator frequency in sinewave and squarewave waveforms, the envelope of the burst signal in all Burst waveforms, a squarewave at the lower IMD frequency in SMPTE IMD waveform, a squarewave at 1/2 the frequency spacing in CCIF IMD waveform, the squarewave IMD signal in DIM IMD waveform, and a pulse at the pseudo-random repetition rate in Pseudo noise modes. There is no signal in Random noise modes.
- 8 AC Mains: the power line frequency.

9	Jitter Gen: The Digital Input/Output Jitter Generator selection provides a trigger at each positive or negative zero crossing for the selected waveform type.
10	Ch1. Fixed Level
11	Ch2. Fixed Level

Description

This command sets the FFT Spectrum Analyzer Trigger Source.

The four channel 1 and channel 2 selections are software triggers, monitoring the signal (which may come from Digital or A/D sources) on the specified channel. channel 1 Fix and channel 2 Fix use a fixed threshold of 1.0%FS (-40 dBFS) on the channel referred to as the triggering threshold, and will trigger on the first signal excursion of the selected slope (Positive or Negative radio button) above that amplitude. The channel 1 and 2 Auto selections will cause triggering at one-half the peak-to-peak value if the selected channel has a signal amplitude greater than digital zero.

Example

See example for `AP.S2CDsp.FFT.Averages`.

AP.S2CDsp.FFT.WfmDisplay**Property**

Syntax	<code>AP.S2CDsp.FFT.WfmDisplay</code>
Data Type	Integer
	0 Interpolate
	1 Display Samples
	2 Peak Values
	3 Absolute Values

Description

This command sets the FFT Spectrum Analyzer generator waveform display mode.

When Interpolate is selected, the DSP module will perform an interpolation calculation based on the assumption that the signal was band-limited by a low-pass filter before sampling. The Interpolate selection produces a much more accurate display of the signal waveform when the signal frequency is high (such as sample rate/100 or higher).

When Display Samples is selected, no processing takes place in the DSP module. At each time value plotted on the X-axis, the DSP simply sends the amplitude of the nearest-in-time acquired sample to the computer for plotting. When the signal frequency is low compared to the sample rate, this may produce an acceptable representation of the original signal waveform. At high signal frequencies, the waveform may be entirely unrecognizable in the Display Samples mode. For example, a 16 kHz sinewave acquired at the 48 kHz sample rate will have each cycle of waveform represented by only three amplitude samples and the result will look very little like a sinewave. The Display Samples mode may be useful when examining the true quantization-limited waveforms of very low amplitude digital domain signals.

When Peak Values is selected, the DSP searches all sample amplitudes in the acquisition buffer between each pair of X-axis time values plotted and returns to the computer the largest positive or negative value in that span, preserving the sign. The intended use of the Peak Values mode is when graphing a relatively long time span on the X-axis, where the combination of Start-to-Stop time span and Steps value on the Sweep panel results in skipping across many actual acquired samples between plotted points. For example, assume a signal is acquired at the 48 kHz sample rate (20.8 microseconds between samples). If the waveform of that signal is being viewed from 0 to 200 milliseconds with 400 steps, the time span between plotted points on the graph X-axis is 0.5 milliseconds (500 microseconds). There are approximately 24 samples between plotted points. If Peak Values or Absolute Values modes are not used, an unfortunate combination of signal frequency, X-axis span, and Points value can make it appear that no waveform, a near-DC signal, or a waveform at a completely different frequency is present. Since Peak Values searches through all sample values within each span between plotted points and sends the largest value to be plotted, signals cannot be missed.

When Absolute Values mode is selected, the DSP searches all sample amplitudes in each plotted-point-to-plotted-point span as it does in Peak Values mode, but takes the absolute value of the largest positive or negative value and always sends a positive number to the computer. The advantage of Absolute Values mode is that logarithms may be computed when all numbers are positive, so a dB unit may be

used on the Y axis to display the waveform. Waveform display with Absolute Values mode can create a wide dynamic range oscilloscope which displays the envelope of an audio signal, calibrated in familiar dB units such as dBV, dBm, dBu, etc. Absolute Values mode is most effective when the X-axis span and Points values are selected to produce approximately two plotted points per cycle of the waveform being plotted. For example, if an envelope display of tone burst waveforms of a 1 kHz signal (1 millisecond period) are being plotted across a 50 millisecond span, the Points value on the Sweep panel should be set to approximately 100.

Example See example for `AP.S2CDsp.FFT.Averages`.

AP.S2CDsp.FFT.Window

Property

Syntax `AP.S2CDsp.FFT.Window`

Data Type Integer

0	Blackman-Harris
1	Hann
2	Flat-Top
3	Equiripple
4	None
5	None, move to bin center
6	Hamming
7	Gaussian
8	Rife-Vincent 4
9	Rife-Vincent 5

Description This command sets the FFT Spectrum Analyzer Window selection. See Appendix C for FFT Window Discriptions.

See Also `AP.S2CDsp.FFT.AveragesType`

Example See example for `AP.S2CDsp.FFT.Averages`.

User Notes

Harmonic Analyzer

AP.S2CDsp.Harmonic.Ch1FreqRdg

Property

Syntax `AP.S2CDsp.Harmonic.Ch1FreqRdg`(ByVal *Unit* As String)

Data Type Double

Parameters

Part	Description
<i>Unit</i>	The following units are available: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM.

Description

This command returns a settled reading for the Harmonic Distortion Analyzer channel 1 Frequency meter and zeros the ready count.

See Also

`AP.S2CDsp.Harmonic.Ch1FreqReady`,
`AP.S2CDsp.Harmonic.Ch1FreqSettling`,
`AP.S2CDsp.Harmonic.Ch1FreqTrig`

Example

```
Sub Main
    AP.Application.NewTest
    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2
    AP.Application.PanelOpen apbPanelDSPSmall
    AP.S2CDsp.Program = 7 'Load Harmonic Distortion _
        Analyzer
    AP.Application.PanelOpen apbPanelDSPLarge
    With AP.S2CDsp.Harmonic
        .InputFormat = 1 'HiRes A/D @65k input
        .Ch1Source = 0 'Analog Analyzer Channel A _
            Input Source
        .Ch1FundSettling 1.000000, 0.000001, "V", 3, _
            0.030000, 1
        .Ch1FreqSettling 0.500000, 0.010000, "Hz", 3, _
            0.030000, 1
    End With
    Wait .5
    .Ch1FundTrig 'Trigger new Fundamental _
        Amplitude reading
    .Ch1FreqTrig 'Trigger new Frequency reading
```

```

    Do 'Wait for new readings
    Loop Until .Ch1FundReady And .Ch1FreqReady
    var1 = .Ch1FundRdg("V")
    var2 = .Ch1FreqRdg("Hz")
End With
Text1$= "Channel 1 Fundamental Amplitude " & _
    Str$(Format(var1, "##.000")) & "V"
Text2$= "Channel 1 Frequency " & _
    Str$(Format(var2, "##.000")) & "Hz"
AP.Prompt.Text = Text1$ & Chr(13) & Text2$ _
    'Text String and New Line
AP.Prompt.ShowWithContinue 'Display Prompt
Stop
End Sub

```

AP.S2CDsp.Harmonic.Ch1FreqReady

Property

Syntax AP.S2CDsp.Harmonic.Ch1FreqReady

Data Type Integer

0 Reading not ready.
>0 Reading ready.

Description This command returns the Harmonic Distortion Analyzer channel 1 Frequency settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the AP.S2CDsp.Harmonic.Ch1FreqRdg or AP.S2CDsp.Harmonic.Ch1FreqTrig commands will zero the ready count.

If the reading is found to be ready, a call to the AP.S2CDsp.Harmonic.Ch1FreqRdg command will be guaranteed to return quickly.

See Also AP.S2CDsp.Harmonic.Ch1FreqRdg,
AP.S2CDsp.Harmonic.Ch1FreqSettling,
AP.S2CDsp.Harmonic.Ch1FreqTrig

Example See example for AP.S2CDsp.Harmonic.Ch1FreqRdg.

AP.S2CDsp.Harmonic.Ch1FreqSettling

Method

Syntax `AP.S2CDsp.Harmonic.Ch1FreqSettling(ByVal Tolerance As Double, ByVal Floor As Double, ByVal FloorUnit As String, ByVal Points As Integer, ByVal Delay As Double, ByVal Algorithm As Integer)`

Parameters See Appendix A for Settling Algorithm and parameter name descriptions.

Description This command sets the settling parameters for the AP.S2CDsp.Harmonic.Ch1FreqRdg command.

See Also AP.S2CDsp.Harmonic.Ch1FreqRdg,
AP.S2CDsp.Harmonic.Ch1FreqReady,
AP.S2CDsp.Harmonic.Ch1FreqTrig

Example See example for AP.S2CDsp.Harmonic.Ch1FreqRdg.

AP.S2CDsp.Harmonic.Ch1FreqTrig

Method

Syntax `AP.S2CDsp.Harmonic.Ch1FreqTrig`

Description Causes a restart of the reading cycle and zeros the ready count for the AP.S2CDsp.Harmonic.Ch1FreqRdg command. The reading in progress is aborted.

See Also AP.S2CDsp.Harmonic.Ch1FreqRdg,
AP.S2CDsp.Harmonic.Ch1FreqReady,
AP.S2CDsp.Harmonic.Ch1FreqSettling

Example See example for AP.S2CDsp.Harmonic.Ch1FreqRdg.

AP.S2CDsp.Harmonic.Ch1FundRdg**Property**

Syntax `AP.S2CDsp.Harmonic.Ch1FundRdg(ByVal Unit As String)`

Data Type Double

Part	Description
<i>Unit</i>	The following units are available: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM.

Description This command returns a settled reading for the Harmonic Analyzer channel 1 Fundamental Amplitude meter and zeros the ready count.

See Also `AP.S2CDsp.Harmonic.Ch1FundReady`,
`AP.S2CDsp.Harmonic.Ch1FundSettling`,
`AP.S2CDsp.Harmonic.Ch1FundTrig`

Example See example for `AP.S2CDsp.Harmonic.Ch1FreqRdg`.

AP.S2CDsp.Harmonic.Ch1FundReady**Property**

Syntax `AP.S2CDsp.Harmonic.Ch1FundReady`

Data Type Integer

<i>0</i>	Reading not ready.
<i>>0</i>	Reading ready.

Description This command returns the Harmonic Distortion Analyzer channel 1 Fundamental Amplitude settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.Harmonic.Ch1FundRdg` or `AP.S2CDsp.Harmonic.Ch1FundTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.Harmonic.Ch1FundRdg` command will be guaranteed to return quickly.

See Also AP.S2CDsp.Harmonic.Ch1FundRdg,
AP.S2CDsp.Harmonic.Ch1FundSettling,
AP.S2CDsp.Harmonic.Ch1FundTrig

Example See example for AP.S2CDsp.Harmonic.Ch1FreqRdg.

AP.S2CDsp.Harmonic.Ch1FundSettling

Method

Syntax **AP.S2CDsp.Harmonic.Ch1FundSettling**(ByVal *Tolerance* As Double, ByVal *Floor* As Double, ByVal *FloorUnit* As String, ByVal *Points* As Integer, ByVal *Delay* As Double, ByVal *Algorithm* As Integer)

Parameters See Appendix A for Settling Algorithm and parameter name descriptions.

Description This command sets the settling parameters for the AP.S2CDsp.Harmonic.Ch1FundRdg command.

See Also AP.S2CDsp.Harmonic.Ch1FundRdg,
AP.S2CDsp.Harmonic.Ch1FundReady,
AP.S2CDsp.Harmonic.Ch1FundTrig

Example See example for AP.S2CDsp.Harmonic.Ch1FreqRdg.

AP.S2CDsp.Harmonic.Ch1FundTrig

Method

Syntax **AP.S2CDsp.Harmonic.Ch1FundTrig**

Description Causes a restart of the reading cycle and zeros the ready count for the AP.S2CDsp.Harmonic.Ch1FundRdg command. The reading in progress is aborted.

See Also AP.S2CDsp.Harmonic.Ch1FundRdg,
AP.S2CDsp.Harmonic.Ch1FundReady,
AP.S2CDsp.Harmonic.Ch1FundSettling

Example See example for AP.S2CDsp.Harmonic.Ch1FreqRdg.

AP.S2CDsp.Harmonic.Ch1Source**Property**

Syntax `AP.S2CDsp.Harmonic.Ch1Source`

Data Type Integer

The following list contains the selections relevant to the AP.S2CDsp.Harmonic.InputFormat command A/D input selections.

0	Anlr-A
1	Anlr-B
2	Anlr Rdg Ampl
3	None

The following list contains the selections relevant to the AP.S2CDsp.Harmonic.InputFormat command Digital input selection.

0	A
1	B
2	None

Description This command sets the Harmonic Distortion Analyzer Channel 1 Input.

See Also `AP.S2CDsp.Harmonic.InputFormat`

Example See example for `AP.S2CDsp.Harmonic.Ch1FreqRdg`.

AP.S2CDsp.Harmonic.Ch1Sum1Harmonics**Property**

Syntax `AP.S2CDsp.Harmonic.Ch1Sum1Harmonics(ByVal HarmonicId As Integer)`

Data Type Boolean

<i>True</i>	Add specified harmonics to RSS summation for channel 1 Sum1 reading.
<i>False</i>	Remove specified harmonics from RSS summation for channel 1 Sum1 reading.

Parameters	Part	Description
	<i>HarmonicId</i>	Single decimal value or the sum of several decimal values representing multiple harmonics or a constant value as defined below. Harmonic decimal value table: 2 = 2 3 = 4 4 = 8 5 = 16 6 = 32 7 = 64 8 = 128 9 = 256 10 = 512 11 = 1024 12 = 2048 13 = 4096 14 = 8192 15 = 16384
Constants	HarmonicId Parameter	Description
	<i>apbEven</i>	Select all even harmonics.
	<i>apbOdd</i>	Select all odd harmonics.
	<i>apbAll</i>	Select all harmonics.
Description	This command adds or removes individual or as a group harmonics from the Harmonic Distortion Analyzer RSS summation #1 for channel 1.	
Example	See example for <code>AP.S2CDsp.Harmonic.Ch1Sum1Rdg</code> .	

AP.S2CDsp.Harmonic.Ch1Sum1Rdg Property

Syntax	<code>AP.S2CDsp.Harmonic.Ch1Sum1Rdg(ByVal Unit As String)</code>
Data Type	Double

Parameters	Part	Description
	<i>Unit</i>	<p>The following units are valid for the "Absolute Units" selection of the AP.S2CDsp.Harmonic.Ch1Sum1RelUnits command: FFS, %FS, dBFS, Bits, V, Vp, Vpp, dBu, dBV, dBr 1, dBr 2.</p> <p>The following units are valid for the "Relative Units" selection of the AP.S2CDsp.Harmonic.Ch1Sum1RelUnits command: %, dB, PPM, X/Y</p>
Description		This command returns a settled reading for the Harmonic Distortion Analyzer channel 1 Sum #1 meter and zeros the ready count.
See Also		AP.S2CDsp.Harmonic.Ch1Sum1Ready, AP.S2CDsp.Harmonic.Ch1Sum1Settling, AP.S2CDsp.Harmonic.Ch1Sum1Trig
Example		<pre> Sub Main AP.Application.NewTest AP.Gen.Output = True AP.Anlr.ChAInput = 2 AP.Application.PanelOpen apbPanelDSPSmall AP.S2CDsp.Program = 7 'Load Harmonic Distortion _ Analyzer AP.Application.PanelOpen apbPanelDSPLarge With AP.S2CDsp.Harmonic .InputFormat = 1 'HiRes A/D @65k input .Ch1Source = 0 'Analog Analyzer Channel A _ Input Source .Ch1Sum1Harmonics(2) = True 'Select harmonic #2 .Ch1Sum2Harmonics(4) = True 'Select harmonic #3 .Ch1Sum1RelUnits = False .Ch1Sum2RelUnits = False .Ch1Sum1Settling 0.500000, -1.200000e+002, _ "dBV", 1, 0.002000, 2 .Ch1Sum2Settling 0.500000, -1.200000e+002, _ "dBV", 1, 0.002000, 2 .Ch1Sum1Trig 'Trigger new Channel 1 reading .Ch1Sum2Trig 'Trigger new Channel 2 reading Do 'Wait for new readings Loop Until .Ch1Sum1Ready And .Ch1Sum2Ready var1 = .Ch1Sum1Rdg("dBV") </pre>

```

        var2 = .Ch1Sum2Rdg("dBV")
    End With
    Text1$= "Channel 1 Sum1 " & _
        Str$(Format(var1, "##.000")) & "dBV"
    Text2$= "Channel 1 Sum2 " & _
        Str$(Format(var2, "##.000")) & "dBV"
    AP.Prompt.Text = Text1$ & Chr$(13) & Text2$ & _
        'Text String and New Line
    AP.Prompt.ShowWithContinue 'Display Prompt
    Stop
End Sub

```

AP.S2CDsp.Harmonic.Ch1Sum1Ready

Property

Syntax `AP.S2CDsp.Harmonic.Ch1Sum1Ready`

Data Type Integer

0 Reading not ready.
 >0 Reading ready.

Description This command returns the Harmonic Distortion Analyzer channel 1 Sum #1 settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.Harmonic.Ch1Sum1Rdg` or `AP.S2CDsp.Harmonic.Ch1Sum1Trig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.Harmonic.Ch1Sum1Rdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.Harmonic.Ch1Sum1Rdg`,
`AP.S2CDsp.Harmonic.Ch1Sum1Settling`,
`AP.S2CDsp.Harmonic.Ch1Sum1Trig`

Example See example for `AP.S2CDsp.Harmonic.Ch1Sum1Rdg`.

AP.S2CDsp.Harmonic.Ch1Sum1RelUnits**Property**

Syntax	<code>AP.S2CDsp.Harmonic.Ch1Sum1RelUnits</code>
Data Type	Boolean
	<i>True</i> Relative Units.
	<i>False</i> Absolute Units.
Description	This command selects the units that are available for the <code>AP.S2CDsp.Harmonic.Ch1Sum1Rdg</code> command
Example	See example for <code>AP.S2CDsp.Harmonic.Ch1Sum1Rdg</code> .

AP.S2CDsp.Harmonic.Ch1Sum1Settling**Method**

Syntax	<code>AP.S2CDsp.Harmonic.Ch1Sum1Settling(ByVal Tolerance As Double, ByVal Floor As Double, ByVal FloorUnit As String, ByVal Points As Integer, ByVal Delay As Double, ByVal Algorithm As Integer)</code>
Parameters	See Appendix A for Settling Algorithm and parameter name descriptions.
Description	This command sets the settling parameters for the <code>AP.S2CDsp.Harmonic.Ch1Sum1Rdg</code> command.
See Also	<code>AP.S2CDsp.Harmonic.Ch1Sum1Rdg</code> , <code>AP.S2CDsp.Harmonic.Ch1Sum1Ready</code> , <code>AP.S2CDsp.Harmonic.Ch1Sum1Trig</code>
Example	See example for <code>AP.S2CDsp.Harmonic.Ch1Sum1Rdg</code> .

AP.S2CDsp.Harmonic.Ch1Sum1Trig**Method**

Syntax	<code>AP.S2CDsp.Harmonic.Ch1Sum1Trig</code>
Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP.S2CDsp.Harmonic.Ch1Sum1Rdg</code> command. The reading in progress is aborted.

See Also AP.S2CDsp.Harmonic.Ch1Sum1Rdg,
 AP.S2CDsp.Harmonic.Ch1Sum1Ready,
 AP.S2CDsp.Harmonic.Ch1Sum1Settling

Example See example for AP.S2CDsp.Harmonic.Ch1Sum1Rdg.

AP.S2CDsp.Harmonic.Ch1Sum2Harmonics Property

Syntax AP.S2CDsp.Harmonic.Ch1Sum2Harmonics(ByVal HarmonicId As Integer)

Data Type Boolean

True Add specified harmonics to RSS summation for channel 1 Sum2 reading.
False Remove specified harmonics from RSS summation for channel 1 Sum2 reading.

Parameters

Part	Description
<i>HarmonicId</i>	Single decimal value or the sum of several decimal values representing multiple harmonics or a constant value as defined below. Harmonic decimal value table: 2 = 2 3 = 4 4 = 8 5 = 16 6 = 32 7 = 64 8 = 128 9 = 256 10 = 512 11 = 1024 12 = 2048 13 = 4096 14 = 8192 15 = 16384

Constants	HarmonicId Parameter	Description
	<i>apbEven</i>	Select all even harmonics.
	<i>apbOdd</i>	Select all odd harmonics.
	<i>apbAll</i>	Select all harmonics.

Description This command adds or removes individual harmonics from the Harmonic Distortion Analyzer RSS summation #2 for channel 1. Use of the optional group parameter allows addition or removal of even, odd, or all harmonics.

Example See example for AP.S2CDsp.Harmonic.Ch1Sum1Rdg.

AP.S2CDsp.Harmonic.Ch1Sum2Rdg

Property

Syntax `AP.S2CDsp.Harmonic.Ch1Sum2Rdg(ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
<i>Unit</i>	The following units are valid for the "Absolute Units" selection of the AP.S2CDsp.Harmonic.Ch1Sum2RelUnits command: FFS, %FS, dBFS, Bits, V, Vp, Vpp, dBu, dBV, dBr 1, dBr 2.
	The following units are valid for the "Relative Units" selection of the AP.S2CDsp.Harmonic.Ch1Sum2RelUnits command: %, dB, PPM, X/Y

Description This command returns a settled reading for the Harmonic Distortion Analyzer channel 1 Sum #2 meter and zeros the ready count.

See Also `AP.S2CDsp.Harmonic.Ch1Sum2Ready`,
`AP.S2CDsp.Harmonic.Ch1Sum2Settling`,
`AP.S2CDsp.Harmonic.Ch1Sum2Trig`

Example See example for AP.S2CDsp.Harmonic.Ch1Sum1Rdg.

AP.S2CDsp.Harmonic.Ch1Sum2Ready**Property**

Syntax	<code>AP.S2CDsp.Harmonic.Ch1Sum2Ready</code>				
Data Type	Integer				
	<table> <tr> <td><i>0</i></td> <td>Reading not ready.</td> </tr> <tr> <td><i>>0</i></td> <td>Reading ready.</td> </tr> </table>	<i>0</i>	Reading not ready.	<i>>0</i>	Reading ready.
<i>0</i>	Reading not ready.				
<i>>0</i>	Reading ready.				
Description	<p>This command returns the Harmonic Distortion Analyzer channel 1 Sum #2 settled reading ready count.</p> <p>Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the <code>AP.S2CDsp.Harmonic.Ch1Sum2Rdg</code> or <code>AP.S2CDsp.Harmonic.Ch1Sum2Trig</code> commands will zero the ready count.</p> <p>If the reading is found to be ready, a call to the <code>AP.S2CDsp.Harmonic.Ch1Sum2Rdg</code> command will be guaranteed to return quickly.</p>				
See Also	<code>AP.S2CDsp.Harmonic.Ch1Sum2Rdg</code> , <code>AP.S2CDsp.Harmonic.Ch1Sum2Settling</code> , <code>AP.S2CDsp.Harmonic.Ch1Sum2Trig</code>				
Example	See example for <code>AP.S2CDsp.Harmonic.Ch1Sum1Rdg</code> .				

AP.S2CDsp.Harmonic.Ch1Sum2RelUnits**Property**

Syntax	<code>AP.S2CDsp.Harmonic.Ch1Sum2RelUnits</code>				
Data Type	Boolean				
	<table> <tr> <td><i>True</i></td> <td>Relative Units.</td> </tr> <tr> <td><i>False</i></td> <td>Absolute Units.</td> </tr> </table>	<i>True</i>	Relative Units.	<i>False</i>	Absolute Units.
<i>True</i>	Relative Units.				
<i>False</i>	Absolute Units.				
Description	This command selects the units that are available for the <code>AP.S2CDsp.Harmonic.Ch1Sum2Rdg</code> command				
Example	See example for <code>AP.S2CDsp.Harmonic.Ch1Sum1Rdg</code> .				

AP.S2CDsp.Harmonic.Ch1Sum2Settling**Method**

Syntax	<code>AP.S2CDsp.Harmonic.Ch1Sum2Settling</code> (ByVal <i>Tolerance</i> As Double, ByVal <i>Floor</i> As Double, ByVal <i>FloorUnit</i> As String, ByVal <i>Points</i> As Integer, ByVal <i>Delay</i> As Double, ByVal <i>Algorithm</i> As Integer)
Parameters	See Appendix A for Settling Algorithm and parameter name descriptions.
Description	This command sets the settling parameters for the <code>AP.S2CDsp.Harmonic.Ch1Sum2Rdg</code> command.
See Also	<code>AP.S2CDsp.Harmonic.Ch1Sum2Rdg</code> , <code>AP.S2CDsp.Harmonic.Ch1Sum2Ready</code> , <code>AP.S2CDsp.Harmonic.Ch1Sum2Trig</code>
Example	See example for <code>AP.S2CDsp.Harmonic.Ch1Sum1Rdg</code> .

AP.S2CDsp.Harmonic.Ch1Sum2Trig**Method**

Syntax	<code>AP.S2CDsp.Harmonic.Ch1Sum2Trig</code>
Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP.S2CDsp.Harmonic.Ch1Sum2Rdg</code> command. The reading in progress is aborted.
See Also	<code>AP.S2CDsp.Harmonic.Ch1Sum2Rdg</code> , <code>AP.S2CDsp.Harmonic.Ch1Sum2Ready</code> , <code>AP.S2CDsp.Harmonic.Ch1Sum2Settling</code>
Example	See example for <code>AP.S2CDsp.Harmonic.Ch1Sum1Rdg</code> .

AP.S2CDsp.Harmonic.Ch2FreqRdg**Property**

Syntax `AP.S2CDsp.Harmonic.Ch2FreqRdg`(ByVal *Unit* As String)

Data Type Double

Parameters

Part	Description
------	-------------

<i>Unit</i>	The following units are available: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM.
-------------	---

Description

This command returns a settled reading for the Harmonic Distortion Analyzer channel 2 Frequency meter and zeros the ready count.

See Also

`AP.S2CDsp.Harmonic.Ch2FreqReady`,
`AP.S2CDsp.Harmonic.Ch2FreqSettling`,
`AP.S2CDsp.Harmonic.Ch2FreqTrig`

Example

```
Sub Main
    AP.Application.NewTest
    AP.Gen.Output = True
    AP.Anlr.ChBInput = 2
    AP.Application.PanelOpen apbPanelDSPSmall
    AP.S2CDsp.Program = 7    'Load Harmonic Distortion _
        Analyzer
    AP.Application.PanelOpen apbPanelDSPLarge
    With AP.S2CDsp.Harmonic
        .InputFormat = 1    'HiRes A/D @65k input
        .Ch2Source = 1    'Analog Analyzer Channel B _
            Input Source
        .Ch2FundSettling 1.000000, 0.000001, "V", _
            3, 0.030000, 1
        .Ch2FreqSettling 0.500000, 0.010000, "Hz", _
            3, 0.030000, 1
    Wait .5
        .Ch2FundTrig    'Trigger new Fundamental _
            Amplitude reading
        .Ch2FreqTrig    'Trigger new Frequency reading
    Do 'Wait for new readings
    Loop Until .Ch2FundReady And .Ch2FreqReady
    var1 = .Ch2FundRdg("V")
    var2 = .Ch2FreqRdg("Hz")
```

```

End With
Text1$= "Channel 2 Fundamental Amplitude " & _
    Str$(Format(var1, "##.000")) & "V"
Text2$= "Channel 2 Frequency " & _
    Str$(Format(var2, "##.000")) & "Hz"
AP.Prompt.Text = Text1$ & Chr(13) & Text2$ _
    'Text String and New Line
AP.Prompt.ShowWithContinue 'Display Prompt
Stop
End Sub

```

AP.S2CDsp.Harmonic.Ch2FreqReady

Property

Syntax `AP.S2CDsp.Harmonic.Ch2FreqReady`

Data Type Integer

0 Reading not ready.
 >0 Reading ready.

Description This command returns the Harmonic Distortion Analyzer channel 2 Frequency settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.Harmonic.Ch2FreqRdg` or `AP.S2CDsp.Harmonic.Ch2FreqTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.Harmonic.Ch2FreqRdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.Harmonic.Ch2FreqRdg`,
`AP.S2CDsp.Harmonic.Ch2FreqSettling`,
`AP.S2CDsp.Harmonic.Ch2FreqTrig`

Example See example for `AP.S2CDsp.Harmonic.Ch2FreqRdg`.

AP.S2CDsp.Harmonic.Ch2FreqSettling**Method**

Syntax	AP.S2CDsp.Harmonic.Ch2FreqSettling (ByVal <i>Tolerance</i> As Double, ByVal <i>Floor</i> As Double, ByVal <i>FloorUnit</i> As String, ByVal <i>Points</i> As Integer, ByVal <i>Delay</i> As Double, ByVal <i>Algorithm</i> As Integer)
Parameters	See Appendix A for Settling Algorithm and parameter name descriptions.
Description	This command sets the settling parameters for the AP.S2CDsp.Harmonic.Ch2FreqRdg command.
See Also	AP.S2CDsp.Harmonic.Ch2FreqRdg, AP.S2CDsp.Harmonic.Ch2FreqReady, AP.S2CDsp.Harmonic.Ch2FreqTrig
Example	See example for AP.S2CDsp.Harmonic.Ch2FreqRdg.

AP.S2CDsp.Harmonic.Ch2FreqTrig**Method**

Syntax	AP.S2CDsp.Harmonic.Ch2FreqTrig
	Description
	Causes a restart of the reading cycle and zeros the ready count for the AP.S2CDsp.Harmonic.Ch1FreqRdg command. The reading in progress is aborted.
See Also	AP.S2CDsp.Harmonic.Ch2FreqRdg, AP.S2CDsp.Harmonic.Ch2FreqReady, AP.S2CDsp.Harmonic.Ch2FreqSettling
Example	See example for AP.S2CDsp.Harmonic.Ch2FreqRdg.

AP.S2CDsp.Harmonic.Ch2FundRdg**Property**

Syntax	AP.S2CDsp.Harmonic.Ch2FundRdg (ByVal <i>Unit</i> As String)
Data Type	Double

Parameters	Part	Description
	<i>Unit</i>	The following units are available: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM.
Description		This command returns a settled reading for the Harmonic Distortion Analyzer channel 2 Fundamental Amplitude meter and zeros the ready count.
See Also		AP.S2CDsp.Harmonic.Ch2FundReady, AP.S2CDsp.Harmonic.Ch2FundSettling, AP.S2CDsp.Harmonic.Ch2FundTrig
Example		See example for AP.S2CDsp.Harmonic.Ch2FreqRdg.

AP.S2CDsp.Harmonic.Ch2FundReady

Property

Syntax AP.S2CDsp.Harmonic.Ch2FundReady

Data Type Integer

0 Reading not ready.
>0 Reading ready.

Description This command returns the Harmonic Distortion Analyzer channel 2 Fundamental Amplitude settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the AP.S2CDsp.Harmonic.Ch2FundRdg or AP.S2CDsp.Harmonic.Ch2FundTrig commands will zero the ready count.

If the reading is found to be ready, a call to the AP.S2CDsp.Harmonic.Ch2FundRdg command will be guaranteed to return quickly.

See Also AP.S2CDsp.Harmonic.Ch2FundRdg,
AP.S2CDsp.Harmonic.Ch2FundSettling,
AP.S2CDsp.Harmonic.Ch2FundTrig

Example See example for `AP.S2CDsp.Harmonic.Ch2FreqRdg`.

AP.S2CDsp.Harmonic.Ch2FundSettling

Method

Syntax `AP.S2CDsp.Harmonic.Ch2FundSettling`(ByVal *Tolerance* As Double, ByVal *Floor* As Double, ByVal *FloorUnit* As String, ByVal *Points* As Integer, ByVal *Delay* As Double, ByVal *Algorithm* As Integer)

Parameters See Appendix A for Settling Algorithm and parameter name descriptions.

Description This command sets the settling parameters for the `AP.S2CDsp.Harmonic.Ch2FundRdg` command.

See Also `AP.S2CDsp.Harmonic.Ch2FundRdg`,
`AP.S2CDsp.Harmonic.Ch2FundReady`,
`AP.S2CDsp.Harmonic.Ch2FundTrig`

Example See example for `AP.S2CDsp.Harmonic.Ch2FreqRdg`.

AP.S2CDsp.Harmonic.Ch2FundTrig

Method

Syntax `AP.S2CDsp.Harmonic.Ch2FundTrig`

Description Causes a restart of the reading cycle and zeros the ready count for the `AP.S2CDsp.Harmonic.Ch2FundRdg` command. The reading in progress is aborted.

See Also `AP.S2CDsp.Harmonic.Ch2FundRdg`,
`AP.S2CDsp.Harmonic.Ch2FundReady`,
`AP.S2CDsp.Harmonic.Ch2FundSettling`

Example See example for `AP.S2CDsp.Harmonic.Ch2FreqRdg`.

AP.S2CDsp.Harmonic.Ch2Source

Property

Syntax `AP.S2CDsp.Harmonic.Ch2Source`

Data Type	Integer								
	The following list contains the selections relevant to the AP.S2CDsp.Harmonic.InputFormat command A/D input selections.								
	<table> <tr> <td>0</td> <td>Anlr-A</td> </tr> <tr> <td>1</td> <td>Anlr-B</td> </tr> <tr> <td>2</td> <td>Anlr Rdg Ampl</td> </tr> <tr> <td>3</td> <td>None</td> </tr> </table>	0	Anlr-A	1	Anlr-B	2	Anlr Rdg Ampl	3	None
0	Anlr-A								
1	Anlr-B								
2	Anlr Rdg Ampl								
3	None								
	The following list contains the selections relevant to the AP.S2CDsp.Harmonic.InputFormat command Digital input selection.								
	<table> <tr> <td>0</td> <td>A</td> </tr> <tr> <td>1</td> <td>B</td> </tr> <tr> <td>2</td> <td>None</td> </tr> </table>	0	A	1	B	2	None		
0	A								
1	B								
2	None								
Description	This command sets the Harmonic Distortion Analyzer Channel 2 Input.								
See Also	AP.S2CDsp.Harmonic.InputFormat								
Example	See example for AP.S2CDsp.Harmonic.Ch2FreqRdg.								

AP.S2CDsp.Harmonic.Ch2Sum1Harmonics

Property

Syntax	AP.S2CDsp.Harmonic.Ch2Sum1Harmonics (ByVal HarmonicId As Integer)	
Data Type	Boolean	
	<i>True</i>	Add specified harmonics to RSS summation for channel 2 Sum1 reading.
	<i>False</i>	Remove specified harmonics from RSS summation for channel 2 Sum1 reading.
Parameters	Part	Description
	<i>HarmonicId</i>	Single decimal value or the sum of several decimal values representing multiple harmonics or a constant value as defined below.
		Harmonic decimal value table:

- 2 = 2
- 3 = 4
- 4 = 8
- 5 = 16
- 6 = 32
- 7 = 64
- 8 = 128
- 9 = 256
- 10 = 512
- 11 = 1024
- 12 = 2048
- 13 = 4096
- 14 = 8192
- 15 = 16384

Constants	HarmonicId Parameter	Description
	<i>apbEven</i>	Select all even harmonics.
	<i>apbOdd</i>	Select all odd harmonics.
	<i>apbAll</i>	Select all harmonics.

Description This command adds or removes individual harmonics from the Harmonic Distortion Analyzer RSS summation #1 for channel 2. Use of the optional group parameter allows addition or removal of even, odd, or all harmonics.

Example See example for `AP.S2CDsp.Harmonic.Ch2Sum1Rdg`.

AP.S2CDsp.Harmonic.Ch2Sum1Rdg Property

Syntax `AP.S2CDsp.Harmonic.Ch2Sum1Rdg(ByVal Unit As String)`

Data Type Double

Parameters	Part	Description
	<i>Unit</i>	The following units are valid for the "Absolute Units" selection of the <code>AP.S2CDsp.Harmonic.Ch2Sum1RelUnits</code> command: FFS, %FS, dBFS, Bits, V, Vp, Vpp, dBu, dBV, dBr 1, dBr 2.

The following units are valid for the "Relative Units" selection of the AP.S2CDsp.Harmonic.Ch2Sum1RelUnits command: %, dB, PPM, X/Y

Description This command returns a settled reading for the Harmonic Distortion Analyzer channel 2 Sum #1 meter and zeros the ready count.

See Also AP.S2CDsp.Harmonic.Ch2Sum1Ready,
AP.S2CDsp.Harmonic.Ch2Sum1Settling,
AP.S2CDsp.Harmonic.Ch2Sum1Trig

Example

```
Sub Main
  AP.Application.NewTest
  AP.Gen.Output = True
  AP.Anlr.ChBInput = 2
  AP.Application.PanelOpen apbPanelDSPSmall
  AP.S2CDsp.Program = 7 'Load Harmonic Distortion _
    Analyzer
  AP.Application.PanelOpen apbPanelDSPLarge
  With AP.S2CDsp.Harmonic
    .InputFormat = 1 'HiRes A/D @65k input
    .Ch2Source = 1 'Analog Analyzer Channel B _
      Input Source
    .Ch2Sum1Harmonics(2) = True 'Select harmonic #2
    .Ch2Sum2Harmonics(4) = True 'Select harmonic #3
    .Ch2Sum1RelUnits = False
    .Ch2Sum2RelUnits = False
    .Ch2Sum1Settling 0.500000, -1.200000e+002, _
      "dBV", 1, 0.002000, 2
    .Ch2Sum2Settling 0.500000, -1.200000e+002, _
      "dBV", 1, 0.002000, 2
    .Ch2Sum1Trig 'Trigger new Channel 1 reading
    .Ch2Sum2Trig 'Trigger new Channel 2 reading
  Do 'Wait for new readings
  Loop Until .Ch2Sum1Ready And .Ch2Sum2Ready
  var1 = .Ch2Sum1Rdg("dBV")
  var2 = .Ch2Sum2Rdg("dBV")
  End With
  Text1$= "Channel 2 Sum1 " & _
    Str$(Format(var1, "##.000")) & "dBV"
  Text2$= "Channel 2 Sum2 " & _
    Str$(Format(var2, "##.000")) & "dBV"
```

```

AP.Prompt.Text = Text1$ & Chr$(13) & Text2$ _
    'Text String and New Line
AP.Prompt.ShowWithContinue 'Display Prompt
Stop
End Sub

```

AP.S2CDsp.Harmonic.Ch2Sum1Ready

Property

Syntax `AP.S2CDsp.Harmonic.Ch2Sum1Ready`

Data Type Integer

0 Reading not ready.
 >0 Reading ready.

Description This command returns the Harmonic Distortion Analyzer channel 2 Sum #1 settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.Harmonic.Ch2Sum1Rdg` or `AP.S2CDsp.Harmonic.Ch2Sum1Trig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.Harmonic.Ch2Sum1Rdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.Harmonic.Ch2Sum1Rdg`,
`AP.S2CDsp.Harmonic.Ch2Sum1Settling`,
`AP.S2CDsp.Harmonic.Ch2Sum1Trig`

Example See example for `AP.S2CDsp.Harmonic.Ch2Sum1Rdg`.

AP.S2CDsp.Harmonic.Ch2Sum1RelUnits

Property

Syntax `AP.S2CDsp.Harmonic.Ch2Sum1RelUnits`

Data Type Boolean

True Relative Units.
False Absolute Units.

Description This command selects the units that are available for the
 AP.S2CDsp.Harmonic.Ch2Sum1Rdg command

Example See example for AP.S2CDsp.Harmonic.Ch2Sum1Rdg.

AP.S2CDsp.Harmonic.Ch2Sum1Settling

Method

Syntax **AP.S2CDsp.Harmonic.Ch2Sum1Settling**(ByVal
Tolerance As Double, ByVal *Floor* As Double, ByVal
FloorUnit As String, ByVal *Points* As Integer,
 ByVal *Delay* As Double, ByVal *Algorithm* As Integer)

Parameters See Appendix A for Settling Algorithm and parameter name
 descriptions.

Description This command sets the settling parameters for the
 AP.S2CDsp.Harmonic.Ch2Sum1Rdg command.

See Also AP.S2CDsp.Harmonic.Ch2Sum1Rdg,
 AP.S2CDsp.Harmonic.Ch2Sum1Ready,
 AP.S2CDsp.Harmonic.Ch2Sum1Trig

Example See example for AP.S2CDsp.Harmonic.Ch2Sum1Rdg.

AP.S2CDsp.Harmonic.Ch2Sum1Trig

Method

Syntax **AP.S2CDsp.Harmonic.Ch2Sum1Trig**

Description Causes a restart of the reading cycle and zeros the ready count for the
 AP.S2CDsp.Harmonic.Ch2Sum1Rdg command. The reading in
 progress is aborted.

See Also AP.S2CDsp.Harmonic.Ch2Sum1Rdg,
 AP.S2CDsp.Harmonic.Ch2Sum1Ready,
 AP.S2CDsp.Harmonic.Ch2Sum1Settling

Example See example for AP.S2CDsp.Harmonic.Ch2Sum1Rdg.

AP.S2CDsp.Harmonic.Ch2Sum2Harmonics**Property**

Syntax `AP.S2CDsp.Harmonic.Ch2Sum2Harmonics (ByVal HarmonicId As Integer)`

Data Type Boolean

True Add specified harmonics to RSS summation for channel 2 Sum1 reading.

False Remove specified harmonics from RSS summation for channel 2 Sum1 reading.

Parameters

Part	Description
<i>HarmonicId</i>	Single decimal value or the sum of several decimal values representing multiple harmonics or a constant value as defined below.
	Harmonic decimal value table:
	2 = 2
	3 = 4
	4 = 8
	5 = 16
	6 = 32
	7 = 64
	8 = 128
	9 = 256
	10 = 512
	11 = 1024
	12 = 2048
	13 = 4096
	14 = 8192
	15 = 16384

Constants

HarmonicId Parameter	Description
<i>apbEven</i>	Select all even harmonics.
<i>apbOdd</i>	Select all odd harmonics.
<i>apbAll</i>	Select all harmonics.

Description

This command adds or removes individual harmonics from the Harmonic Distortion Analyzer RSS summation #2 for channel 2. Use

of the optional group parameter allows addition or removal of even, odd, or all harmonics.

Example See example for `AP.S2CDsp.Harmonic.Ch2Sum1Rdg`.

AP.S2CDsp.Harmonic.Ch2Sum2Rdg

Property

Syntax `AP.S2CDsp.Harmonic.Ch2Sum2Rdg (ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
<i>Unit</i>	The following units are valid for the "Absolute Units" selection of the <code>AP.S2CDsp.Harmonic.Ch2Sum2RelUnits</code> command: FFS, %FS, dBFS, Bits, V, Vp, Vpp, dBu, dBV, dBr 1, dBr 2.
	The following units are valid for the "Relative Units" selection of the <code>AP.S2CDsp.Harmonic.Ch2Sum2RelUnits</code> command: %, dB, PPM, X/Y

Description This command returns a settled reading for the Harmonic Distortion Analyzer channel 2 Sum #2 meter and zeros the ready count.

See Also `AP.S2CDsp.Harmonic.Ch2Sum2Ready`,
`AP.S2CDsp.Harmonic.Ch2Sum2Settling`,
`AP.S2CDsp.Harmonic.Ch2Sum2Trig`

Example See example for `AP.S2CDsp.Harmonic.Ch2Sum1Rdg`.

AP.S2CDsp.Harmonic.Ch2Sum2Ready

Property

Syntax `AP.S2CDsp.Harmonic.Ch2Sum2Ready`

Data Type Integer

0 Reading not ready.

>0 Reading ready.

Description	<p>This command returns the Harmonic Distortion Analyzer channel 1 Sum #2 settled reading ready count.</p> <p>Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the <code>AP.S2CDsp.Harmonic.Ch2Sum2Rdg</code> or <code>AP.S2CDsp.Harmonic.Ch2Sum2Trig</code> commands will zero the ready count.</p> <p>If the reading is found to be ready, a call to the <code>AP.S2CDsp.Harmonic.Ch2Sum2Rdg</code> command will be guaranteed to return quickly.</p>
See Also	<p><code>AP.S2CDsp.Harmonic.Ch2Sum2Rdg</code>, <code>AP.S2CDsp.Harmonic.Ch2Sum2Settling</code>, <code>AP.S2CDsp.Harmonic.Ch2Sum2Trig</code></p>
Example	<p>See example for <code>AP.S2CDsp.Harmonic.Ch2Sum1Rdg</code>.</p>

AP.S2CDsp.Harmonic.Ch2Sum2RelUnits

Property

Syntax	<code>AP.S2CDsp.Harmonic.Ch2Sum2RelUnits</code>				
Data Type	Boolean				
	<table> <tr> <td><i>True</i></td> <td>Relative Units</td> </tr> <tr> <td><i>False</i></td> <td>Absolute Units</td> </tr> </table>	<i>True</i>	Relative Units	<i>False</i>	Absolute Units
<i>True</i>	Relative Units				
<i>False</i>	Absolute Units				
Description	<p>This command selects the units that are available for the <code>AP.S2CDsp.Harmonic.Ch2Sum2Rdg</code> command</p>				
Example	<p>See example for <code>AP.S2CDsp.Harmonic.Ch2Sum1Rdg</code>.</p>				

AP.S2CDsp.Harmonic.Ch2Sum2Settling

Method

Syntax	<pre>AP.S2CDsp.Harmonic.Ch2Sum2Settling(ByVal Tolerance As Double, ByVal Floor As Double, ByVal FloorUnit As String, ByVal Points As Integer, ByVal Delay As Double, ByVal Algorithm As Integer)</pre>
---------------	--

Parameters	See Appendix A for Settling Algorithm and parameter name descriptions.
Description	This command sets the settling parameters for the <code>AP.S2CDsp.Harmonic.Ch2Sum2Rdg</code> command.
See Also	<code>AP.S2CDsp.Harmonic.Ch2Sum2Rdg</code> , <code>AP.S2CDsp.Harmonic.Ch2Sum2Ready</code> , <code>AP.S2CDsp.Harmonic.Ch2Sum2Trig</code>
Example	See example for <code>AP.S2CDsp.Harmonic.Ch2Sum1Rdg</code> .

AP.S2CDsp.Harmonic.Ch2Sum2Trig

Method

Syntax	<code>AP.S2CDsp.Harmonic.Ch2Sum2Trig</code>
Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP.S2CDsp.Harmonic.Ch2Sum2Rdg</code> command. The reading in progress is aborted.
See Also	<code>AP.S2CDsp.Harmonic.Ch2Sum2Rdg</code> , <code>AP.S2CDsp.Harmonic.Ch2Sum2Ready</code> , <code>AP.S2CDsp.Harmonic.Ch2Sum2Settling</code>
Example	See example for <code>AP.S2CDsp.Harmonic.Ch2Sum1Rdg</code> .

AP.S2CDsp.Harmonic.Freq

Property

Syntax	<code>AP.Gen.ChAFreq(ByVal <i>Unit</i> As String)</code>	
Data Type	Double	
Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz, F/R, dHz, %Hz, cent, octs, decs, d%, dPPM
Description	This command sets the Harmonic Distortion Analyzer tuning frequency.	
See Also	<code>AP.S2CDsp.Harmonic.Tuning</code>	

Example

```

Sub Main
  AP.Application.NewTest
  AP.Gen.Output = True
  AP.Anlr.ChAInput = 2
  AP.Application.PanelOpen apbPanelDSPSmall
  AP.S2CDsp.Program = 7 'Load Harmonic Distortion _
    Analyzer
  AP.S2CDsp.Harmonic.Ch1Sum1Harmonics(2) = True
  'Measure second harmonic only
AP.S2CDsp.Harmonic.Tuning = 4 'Set Tuning mode to Fixed
AP.S2CDsp.Harmonic.Freq("Hz") = 1000.0 'Set Tuning _
  Frequencu to 1kHz
AP.S2CDsp.Harmonic.Selectivity = 0 'Hi-Speed _
  measurement mode
  AP.Sweep.Data1.Id = 6129
  AP.Sweep.Data1.Top("FFS") = 1.0
  AP.Sweep.Data1.Bottom("FFS") = 0.0
  AP.Sweep.Source1.Start("Hz") = 1000.0
  AP.Sweep.Source1.Stop("Hz") = 3000.0
  AP.Sweep.Source1.LogLin = 1
  AP.Sweep.Source1.Steps = 100
  AP.Sweep.Start
AP.S2CDsp.Harmonic.Selectivity = 1 'Hi-Accuracy _
  measurement mode
  AP.Sweep.Append = True
  AP.Sweep.Start
  AP.Graph.Comment = "Hi-Speed and Hi-Accuracy _
    measurement bandwidth."
End Sub

```

AP.S2CDsp.Harmonic.InputFormat**Property**

Syntax	AP.S2CDsp.Harmonic.InputFormat
Data Type	Integer
	0 Digital @ ISR
	1 HiRes A/D @65536
	2 HiBW A/D @131072

3	HiBW A/D @262144
4	HiRes A/D @OSR
5	HiBW A/D @2xOSR

Description This command sets the Harmonic Distortion Analyzer Input Format.

Example See example for `AP.S2CDsp.Harmonic.Ch1FreqRdg`.

AP.S2CDsp.Harmonic.Selectivity

Property

Syntax `AP.S2CDsp.Harmonic.Selectivity`

Data Type Integer

0	Hi-Speed: In this mode measurement selectivity is reduced to improve measurement speed. Rapid readings of changing conditions are best done in this mode.
1	Hi-Accuracy: In this mode measurement selectivity is increased to reduce the noise contribution to the measurement. Frequency and amplitude accuracy is improved.

Description This command sets the Harmonic Distortion Analyzer measurement mode.

Example See example for `AP.S2CDsp.Harmonic.Freq`.

AP.S2CDsp.Harmonic.Tuning

Property

Syntax `AP.S2CDsp.Harmonic.Tuning`

Data Type Integer

0	Counter Tuned: the frequency value measured by the Harmonic Distortion Analyzer Frequency counter is the filter steering source. This function would be selected when making measurements from an external signal such as reproduction of a Compact Disc or digital audio tape or reception of a digital signal from a distant source.
---	--

- 1 Sweep Track: the filter tracks the frequency of whichever generator is selected in the Source 1 or Source 2 fields of the Sweep panel.
- 2 AGen Track: harmonic measurements track the frequency of the Analog Generator. This mode is useful for testing A/D converters driven from an analog output.
- 3 DGen Track: harmonic measurements track the frequency of the Digital Generator. This mode would normally be used when sweeping digital input- digital output devices with stimulus coming from the Digital Generator.
- 4 Fixed: harmonic measurements are fixed, based on the frequency entered in the Freq field, unless the filter is being deliberately varied as part of a sweep test.

Description This command sets the Harmonic Distortion Analyzer Tuning Source.

Example See example for `AP.S2CDsp.Harmonic.Freq`.

User Notes

Digital Interface Analyzer

AP.S2CDsp.Intervu.AmplVsTime

Property

Syntax
`AP.S2CDsp.Intervu.AmplVsTime`
Data Type

Integer

- | | |
|----------|--|
| <i>0</i> | Interpolate: the DSP module will perform an interpolation calculation based on the fact that the signal was band-limited by an internal 30 MHz low-pass filter before sampling. |
| <i>1</i> | Display Samples: no processing takes place in the DSP module. At each time value plotted on the X-axis, the DSP simply sends the amplitude of the nearest-in-time acquired sample of the digital interface waveform to the computer for plotting. When displaying only a few pulses of the digital interface waveform, this is typically the best mode to use. |
| <i>2</i> | Peak Values: the DSP searches all sample amplitudes in the acquisition buffer between each pair of horizontal axis time values plotted and sends to the computer for plotting the largest positive or negative value in that span, preserving the plus or minus sign. The intended use of the Peak Values mode is when graphing pulse width histograms or a relatively long time span on the X-axis, where the combination of Start-to-Stop time span and Steps value on the Sweep panel results in skipping across many actual acquired samples between plotted points. If Peak Values mode is not used, an unfortunate combination of signal, X-axis span, and Points value can make it appear that no waveform, a near-DC signal, or a waveform at a completely different frequency is present. Since Peak Values searches through all sample values within each span between plotted points and sends the largest value to be plotted, signals cannot be missed. |
| <i>3</i> | Eye Pattern: Following acquisition of the digital interface signal and extraction of an average clock signal from it, the worst-case (nearest to zero Volts) amplitude is determined for each time increment relative to the beginning of each data cell. These values are plotted when Upper Eye Opening and |

Lower Eye Opening are selected as Data parameters, resulting in a plot of the worst-case inside of the eye.

Description

This command provides four modes for processing the amplitude-versus-time relationship of a sampled digital interface signal before displaying the waveform. These modes are applicable to digital storage oscilloscope operation (amplitude versus time graphs) and histograms, but have no effect on FFT spectrum analysis.

Example

```
Sub Main
  AP.File.OpenTest "INTERVU1.AT2C"
  With AP.S2CDsp.Intervu
    .AmplVsTime = 0      'Set to Interpolate
    .AudioMonitor = 0   'Set Audio Monitor
    .JitterDetection = 0 'Set Jitter Detection to_
                        Stable Bits
    .TrigSource = 3     'Set Trigger Ch B Transmit _
                        Preamble
    .Window = 0        'Set Blackman-Harris Window
  End With
  AP.Sweep.Start
End Sub
```

AP.S2CDsp.Intervu.AudioMonitor**Property****Syntax**

AP.S2CDsp.Intervu.AudioMonitor

Data Type

Integer

0 Audio Monitor: Monitor the imbedded digital audio signal.
1 Jitter Signal: Monitor the demodulated jitter signal.

Description

This command determines the audio signal that is sent to the headphone output from the Digital Interface Analyzer.

See Also

AP.Speaker.Mode, AP.Speaker.Source

Example

See example for AP.S2CDsp.Intervu.AmplVsTime.

AP.S2CDsp.Intervu.Averages**Property****Syntax** **AP.S2CDsp.Intervu.Averages****Data Type** Integer

0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

Description This command sets the Digital Interface Analyzer number of acquisition-and-processing cycles to average.

Example Sub Main

```

AP.Application.NewTest
AP.Application.PanelClose apbPanelAnalogGenSmall
AP.Application.PanelClose apbPanelAnlrSmall
AP.DGen.Output = True
AP.S2CDsp.Program = 3     'Digital Interface Analyzer
AP.S2CDsp.Intervu.Averages = 4     '16 Averages
AP.S2CDsp.Intervu.Window = 4     'No Window
AP.Sweep.Data1.Id = 6055
AP.Sweep.Source1.Id = 5613
AP.S2CDio.OutJitterType = 1         'Sine Jitter
AP.S2CDio.OutJitterAmpl("sec") = 100e-9
AP.S2CDio.InFormat = 3
AP.Sweep.Start
AP.Graph.OptimizeLeft
End Sub

```


AP.S2CDsp.Intervu.JitterDetection

Property

Syntax	<code>AP.S2CDsp.Intervu.JitterDetection</code>	
Data Type	Integer	
	<i>0</i>	<p>Stable Bits: derives the stable reference clock at 1/4 the actual cell (bit) rate, synchronized to the beginning transition of the preamble. The serial signal consists of 32 cells (bits) per subframe and two subframes (left and right channels) per frame. The frame rate is equal to the sample rate of the embedded audio. Thus, there are 64 cells (bits) in a complete frame and the cell rate is 1/64 the audio sample rate. The first four cells of each subframe are the preamble. The preamble always starts with a three UI (1 1/2 cell) wide pulse followed by sequences of one UI, two UI, and three UI pulses which are different among the three possible preambles. There is no cell transition time within the preamble which is common to all three preambles. The highest rate at which transitions can be guaranteed to occur regularly is at 1/4 the cell rate, which includes the beginning and end of each preamble but no transitions within the preamble. This rate is 16 times the audio sample rate, so the effective jitter measurement bandwidth is eight times the audio sample rate (384 kHz at a 48 kHz sample rate).</p>
	<i>1</i>	<p>All Bits: derives the stable reference clock at the actual cell (bit) rate. Since there are 64 cells per frame and the frame rate is the audio sample rate, the reference clock is at 64 times the sample rate and the effective jitter measurement bandwidth is 32 times the audio sample rate (1.536 MHz at a 48 kHz sample rate). Since the preamble of each sub-frame will not have transitions at every cell boundary due to its three-UI-wide pulses (violations of bi-phase coding), the DSP interpolates where transitions would have occurred if the preamble did not violate bi-phase coding.</p>
	<i>2</i>	<p>Preambles: the average rate of the trailing edge of the first three-UI-wide pulse in each preamble as the stable clock reference. Each actual transition at the trailing edge of the first three-UI-wide preamble pulse is then compared to that reference (average value) to obtain jitter values for display as</p>

3

jitter waveform, histogram of jitter, or FFT spectrum analysis of jitter. The three-UI pulse in a preamble is the most robust portion of the digital interface signal, since it is least affected by reduced bandwidth in the cable or system. Therefore, jitter measurements made with the Preamble Jitter Detection selection tend to be measurements of the intrinsic jitter in the transmitting device clock and are relatively unaffected by data jitter caused by reduced bandwidth. Since this derived reference clock rate is low (twice the audio sample rate), the effective jitter measurement bandwidth equals the audio sample rate when Preamble is selected.

Squarewave Rising: In addition to measuring jitter on an AES/EBU or SPDIF/EIAJ serial digital input signal, INTERVU can also measure jitter on any 28 kHz-13 MHz squarewave connected to the BNC digital input connector. This feature permits direct measurement of clock jitter on A/D and D/A converters.

The Squarewave Rising selection measures jitter on rising edges of the Squarewave signal.

Jitter is a measurement of the time deviation of zero crossings of a waveform compared to a reference perfect clock of the same average frequency. For AES/EBU and SPDIF/EIAJ waveforms, System Two determines the average clock frequency by measuring the frame rate of the serial digital input signal with a frequency counter. This frame frequency extraction circuitry is not functional for a squarewave signal, so the DIO panel Sample Rate field is not useful with squarewave input. INTERVU determines the average clock frequency to the best of its ability from its acquired signal.

Since the acquired signal duration is approximately four milliseconds, the resulting frequency measurement is limited in resolution. The result is that the initial time domain graph of jitter of a squarewave input clock, plotted across the approximately four millisecond record duration, may appear as a ramp. The desired jitter signal is the deviation from this ramp. The Compute Linearity function is used to extract variations from an underlying systematic variation such as this ramp.

4 Squarewave Falling: The Squarewave Falling selection measures jitter on falling edges of the Squarewave signal. Note: See additional text in Squarewave Rising selection above.

Description This command determines at which transitions the clock timing is compared to the interface signal..

Example See example for `AP.S2CDsp.Intervu.AmplVsTime`.

AP.S2CDsp.Intervu.TrigErrorCoding

Property

Syntax `AP.S2CDsp.Intervu.TrigErrorCoding`

Data Type Boolean

True Enable Data Acquisition trigger generation on a Coding error.
False Disable Data Acquisition trigger generation on a Coding error.

Description This command enables or disables generation of a Data Acquisition trigger for a Coding Error.

Coding indicates a deviation from proper biphase coding in the input serial stream (ignoring preambles). Proper biphase signals can never remain at a logic high or logic low level for more than two consecutive Unit Intervals (UI) except in the preamble. The preamble deliberately deviates from biphase coding in order to provide a unique frame synchronization signal, so preambles are excluded from the function of the Coding indicators.

Example See example for `AP.S2CDsp.Intervu.TrigErrorParity`.

AP.S2CDsp.Intervu.TrigErrorConfidence

Property

Syntax `AP.S2CDsp.Intervu.TrigErrorConfidence`

Data Type Boolean

True Enable Data Acquisition trigger generation on a Confidence error.

False Disable Data Acquisition trigger generation on a Confidence error.

Description

This command enables or disables generation of a Data Acquisition trigger for a Confidence Error.

The Confidence error occurs when the ratio between the amplitude of the three UI long pulse and the following one UI-long pulse in a preamble becomes large enough to cause an increasing probability of errors when slicing the received signal into logic high and low values. This large ratio occurs when the transmission bandwidth has been reduced to marginal or unacceptable values. Under these conditions, selection of hardware input equalization (XLR with EQ or BNC with EQ rather than XLR or BNC selections of the Input Format field) will often compensate for the cable bandwidth reduction, and provide reliable measurements.

Example

See example for `AP.S2CDsp.Intervu.TrigErrorParity`.

AP.S2CDsp.Intervu.TrigErrorLock**Property****Syntax**

`AP.S2CDsp.Intervu.TrigErrorLock`

Data Type

Boolean

True Enable Data Acquisition trigger generation on a Lock error.

False Disable Data Acquisition trigger generation on a Lock error.

Description

This command enables or disables generation of a Data Acquisition trigger for a Lock Error.

The Lock error occurs when the digital input phase-locked loop is unable to lock to the incoming signal.

Example

See example for `AP.S2CDsp.Intervu.TrigErrorParity`.

AP.S2CDsp.Intervu.TrigErrorParity**Property****Syntax**

`AP.S2CDsp.Intervu.TrigErrorParity`

Data Type

Boolean

True Enable Data Acquisition trigger generation on a Parity error.
False Disable Data Acquisition trigger generation on a Parity error.

Description

This command enables or disables generation of a Data Acquisition trigger for a Parity Error.

The Parity error indicates a parity error in either subframe. Correct parity is determined by comparing the P (parity) bit with the sum of the remaining 31 bits in each subframe. Any single bit error or odd number of bit errors introduced in transmission within a subframe will cause a Parity error indication, but even numbers of bit errors cannot be detected by this technique.

Example

```
Sub Main
  AP.Application.NewTest
  AP.Application.PanelClose apbPanelAnalogGenSmall
  AP.Application.PanelClose apbPanelAnlrSmall
  AP.S2CDsp.Program = 3
  'Configure Digital Interface Analyzer to
  AP.S2CDsp.Intervu.TrigSource = 4     ' trigger on _
      an error
  AP.S2CDsp.Intervu.TrigPolarity = 0   ' trigger on _
      positive polarity
  AP.S2CDsp.Intervu.AcquisitionPosition = 1 ' retain _
      acquire data before the error
  AP.S2CDsp.Intervu.TrigErrorConfidence = False
  AP.S2CDsp.Intervu.TrigErrorCoding = False
  AP.S2CDsp.Intervu.TrigErrorLock = False
  AP.S2CDsp.Intervu.TrigErrorParity = True   ' trigger _
      on a Parity Error.

  AP.S2CDio.InFormat = 3

  AP.Sweep.Data1.Id = 6053
  AP.Sweep.Source1.Id = 5612
  AP.Sweep.Source1.Start("sec") = -5.0e-006
  AP.Sweep.Source1.Stop("sec") = 5.0e-006

  AP.Application.SetWatchDogTimer1(5.0,False) 'Fire _
      timer to create Parity Error in 5 seconds.
  AP.Sweep.StartNoWait
  AP.Graph.Comment = "Wait for Parity Error to occur."
```

```

'The Sweep will proceed automatically when the Parity _
  Error occurs.

  Do
    Loop While AP.S2CDio.FlagParityRdg = False
'Wait here for Parity Error to be detected on the _
  Digital Input/Output panel.

  AP.Graph.Comment = "Parity Error detected and _
    waveform display updated."
  AP.Graph.OptimizeLeft

End Sub
Sub APEvent_OnWatchDogTimeout(ByVal Id As Long)
  If Id = 1 Then
    AP.S2CDio.OutParityError = True
  End If
End Sub

```

AP.S2CDsp.Intervu.TrigPolarity

Property

Syntax	<code>AP.S2CDsp.Intervu.TrigPolarity</code>
Data Type	Integer
	<p><i>0</i> Positive: The data acquisition will begin on the first positive-going zero crossing of the signal selected in the Trigger Source field.</p> <p><i>1</i> Negative: The data acquisition will begin on the first negative-going zero crossing of the selected signal selected in the Trigger Source field.</p>
Description	This command sets the Digital Interface Analyzer Trigger Polarity for the for the Trigger Source field Jitter Generator, External, Common Mode Signal, Interfering Noise, Digital Generator, and Input Zero Crossing selections.
See Also	<code>AP.S2CDsp.Intervu.TrigSource</code>
Example	See example for <code>AP.S2CDsp.Intervu.TrigErrorParity</code> .

AP.S2CDsp.Intervu.TrigSource**Property****Syntax** `AP.S2CDsp.Intervu.TrigSource`**Data Type**

Integer

0

Ch. A Receive Preamble: cause signal to be acquired at the first Channel A (left) Preamble which occurs after Go is clicked or the F9 function key is pressed. The Channel A Preamble is known as the X Preamble in the AES/EBU standard and the M Preamble in the Consumer standard. The first information acquired will be the last four Unit Intervals of the selected preamble, followed by the LSB of the audio signal if full 24-bit resolution audio is transmitted, or the beginning of the 4-bit Auxiliary data if audio is restricted to 20 bits or less.

1

Ch. A Transmit Preamble: cause signal to be acquired beginning at the start of the first Channel A Preamble which is transmitted from System Two after the `AP.Sweep.Start` command is executed. The first information acquired includes the entire preamble, followed by audio or Auxiliary data. This triggering selection permits measurement of time delay through a digital device or system under test.

2

Ch. B Receive Preamble: cause signal to be acquired at the first Channel B (right) Preamble which occurs after Go is clicked or the F9 function key is pressed. The Channel B Preamble is known as the Y Preamble (AES/EBU) or W Preamble (consumer). The first information acquired will be the last four Unit Intervals of the selected preamble, followed by the LSB of the audio signal if full 24-bit resolution audio is transmitted, or the beginning of the 4-bit Auxiliary data if audio is restricted to 20 bits or less.

3

Ch. B Transmit Preamble: cause signal to be acquired beginning at the start of the first Channel B Preamble which is transmitted from System Two after `AP.Sweep.Start` command is executed. The first information acquired includes the entire preamble, followed by audio or Auxiliary data. This triggering selection permits measurement of time delay through a digital device or system under test.

- 4 Receive Error: selection is a pre-trigger, causing the 256k samples (about 3.9 milliseconds) immediately preceding an interface Error Flag to be retained (approximately 39 microseconds of signal following the occurrence of the error will also be retained. The interface Error Flags are generated by the AES/EBU receiver chip of the DIO, and their status is indicated by the Parity, Coding, Lock, or Confidence indicators at the right of the DIO panel. If this acquisition trigger selection is in use and a Parity error, Coding error, Lock error, or Confidence error occurs, the last (approximately) 3.9 milliseconds of interface signal preceding the error will be retained in the INTERVU buffer for examination via waveform display, spectrum analysis, or probability histograms. The Invalid indicator is not considered an interface error and thus will not result in an acquisition into INTERVU.
- 5 Receive Block: causes signal to be acquired beginning at the end of the first Channel Status Block Preamble received after Go is clicked or the F9 function key is pressed. This is known as the Z Preamble in the AES/EBU standard and the B Preamble in the Consumer standard. The first information displayed will be the last four UIs of the Z preamble, followed by the LSB of the Channel A audio signal if full 24-bit resolution audio is transmitted, or the beginning of the 4-bit Auxiliary data if audio is restricted to 20 bits or less, of the frame which marks the beginning of a new Channel Status Block. Channel Status Blocks are 192 frames long, with the C (Channel Status) bit of each of these 192 frames being assembled into the 24 Channel Status Bytes defined in the AES/EBU and Consumer standards.
- 6 Jitter Generator: causes a trigger at every zero crossing of the sinewave, squarewave, or noise signal generated by the DIO jitter generator. This selection provides a stable display of the received jitter waveform when measuring jitter gain or loss through a digital device.
- 7 External:
- 8 Common Mode Signal:
- 9 Interfering Noise:
- 10 Digital Generator:
- 11 Transmit Block:

12	Ch.A Sync Preamble:
13	Ch.B Sync Preamble:
14	Sync Error:
15	Sync Block:
16	Input ZeroCrossing:

Description This command defines the trigger source that is used to trigger an acquisition.

Example See example for `AP.S2CDsp.Intervu.AmplVsTime`.

AP.S2CDsp.Intervu.Window

Property

Syntax `AP.S2CDsp.Intervu.Window`

Data Type Integer

0	Blackman-Harris
1	Hann
2	Flat-Top
3	Equiripple
4	None

Description This command sets the Digital Interface Analyzer Window selection. See Appendix C for FFT Window Discriptions.

Example See example for `AP.S2CDsp.Intervu.AmplVsTime`.

User Notes

User Notes

Quasi-Anechoic Acoustical Tester

AP.S2CDsp.MLS.Ch1Rdg

Property

Syntax `AP.S2CDsp.MLS.Ch1Rdg(ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, Bits.

Description

This command returns a unsettled reading for the Quasi-Anechoic Acoustical Tester channel 1 Peak Monitor meter and zeros the ready count.

See Also

AP.S2CDsp.MLS.Ch1Ready, AP.S2CDsp.MLS.Ch1Trig

Example

```
Sub Main
    AP.Application.NewTest
    AP.Gen.Wfm 7, 0
    AP.Gen.Output = True
    AP.Anlr.ChAInput = 2
    AP.Anlr.ChBInput = 2
    With AP.S2Dsp
        .Program = 5
        .Mls.InputFormat = 1           'HiRes A/D @65k input
        .Mls.Ch1Source = 0             'Analyzer A input
        .Mls.Ch2Source = 1             'Analyzer B input
    End With

    Wait 0.5
    .Mls.Ch1Trig                       'Trigger Ch 1 reading
    .Mls.Ch2Trig                       'Trigger Ch 2 reading
    Do
        Ready1 = .Mls.Ch1Ready         'Check status
        Ready2 = .Mls.Ch2Ready         'Ccheck status
    Loop Until Ready1 > 0 And Ready2 > 0

    Reading1 = .Mls.Ch1Rdg("FFS")     'Get reading
    Reading2 = .Mls.Ch2Rdg("FFS")     'Get reading
```

```

End With
NewLine$ = Chr(13)
a$= "Ch1 Peak Mon "+Left(Str$(Reading1),6)+" FFS"
b$= "Ch2 Peak Mon "+Left(Str$(Reading2),6)+" FFS"
AP.Prompt.Text = a$ + NewLine$ + b$ + NewLine
AP.Prompt.ShowWithContinue
Stop
End Sub

```

AP.S2CDsp.MLS.Ch1Ready

Property

Syntax `AP.S2CDsp.MLS.Ch1Ready`

Data Type Integer

0 Reading not ready.
 >0 Reading ready.

Description This command returns the Quasi-Anechoic Acoustical Tester channel 1 Peak Monitor meter unsettled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.MLS.Ch1Rdg` or `AP.S2CDsp.MLS.Ch1Trig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.MLS.Ch1Rdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.MLS.Ch1Rdg`, `AP.S2CDsp.MLS.Ch1Trig`

Example See example for `AP.S2CDsp.MLS.Ch1Rdg`.

AP.S2CDsp.MLS.Ch1Source**Property****Syntax** `AP.S2CDsp.MLS.Ch1Source`**Data Type** Integer

The following list contains the selections relevant to the `AP.S2CDsp.MLS.InputFormat` command A/D input selection.

0	Anlr-A
1	Anlr-B
2	Anlr Reading Ampl
3	Anlr Reading Ratio
4	Ch. A Generator
5	Ch. B Generator
6	Jitter Signal (UI)
7	None
8	Jitter Signal (sec)

The following list contains the selections relevant to the `AP.S2CDsp.MLS.InputFormat` command Digital input selection.

0	A
1	B
2	None

Description This command sets the Quasi-Anechoic Acoustical Tester Channel 1 Input.**Example** See example for `AP.S2CDsp.MLS.Ch1Rdg`.**AP.S2CDsp.MLS.Ch1Trig****Method****Syntax** `AP.S2CDsp.MLS.Ch1Trig`**Description** Causes a restart of the reading cycle and zeros the ready count for the `AP.S2CDsp.MLS.Ch1Rdg` command. The reading in progress is aborted.

See Also `AP.S2CDsp.MLS.Ch1Rdg`, `AP.S2CDsp.MLS.Ch1Ready`

Example See example for `AP.S2CDsp.MLS.Ch1Rdg`.

AP.S2CDsp.MLS.Ch2Rdg

Property

Syntax `AP.S2CDsp.MLS.Ch2Rdg(ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dBFS, Bits.

Description This command returns a unsettled reading for the Quasi-Anechoic Acoustical Tester channel 2 Peak Monitor meter and zeros the ready count.

See Also `AP.S2CDsp.MLS.Ch2Ready`, `AP.S2CDsp.MLS.Ch2Trig`

Example See example for `AP.S2CDsp.MLS.Ch1Rdg`.

AP.S2CDsp.MLS.Ch2Ready

Property

Syntax `AP.S2CDsp.MLS.Ch2Ready`

Data Type Integer

<i>0</i>	Reading not ready.
<i>>0</i>	Reading ready.

Description This command returns the Quasi-Anechoic Acoustical Tester channel 2 Peak Monitor meter unsettled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.S2CDsp.MLS.Ch2Rdg` or `AP.S2CDsp.MLS.Ch2Trig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2CDsp.MLS.Ch2Rdg` command will be guaranteed to return quickly.

See Also `AP.S2CDsp.MLS.Ch2Rdg`, `AP.S2CDsp.MLS.Ch2Trig`

Example See example for `AP.S2CDsp.MLS.Ch1Rdg`.

AP.S2CDsp.MLS.Ch2Source

Property

Syntax `AP.S2CDsp.MLS.Ch2Source`

Data Type Integer

The following list contains the selections relevant to the `AP.S2CDsp.MLS.InputFormat` command A/D input selection.

0	Anlr-A
1	Anlr-B
2	Anlr Reading Ampl
3	Anlr Reading Ratio
4	Ch. A Generator
5	Ch. B Generator
6	Jitter Signal (UI)
7	None
8	Jitter Signal (sec)

The following list contains the selections relevant to the `AP.S2CDsp.MLS.InputFormat` command Digital input selection.

0	A
1	B
2	None

Description This command sets the Quasi-Anechoic Acoustical Tester Channel 2 Input.

Example See example for `AP.S2CDsp.MLS.Ch1Rdg`.

AP.S2CDsp.MLS.Ch2Trig

Method

Syntax	<code>AP.S2CDsp.MLS.Ch2Trig</code>
Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP.S2CDsp.MLS.Ch2Rdg</code> comand. The reading in progress is aborted.
See Also	<code>AP.S2CDsp.MLS.Ch2Rdg</code> , <code>AP.S2CDsp.MLS.Ch2Ready</code>
Example	See example for <code>AP.S2CDsp.MLS.Ch1Rdg</code> .

AP.S2CDsp.MLS.InputFormat

Property

Syntax	<code>AP.S2CDsp.MLS.InputFormat</code>												
Data Type	Integer												
	<table> <tr> <td><code>0</code></td> <td>Digital @ ISR:</td> </tr> <tr> <td><code>1</code></td> <td>HiRes A/D @65536</td> </tr> <tr> <td><code>2</code></td> <td>HiBW A/D @131072</td> </tr> <tr> <td><code>3</code></td> <td>HiBW A/D @262144</td> </tr> <tr> <td><code>4</code></td> <td>HiRes A/D @OSR</td> </tr> <tr> <td><code>5</code></td> <td>HiBW A/D @2xOSR</td> </tr> </table>	<code>0</code>	Digital @ ISR:	<code>1</code>	HiRes A/D @65536	<code>2</code>	HiBW A/D @131072	<code>3</code>	HiBW A/D @262144	<code>4</code>	HiRes A/D @OSR	<code>5</code>	HiBW A/D @2xOSR
<code>0</code>	Digital @ ISR:												
<code>1</code>	HiRes A/D @65536												
<code>2</code>	HiBW A/D @131072												
<code>3</code>	HiBW A/D @262144												
<code>4</code>	HiRes A/D @OSR												
<code>5</code>	HiBW A/D @2xOSR												
Description	This command sets the Quasi-Anechoic Acoustical Tester Input Format.												
Example	See example for <code>AP.S2CDsp.MLS.Ch1Rdg</code> .												

AP.S2CDsp.MLS.TimeDelay

Property

Syntax	<code>AP.S2CDsp.MLS.TimeDelay(ByVal Unit As String)</code>				
Data Type	Double				
Parameters	<table> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><code>Unit</code></td> <td>String that designates the desired unit. The following unit is valid for this command: sec</td> </tr> </tbody> </table>	Name	Description	<code>Unit</code>	String that designates the desired unit. The following unit is valid for this command: sec
Name	Description				
<code>Unit</code>	String that designates the desired unit. The following unit is valid for this command: sec				
Description	This command sets the Quasi-Anechoic Acoustical Tester Time Delay.				

The Time Delay field is used to tell the DSP the distance from the speaker under test to the measurement microphone as a reference for the phase measurements. This information allows the DSP to subtract out the transit time delay from the phase readings. As the Time Delay value is adjusted the phase response will slope up or down reflecting the constant time delay component of the data. The initial value of Time Delay may be estimated from a measurement of the distance between loudspeaker and microphone. The proper final Time Delay value may be determined experimentally as the peak amplitude on a time domain graph or to obtain the smallest slope on phase.

Example

Sub Main

```

AP.Prompt.FontSize = 8 'Set font size to 8 point.
AP.Prompt.Position(-1,-1,220,130) 'Set location and _
    size.
AP.Application.NewTest
AP.Gen.Wfm 7, 0
AP.Gen.Output = True
AP.Anlr.ChAInput = 2
AP.Anlr.ChBInput = 2
AP.Application.PanelOpen apbPanelDSPSmall
AP.S2CDsp.Program = 5
AP.S2CDsp.Mls.InputFormat = 1
AP.S2CDsp.Mls.TimeDisplay = 1 'Energy-Time display
AP.S2CDsp.Mls.WindowETime = 1 'Select Half Hann _
    Energy-Time Window
AP.S2CDsp.Mls.TrigSource = 1 'Analog Generator
AP.S2CDsp.Mls.WfmDisplay = 0 'Interpolate
AP.Application.PanelOpen apbPanelSweepSmall
AP.Sweep.Data1.Id = 6326
AP.Sweep.Source1.Id = 5582
AP.Sweep.Stereo = True
AP.Sweep.Source1.Stop("sec") = 0.005
AP.Sweep.Start 'Run Sweep
AP.Graph.OptimizeLeft 'Optimize display _
    for Data 1

AP.Prompt.Text = "Energy-Time Response."
AP.Prompt.ShowWithContinue 'Display prompt with _
    Continue button.
Stop 'Stop macro.

```

```

AP.Sweep.Stereo = False
AP.Sweep.Source1.Id = 5581      'Amplitude
AP.Sweep.Data1.Top("dBV") = 26.020600
AP.Sweep.Source1.LogLin = 0
AP.Sweep.Data2.Id = 6046      'Phase
AP.Sweep.Stereo = True
AP.S2CDsp.Mls.TimeDelay("sec") = 38.74e-6
AP.S2CDsp.Mls.WindowStart = 0 'None
AP.S2CDsp.Mls.WindowStop = 0 'None
AP.Sweep.Retransform          'Retransform FFT
AP.Graph.OptimizeIndividually 'Optimize display

AP.Prompt.Text = "Frequency and Phase Response."
AP.Prompt.ShowWithContinue   'Display prompt with _
    Continue button.

    Stop                      'Stop macro.
End Sub

```

AP.S2CDsp.MLS.TimeDisplay

Property

Syntax `AP.S2CDsp.MLS.TimeDisplay`

Data Type Integer

0 Impulse Response: will show the results of the MLS correlation which is the actual impulse response of the device under test.

1 Energy-Time: will display what is commonly called an energy-time curve. The energy-time curve computation process involves transforming the impulse response to the frequency domain, doing further processing in the frequency domain, and transforming the result back to the time domain. A frequency window may be used for the conversion from frequency domain back to time domain. The frequency window is selected in the Energy-Time Window field.

Description This command sets the Quasi-Anechoic Acoustical Tester Time Domain Display type.

Example See example for `AP.S2CDsp.Mls.TimeDelay`.

AP.S2CDsp.MLS.TrigSource

Property

Syntax	<code>AP.S2CDsp.MLS.TrigSource</code>
Data Type	Integer
	0 Analog Generator
	1 Digital Generator
Description	This command sets the Quasi-Anechoic Acoustical Tester Trigger Source.
Example	See example for <code>AP.S2CDsp.Mls.TimeDelay</code> .

AP.S2CDsp.MLS.WfmDisplay

Property

Syntax	<code>AP.S2CDsp.MLS.WfmDisplay</code>
Data Type	Integer
	0 Interpolate
	1 Display Samples
	2 Peak Values
Description	This command sets the Quasi-Anechoic Acoustical Tester waveform display mode.

When Interpolate is selected, the DSP will compute the data value, interpolated from the nearby measured values. This smooths out the stair-step appearance of frequency response curves at low frequencies with a Log horizontal axis, where the bin width (usually 2.93 Hz at the 48 kHz sample rate) occupies a significant portion of the screen.

When Display Samples is selected, the DSP will return the closest actual measured value without altering the data. Normal is the recommended display mode for frequency response data with a Linear horizontal axis or with a Log axis above 100 to 300 Hz. In these cases, the jagged lines caused by the FFT bin width are not usually noticeable.

When Peak Values is selected, The Peak mode will return the largest value between the last requested sweep point and the current one. Peak is recommended for time domain MLS displays (Impulse

Response and Energy-Time). Peak mode would not normally be used for frequency response displays with MLS.AZ1, since high values are of no more interest than low values when plotting frequency response.

Example

See example for `AP.S2CDsp.Mls.TimeDelay`.

AP.S2CDsp.MLS.WindowETime**Property****Syntax**

`AP.S2CDsp.MLS.WindowETime`

Data Type

Integer

0

No Window: will perform the required transformations with all frequency components of the signal included in the computations.

1

Half Hann: reduces the contribution of high frequencies. The low frequency information remains unchanged. When operating at the 48 kHz sample rate this window filters out energy above 12 kHz.

2

Hann: reduces both high and low frequency energy, concentrating on arrivals at the center of the frequency range. Since the processing occurs on a linear frequency scale, this will focus analysis on signals around one quarter of the sample rate. At 48 kHz this will result in the 12 kHz energy dominating the energy-time display. This selection is not fundamentally useful for most applications, but is included for correlation to measurements by other manufacturers

3

<240Hz >8kHz: filters out energy below 240 Hz and above 8 kHz, producing equal sensitivity to signals over a 5 octave range.

4

<124Hz >16kHz: spreads the analysis over a 7 octave range.

Description

This command sets the Quasi-Anechoic Acoustical Tester Energy-Time Window selection.

Example

See example for `AP.S2CDsp.Mls.TimeDelay`.

AP.S2CDsp.MLS.WindowStart**Property****Syntax** `AP.S2CDsp.MLS.WindowStart`**Data Type** Integer

0	None:
1	<5%
2	<10%
3	<20%
4	<30%

Description This command sets the Quasi-Anechoic Acoustical Tester Start Time Window selection.

When a section of the impulse response (direct arrival signal before reflections, for example) is isolated and transformed into the frequency domain, the impulse amplitude at the beginning and ending of that section will generally not be exactly the same and thus will not splice smoothly. The sharp edges introduced into the impulse response by splicing unequal amplitudes will produce ripples in the resulting frequency response plot. Windowing the time domain data by attenuating the amplitude at the beginning and end of the section to be transformed will reduce this rippling, but also reduces the steepness of transitions in the frequency response plots. The Time Start Window and Time Stop Window fields select the window applied to the impulse response (time domain) when transforming it to the frequency domain.

The time window is made up of two half-windows. The first half is selected in the Time Start Window field and is used to process the first portion of data, beginning at the Source 1 Start time on the Sweep panel. The second half-window is selected in the Time Stop Window field and processes the later portion of data, ending at the selected Stop time on the Sweep panel. Separate selection of the Source 1 Start and Stop half-windows permits creation of asymmetrical windows, which provide the optimum match to the asymmetrical shape of the typical impulse response. To change selections, click on the down arrow at the right of the field and click on the desired selection in the list which is displayed. The available selections at both the Time Start Window and Time Stop Window fields are a family of half-cycle raised cosine functions labeled NONE, <5%, <10%, <20% and <30%. The

numeric value refers to the amount of the data record (time span multiplied by sample period) taken up by the window's transition from zero to full amplitude. The Time Start Window half-window starts with an amplitude of zero at the Sweep panel Start time and climbs to an amplitude of 1.00 (no attenuation) at or before the selected percentage of the record. The Time Stop Window half-window starts with an amplitude of 1.00 at or following a point during the record which is within the selected percentage of the record end, and falls to zero at the Sweep panel Stop time. The windows with a steeper transition will alter the data less but will also have less impact on the frequency response ripples. The more gradual transitions have greater ripple reduction but alter the data more.

Example See example for `AP.S2CDsp.Mls.TimeDelay`.

AP.S2CDsp.MLS.WindowStop

Property

Syntax `AP.S2CDsp.MLS.WindowStop`

Data Type Integer

<i>0</i>	None:
<i>1</i>	<5%
<i>2</i>	<10%
<i>3</i>	<20%
<i>4</i>	<30%

Description This command sets the Quasi-Anechoic Acoustical Tester Stop Time Window selection.

See Also `AP.S2CDsp.MLS.WindowStart`

Example See example for `AP.S2CDsp.Mls.TimeDelay`.

User Notes

User Notes

System Two DSP Program & Reference

AP.S2CDsp.Program

Property

Syntax	<code>AP.S2CDsp.Program</code>	
Data Type	Integer	
	0	None: No Digital Analyzer selected.
	1	DSP Audio Analyzer (ANALYZER): usable for analog or digital domain input signals. Measures frequency, amplitude, and one of a number of more sophisticated parameters (2-channel amplitude ratio, selective amplitude, crosstalk, THD+N, IMD, and phase) on both stereo channels simultaneously. For analog domain signals, this DSP-implemented analyzer provides certain advantages over the hardware Analog Analyzer. These include faster testing speed, sharper selectivity of its bandpass filter, a number of standard weighting filters, and a very steep cutoff 400 Hz highpass filter for quantization noise and distortion measurements.
	2	FFT spectrum analyzer (FFT): usable for analog or digital domain input signals. Provides general-purpose time domain (oscilloscope) display of waveforms or frequency domain (spectrum analyzer) display of signals, including the received jitter signal on Cascade Dual Domain units. Features include double precision transforms for better than 140 dB dynamic range, pre-trigger, a variety of selectable transform lengths up through 32k, acquisition memory up to 256k, the ability to position the start of the transformed section anywhere in the acquired record, both synchronous (time domain) and FFT spectrum (power-law) averaging, four windowing functions, a frequency correction technique which adjusts single sinewave signals to bin center so that no window is necessary, and several types of waveform processing for display. FFT for System Two Cascade combines the functions of FFTSLIDE.DSP and the analysis functions of FFTGEN.DSP

for System One, with greatly improved dynamic range due to double precision FFT computations, an additional windowing function not available with System One, and both FFT and time domain averaging for improved accuracy on noisy signals.

3 Digital interface analyzer (INTERVU): analyzes the AES/EBU or consumer digital interface input signal of digital interface models via a 80 MHz sample rate A/D converter. Displays eye patterns, waveform display or spectrum analysis of the digital interface signal, waveform display or spectrum analysis of the recovered jitter signal, triggers on interface errors or on selected sections of the signal including received or transmitted preambles or received channel status blocks, measures jitter of the entire signal or selected sections such as preambles, and performs statistical analysis and histogram display of parameters including amplitude, pulse width, and jitter.

4 Multitone audio analyzer (FASTTEST): usable for analog signals with Cascade + DSP models or both analog and digital domain signals with Cascade Dual Domain models. Provides time or frequency domain views of the signal. With multitone test signals, performs post-FFT processing to measure frequency response, total distortion and noise, noise in the presence of test signal, crosstalk, and generates psychoacoustic masking curves. Trigger modes include external and free-running, or triggering only upon receipt of the specific multitone signal matching the reference signal presently loaded into the digital generator. Variable trigger delay may be set to allow audio processors to settle. Frequency error correction compensates for multitone signals coming from other Audio Precision test instruments, played back from digital reproducers with different clock rates, or recorded and reproduced from analog recorders with speed errors up to 3%. FASTTEST also tests low-bit-rate perceptual coders with multitone signals by summing quantization noise and distortion in critical bands and comparing the results to an embedded psychoacoustic model of the frequency masking effect in humans. FASTTEST for System Two Cascade

- combines the features of FASTTEST.DSP, FASTTRIG.DSP, and CODEC.DSP for System One.
- 5 Quasi-Anechoic Acoustical Tester (MLS): The Quasi-Anechoic Acoustical Tester (MLS) program for the Digital Analyzer uses Maximum Length Sequence (MLS) testing to characterize the linear response of acoustical and electronic devices. It permits time-selective measurements in which one signal, such as the direct sound from a loudspeaker, may be separated from another similar signal, such as a room reflection. The time window may be adjusted to allow measurement of any arrival in a complex reverberation pattern. These signals may be examined in the time domain (showing energy as a function of time) or in the frequency domain (amplitude and phase vs. frequency). Impulse responses may be saved to disk for later down-load to the DSP and further analysis.
- 6 Digital data analyzer (BITTEST). In conjunction with certain specific signals that the digital generator creates, this program measures the returned digital audio signals for bit errors. The audio test signal may be a pseudo-random noise sequence, constant valued samples ("digital dc"), a sinewave of selectable amplitude and frequency, or walking bit patterns. The measurement displays both real-time received data and errors in the received data sequence. Any amount of delay between transmitted and received signals is permissible, allowing testing of devices and transmission links with large amounts of delay or even recorder-reproducers. BITTEST is useful for investigating the integrity of digital audio data links, recorders, etc. It is also invaluable for design test of digital interfaces. BITTEST operates only with digital domain input and output.
- 7 Harmonic Distortion Analyzer(DISTORT): permits flexible, highly selective measurement of the amplitude of user-specified harmonic orders. The user may choose to measure any individual harmonic through the 15th, or the sum of any arbitrary selection of harmonic distortion products from 2nd through 15th. Harmonic Distortion Analyzer may be used with either analog or digital domain signals. It is a two-channel program with four measurement meters per

channel: a selective amplitude meter for the fundamental component of the signal, a frequency counter for the fundamental component, and two identical distortion product summing meters. Each of these distortion summing meters can be set to include any desired combination of harmonic distortion products (through the 15th) as long as each product is within the bandwidth limitations determined by the sample rate. Harmonic Distortion Analyzer can thus measure THD (Total Harmonic Distortion) without noise for any specified set of harmonic products. This analyzer effectively operates as a real-time program, even though it is internally based on FFT technology.

Both channels of Harmonic Distortion Analyzer may be set to measure one signal, such as a single-channel signal or one channel of a stereo signal. This configuration provides four Distortion summing meters. Each meter sends data to a different trace on the graph. This permits, for example, simultaneously plotting the fundamental signal amplitude and the 2nd, 3rd, 4th, and 5th harmonic amplitudes as five graph traces during a single frequency sweep.

Description

This command selects a System Two Cascade Digital Analyzer type.

Example

Sub Main

```
AP.Application.NewTest
```

```
AP.Gen.Output = True
```

```
AP.Application.PanelClose(apbPanelAnalogGenSmall)
```

```
AP.Application.PanelClose(apbPanelAnlrSmall)
```

```
AP.S2CDsp.Program = 2 'Select FFT Spectrum Analyzer
```

```
AP.S2CDsp.FFT.InputFormat = 3 'Select HiBW A/D input
```

```
AP.S2CDsp.FFT.TransformLength = 7'Transform length _  
32768
```

```
AP.S2CDsp.Anlr.AcquireLength = 0 'Track Transform _  
length
```

```
AP.S2CDsp.FFT.Window = 5'No Window move to bin center
```

```

    AP.Sweep.Source1.Start("Hz") = 100000
    AP.Sweep.Data1.Id = 6024
    AP.Sweep.Data1.Top("dBV") = 26.020600
    AP.Sweep.Source1.Id = 5515
    AP.Sweep.Start
    AP.Graph.OptimizeLeft
End Sub

```

AP.S2CDsp.RefCh1dBr

Property

Syntax `AP.S2CDsp.RefCh1dBr(ByVal Unit As String)`

Data Type Double

Parameters

Name	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dbFS, Bits, V, Vp, Vpp, dBu, dBV

Description

This command sets the dBr1 value as the reference for the dBr1 unit selectable at the Level Monitor meter and at the main Function meter in absolute functions. When analog domain units are selected for the dBr1 unit, is are converted into the digital domain via the V/FS Reference value.

Example

```

Sub Main
    AP.File.OpenTest "Ref1.at2"
    AP.S2CDsp.RefVFS("V") = 2
    AP.S2CDsp.RefFreq("Hz") = 2000
    AP.S2CDsp.RefCh1dBr("FFS") = .5
    AP.S2CDsp.RefCh2dBr("FFS") = .75
    Wait .5
    'Get new Ch A Freq reading
    F_reading1 = AP.S2CDsp.Analyzer.ChAFreqRdg("%Hz")
    'Get new Ch A Level reading
    L_reading1 = AP.S2CDsp.Analyzer.ChALevelRdg("dBr2")
    A_reading1 = AP.S2CDsp.Analyzer.FuncRdg("dBr1")
    V_reading1 = AP.S2CDsp.Analyzer.FuncRdg("V")
    NewLine$ = Chr(13)
    a$= "Ch A Level Reading _

```

```

        "+Left(Str$(L_reading1),6)+"dBr 2"
    b$= "Ch A Freq Reading "+Left(Str$(F_reading1),6) _
        +"%Hz"
    c$= "Function Meter Reading _
        "+Left(Str$(A_reading1),6)+"dBr 1"
    d$= "Function Meter Reading _
        "+Left(Str$(V_reading1),6)+"V/FS"
    AP.Prompt.Text = a$ + NewLine$ + b$ + NewLine$ + c$ _
        + NewLine$ + d$
    AP.Prompt.ShowWithContinue
    Beep
    Stop
End Sub

```

AP.S2CDsp.RefCh2dBr

Property

Syntax `AP.S2CDsp.RefCh2dBr (ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: FFS, %FS, dbFS, Bits, V, Vp, Vpp, dBu, dBV

Description This command sets the dBr2 value as the reference for the dBr2 unit selectable at the Level Monitor meter and at the main Function meter in absolute functions. When analog domain units are selected for the dBr2 unit, is are converted into the digital domain via the V/FS Reference value.

Example See example for AP.S2CDsp.RefCh1dBr.

AP.S2CDsp.RefFreq**Property**

Syntax `AP.S2CDsp.RefFreq(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz

Description This command sets the Frequency value for the relative frequency units (octaves, decades, %Hz, etc) of the Digital Analyzer Frequency counter.

Example See example for `AP.S2CDsp.RefCh1dBr`.

AP.S2CDsp.RefVFS**Property**

Syntax `AP.S2CDsp.RefVFS(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: V

Description This command sets the V/FS value is the analog-to-digital scaling value. When testing an external Analog to Digital converter (A/D), the value of analog input voltage which produces digital full scale output may be typed into this field. The Level Monitor or Reading meter units may then be selected as V, Vp, Vpp, dBu, dBV, dBr1, or dBr2 to express the measured digital amplitude in terms of the analog input value to the converter.

Example See example for `AP.S2CDsp.RefCh1dBr`.

User Notes

Serial Interface

AP.SI.RecClock.Bit.Freq

Property

Syntax `AP.SI.RecClock.Bit.Freq(ByVal Unit As String)`

Data Type Double

Parameters

Name	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz

Description This command sets the Serial Interface Receive Bit Clock Frequency value.

AP.SI.RecClock.Bit.Invert

Property

Syntax `AP.SI.RecClock.Bit.Invert`

Data Type Boolean

<i>True</i>	Invert Bit Clock.
<i>False</i>	Normal non-inverting Bit Clock.

Description This command sets the Serial Interface Receive Bit Clock input to normal polarity or inverted polarity.

AP.SI.RecClock.Bit.Source

Property

Syntax `AP.SI.RecClock.Bit.Source`

Data Type Integer

<i>0</i>	Internal Source
<i>1</i>	External Source

Description This command selects the Serial Interface Receive Bit Clock input source.

AP.SI.RecClock.Frame.Freq**Property****Syntax** `AP.SI.RecClock.Frame.Freq(ByVal Unit As String)`**Data Type** Double**Parameters**

Name	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz

Description

This command sets the Serial Interface Receive Frame Clock Frequency value.

AP.SI.RecClock.Frame.Invert**Property****Syntax** `AP.SI.RecClock.Frame.Invert`**Data Type** Boolean*True*

Invert Frame Clock.

False

Normal non-inverting Frame Clock.

Description

This command sets the Serial Interface Receive Frame Clock input to normal polarity or inverted polarity.

AP.SI.RecClock.Frame.Shift**Property****Syntax** `AP.SI.RecClock.Frame.Shift`**Data Type** Boolean*True*

Shift the beginning of the frame clock one bit before the start of the frame.

False

Normal.

Description

This command sets the Serial Interface Receive Frame Clock position.

AP.SI.RecClock.Frame.Source

Property

Syntax	<code>AP.SI.RecClock.Frame.Source</code>	
Data Type	Integer	
	0	Internal Source
	1	External Source
Description	This command selects the Serial Interface Receive Frame Clock input source.	

AP.SI.RecClock.Frame.StartPosition

Property

Syntax	<code>AP.SI.RecClock.Frame.StartPosition</code>	
Data Type	Integer	1 to SubFrameToFrame Ratio as specified by the AP.SI.RecClock.Ratio.SubFrameToFrame command - 1.
Description	This command sets the Serial Interface Receive Frame Clock Start Position in SubFrames.	

AP.SI.RecClock.Frame.StopPosition

Property

Syntax	<code>AP.SI.RecClock.Frame.StopPosition</code>	
Data Type	Integer	Start Position value + 1 to SubFrameToFrame Ratio as specified by the AP.SI.RecClock.Ratio.SubFrameToFrame command.
Description	This command sets the Serial Interface Receive Frame Clock Stop Position in SubFrames.	

AP.SI.RecClock.Master.Freq

Property

Syntax	<code>AP.SI.RecClock.Master.Freq(ByVal Unit As String)</code>	
Data Type	Double	

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz
Description	This command sets the Serial Interface Receive Master Clock Frequency value.	

AP.SI.RecClock.Master.Invert

Property

Syntax	<code>AP.SI.RecClock.Master.Invert</code>	
Data Type	Boolean	
	<i>True</i>	Invert Master Clock.
	<i>False</i>	Normal non-inverting Master Clock.
Description	This command sets the Serial Interface Receive Master Clock input to normal polarity or inverted polarity.	

AP.SI.RecClock.Master.Source

Property

Syntax	<code>AP.SI.RecClock.Master.Source</code>	
Data Type	Integer	
	<i>0</i>	Internal Source
	<i>1</i>	External Source
Description	This command selects the Serial Interface Receive Master Clock input source.	

AP.SI.RecClock.Ratio.BitToSubFrame

Property

Syntax	<code>AP.SI.RecClock.Ratio.BitToSubFrame</code>	
Data Type	Integer	1-256
Description	This command sets the Serial Interface Receive Bit Clock to SubFrame Clock Frequency Ratio.	

AP.SI.RecClock.Ratio.MasterToBit **Property**

Syntax	<code>AP.SI.RecClock.Ratio.MasterToBit</code>	
Data Type	Integer	1-16
Description	This command sets the Serial Interface Receive Master Clock to Bit Clock Frequency Ratio.	

AP.SI.RecClock.Ratio.SubFrameToFrame **Property**

Syntax	<code>AP.SI.RecClock.Ratio.SubFrameToFrame</code>	
Data Type	Integer	1-256
Description	This command sets the Serial Interface Receive SubFrame Clock to Frame Clock Frequency Ratio.	

AP.SI.RecClock.Ratio.SystemToMaster **Property**

Syntax	<code>AP.SI.RecClock.Ratio.SystemToMaster</code>	
Data Type	Integer	1-16
Description	This command sets the Serial Interface Receive System Clock to Master Clock Frequency Ratio.	

AP.SI.RecClock.SubFrame.Freq **Property**

Syntax	<code>AP.SI.RecClock.SubFrame.Freq(ByVal <i>Unit</i> As String)</code>	
Data Type	Double	
Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz
Description	This command sets the Serial Interface Receive SubFrame Clock Frequency value.	

AP.SI.RecClock.SubFrame.Invert**Property**

Syntax	<code>AP.SI.RecClock.SubFrame.Invert</code>	
Data Type	Boolean	
	<i>True</i>	Invert SubFrame Clock.
	<i>False</i>	Normal non-inverting SubFrame Clock.
Description	This command sets the Serial Interface Receive SubFrame Clock input to normal polarity or inverted polarity.	

AP.SI.RecClock.SubFrame.Source**Property**

Syntax	<code>AP.SI.RecClock.SubFrame.Source</code>	
Data Type	Integer	
	<i>0</i>	Internal Source
	<i>1</i>	External Source
Description	This command selects the Serial Interface Receive SubFrame Clock input source.	

AP.SI.RecClock.SubFrame.Width**Property**

Syntax	<code>AP.SI.RecClock.SubFrame.Width</code>	
Data Type	Boolean	
	<i>True</i>	Bit wide pulse
	<i>False</i>	Normal 50% duty cycle signal.
Description	This command sets the Serial Interface Receive SubFrame Clock Width.	

AP.SI.RecClock.System.Freq**Property**

Syntax	<code>AP.SI.RecClock.System.Freq(ByVal Unit As String)</code>
---------------	---

Data Type	Double				
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>Unit</i></td> <td>String that designates the desired unit. The following units are valid for this command: Hz</td> </tr> </tbody> </table>	Name	Description	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz
Name	Description				
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz				
Description	This command sets the Serial Interface Receive System Clock Frequency value.				

AP.SI.RecClock.System.Invert

Property

Syntax	<code>AP.SI.RecClock.System.Invert</code>				
Data Type	Boolean				
	<table> <tr> <td><i>True</i></td> <td>Invert System Clock.</td> </tr> <tr> <td><i>False</i></td> <td>Normal non-inverting System Clock.</td> </tr> </table>	<i>True</i>	Invert System Clock.	<i>False</i>	Normal non-inverting System Clock.
<i>True</i>	Invert System Clock.				
<i>False</i>	Normal non-inverting System Clock.				
Description	This command sets the Serial Interface Receive System Clock input to normal polarity or inverted polarity.				

AP.SI.RecClock.System.Source

Property

Syntax	<code>AP.SI.RecClock.System.Source</code>				
Data Type	Integer				
	<table> <tr> <td><i>0</i></td> <td>Internal Source</td> </tr> <tr> <td><i>1</i></td> <td>External Source</td> </tr> </table>	<i>0</i>	Internal Source	<i>1</i>	External Source
<i>0</i>	Internal Source				
<i>1</i>	External Source				
Description	This command selects the Serial Interface Receive System Clock input source.				

AP.SI.RecData.ChASubFrame

Property

Syntax	<code>AP.SI.RecData.ChASubFrame</code>
Data Type	Integer
	0 to the SubFrame To Frame Ratio as specified by the AP.SI.RecClock.Ratio.SubFrameToFrame command -1

Description This command sets the Serial Interface Receive channel A input to the selected SubFrame.

Note: To specify no data connection to channel A, the special case of zero (0) is used. For monaural data, channel A is set to zero (0) and channel B to the desired SubFrame number.

See Also `AP.SI.RecClock.Ratio.SubFrameToFrame`

AP.SI.RecData.ChBSubFrame

Property

Syntax `AP.SI.RecData.ChBSubFrame`

Data Type Integer Channel A value as specified by the `AP.SI.RecData.ChASubFrame` command + 1 to `SubFrame To Frame Ratio` as specified by the `AP.SI.RecClock.Ratio.SubFrameToFrame` command.

Description This command sets the Serial Interface Receive channel B input to the selected SubFrame.

Note: To specify no data connection to channel A, the special case of zero (0) is used. For monaural data, channel A is set to zero (0) and channel B to the desired SubFrame number.

See Also `AP.SI.RecData.ChASubFrame`,
`AP.SI.RecClock.Ratio.SubFrameToFrame`

AP.SI.RecData.Format.Decode

Property

Syntax `AP.SI.RecData.Format.Decode`

Data Type Integer

0	No data cpmression applied.
1	Apply Mu-Law decoding to data signal.
2	Apply A-Law decoding to data signal.

Description This command selects the Serial Interface Receive Data Format Compression.

AP.SI.RecData.Format.LSBFirst**Property****Syntax** `AP.SI.RecData.Format.LSBFirst`**Data Type** Boolean*True* MSB first*False* LSB first**Description** This command sets the Serial Interface Receive Data Format to Least Significant Bit first (LSB) or Most Significant Bit (MSB) first.

AP.SI.RecData.Format.NumBits**Property****Syntax** `AP.SI.RecData.Format.NumBits`**Data Type** Integer 1 to the number of bits per SubFrame as specified by the AP.SI.RecClock.Ratio.BitToSubFrame command.**Description** This command sets the Serial Interface Receive Data Format Number of Data Bts in each SubFrame.**See Also** `AP.SI.RecClock.Ratio.BitToSubFrame`

AP.SI.RecData.Format.StartPosition**Property****Syntax** `AP.SI.RecData.Format.StartPosition`**Data Type** Integer 1 to the number of bits per SubFrame - the number of Data Bits as defined by the AP.SI.RecData.Format.NumBits command + 1.**Description** This command sets the Serial Interface Receive Data Format Number of Data Bts in each SubFrame.**See Also** `AP.SI.RecData.Format.NumBits`

AP.SI.TransClock.Bit.Freq**Property**

Syntax	<code>AP.SI.TransClock.Bit.Freq(ByVal Unit As String)</code>	
Data Type	Double	
Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz
Description	This command sets the Serial Interface Transmit Bit Clock Frequency value.	

AP.SI.TransClock.Bit.Invert**Property**

Syntax	<code>AP.SI.TransClock.Bit.Invert</code>	
Data Type	Boolean	
	<i>True</i>	Invert Bit Clock.
	<i>False</i>	Normal non-inverting Bit Clock.
Description	This command sets the Serial Interface Transmit Bit Clock input to normal polarity or inverted polarity.	

AP.SI.TransClock.Bit.Source**Property**

Syntax	<code>AP.SI.TransClock.Bit.Source</code>	
Data Type	Integer	
	<i>0</i>	Internal Source
	<i>1</i>	External Source
Description	This command selects the Serial Interface Transmit Bit Clock input source.	

AP.SI.TransClock.Frame.Freq Property

Syntax `AP.SI.TransClock.Frame.Freq(ByVal Unit As String)`

Data Type Double

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz

Description This command sets the Serial Interface Transmit Frame Clock Frequency value.

AP.SI.TransClock.Frame.Invert Property

Syntax `AP.SI.TransClock.Frame.Invert`

Data Type Boolean

<i>True</i>	Invert Frame Clock.
<i>False</i>	Normal non-inverting Frame Clock.

Description This command sets the Serial Interface Transmit Frame Clock input to normal polarity or inverted polarity.

AP.SI.TransClock.Frame.Shift Property

Syntax `AP.SI.TransClock.Frame.Shift`

Data Type Boolean

<i>True</i>	Shift the beginning of the frame clock one bit before the start of the frame.
<i>False</i>	Normal.

Description This command sets the Serial Interface Transmit Frame Clock position.

AP.SI.TransClock.Frame.Source

Property

Syntax	<code>AP.SI.TransClock.Frame.Source</code>	
Data Type	Integer	
	<code>0</code>	Internal Source
	<code>1</code>	External Source
Description	This command selects the Serial Interface Transmit Frame Clock input source.	

AP.SI.TransClock.Frame.StartPosition

Property

Syntax	<code>AP.SI.TransClock.Frame.StartPosition</code>	
Data Type	Integer	1 to SubFrameToFrame Ratio as specified by the AP.SI.TransClock.Ratio.SubFrameToFrame command - 1.
Description	This command sets the Serial Interface Transmit Frame Clock Start Position in SubFrames.	

AP.SI.TransClock.Frame.StopPosition

Property

Syntax	<code>AP.SI.TransClock.Frame.StopPosition</code>	
Data Type	Integer	Start Position value + 1 to SubFrameToFrame Ratio as specified by the AP.SI.TransClock.Ratio.SubFrameToFrame command.
Description	This command sets the Serial Interface Transmit Frame Clock Stop Position in SubFrames.	

AP.SI.TransClock.Master.Freq

Property

Syntax	<code>AP.SI.TransClock.Master.Freq(ByVal Unit As String)</code>	
Data Type	Double	

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz
Description	This command sets the Serial Interface Transmit Master Clock Frequency value.	

AP.SI.TransClock.Master.Invert

Property

Syntax	<code>AP.SI.TransClock.Master.Invert</code>	
Data Type	Boolean	
	<i>True</i>	Invert Master Clock.
	<i>False</i>	Normal non-inverting Master Clock.
Description	This command sets the Serial Interface Transmit Master Clock input to normal polarity or inverted polarity.	

AP.SI.TransClock.Master.Source

Property

Syntax	<code>AP.SI.TransClock.Master.Source</code>	
Data Type	Integer	
	<i>0</i>	Internal Source
	<i>1</i>	External Source
Description	This command selects the Serial Interface Transmit Master Clock input source.	

AP.SI.TransClock.Ratio.BitToSubFrame

Property

Syntax	<code>AP.SI.TransClock.Ratio.BitToSubFrame</code>	
Data Type	Integer	1-256
Description	This command sets the Serial Interface Transmit Bit Clock to SubFrame Clock Frequency Ratio.	

AP.SI.TransClock.Ratio.MasterToBit **Property**

Syntax	<code>AP.SI.TransClock.Ratio.MasterToBit</code>
Data Type	Integer 1-16
Description	This command sets the Serial Interface Transmit Master Clock to Bit Clock Frequency Ratio.

AP.SI.TransClock.Ratio.SubFrameToFrame **Property**

Syntax	<code>AP.SI.TransClock.Ratio.SubFrameToFrame</code>
Data Type	Integer 1-256
Description	This command sets the Serial Interface Transmit SubFrame Clock to Frame Clock Frequency Ratio.

AP.SI.TransClock.Ratio.SystemToMaster **Property**

Syntax	<code>AP.SI.TransClock.Ratio.SystemToMaster</code>
Data Type	Integer 1-16
Description	This command sets the Serial Interface Transmit System Clock to Master Clock Frequency Ratio.

AP.SI.TransClock.SubFrame.Freq **Property**

Syntax	<code>AP.SI.TransClock.SubFrame.Freq(ByVal <i>Unit</i> As String)</code>	
Data Type	Double	
Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz

Description This command sets the Serial Interface Transmit SubFrame Clock Frequency value.

AP.SI.TransClock.SubFrame.Invert Property

Syntax `AP.SI.TransClock.SubFrame.Invert`

Data Type Boolean

True Invert SubFrame Clock.

False Normal non-inverting SubFrame Clock.

Description This command sets the Serial Interface Transmit SubFrame Clock input to normal polarity or inverted polarity.

AP.SI.TransClock.SubFrame.Source Property

Syntax `AP.SI.TransClock.SubFrame.Source`

Data Type Integer

0 Internal Source

1 External Source

Description This command selects the Serial Interface Transmit SubFrame Clock input source.

AP.SI.TransClock.SubFrame.Width Property

Syntax `AP.SI.TransClock.SubFrame.Width`

Data Type Boolean

True Bit wide pulse

False Normal 50% duty cycle signal.

Description This command sets the Serial Interface Transmit SubFrame Clock Width.

AP.SI.TransClock.System.Freq**Property****Syntax** `AP.SI.TransClock.System.Freq(ByVal Unit As String)`**Data Type** Double**Parameters**

Name	Description
<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: Hz

Description

This command sets the Serial Interface Transmit System Clock Frequency value.

AP.SI.TransClock.System.Invert**Property****Syntax** `AP.SI.TransClock.System.Invert`**Data Type** Boolean*True*

Invert System Clock.

False

Normal non-inverting System Clock.

Description

This command sets the Serial Interface Transmit System Clock input to normal polarity or inverted polarity.

AP.SI.TransClock.System.Source**Property****Syntax** `AP.SI.TransClock.System.Source`**Data Type** Integer*0*

Internal Source

1

External Source

Description

This command selects the Serial Interface Transmit System Clock input source.

AP.SI.TransData.ChASubFrame**Property**

Syntax	<code>AP.SI.TransData.ChASubFrame</code>	
Data Type	Integer	0 to the SubFrame To Frame Ratio as specified by the <code>AP.SI.TransClock.Ratio.SubFrameToFrame</code> command -1
Description	This command sets the Serial Interface Transmit channel A input to the selected SubFrame. Note: To specify no data connection to channel A, the special case of zero (0) is used. For monaural data, channel A is set to zero (0) and channel B to the desired SubFrame number.	
See Also	<code>AP.SI.TransClock.Ratio.SubFrameToFrame</code>	

AP.SI.TransData.ChBSubFrame**Property**

Syntax	<code>AP.SI.TransData.ChBSubFrame</code>	
Data Type	Integer	Channel A value as specified by the <code>AP.SI.TransData.ChASubFrame</code> command + 1 to SubFrame To Frame Ratio as specified by the <code>AP.SI.TransClock.Ratio.SubFrameToFrame</code> command.
Description	This command sets the Serial Interface Transmit channel B input to the selected SubFrame. Note: To specify no data connection to channel A, the special case of zero (0) is used. For monaural data, channel A is set to zero (0) and channel B to the desired SubFrame number.	
See Also	<code>AP.SI.TransData.ChASubFrame</code> , <code>AP.SI.TransClock.Ratio.SubFrameToFrame</code>	

AP.SI.TransData.Format.Encode**Property**

Syntax	<code>AP.SI.TransData.Format.Encode</code>	
Data Type	Integer	

0	No data compression applied.
1	Apply Mu-Law decoding to data signal.
2	Apply A-Law decoding to data signal.

Description This command selects the Serial Interface Transmit Data Format Compression.

AP.SI.TransData.Format.LSBFirst

Property

Syntax `AP.SI.TransData.Format.LSBFirst`

Data Type Boolean

<i>True</i>	MSB first
<i>False</i>	LSB first

Description This command sets the Serial Interface Transmit Data Format to Least Significant Bit first (LSB) or Most Significant Bit (MSB) first.

AP.SI.TransData.Format.NumBits

Property

Syntax `AP.SI.TransData.Format.NumBits`

Data Type Integer 1 to the number of bits per SubFrame as specified by the `AP.SI.TransClock.Ratio.BitToSubFrame` command.

Description This command sets the Serial Interface Transmit Data Format Number of Data Bts in each SubFrame.

See Also `AP.SI.TransClock.Ratio.BitToSubFrame`

AP.SI.TransData.Format.StartPosition

Property

Syntax `AP.SI.TransData.Format.StartPosition`

Data Type Integer 1 to the number of bits per SubFrame - the number of Data Bits as defined by the `AP.SI.TransData.Format.NumBits` command + 1.

Description This command sets the Serial Interface Transmit Data Format Number of Data Bts in each SubFrame.

See Also `AP.SI.TransData.Format.NumBits`

User Notes

Speaker

AP.Speaker.Mode

Property

Syntax `AP.Speaker.Mode`

Data Type

Integer

0 Mono: a single signal is fed to both left and right headphones and to the internal loudspeaker.

1 Stereo: different signals are fed to the left and right headphones (in most cases). Both these signals are summed into the internal monaural loudspeaker located in the bottom of the instrument.

Description

This command selects the output configuration for the speaker output jack.

See Also

`AP.Speaker.Source`

Example

```
Sub Main
  AP.Application.NewTest
  AP.Gen.Wfm 1, 2
  AP.Gen.ChBFreq("Hz") = 2000.000000
Start:
  Begin Dialog UserDialog 280,133
    PushButton 20,14,240,28,"Monitor Analog Generator _
      Channel A",.PushButton1
    PushButton 20,49,240,28,"Monitor Analog Generator _
      Channel B",.PushButton2
    PushButton 20,91,240,28,"EXIT (Speaker _
      OFF)",.PushButton3
  End Dialog
  Dim dlg As UserDialog

  Dim MainMenu As UserDialog
  Select Case Dialog(MainMenu)
  Case 1
    AP.Speaker.Mode = 0      'Mono
    AP.Speaker.Source = 2    'Analog Generator Ch A
```

```

Case 2
    AP.Speaker.Mode = 0
    AP.Speaker.Source = 3    'Analog Generator Ch B
Case Else
    AP.Speaker.Mode = 0
    AP.Speaker.Source = 0    'Speaker OFF
End
End Select
GoTo Start:
End Sub

```

AP.Speaker.Source

Property

Syntax

AP.Speaker.Source

Data Type

Integer

The following list contains the selections relevant to the `AP.Speaker.Mode` command Mono Configuration.

0	Off: disables audible monitoring.
1	Analog Analyzer Reading: is the final analog signal in the Analog Analyzer, following all filtering (and following the wow and flutter discriminator or IMD detectors if the reading meter is in W&F or IMD modes).
2	Analog Generator A
3	Analog Generator B
4	DSP Monitor A
5	DSP Monitor B
6	Analog Input A
7	Analog Input B

The following list contains the selections relevant to the `AP.Speaker.Mode` command Stereo Configuration.

0	Off: disables audible monitoring.
1	Analog Analyzer Reading: is the final analog signal in the Analog Analyzer, following all filtering (and following the wow

	and flutter discriminator or IMD detectors if the reading meter is in W&F or IMD modes).
2	Generator Monitor
3	DSP Monitor A&B
4	Analog Input

Description This command selects a monitoring location(s) for the speaker and headphone jack outputs.

See Also AP .Speaker .Mode

Example See example for AP .Speaker .Mode.

User Notes

Sweep

AP.Sweep.AbortTime

Property

Syntax `AP.Sweep.AbortTime`

Data Type Double Time in seconds. Setting an abort time of zero seconds disables the abort function.

Description This command defines the maximum time allowed for a sweep to complete after a sweep is started using any OLE command. If the abort time is exceeded the current sweep is terminated. If the `AP.Data.ColSize` command returns a value less than the number of steps in the sweep then the sweep was aborted. This setting is not routinely monitored therefore accuracy may be in the seconds.

Note: This command is global and affects all subsequent sweeps. Care should be taken when using this command to disable it when finished.

See Also `AP.Sweep.Start`, `AP.Sweep.StartWithAppend`,
`AP.Sweep.StartWithRepeat`, `AP.Data.ColSize`

Example Sub Main
Dim Steps As Integer

```
Steps = 100
AP.Application.NewTest
AP.Sweep.Source1.Steps = Steps
```

```
AP.Sweep.AbortTime = 5.0
AP.Sweep.Start
```

```
If AP.Data.ColSize(0, 0) < Steps Then
  With AP.Prompt
    .Text = Chr$(13) & "Normal Sweep Time exceeded"
           & Chr$(13) & Chr$(13) & "Sweep Terminated"
    .FontSize = 8
    .Position(-1,-1,220,150)
    .Show
  Wait 3
```

```

        .Hide
    End With
End If
AP.Sweep.AbortTime = 0.0
End Sub

```

AP.Sweep.Append

Property

Syntax `AP.Sweep.Append`

Data Type Boolean

True Append data to current data in memory.
False Replace current data in memory.

Description This command enables or disables appending data to the end of measurements contained in memory. If append is enabled the measurements in memory are retained and the next sweep will add additional measurements to memory. If append is disabled the measurements in memory are replaced by the next sweep data.

See Also `AP.Sweep.Repeat`, `AP.Sweep.StartWithAppend`,
`AP.Sweep.StartWithRepeat`

Example

```

Sub Main
    AP.Application.NewTest 'Reset panels
    AP.Sweep.CreateGraph = 1
    AP.Sweep.CreateTable = 0
    AP.Sweep.GraphType = 0
    AP.Gen.Output = 1
    AP.Anlr.ChAInput = 2
    AP.Anlr.FuncFilterHP = 3
    AP.Anlr.FuncFilterLP = 0

    'The commands in the following section could be
    ' replaced with commands for Data1-6
    AP.Sweep.Data1.Id = 5906
    AP.Sweep.Data1.Limits("None",1,1)

    'The commands in the following section could be
    ' replaced with commands for Data2

```

```

AP.Sweep.Data1.AutoDiv = 0
AP.Sweep.Data1.Div = 1
AP.Sweep.Data1.Autoscale = 1
AP.Sweep.Data1.LogLin = 1
AP.Sweep.Data1.Top("V") = 1
AP.Sweep.Data1.Bottom("V") = 0

AP.Sweep.Source1.Start("Hz") = 20.0
AP.Sweep.Source1.Stop("Hz") = 200000.0
AP.Sweep.PreSweepDelay = 0.2

AP.Sweep.Start
AP.Sweep.Append = True

AP.Anlr.FuncFilterHP = 2
AP.Anlr.FuncFilterLP = 1
AP.Sweep.Start
AP.Anlr.FuncFilterHP = 1
AP.Anlr.FuncFilterLP = 2
AP.Sweep.Start
AP.Anlr.FuncFilterHP = 0
AP.Anlr.FuncFilterLP = 3

AP.Data.OptimizeDisplay(0)
End Sub

```

AP.Sweep.CopyData1To2

Method

Syntax

AP.Sweep.CopyData1To2

Description

This command copies the Sweep panel Data 1 settings to Data 2

Example

```

Sub Main
  AP.Application.NewTest 'New Test
  AP.Gen.Output = True 'Generator Output ON
  AP.Anlr.ChAInput = 2 'Ch A Input to GenMon
  AP.Anlr.ChBInput = 2 'Ch B Input to GenMon
  AP.Anlr.FuncMode = 3 'Func Meter to THD+N Ampl

  AP.S2CDsp.Program = 2 'Select FFT Digital Analyzer

```

```

AP.S2CDsp.FFT.InputFormat = 1'Select Low BW A/D Input
AP.S2CDsp.FFT.Ch1Source = 2   'Digital Analyzer Ch 1 _
    Source to Anlr Rdg Ampl

AP.Sweep.Data1.Id = 6024   'Select Fft.Ch.1 Ampl _
    for Data 1
AP.Sweep.Data2.Id = 6027   'Select Fft.Ch.2 Ampl _
    for Data 2
AP.Sweep.Source1.Id = 5515 'Select Fft.FFT Freq. _
    for Source 1
AP.Sweep.Start             'Acquire waveform
'Display data so that the vertical scaling is _
    relative to optimized data for Data 1
AP.Graph.OptimizeLeft      'Optimize Data 1
AP.Graph.CopyToSweepPanel 'Copy Left and Right _
    graph vertical scale information to Sweep Panel
AP.Sweep.CopyData1to2     'Copy Data 1 settings _
    to Data 2
Wait 5
'Display data so that the vertical scaling is _
    relative to optimized data for Data 2
AP.Graph.OptimizeRight     'Optimize Data 2
AP.Graph.CopyToSweepPanel 'Copy Left and Right _
    graph vertical scale information to Sweep Panel
AP.Sweep.CopyData2to1     'Copy Data 2 settings
    to Data 1
End Sub

```

AP.Sweep.CopyData2To1

Method

Syntax	AP.Sweep.CopyData2To1
Description	This command copies the Sweep panel Data 2 settings to Data 1
Example	See example for AP.Sweep.CopyData1To2.

AP.Sweep.CreateGraph

Property

Syntax	AP.Sweep.CreateGraph	
Data Type	Boolean	
	<i>True</i>	Display a graph window when starting the sweep if a graph window is not displayed.
	<i>False</i>	Do not create a graph window when starting the sweep.
Description	This command enables or disables creation of the graph window when a sweep is run.	
See Also	AP.Sweep.CreateTable, AP.Sweep.GraphType	
Example	See example for AP.Sweep.Append.	

AP.Sweep.CreateTable

Property

Syntax	AP.Sweep.CreateTable	
Data Type	Boolean	
	<i>True</i>	Display a Data Table window when starting the sweep if a Data Table window is not displayed.
	<i>False</i>	Do Not create a Data Table window when starting the sweep.
Description	This command enables or disables creation of the Data Table window when a sweep is run.	
See Also	AP.Sweep.CreateGraph, AP.Sweep.GraphType	
Example	See example for AP.Sweep.Append.	

AP.Sweep.Data1.AutoDiv

Property

Syntax	AP.Sweep.Data1.AutoDiv	
Data Type	Boolean	
	<i>True</i>	Automatically select the number of divisions.

False Use the number of divisions defined by the AP.Sweep.Data1.Div command.

Description This command enables or disables automatic selection of the number of linear vertical axis divisions displayed for Data 1.

See Also AP.Sweep.Data1.Div, AP.Sweep.Data1.LogLin

Example See example for AP.Sweep.Append.

AP.Sweep.Data1.Autoscale

Property

Syntax AP.Sweep.Data1.Autoscale

Data Type Boolean

True Autoscale graph vertical axis for Data 1.

False Do Not Autoscale graph vertical axis for Data 1.

Description This command enables or disables automatic scaling of the graph vertical axis Top and Bottom values for Data 1. The Data 1 vertical axis is shown on the left side of the graph.

Example See example for AP.Sweep.Append.

AP.Sweep.Data1.Bottom

Property

Syntax AP.Sweep.Data1.Bottom(*ByVal Unit As String*)

Data Type Double Enter a value that is to be displayed at the bottom of the graph left axis.

Parameters	Name	Description
	<i>Unit</i>	Refer to the setting or reading defined by the AP.Sweep.Data1.Id command to determine the appropriate unit selections.

Description This command defines the bottom value on the graph vertical axis located on the left side of the graph window.

See Also AP.Sweep.Data1.Top

Example See example for `AP.Sweep.Append`.

AP.Sweep.Data1.Div

Property

Syntax `AP.Sweep.Data1.Div`

Data Type Long Number of divisions displayed.

Description This command sets the number of divisions that are to be displayed for a linear vertical axis defined on Data 1. The `AP.Sweep.Data1.AutoDiv` must be disabled.

See Also `AP.Sweep.Data1.AutoDiv`, `AP.Sweep.Data1.LogLin`

Example See example for `AP.Sweep.Append`.

AP.Sweep.Data1.Id

Property

Syntax `AP.Sweep.Data1.Id`

Data Type Long Instrument Parameter ID#.

Description This command is used to select the instrument parameter, which will return readings for Data 1.

Refer to Appendix B to obtain instrument parameter identification numbers.

Example See example for `AP.Sweep.Append`.

AP.Sweep.Data1.Limits

Method

Syntax `AP.Sweep.Data1.Limits(ByVal PathName As String, ByVal Column As Integer, ByVal Upper As Boolean)`

Parameters

Name	Description
<i>PathName</i>	Any valid DOS path and file name. The file must be an APWIN limit file (.adl). Enter "None" for the file name to remove the limit file from Data 1.

Column 1 = Data 1 measurements.
 2 = Data 2 measurements.
 3 = Data 3 measurements.
 4 = Data 4 measurements.
 5 = Data 5 measurements.
 6 = Data 6 measurements.

Upper True = Upper Limit.
 False = Lower Limit.

Result

Boolean

True File attachment successful.

False File attachment failed.

Description

This command attaches or removes a limit file from Data 1 for upper or lower limit comparisons.

See Also

AP.Sweep.Recompare

Example

Sub Main

```

AP.Log.Enable = 0                            'Disable log file
AP.File.OpenTest "CODEC.AT2" 'Open test to create a _
                                            masking curve limit file.
AP.Sweep.Start                                'Start sweep.
AP.File.SaveDataAs "MASK.ADL" 'Save masking curve as _
                                            MASK.ADL limit file.
AP.Application.NewData                        'Remove Masking curve _
                                            data from memory.
AP.S2CDsp.FastTest.Mode = 0                'Set the DSP panel _
                                            measurement field to spectrum mode.
AP.Sweep.Reprocess                            'Reprocess the acquired _
                                            waveform and display spectrum results.
AP.Sweep.Sourcel.Table ("CODEC.ADS",0)
                                            'Attach Sweep table.
AP.S2CDsp.FastTest.Mode = 1                'Set the DSP panel _
                                            measurement field to response mode.
AP.Sweep.Reprocess                            'Reprocess the acquired _
                                            waveform and display response results.

'Attach upper limits to Data1 & Data3.
   AP.Sweep.Data1.Limits("MASK.ADL", 1, True)
   AP.Sweep.Data3.Limits("Mask.ADL", 1, True)

```

```

AP.S2CDsp.FastTest.Mode = 2    'Set DSP measurement _
    mode to distortion.
AP.Sweep.Reprocess             'Reprocess the acquired _
    waveform and display distortion results.
AP.S2CDsp.FastTest.Mode = 3    'Set the DSP panel _
    measurement field to noise mode.
AP.Sweep.Reprocess             'Reprocess the acquired _
    waveform and display noise results.
End Sub

```

AP.Sweep.Data1.LogLin

Property

Syntax	<code>AP.Sweep.Data1.LogLin</code>				
Data Type	Integer				
	<table> <tr> <td><i>0</i></td> <td>Logarithmic vertical axis.</td> </tr> <tr> <td><i>1</i></td> <td>Linear vertical axis.</td> </tr> </table>	<i>0</i>	Logarithmic vertical axis.	<i>1</i>	Linear vertical axis.
<i>0</i>	Logarithmic vertical axis.				
<i>1</i>	Linear vertical axis.				
Description	This command determines the Data 1 vertical axis data scaling type.				
See Also	<code>AP.Sweep.Data1.Div</code> , <code>AP.Sweep.Data1.AutoDiv</code>				
Example	See example for <code>AP.Sweep.Append</code> .				

AP.Sweep.Data1.Top

Property

Syntax	<code>AP.Sweep.Data1.Top(ByVal Unit As String)</code>				
Data Type	Double	Enter a value that is to be displayed at the top of the graph left axis.			
Parameters	<table> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>Unit</i></td> <td>Refer to the setting or reading defined by the <code>AP.Sweep.Data1.Id</code> command to determine the appropriate unit selections.</td> </tr> </tbody> </table>	Name	Description	<i>Unit</i>	Refer to the setting or reading defined by the <code>AP.Sweep.Data1.Id</code> command to determine the appropriate unit selections.
Name	Description				
<i>Unit</i>	Refer to the setting or reading defined by the <code>AP.Sweep.Data1.Id</code> command to determine the appropriate unit selections.				
Description	This command defines the top value on the graph vertical axis located on the left side of the graph window.				

See Also	AP.Sweep.Data1.Bottom
Example	See example for AP.Sweep.Append.

AP.Sweep.Data2.AutoDiv

Property

Syntax	AP.Sweep.Data2.AutoDiv
Data Type	Boolean
	<i>True</i> Automatically select the number of divisions.
	<i>False</i> Use the number of divisions defined by the AP.Sweep.Data2.Div command.
Description	This command enables or disables automatic selection of the number of linear vertical axis divisions displayed for Data 2.
See Also	AP.Sweep.Data2.Div, AP.Sweep.Data2.LogLin
Example	See example for AP.Sweep.Append.

AP.Sweep.Data2.Autoscale

Property

Syntax	AP.Sweep.Data2.Autoscale
Data Type	Boolean
	<i>True</i> Autoscale graph vertical axis for Data 2.
	<i>False</i> Do Not Autoscale graph vertical axis for Data 2.
Description	This command enables or disables automatic scaling of the graph vertical axis Top and Bottom values for Data 2. The Data 2 vertical axis is shown on the right side of the graph.
Example	See example for AP.Sweep.Append.

AP.Sweep.Data2.Bottom

Property

Syntax	AP.Sweep.Data2.Bottom (ByVal <i>Unit</i> As String)	
Data Type	Double	Enter a value that is to be displayed at the bottom of the graph right axis.
Parameters	Name	Description
	<i>Unit</i>	Refer to the setting or reading defined by the <code>AP.Sweep.Data2.Id</code> command to determine the appropriate unit selections.
Description	This command defines the bottom value on the graph vertical axis located on the right side of the graph window.	
See Also	<code>AP.Sweep.Data2.Top</code>	
Example	See example for <code>AP.Sweep.Append</code> .	

AP.Sweep.Data2.Div

Property

Syntax	AP.Sweep.Data2.Div	
Data Type	Long	Number of divisions displayed.
Description	This command sets the number of divisions that are to be displayed for a linear vertical axis defined on Data 2. The <code>AP.Sweep.Data2.AutoDiv</code> must be disabled.	
See Also	<code>AP.Sweep.Data2.AutoDiv</code> , <code>AP.Sweep.Data2.LogLin</code>	
Example	See example for <code>AP.Sweep.Append</code> .	

AP.Sweep.Data2.Id

Property

Syntax	AP.Sweep.Data2.Id	
Data Type	Long	Instrument Parameter ID#.
Description	This command is used to select the instrument parameter, which will return readings for Data 2.	

Refer to Appendix B to obtain instrument parameter identification numbers.

Example See example for `AP.Sweep.Append`.

AP.Sweep.Data2.Limits

Method

Syntax `AP.Sweep.Data2.Limits(ByVal PathName As String, ByVal Column As Integer, ByVal Upper As Boolean)`

Parameters	Name	Description
	<i>PathName</i>	Any valid DOS path and file name. The file must be an APWIN limit file (.adl). Enter "None" for the file name to remove the limit file from Data 2.
	<i>Column</i>	1 = Data 1 measurements. 2 = Data 2 measurements. 3 = Data 3 measurements. 4 = Data 4 measurements. 5 = Data 5 measurements. 6 = Data 6 measurements.
	<i>Upper</i>	True = Upper Limit. False = Lower Limit.
Result	Boolean	
	<i>True</i>	File attachment successful.
	<i>False</i>	File attachment failed.

Description This command attaches or removes a limit file from Data 2 for upper or lower limit comparisons.

See Also `AP.Sweep.Recompare`

Example See example for `AP.Sweep.Data1.Limits`.

AP.Sweep.Data2.LogLin

Property

Syntax `AP.Sweep.Data2.LogLin`

Data Type Integer

0 Logarithmic vertical axis.

1 Linear vertical axis.

Description This command determines the Data 2 vertical axis data scaling type.

See Also `AP.Sweep.Data2.Div`, `AP.Sweep.Data2.AutoDiv`

Example See example for `AP.Sweep.Append`.

AP.Sweep.Data2.Top

Property

Syntax `AP.Sweep.Data2.Top`(ByVal *Unit* As String)

Data Type Double Enter a value that is to be displayed at the top of the graph right axis.

Parameters	Name	Description
	<i>Unit</i>	Refer to the setting or reading defined by the <code>AP.Sweep.Data2.Id</code> command to determine the appropriate unit selections.

Description This command defines the top value on the graph vertical axis located on the right side of the graph window.

See Also `AP.Sweep.Data2.Bottom`

Example See example for `AP.Sweep.Append`.

AP.Sweep.Data3.Id

Property

Syntax `AP.Sweep.Data3.Id`

Data Type Long Instrument Parameter ID#.

Description This command is used to select the instrument parameter, which will return readings for Data 3.

Refer to Appendix B to obtain instrument parameter identification numbers.

Example See example for AP.Sweep.Append.

AP.Sweep.Data3.Limits

Method

Syntax `AP.Sweep.Data3.Limits(ByVal PathName As String, ByVal Column As Integer, ByVal Upper As Boolean)`

Parameters	Name	Description
	<i>PathName</i>	Any valid DOS path and file name. The file must be an APWIN limit file (.adl). Enter "None" for the file name to remove the limit file from Data 3.
	<i>Column</i>	1 = Data 1 measurements. 2 = Data 2 measurements. 3 = Data 3 measurements. 4 = Data 4 measurements. 5 = Data 5 measurements. 6 = Data 6 measurements.
	<i>Upper</i>	True = Upper Limit. False = Lower Limit.
Result	Boolean	
	<i>True</i>	File attachment successful.
	<i>False</i>	File attachment failed.

Description This command attaches or removes a limit file from Data 3 for upper or lower limit comparisons.

See Also AP.Sweep.Recompare

Example See example for AP.Sweep.Data1.Limits.

AP.Sweep.Data4.Id

Property

Syntax `AP.Sweep.Data4.Id`

Data Type Long Instrument Parameter ID#.

Description This command is used to select the instrument parameter, which will return readings for Data 4.

Refer to Appendix B to obtain instrument parameter identification numbers.

Example See example for AP.Sweep.Append.

AP.Sweep.Data4.Limits

Method

Syntax `AP.Sweep.Data4.Limits(ByVal PathName As String, ByVal Column As Integer, ByVal Upper As Boolean)`

Parameters

Name	Description
<i>PathName</i>	Any valid DOS path and file name. The file must be an APWIN limit file (.adl). Enter "None" for the file name to remove the limit file from Data 4.
<i>Column</i>	1 = Data 1 measurements. 2 = Data 2 measurements. 3 = Data 3 measurements. 4 = Data 4 measurements. 5 = Data 5 measurements. 6 = Data 6 measurements.
<i>Upper</i>	True = Upper Limit. False = Lower Limit.

Result

Boolean	
<i>True</i>	File attachment successful.
<i>False</i>	File attachment failed.

Description This command attaches or removes a limit file from Data 4 for upper or lower limit comparisons.

See Also AP.Sweep.Recompare

Example See example for AP.Sweep.Append.

AP.Sweep.Data5.Id

Property

Syntax	<code>AP.Sweep.Data5.Id</code>
Data Type	Long Instrument Parameter ID#.
Description	This command is used to select the instrument parameter, which will return readings for Data 5. Refer to Appendix B to obtain instrument parameter identification numbers.
Example	See example for <code>AP.Sweep.Append</code> .

AP.Sweep.Data5.Limits

Method

Syntax `AP.Sweep.Data5.Limits(ByVal PathName As String, ByVal Column As Integer, ByVal Upper As Boolean)`

Parameters	Name	Description
	<i>PathName</i>	Any valid DOS path and file name. The file must be an APWIN limit file (.adl). Enter "None" for the file name to remove the limit file from Data 5.
	<i>Column</i>	1 = Data 1 measurements. 2 = Data 2 measurements. 3 = Data 3 measurements. 4 = Data 4 measurements. 5 = Data 5 measurements. 6 = Data 6 measurements.
Result	<i>Upper</i>	True = Upper Limit. False = Lower Limit.
	<i>True</i>	File attachment successful.
	<i>False</i>	File attachment failed.
Description	This command attaches or removes a limit file from Data 5 for upper or lower limit comparisons.	
See Also	<code>AP.Sweep.Recompare</code>	

Example See example for AP . Sweep . Append.

AP.Sweep.Data6.Id

Property

Syntax `AP . Sweep . Data6 . Id`

Data Type Long Instrument Parameter ID#.

Description This command is used to select the instrument parameter, which will return readings for Data 6.

Refer to Appendix B to obtain instrument parameter identification numbers.

Example See example for AP . Sweep . Append.

AP.Sweep.Data6.Limits

Method

Syntax `AP . Sweep . Data6 . Limits (ByVal PathName As String, ByVal Column As Integer, ByVal Upper As Boolean)`

Data Type Boolean

Parameters

Name	Description
<i>PathName</i>	Any valid DOS path and file name. The file must be an APWIN limit file (.adl). Enter "None" for the file name to remove the limit file from Data 6.
<i>Column</i>	1 = Data 1 measurements. 2 = Data 2 measurements. 3 = Data 3 measurements. 4 = Data 4 measurements. 5 = Data 5 measurements. 6 = Data 6 measurements.
<i>Upper</i>	True = Upper Limit. False = Lower Limit.
Result	Boolean
<i>True</i>	File attachment successful.

False File attachment failed.

Description This command attaches or removes a limit file from Data 6 for upper or lower limit comparisons.

See Also `AP.Sweep.Recompare`

Example See example for `AP.Sweep.Append`.

AP.Sweep.GraphType

Property

Syntax `AP.Sweep.GraphType`

Data Type Integer

0 X - Y mode. Data 1-6 measurements are displayed on the vertical axis and Source settings are displayed on the horizontal axis.

1 X - Y Data2 on X mode. Data 1, and 3-6 measurements are displayed on the vertical axis and Data 2 readings are displayed on the horizontal axis.

Description This command selects the graph display mode. The `AP.Sweep.Data2.Id` must be defined.

See Also `AP.Sweep.Data2.Id`

Example See example for `AP.Sweep.Append`.

AP.Sweep.IsRunning

Property

Syntax `AP.Sweep.IsRunning`

Result Boolean

True Sweep process running.

False Sweep process not running.

Description This command returns the status of the Sweep process.

Example See example for `AP.Sweep.Stop`.

AP.Sweep.PreSweepDelay

Property

Syntax	<code>AP.Sweep.PreSweepDelay</code>
Data Type	Double 0.0 to 3.0 sec.
Description	<p>This command sets a user-controllable time delay value inserted after the <code>AP.Sweep.Start</code> command is executed, before the first data point is taken. This can be valuable when the device under test needs a certain amount of setup time before it operates normally, or to allow for full autoranging and other time within the instrument. In nested sweeps, this Pre-Sweep Delay is inserted before the start of each sweep of the test.</p> <p>The Pre-Sweep Delay field is located on the right half of the large version of the Sweep panel, below the Data 3-Data 6 Limits buttons.</p>
See Also	<code>AP.Sweep.Start</code>
Example	See example for <code>AP.Sweep.Append</code> .

AP.Sweep.Recompare

Method

Syntax	<code>AP.Sweep.Recompare</code>
Result	Boolean
	<p><i>True</i> Recompare successful.</p> <p><i>False</i> Recompare failed.</p>
Description	<p>This command causes any sweep result currently in memory to be regraphed and compared to limits if limit files are attached to any Data (Data 1 - Data 6) variable via the test configuration or usage of the <code>AP.Sweep.Data(n).Limits</code> command.</p> <p>This command is equivalent to F7 in APWIN.</p>
See Also	<code>AP.Sweep.Data1.Limits</code> , <code>AP.Sweep.Data2.Limits</code> , <code>AP.Sweep.Data3.Limits</code> , <code>AP.Sweep.Data4.Limits</code> , <code>AP.Sweep.Data5.Limits</code> , <code>AP.Sweep.Data6.Limits</code>

AP.Sweep.Repeat

Property

Syntax `AP.Sweep.Repeat`

Data Type Boolean

True Repeat sweep continuously.

False Do not repeat sweep continuously.

Description This command enables or disables repeating the currently defined sweep indefinitely.

See Also `AP.Sweep.Append`, `AP.Sweep.StartWithAppend`,
`AP.Sweep.StartWithRepeat`

Example

```

Sub Main
    AP.Application.NewTest      'Start with New Test
    AP.Gen.Output = True       'Generator Output ON
    AP.Anlr.ChAInput = 2
    AP.Application.PanelOpen apbPanelSweepSmall _
        'Display Sweep Panel

    AP.Prompt.Text = "Press Continue to Stop _
        Sweep." 'Prompt text
    AP.Prompt.FontSize = 8     'Set font size to 8 point
    AP.Prompt.Position(-1,-1,190,120)'Location and size

    Begin Dialog UserDialog 310,154,"Sweep Controler"
        PushButton 100,7,100,21,"Single sweep",.PushButton1
        PushButton 30,35,250,21,"Single sweep and _
            Append",.PushButton3
        PushButton 30,56,250,21,"Start repeating _
            sweep",.PushButton2
        PushButton 30,77,250,21,"Start repeating sweep _
            with Append",.PushButton4
        CancelButton 60,119,190,21
    End Dialog
    Dim dlg As UserDialog

    DisplayDialog:
    Select Case Dialog (dlg)
        Case 0
            End

```

```

        AP.Sweep.Append = False
        AP.Sweep.Repeat = False
    Case 1    'Run single sweep
        AP.Sweep.Append = False
        AP.Sweep.Repeat = False
        Sweep.Start
    Case 2    'Run sweep and append data
        AP.Sweep.Repeat = False
        AP.Sweep.StartWithAppend
    Case 3    'Run repeating sweep
        AP.Sweep.Append False
'Display prompt
        AP.Prompt.ShowWithContinueAndStopSweep
        AP.Sweep.StartWithRepeat           'Start sweep
        AP.Sweep.Repeat = False
    Case 4    'Run repeating sweep and append data
        AP.Sweep.Append = True
        AP.Sweep.Repeat = True
'Display prompt
        AP.Prompt.ShowWithContinueAndStopSweep
        AP.Sweep.Start           'Start sweep
        AP.Sweep.Repeat = False
End Select
GoTo DisplayDialog
End Sub

```

AP.Sweep.Reprocess

Method

Syntax

AP.Sweep.Reprocess

Result

Boolean

True Reprocess successful.

False Reprocess failed.

Description

This command instructs APWIN to cause the third phase of the following process to be performed.

FFT-based (batch mode) DSP programs have three distinct, sequential phases to their operation.

First, data is accumulated into the acquisition buffer until the buffer is filled to the specified acquisition length.

Second, a Fast Fourier Transform (FFT) is performed to obtain amplitude (and sometimes phase) versus frequency data which is stored in a different memory buffer from the acquired signal (amplitude versus time).

Third, a post-processed version of the amplitude versus time or amplitude versus frequency data (depending upon sweep Source 1 and Data 1 or 2) is transmitted from the DSP module in the test system to the computer for graphing by APWIN software.

This command is equivalent to Ctrl+F6 in APWIN.

Example

```
Sub Main
  AP.Log.Enable = 0      'Disable log file
  AP.File.OpenTest "CODEC.AT2" 'Open test to create a _
    masking curve limit file.
  AP.Sweep.Start          'Start sweep.
  AP.File.SaveDataAs "MASK.ADL" 'Save masking curve as _
    MASK.ADL limit file.
  AP.Application.NewData 'Remove Masking curve data _
    from memory.
  AP.S2CDsp.FastTest.Mode = 0 'Set the DSP panel _
    measurement field to spectrum mode.
  AP.Sweep.Reprocess          'Reprocess the acquired _
    waveform and display spectrum results.
                                'Attach Sweep table.
  AP.Sweep.Source1.Table ("CODEC.ADS",0)
  AP.S2CDsp.FastTest.Mode = 1 'Set the DSP panel _
    measurement field to response mode.
  AP.Sweep.Reprocess          'Reprocess the acquired _
    waveform and display response results.

  'Attach upper limits to Data1 & Data3.
  AP.Sweep.Data1.Limits("MASK.ADL", 1, True)
  AP.Sweep.Data3.Limits("Mask.ADL", 1, True)

  AP.S2CDsp.FastTest.Mode = 2 'Set DSP measurement _
    mode to distortion.
```

```

AP.Sweep.Reprocess           'Reprocess the acquired _
                               waveform and display distortion results.
AP.S2CDsp.FastTest.Mode = 3     'Set the DSP panel _
                               measurement field to noise mode.
AP.Sweep.Reprocess           'Reprocess the acquired _
                               waveform and display noise results.
End Sub

```

AP.Sweep.Retransform

Method

Syntax **AP.Sweep.Retransform**

Result Boolean

True Retransform successful.
False Retransform failed.

Description

This command instructs APWIN to cause the second and third phases of the following process to be performed.

FFT-based (batch mode) DSP programs have three distinct, sequential phases to their operation.

First, data is accumulated into the acquisition buffer until the buffer is filled to the specified acquisition length.

Second, a Fast Fourier Transform (FFT) is performed to obtain amplitude (and sometimes phase) versus frequency data which is stored in a different memory buffer from the acquired signal (amplitude versus time).

Third, a post-processed version of the amplitude versus time or amplitude versus frequency data (depending upon sweep Source 1 and Data 1 or 2) is transmitted from the DSP module in the test system to the computer for graphing by APWIN software.

This command is equivalent to F6 in APWIN.

Example

```

Sub Main
  AP.Application.NewTest
  AP.Gen.Output = 1

```



```

AP.Anlr.ChAInput = 2
AP.S2CDsp.Program = 2      'Select FFT program
AP.S2CDsp.FFT.InputFormat = 1'Select Low BW(A/D) input
AP.S2CDsp.FFT.Length = 6  'Set FFT length to 16384
AP.S2CDsp.FFT.Window = 0  'Set FFT window to BH4
AP.Sweep.Data1.Id = 6023  'Set sweep panel Data 1 _
    to Fft.Ch.1 Ampl
AP.Sweep.Source1.Id = 5515 'Set sweep panel Source 1 _
    to Fft.FFT Freq
AP.Sweep.Start
AP.Sweep.Append = 1
AP.S2CDsp.FFT.Window = 1'Set FFT window to Hann
AP.Sweep.Rettransform
AP.S2CDsp.FFT.Window = 2'Set FFT window to Flat-Top
AP.Sweep.Rettransform
AP.S2CDsp.FFT.Window = 3'Set FFT window to Equiripple
AP.Sweep.Rettransform
AP.S2CDsp.FFT.Window = 4'Set FFT window to None
AP.Sweep.Rettransform
AP.Data.OptimizeDisplay 0
End Sub

```

AP.Sweep.ReverseChannels

Method

Syntax `AP.Sweep.ReverseChannels (ByVal Reversed As Boolean)`

Parameters	Name	Description
	<i>Reversed</i>	True = Change channel to alternate channel. False = Return channel to previous state.

Description This command selects the alternate channel from the present settings for the generator output and analyzer Function meter input selection. If channel A is selected for the generator output and the analyzer Function meter and this command is executed using a 1 for the command argument channel B will be selected for the generator and analyzer Function meter. To revert to the previous state use the command argument 0.

AP.Sweep.SinglePoint

Property**Syntax** `AP.Sweep.SinglePoint`**Data Type** Boolean*True* Enable single point sweep.*False* Disable single point sweep.**Description** This command sets the Source 1 Sweep to Single Point mode. When a sweep is initiated (`AP.Sweep.Start`) the Data Editor will be automatically displayed and a single measurement taken at the Sweep Start value of Source 1.**See Also** `AP.Sweep.Source1.Start`**Example** Sub Main

```

AP.File.OpenTest "SWEEPFFT.AT2"
AP.Sweep.Repeat = False
AP.Sweep.SinglePoint = False
AP.Sweep.Stereo = False
AP.Sweep.Timeout("sec") = 3
AP.Sweep.Source1.Id = 5051     'Set Source 1 Gen Freq
AP.Sweep.Source1.LogLin = 1
AP.Sweep.Source1.Start("Hz") = 20000
AP.Sweep.Source1.Stop("Hz") = 20
AP.Sweep.Source1.Steps = 15
AP.Sweep.Source1.AutoDiv = False
AP.Sweep.Source1.Div = 10
AP.Sweep.Source2.Id = 5052 'Set Source 2 Gen Ampl A
AP.Sweep.Source2.LogLin = 1
AP.Sweep.Source2.Start("Vrms") = 5
AP.Sweep.Source2.Steps = 2
AP.Sweep.Source2.Stop("Vrms") = 1
AP.Sweep.Start

```

End Sub

AP.Sweep.Source1.AutoDiv

Property**Syntax** `AP.Sweep.Source1.AutoDiv`**Data Type** Boolean

True automatically select the number of divisions.
False Use the number of divisions defined by the
 AP.Sweep.Source1.Div command.

Description This command enables or disables automatic selection of the number of linear horizontal axis divisions displayed for Source 1 sweeps.

See Also AP.Sweep.Source1.Div, AP.Sweep.Source1.LogLin

Example See example for AP.Sweep.SinglePoint.

AP.Sweep.Source1.Div

Property

Syntax AP.Sweep.Source1.Div

Data Type Long Number of divisions displayed.

Description This command sets the number of divisions that are to be displayed for a linear horizontal axis for a Source 1. The AP.Sweep.Source1.AutoDiv must be disabled.

See Also AP.Sweep.Source1.AutoDiv, AP.Sweep.Source1.LogLin

Example See example for AP.Sweep.SinglePoint.

AP.Sweep.Source1.EndOn

Property

Syntax AP.Sweep.Source1.EndOn(ByVal Unit As String)

Data Type Double Refer to the setting or reading defined by the AP.Sweep.Source1.Id command to determine the appropriate range of acceptable values.

Parameters	Name	Description
	<i>Unit</i>	Refer to the setting or reading defined by the AP.Sweep.Source1.Id command to determine the appropriate unit selections.

Description This command sets the Sweep End value for an external sweep. The sweep will be considered to have finished when the Source 1

parameter reverses its direction (starts to change in the direction from Stop to Start) to the End On value.

It is frequently necessary to make and graph a series of measurements where some external, uncontrollable source is the independent variable. Common examples include frequency response measurements or other swept tests where the sweeping signal is pre-recorded on a test tape or test CD, or testing of a transmission link where a remote generator (not under control of APWIN software) is providing the signal. In these cases, APWIN software cannot control the values, direction of progression (high to low versus low to high), or dwell times of the signal. APWIN can, however, measure the changing parameter of the incoming signal (usually frequency but sometimes level) and use those measurements as the X-axis calibration. This mode of operation, where a measurement (Reading) drives the data-taking process and calibrates the X-axis, is called External Sweep.

Example

```
Sub Main
'This test requires an external sweep source.
  AP.File.OpenTest "SweepD.at2"
  AP.Sweep.Source1.Id = 5901 'Set Source 1 to Anlr.FreqA
  AP.Sweep.Source1.EndOn("Hz") = 2500 'Set Sweep End _
    On to 2.5kHz
  AP.Sweep.Source1.MinLevelID = 5903 'Select _
    Anlr.LevelA for MinLevel
  AP.Sweep.Source1.MinLevel("dBu") = -40 'Set Min _
    Level to 100mV
  AP.Sweep.Source1.Spacing("%") = 3 'Set Spacing to 3%
  AP.Sweep.Start           'Wait for external sweep
End Sub
```

AP.Sweep.Source1.Id

Property

Syntax	<code>AP.Sweep.Source1.Id</code>
Data Type	Long Instrument Parameter ID#.

Description This command is used to select the instrument parameter which will define settings or return readings, in the case of external sweeps, for Source 1.

Refer to Appendix B to obtain instrument parameter identification numbers.

Example See example for `AP.Sweep.SinglePoint`.

AP.Sweep.Source1.LogLin

Property

Syntax `AP.Sweep.Source1.LogLin`

Data Type Integer

0 Logarithmic horizontal axis and step type.

1 Linear horizontal axis and step type.

Description This command determines the Source 1 horizontal axis type and the sweep step type.

See Also `AP.Sweep.Source1.Div`, `AP.Sweep.Source1.AutoDiv`

Example See example for `AP.Sweep.SinglePoint`.

AP.Sweep.Source1.MinLevel

Property

Syntax `AP.Sweep.Source1.MinLevel(ByVal Unit As String)`

Data Type Double Refer to the reading defined by the `AP.Sweep.Source1.MinLevelId` command to determine the appropriate range of acceptable values.

Parameters	Name	Description
	<i>Unit</i>	Refer to the reading defined by the <code>AP.Sweep.Source1.MinLevelId</code> command to determine the appropriate unit selections.

Description This command sets the minimum input signal level at which measurements will be taken during an external sweep (reading instead of setting at Source 1). The purpose of this command is to avoid taking

measurements during the "dead time" between tracks of a test tape or test CD, when noise still produces some finite signal level.

It is frequently necessary to make and graph a series of measurements where some external, uncontrollable source is the independent variable. Common examples include frequency response measurements or other swept tests where the sweeping signal is pre-recorded on a test tape or test CD, or testing of a transmission link where a remote generator (not under control of APWIN software) is providing the signal. In these cases, APWIN software cannot control the values, direction of progression (high to low versus low to high), or dwell times of the signal. APWIN can, however, measure the changing parameter of the incoming signal (usually frequency but sometimes level) and use those measurements as the X-axis calibration. This mode of operation, where a measurement (Reading) drives the data-taking process and calibrates the X-axis, is called External Sweep.

See Also

AP.Sweep.Data2.Id, AP.Sweep.MinLevelSource

AP.Sweep.Source1.MinLevelId

Property**Syntax**

AP.Sweep.Source1.MinLevelId

Data Type

Long Instrument Parameter ID#.

Description

This command is used to select the instrument parameter which will define settings or return readings, in the case of external sweeps, for Source 1.

Refer to Appendix B to obtain instrument parameter identification numbers.

AP.Sweep.Source1.Multiply

Property**Syntax**

AP.Sweep.Source1.Multiply

Data Type

Double

Description This command sets the Source 1 Log Sweep Multiply factor used to determine the next Source 1 sweep setting.

See Also `AP.Sweep.Source1.Start`, `AP.Sweep.Source1.Stop`, `AP.Sweep.Source1.LogLin`, `AP.Sweep.Source1.Steps`

AP.Sweep.Source1.Spacing

Property

Syntax `AP.Sweep.Source1.Spacing(ByVal Unit As String)`

Data Type Double

Parameters

Name	Description
<i>Unit</i>	% unit only.

Description This command sets the minimum change of the Source 1 Reading (ID#) Property parameter required to allow an additional external sweep measurement to be taken. This setting is only available for external sweeps.

Example See example for `AP.Sweep.Source1,EndOn`.

AP.Sweep.Source1.Start

Property

Syntax `AP.Sweep.Source1.Start(ByVal Unit As String)`

Data Type Double Refer to the setting or reading defined by the `AP.Sweep.Source1.Id` command to determine the appropriate range of acceptable values.

Parameters

Name	Description
<i>Unit</i>	Refer to the setting or reading defined by the <code>AP.Sweep.Source1.Id</code> command to determine the appropriate unit selections.

Description This command sets the first setting value to be sent to the instrument parameter specified as Source 1 and to be displayed on the graph horizontal axis. In the case of an external sweep (a reading selected at Source 1 instead of a setting), this value determines the graph horizontal axis end point and, in conjunction with the

`AP.Sweep.Source1.Stop` command, defines the expected direction of change of the Source 1 parameter during the sweep.

See Also `AP.Sweep.Source1.Stop`

Example See example for `AP.Sweep.Append`.

AP.Sweep.Source1.Steps

Property

Syntax `AP.Sweep.Source1.Steps`

Data Type Long

Description This command sets the number of Source 1 steps that a log or linear sweep makes between the Source 1 Start and Stop values.

See Also `AP.Sweep.Source1.Start`, `AP.Sweep.Source1.Stop`, `AP.Sweep.Source1.LogLin`, `AP.Sweep.Source1.StepSize`

Example See example for `AP.Sweep.SinglePoint`.

AP.Sweep.Source1.StepSize

Property

Syntax `AP.Sweep.Source1.StepSize(ByVal Unit As String)`

Data Type Double Source 1 step size.

Description This command sets the Source 1 Linear Sweep Step Size used to determine the next Source 1 sweep setting.

Name	Description
<i>Unit</i>	Refer to the setting or reading defined by the <code>AP.Sweep.Source1.Id</code> command to determine the appropriate unit selections.

See Also `AP.Sweep.Source1.Start`, `AP.Sweep.Source1.Stop`, `AP.Sweep.Source1.LogLin`, `AP.Sweep.Source1.Steps`

AP.Sweep.Source1.Stop

Property

Syntax	AP.Sweep.Source1.Stop (ByVal <i>Unit</i> As String)	
Data Type	Double	Refer to the setting or reading defined by the <code>AP.Sweep.Source1.Id</code> command to determine the appropriate range of acceptable values.
Parameters	Name	Description
	<i>Unit</i>	Refer to the setting or reading defined by the <code>AP.Sweep.Source1.Id</code> command to determine the appropriate unit selections.
Description	This command sets the last setting value to be sent to the instrument parameter specified as Source 1 and to be displayed on the graph horizontal axis. In the case of an external sweep (a reading selected at Source 1 instead of a setting), this value determines the graph horizontal axis end point and, in conjunction with the <code>AP.Sweep.Source1.Start</code> command, defines the expected direction of change of the Source 1 parameter during the sweep.	
See Also	<code>AP.Sweep.Source1.Start</code>	
Example	See example for <code>AP.Sweep.Append</code> .	

AP.Sweep.Source1.Table

Method

Syntax	AP.Sweep.Source1.Table (ByVal <i>FileName</i> As String, ByVal <i>Column</i> As Integer)	
Parameters	Name	Description
	<i>FileName</i>	Any valid DOS path and file name. The file must be an APWIN sweep file (.ads). Enter "None" for the file name to remove the sweep file from Source1.
	<i>Column</i>	0 = Source 1 settings. 1 = Data 1 measurements. 2 = Data 2 measurements. 3 = Data 3 measurements. 4 = Data 4 measurements. 5 = Data 5 measurements.

6 = Data 6 measurements.

7 = Source 2 settings.

Result	Boolean
	<i>True</i> File attachment successful.
	<i>False</i> File attachment failed.

Description This command attaches a sweep file to Source 1. Values in the file will be used as Source 1 settings, rather than Start, Stop, Steps, and Multiply, or Stepsize values. The Start and Stop values will continue to be used to define the horizontal end points of the graph.

AP.Sweep.Source2.Id

Property

Syntax `AP.Sweep.Source2.Id`

Data Type Long Instrument Parameter ID#.

Description This command is used to select the instrument parameter, which will define settings for Source 2.

Refer to Appendix B to obtain instrument parameter identification numbers.

Example See example for `AP.Sweep.SinglePoint`.

AP.Sweep.Source2.LogLin

Property

Syntax `AP.Sweep.Source2.LogLin`

Data Type Integer

<i>0</i>	Logarithmic step type.
<i>1</i>	Linear step type.

Description This command determines if the sweep steps will be Logarithmically or linear spaced.

Example See example for `AP.Sweep.SinglePoint`.

AP.Sweep.Source2.Multiply

Property

Syntax	<code>AP.Sweep.Source2.Multiply</code>
Data Type	Double
Description	This command sets the Source 2 Log Sweep multiply factor used to determine the next Source 2 Sweep setting.
See Also	<code>AP.Sweep.Source2.Start</code> , <code>AP.Sweep.Source2.Stop</code> , <code>AP.Sweep.Source2.LogLin</code> , <code>AP.Sweep.Source2.Steps</code>

AP.Sweep.Source2.Start

Property

Syntax	<code>AP.Sweep.Source2.Start (ByVal Unit As String)</code>	
Data Type	Double	Refer to the setting defined by the <code>AP.Sweep.Source2.Id</code> command to determine the appropriate range of acceptable values.
Parameters	Name	Description
	<i>Unit</i>	Refer to the setting defined by the <code>AP.Sweep.Source2.Id</code> command to determine the appropriate range of acceptable values.
Description	This command sets the first setting to be used in the Source 2 sweep.	
See Also	<code>AP.Sweep.Source2.Stop</code>	
Example	See example for <code>AP.Sweep.SinglePoint</code> .	

AP.Sweep.Source2.Steps

Property

Syntax	<code>AP.Sweep.Source2.Steps</code>
Data Type	Long
Description	This command sets the number of Source 2 steps that a log or linear sweep makes between the Source 2 Start and Stop values.

See Also `AP.Sweep.Source2.Start`, `AP.Sweep.Source2.Stop`,
`AP.Sweep.Source2.LogLin`, `AP.Sweep.Source2.StepSize`

Example See example for `AP.Sweep.SinglePoint`.

AP.Sweep.Source2.StepSize

Property

Syntax `AP.Sweep.Source2.StepSize`(ByVal *Unit* As String)

Data Type Double Refer to the setting defined by the
`AP.Sweep.Source2.Id` command to determine the
appropriate range of acceptable values.

Parameters	Name	Description
	<i>Unit</i>	Refer to the setting defined by the <code>AP.Sweep.Source2.Id</code> command to determine the appropriate unit selections.

Description This command sets the Source 2 Linear Sweep Step Size used to
determine the next Source 2 sweep setting.

See Also `AP.Sweep.Source2.Start`, `AP.Sweep.Source2.Stop`,
`AP.Sweep.Source2.LogLin`, `AP.Sweep.Source2.Steps`

AP.Sweep.Source2.Stop

Property

Syntax `AP.Sweep.Source2.Stop`(ByVal *Unit* As String)

Data Type Double Refer to the setting defined by the
`AP.Sweep.Source2.Id` command to determine the
appropriate range of acceptable values.\

Parameters	Name	Description
	<i>Unit</i>	Refer to the setting defined by the <code>AP.Sweep.Source2.Id</code> command to determine the appropriate unit selections.

Description This command sets the last setting to be used in the Source 2 sweep.

See Also `AP.Sweep.Source2.Start`

AP.Sweep.Spectrum

Method**Syntax** `AP.Sweep.Spectrum`**Result** Boolean*True* Change to Spectrum Display successful.*False* Display change not successful.**Description** This command configures the Sweep Panel to produce a Spectrum display when a sweep is run.**New Test configuration functionality:**

In this situation the user has not defined the Sweep Panel to display a Spectrum but has selected from one of the Digital Analyzer selections listed below. If the user has not selected from one of the Digital Analyzer selections listed below this command is not active.

When this command is executed default values are automatically entered into the sweep panel settings to setup the sweep to display the default Spectrum when run. Each Digital Analyzer selection listed below has it's own default sweep panel settings for a frequency domain display.

User defined test functionality:

In this situation the user has loaded a previously saved test. If the user has redefined any of the default sweep panel settings for any or all of the Digital Analyzer selections and then saved the settings as a test then all of the settings for all of the Digital Analyzer selections will be restored when the test is loaded. The user can then switch between any of the Digital Analyzer selections listed below and the previously defined settings will be restored.

Digital Analyzer selections:

FFT spectrum analyzer (fft)

Digital interface analyzer (intervu)

Multitone audio analyzer (fasttest)

Quasi-anechoic acoustical tester (mls)

See Also `AP.Sweep.Waveform`**Example** `Sub Main`

```

AP.Application.NewTest
AP.Gen.Output = True 'Turn Analog Generator Output ON
AP.Anlr.ChAInput = 2
AP.Anlr.ChBInput = 2
AP.S2CDsp.Program = 2 'Select FFT Spectrum Analyzer
AP.S2CDsp.FFT.InputFormat = 1 'HiRes A/D @65536 Input

AP.Sweep.Spectrum 'Setup Sweep panel for Spectrum _
    Display
AP.Application.Page = 2 'Display Graph on Page 2
AP.Sweep.Start 'Acquire and display Spectrum data
Wait 5
AP.Sweep.Waveform 'Retransform and display _
    Waveform data
End Sub

```

AP.Sweep.Start

Method

Syntax

AP.Sweep.Start

Result

Boolean

True Sweep completed successfully.

False Sweep terminated abnormally.

Description

This command initiates a sweep.

Note: When using this command from an external application execution of additional commands will not be held off if the `AP.Sweep.Repeat` command is set to `True`. The `AP.Sweep.Repeat` command is also affected by the `AP.Sweep.StartWithRepeat` command.

See Also

`AP.Sweep.StartWithAppend`, `AP.Sweep.StartWithRepeat`

Example

```

Sub Main
    AP.File.OpenTest "FRQ-RESP.AT2" 'Open frequency _
        response test.
    AP.Sweep.Start 'Start sweep.
    AP.File.SaveDataAs "FRQ-RESP.DAT" 'Save data.
End Sub

```

```

AP.File.OpenTest "THD-FRQ.AT2"      'Open total _
    harmonic distortion + noise test.
AP.Sweep.Start                    'Start sweep.
AP.File.SaveDataAs "THD-FRQ.DAT"    'Save data.

AP.File.OpenTest "RESIDNOI.AT2"    'Open residual _
    noise test.
AP.Sweep.Start                    'Start sweep.
AP.File.SaveDataAs "RESIDNOI.DAT"   'Save data.
End Sub

```

AP.Sweep.StartNoWait

Method

Syntax	<code>AP.Sweep.StartNoWait</code>
Result	Boolean
	<i>True</i> Sweep process started successfully.
	<i>False</i> Sweep process not started successfully.
Description	This command initiates a sweep process and then continues macro execution.
Example	See example for <code>AP.Sweep.Stop</code> .

AP.Sweep.StartWithAppend

Method

Syntax	<code>AP.Sweep.StartWithAppend</code>
Description	This command initiates a sweep in append mode which is equivalent to pressing the Ctrl+F9 function key.
See Also	<code>AP.Sweep.Start</code> , <code>AP.Sweep.StartWithRepeat</code>
Example	See example for <code>AP.Sweep.SinglePoint</code> .

AP.Sweep.StartWithRepeat

Method

Syntax	<code>AP.Sweep.StartWithRepeat</code>
Description	This command initiates a sweep in repeat mode which is equivalent to pressing the Alt+F9 function key.
See Also	<code>AP.Sweep.Start</code> , <code>AP.Sweep.StartWithAppend</code>
Example	See example for <code>AP.Sweep.Repeat</code> .

AP.Sweep.Stereo

Property

Syntax	<code>AP.Sweep.Stereo</code>
Data Type	Boolean
	<i>True</i> Enable Stereo Sweep
	<i>False</i> Disable Stereo Sweep.
Description	This command enables or disables the stereo sweep feature on the Sweep panel.
Example	See example for <code>AP.Sweep.SinglePoint</code> .

AP.Sweep.Stop

Method

Syntax	<code>AP.Sweep.Stop</code>
Result	Boolean
	<i>True</i> Sweep terminated successfully.
	<i>False</i> Sweep not terminated.
Description	This command terminates a running sweep.
See Also	<code>AP.Sweep.IsRunning</code>
Example	<pre>Dim Halt As Boolean Sub Main Halt = False</pre>


```

AP.Application.NewTest
AP.Gen.Output = True
AP.Anlr.ChAInput = 2
AP.Sweep.Source1.Steps = 200

AP.Application.SetWatchDogTimer1(5.0,False)

AP.Sweep.StartNoWait

Do
    'nothing
Loop While Halt = False

End Sub
Sub APEvent_OnWatchDogTimeout(ByVal Id As Long)
    If Id = 1 Then
        Halt = True
        If AP.Sweep.IsRunning = True Then
            AP.Sweep.Stop
            Debug.Print "Sweep Stopped"
        End If
    End If
End Sub

```

AP.Sweep.Timeout

Property

Syntax `AP.Sweep.Timeout`(ByVal *Unit* As String)

Data Type Double Timeout values of 0 to 3000 seconds (50 minutes) are allowed.

Parameters	Name	Description
	<i>Unit</i>	Sec unit only.

Description This command sets the timeout used during settling comparisons. If settling cannot be achieved during the Timeout duration, the average of its last 6 readings is computed and returned. Timeout serves as a "safety valve" to avoid excessive delays or hang-up when the data has more variation that present settling parameters will accept.

In a graph display, each timeout point is indicated by a white T at the upper margin of the graph, directly above the plotted point. In the Data Editor, each timeout point is indicated by the letter T following the data. In the Log File, the Pass/Fail message (if enabled) shows the total number of timeouts which occurred during a sweep. However, a timeout is not treated as a failure if the eventual averaged data was within limits. The Log File may also includes a line for each measured point which timed out durring the sweep resulting in a row showing the measured value and a letter T.

See Appendix A for Settling Algorithm and parameter name descriptions.

Example See example for `AP.Sweep.SinglePoint`.

AP.Sweep.Waveform

Method

Syntax `AP.Sweep.Waveform`

Result Boolean

True Change to Waveform Display successful.
False Display change not successful.

Description This command configures the Sweep Panel to produce a Waveform display when a sweep is run.

New Test configuration functionality:

In this situation the user has not defined the Sweep Panel to display a Waveform but has selected from one of the Digital Analyzer selections listed below. If the user has not selected from one of the Digital Analyzer selections listed below this command is not active.

When this command is executed default values are automatically entered into the sweep panel settings to setup the sweep to display the default Waveform when run. Each Digital Analyzer selection listed below has it's own default sweep panel settings for a time domain display.

User defined test functionality:

In this situation the user has loaded a previously saved test. If the user has redefined any of the default sweep panel settings for any or all of the Digital Analyzer selections and then saved the settings as a test then all of the settings for all of the Digital Analyzer selections will be restored when the test is loaded. The user can then switch between any of the Digital Analyzer selections listed below and the previously defined settings will be restored.

Digital Analyzer selections:

FFT spectrum analyzer (fft)
Digital interface analyzer (intervu)
Multitone audio analyzer (fasttest)
Quasi-anechoic acoustical tester (mls)

See Also

AP.Sweep.Spectrum

Example

See example for AP.Sweep.Spectrum.

User Notes

User Notes

Switcher

AP.SWR.ChABIn

Property

Syntax `AP.SWR.ChABIn`

Data Type Long 0 - 192

Description This command sets the channel A and B connections of the Input switchers simultaneously. The channel A input is set to the specified channel number. The channel B input differs from the specified channel number by the value of the `AP.SWR.ChBOffset` command.

Channel numbers 1 to 192 are available, where 0 means all channels off. Any other number results in no action taken.

See Also `AP.SWR.ChBOffset`

Example See example for `AP.SWR.Mode`.

AP.SWR.ChABInOut

Property

Syntax `AP.SWR.ChABInOut`

Data Type Long 0 - 192

Description This command sets the channel A and B connections of the Input and Output switchers simultaneously. The channel A input is set to the specified channel number. The channel B input differs from the specified channel number by the value of the `AP.SWR.ChBOffset` command. The channel A output differs from the specified channel number by the value of the `AP.SWR.OutOffset` command. The channel B output differs from the specified channel number by the sum of the values of the `AP.SWR.ChBOffset` and the `AP.SWR.OutOffset` commands.

Channel numbers 1 to 192 are available, where 0 means all channels off. Any other number results in no action taken.

See Also `AP.SWR.ChBOffset`, `AP.SWR.OutOffset`

Example See example for `AP.SWR.Mode`.

AP.SWR.ChABOut

Property

Syntax `AP.SWR.ChABOut`

Data Type Long 0 - 192

Description This command sets the channel A and B connections of the Output switchers simultaneously. The channel A output is set to the specified channel number. The channel B output differs from the specified channel number by the value of the `AP.SWR.ChBOffset` command.

Channel numbers 1 to 192 are available, where 0 means all channels off. Any other number results in no action taken.

See Also `AP.SWR.ChBOffset`

Example See example for `AP.SWR.Mode`.

AP.SWR.ChAIn

Property

Syntax `AP.SWR.ChAIn`

Data Type Long 0 - 192

Description This command sets the switcher channel A input channel.

Channel numbers 1 to 192 are available, where 0 means all channels off. Any other number results in no action taken.

Example See example for `AP.SWR.Mode`.

AP.SWR.ChAInOut

Property

Syntax `AP.SWR.ChAInOut`

Data Type Long 0 - 192

Description	This command sets the channel A connections of the Input and Output switchers simultaneously. The channel A input is set to the specified channel number. The channel A output differs from the specified channel number by the value of the <code>AP.SWR.OutOfSet</code> command. Channel numbers 1 to 192 are available, where 0 means all channels off. Any other number results in no action taken.
Example	See example for <code>AP.SWR.Mode</code> .

AP.SWR.ChAOut

Property

Syntax	<code>AP.SWR.ChAOut</code>
Data Type	Long 0 - 192
Description	This command sets the switcher channel A Output channel. Channel numbers 1 to 192 are available, where 0 means all channels off. Any other number results in no action taken.
Example	See example for <code>AP.SWR.Mode</code> .

AP.SWR.ChBIn

Property

Syntax	<code>AP.SWR.ChBIn</code>
Data Type	Long 0 - 192
Description	This command sets the switcher channel B Input channel. Channel numbers 1 to 192 are available, where 0 means all channels off. Any other number results in no action taken.
Example	See example for <code>AP.SWR.Mode</code> .

AP.SWR.ChBinOut**Property****Syntax** `AP.SWR.ChBinOut`**Data Type** Long 0 - 192**Description** This command sets the channel B connections of the Input and Output switchers simultaneously. The channel B input is set to the specified channel number. The channel B output differs from the specified channel number by the value of the `AP.SWR.OutOffset` command.

Channel numbers 1 to 192 are available, where 0 means all channels off. Any other number results in no action taken.

See Also `AP.SWR.OutOffset`**Example** See example for `AP.SWR.Mode`.**AP.SWR.ChBOffset****Property****Syntax** `AP.SWR.ChBOffset`**Data Type** Long 1 - 192**Description** This command determines the channel number difference between channel B and the specified channel A.**See Also** `AP.SWR.ChABIn`, `AP.SWR.ChABInOut`, `AP.SWR.ChABOut`**Example** See example for `AP.SWR.Mode`.**AP.SWR.ChBOut****Property****Syntax** `AP.SWR.ChBOut`**Data Type** Long 0 - 192**Description** This command sets the channel B output channel.

Channel numbers 1 to 192 are available, where 0 means all channels off. Any other number results in no action taken.

Example See example for `AP . SWR . Mode .`

AP.SWR.Mode

Property

Syntax `AP . SWR . Mode`

Data Type Integer

<i>0</i>	B independent from A: when selected, channels A and B may be independently set to any channel number within their range. This is the normal mode for most operation.
<i>1</i>	B = All outputs driven, A = off: when selected, the switcher B common input is connected to all 12 outputs on each Output switcher module and the A common input is disconnected. Both the A and B output fields will be gray and unavailable for settings in this mode since all connections are defined by the mode itself. This mode enables connection of a single generator signal to all device inputs, which may be a requirement of a burn-in rack or a life test.
<i>2</i>	B = All outputs driven except # selected for A: when selected, the A common input connects to the channel number entered in the A output field and the B common input connects to the remaining 11 channels on that switcher and to all 12 channels of all other Output switchers connected. The purpose of this mode is for worst-case crosstalk measurements, so that all except one channels of a multi-track or multi-channel recorder or mixing console are driven while the output signal from the one un-driven channel is measured. This mode is normally used with a nested sweep with Source 2 on the Sweep panel set to scan channel A input and output through all possible device channels while Source 1 is commonly set for a frequency sweep to measure selective crosstalk across the audio spectrum.

Description This command sets the switcher output configuration

Example

```

Const INDEPENDENT As Integer = 0      'B independent of A
Const B_ONLY_A_OFF As Integer = 1    'All B on, All A off
Const COMPLEMENT As Integer = 2     'All B except _
    channel specified by A
Sub Main
    Dim switch As Integer, signal As Double, msg As String

    signal = 1.0                      'Use 1 V signal
    AP.Gen.ChAAmpl("V") = signal      'Set gen out level
    AP.Gen.Output = True               'Turn output on
    AP.SWR.Mode = INDEPENDENT         'Set Mode
    AP.SWR.OutOffset = 1             'ChAOut = ChAIn + 1
    AP.SWR.ChBOffset = 2            'ChB = ChA + 2
    For Switch = 1 To 6               'Sweep switches 1 to 6
        AP.SWR.ChAIn = Switch
        'AP.SWR.ChBIn = Switch        'Any of these switch
        'AP.SWR.ChABIn = Switch       ' commands can be used
        'AP.SWR.ChAOut = Switch       ' to sweep the channel
        'AP.SWR.ChBOut = Switch       ' A and/or B
        'AP.SWR.ChABOut = Switch     ' input and/or output
        'AP.SWR.ChAInOut = Switch     ' switches
        'AP.SWR.ChBInOut = Switch     ' with the appropriate
        'AP.SWR.ChABInOut = Switch    ' offsets
        AP.Anlr.ChALevelTrig
        While AP.Anlr.ChALevelReady = 0
            Wend
            rdg = AP.Anlr.ChALevelRdg("V")
            If rdg > 0.5 * signal Then 'any signal
                msg = msg & "Ch A In " & Switch & "<-> Ch A _
                    Out " & Switch + AP.SWR.OutOffset & Chr(13)
            End If
        Next Switch
        AP.Prompt.Text = msg
        AP.Prompt.ShowWithContinue
        Stop
    End Sub

```

AP.SWR.OutOffset**Property**

Syntax	<code>AP.SWR.OutOffset</code>
Data Type	Long 0 - 192
Description	This command determines the channel number difference between the specified channel number and the output switcher channel.
See Also	<code>AP.SWR.ChAInOut</code> , <code>AP.SWR.ChBInOut</code> , <code>AP.SWR.ChABInOut</code>
Example	See example for <code>AP.SWR.Mode</code> .

User Notes

Sync/Ref Input

AP.Sync.DelayRdg

Property

Syntax `AP.Sync.DelayRdg(ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
------	-------------

<i>Unit</i>	The following units are available, sec.
-------------	---

Description

This command returns a settled reading for the Sync Delay, In from Ref In field on the Sync/Ref Input panel. The reading is the time (phase) delay of the selected front panel XLR, BNC, or optical connector with respect to the selected rear panel AES/EBU Reference (sync) input signal. This feature is not relevant with general purpose serial or parallel formats.

Example

```
Const NOT_READY As Boolean = False
Const FLAT As Integer = 2
Const AES As Integer = 0
Const Z110 As Integer = 1

Sub Main
    Dim delay As Double

    AP.Application.NewTest      'Reset panels
    AP.Sync.SourceInput = AES  'Set Sync input source
    AP.Sync.Impedance = Z110   '110 ohm input impedance
    AP.Sync.Source = True      'Turn on source
    AP.Sync.DelaySettling 10e-3, 100e-9, "Sec", 3, 0.0,
        NONE
    AP.Sync.DelayTrig          'Trigger a new reading
    While AP.Sync.DelayReady = NOT_READY 'wait for _
        reading to settle
    Wend
    delay = AP.Sync.DelayRdg("SEC") 'Measure the _
        sync-signal delay
    'now that we have delay, use this to output test signal
    AP.Sync.Source = False      'Generate sync
```

```

AP.Sync.OutDelay = True      'Enable output delay
AP.Sync.OutDelayFromRef("SEC") = delay 'Match _
    measured output delay
'now perform further testing on DUT ...
End Sub

```

AP.Sync.DelayReady

Property

Syntax `AP.Sync.DelayReady`

Data Type Integer

0 Reading not ready.
 >0 Reading ready.

Description This command returns the Sync In Delay settled settled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the `AP.Sync.DelayRdg` command will zero the ready count.

If the reading is found to be ready, a call to the Frequency A command will be guaranteed to return quickly.

Note that readings free run at the selected measurement rate and eventually become ready without a call to the `AP.Sync.DelayTrig` command.

See Also `AP.Sync.DelayRdg`, `AP.Sync.DelaySettling`, `AP.Sync.DelayTrig`

Example See example for `AP.Sync.DelayRdg`.

AP.Sync.DelaySettling

Method

Syntax `AP.Sync.DelaySettling`(ByVal *Tolerance* As Double, ByVal *Floor* As Double, ByVal *FloorUnit* As String,

ByVal *Points* As Integer, ByVal *Delay* As Double,
ByVal *Algorithm* As Integer)

Parameters	See Appendix A for Settling Algorithm and parameter name descriptions.
Description	This command sets the settling parameters for the AP.Sync.DelaySettling command.
See Also	AP.Sync.DelayRdg, AP.Sync.DelayReady, AP.Sync.DelayTrig
Example	See example for AP.Sync.DelayRdg.

AP.Sync.DelayTrig

Method

Syntax	AP.Sync.DelayTrig
Description	Causes a restart of the reading cycle and zeros the ready count for the AP.Sync.DelayRdg command. The reading in progress is aborted.
See Also	AP.Sync.DelayRdg, AP.Sync.DelayReady, AP.Sync.DelaySettling
Example	See example for AP.Sync.DelayRdg.

35
sync/Ref

AP.Sync.FrameLock

Property

Syntax	AP.Sync.FrameLock
Data Type	Boolean
	<i>True</i> Enable
	<i>False</i> Disable
Description	This command synchronizes the output AES/EBU frame (status bits preambles) to the AES/EBU Ref Input frame (status bits preambles). The Digital Output Sample Rate (OSR) command AP.S2CDio.OutRate and Sync Reference Frequency command AP.Sync.Freq must be set identical for this control to operate correctly.

See Also AP.Sync.OutRate

AP.Sync.Freq

Property

Syntax AP.Sync.Freq(ByVal Unit As String)

Data Type Double 8kHz - 54kHz

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following unit is valid for this command: Hz

Description This command specifies the exact Sync Input rate to be assumed by the phase-locked loop which locks the internal crystal oscillator to the reference. The Internal Sample Rate is then derived from the internal crystal oscillator. Normally, the user will enter the known reference frequency. If the value entered differs by small amounts (less than 15 ppm) from the actual Reference frequency, all System Two sample rates will be shifted by the percentage error. If the value entered differs by more than +/-15 ppm from the actual Reference signal frequency, the internal crystal oscillator will not lock to the reference. When either of the two video sync functions (NTSC or PAL/SECAM) is selected, the normal horizontal sync rate for the selected video standard is automatically typed into the Frequency field.

See Also AP.Sync.

Example

```

Const NTSC As Integer = 2
Const Z75 As Integer = 1

Sub Main
    AP.Sync.Source = NTSC           'Set Sync input source
    AP.Sync.Impedance = Z75        '75 ohm input impedance
    AP.Sync.Source = False        'Turn off sourcing

    rdg = AP.Sync.Freq("Hz")      'get input sync frequency
    If (rdg < lower_limit) Or (rdg > upper_limit) Then
        'input sync freq not close enough, flag an error _
        and ...
    End

```

```

Else
  If AP.Sync.OutRangeRdg Then
    'internal clock not sync'ed, flag an error _
    and ...
  End
Else
  AP.Sync.Source = True    'Turn on sourcing
  'now perform further testing on DUT...
End If
End If
End Sub

```

AP.Sync.FreqRdg

Property

Syntax `AP.Sync.FreqRdg (ByVal Unit As String)`

Data Type Double

Parameters

Part	Description
<i>Unit</i>	The following units are available, Hz.

Description

This command returns a unsettled reading for the Sync Delay Input Frequency for the signal selected in the Sync Source field when the ON/OFF button is OFF. This is intended as a verification of a proper sync input connection. The Reference frequency value is usually known to a greater accuracy than it can be measured by System Two (whose accuracy is typically about 1 ppm), in which case the known value should be entered in the Sync Input Frequency entry field. For example, a measured and displayed value of 47.9998 kHz almost certainly indicates an actual 48 kHz reference frequency, and 48.0000 kHz is the value which should be entered in the Input Frequency entry field. When the ON/OFF button is turned ON, the display field is blanked since the reading will be identical to the value in the Frequency Entry field.

See Also

AP.Sync.FreqReady, AP.Sync.FreqTrig

Example

```

Sub Main
  AP.Sync.Source = False  'Freq rdg only w/src off
  AP.Sync.FreqTrig       'Start a new reading

```

```

While AP.Sync.FreqReady = False 'Wait for reading
  'do other tasks while waiting for reading ...
Wend
reading1 = AP.Sync.FreqRdg("Hz")
Debug.Print "Sync Input Frequency = "; _
  Format(Reading1, "#.0000");" Hz"
End Sub

```

Example Output Sync Input Frequency = 48000.0017 Hz

AP.Sync.FreqReady

Property

Syntax AP.Sync.FreqReady

Data Type Integer

0 Reading not ready.

>0 Reading ready.

Description This command returns the Sync Frequency unsettled reading ready count.

Because readings do not return until a reading is ready, this command may be used to avoid waiting for a reading. This command does NOT zero the ready count and so may be called any number of times. Only a call to the AP.Sync.FreqRdg command will zero the ready count.

If the reading is found to be ready, a call to the AP.Sync.FreqRdg command will be guaranteed to return quickly.

Note that readings free run at the selected measurement rate and eventually become ready without a call to the AP.Sync.FreqTrig command.

See Also AP.Sync.FreqRdg, AP.Sync.FreqTrig

Example See example for AP.Sync.FreqRdg.

AP.Sync.FreqTrig**Method**

Syntax	<code>AP.Sync.FreqTrig</code>
Description	Causes a restart of the reading cycle and zeros the ready count for the <code>AP.Sync.FreqRdg</code> command. The reading in progress is aborted.
See Also	<code>AP.Sync.FreqRdg</code> , <code>AP.Sync.FreqReady</code>
Example	See example for <code>AP.Sync.FreqRdg</code> .

AP.Sync.Impedance**Property**

Syntax	<code>AP.Sync.Impedance</code>				
Data Type	Integer				
	The following list contains the selections relevant to the <code>AP.Sync.Source</code> command for the AES Sync Rate selection.				
	<table> <tr> <td><i>0</i></td> <td>Hi Impedance</td> </tr> <tr> <td><i>1</i></td> <td>110 Ohms</td> </tr> </table>	<i>0</i>	Hi Impedance	<i>1</i>	110 Ohms
<i>0</i>	Hi Impedance				
<i>1</i>	110 Ohms				
	The following list contains the selections relevant to the <code>AP.Sync.Source</code> command for the Squarewave, NTSC Video Sync Horz Rate, PAL / SECAM Video Sync Horz Rate selections.				
	<table> <tr> <td><i>0</i></td> <td>Hi Impedance</td> </tr> <tr> <td><i>1</i></td> <td>75 Ohms</td> </tr> </table>	<i>0</i>	Hi Impedance	<i>1</i>	75 Ohms
<i>0</i>	Hi Impedance				
<i>1</i>	75 Ohms				
Description	This command controls the input impedance for Balanced and Un-Balanced Sync Input configurations.				
See Also	<code>AP.Sync.Source</code>				
Example	See example for <code>AP.Sync.DelayRdg</code> .				

AP.Sync.OutDelay

Property

Syntax `AP.Sync.OutDelay`

Data Type Boolean

True Enable delay.

False Disable delay.

Description This command enables or disables the specified generator output delay relative to the Ref Out as defined by the `AP.Sync.OutDelayFromRef` command. When delay is not required, this feature should be disabled rather than set the `AP.Sync.OutDelayFromRef` command to a zero value, since residual jitter is slightly higher when the output delay feature is enabled.

The ON/OFF button at the left of the Source selection field connects the selected source signal to System Twos internal phase locked loops. The Input Frequency field will be blanked when the switch is turned on, since the reading will be redundant if lock is achieved and will be incorrect if lock is not possible.

See Also `AP.Sync.OutDelayFromRef`

Example See example for `AP.Sync.DelayRdg`.

AP.Sync.OutDelayFromRef

Property

Syntax `AP.Sync.OutDelayFromRef(ByVal Unit As String)`

Data Type Double -10.42 to 10.34 sec

Parameters	Name	Description
	<i>Unit</i>	String that designates the desired unit. The following units are valid for this command: UI, sec

Description This command controls the time (phase) delay of the Digital Generator (front panel) output relative to the rear panel AES/EBU REF OUT XLR connector. To use this feature, the Ref Out signal would be connected to a digital device under test as house sync while System Twos Digital Generator drives the devices digital signal input. The devices tolerance to delay from reference may then be tested by entering different values

into the Output Delay from Ref value. When delay is not required, this feature should be turned off via the `AP.Sync.OutDelayFromRef` command or manually via the On/Off button at the right of the field rather than set to a zero value with the button On, since residual jitter is slightly higher when the output delay feature is on.

Example See example for `AP.Sync.DelayRdg`.

AP.Sync.OutFrameSyncInvert

Property

Syntax `AP.Sync.OutFrameSyncInvert`

Result Boolean

True Invert Transmit Frame Sync output.
False Normal non-inverting output.

Description This command sets the Transmit Frame Sync to normal or inverted polarity.

AP.Sync.OutJitterClockOutputs

Property

Syntax `AP.Sync.OutJitterClockOutputs`

Result Boolean

True Enable
False Disable

Description This command enables or disables the addition of jitter to the Master Clock and Transmit Frame Sync Clock outputs.

AP.Sync.OutOfRangeRdg

Property

Syntax	<code>AP.Sync.OutOfRangeRdg</code>	
Result	Boolean	
	<i>True</i>	In Range
	<i>False</i>	Out Of Range
Description	This command returns a unsettled reading for the Sync Out Of Range indicator.	
Example	See example for <code>AP.Sync.Freq.</code>	

AP.Sync.Source

Property

Syntax	<code>AP.Sync.Source</code>	
Data Type	Boolean	
	<i>True</i>	Enable.
	<i>False</i>	Disable.
Description	This command enables or disables the external sync input.	
See Also	<code>AP.Sync.SourceInput</code>	
Example	See example for <code>AP.Sync.FreqRdg.</code>	

AP.Sync.SourceInput

Property

Syntax	<code>AP.Sync.SourceInput</code>	
Data Type	Integer	
	0	AES Sync Rate:
	1	Squarewave:
	2	NTSC Video Sync Horz Rate:
	3	PAL / SECAM Video Sync Horz Rate:
Description	This command sets the input type for the external sync input.	

See Also AP.Sync.Source

Example See example for AP.Sync.DelayRdg.

User Notes

Appendix A Settling Algorithm

Description The general concept of the Sweep Settling Exponential and Flat algorithms is to discard all meter readings during the Delay interval, then to compare the number of successive readings equal to the Points value against the Tolerance or Floor values. Only when the specified (Points) number of consecutive readings agree with one another within the specified Tolerance or Floor values will the data be considered settled. It is then accepted for plotting and the Source parameter permitted to proceed to the next step.

Settling Parameter Discriptions

Name	Discription
<i>Tolerance</i>	The Tolerance value which should be entered is the amount of variability the user is willing to accept from test to test. A Tolerance value of 0.1% (about 0.01 dB) or even slightly smaller may be appropriate when making frequency response measurements on the test system itself or on an external device known to be very flat and being measured under excellent signal-to-noise conditions. At the other extreme, Tolerance values of 10% to 25% (1 to 2 dB) may be required to obtain data under noisy conditions, or when making measurements with a random noise signal as the stimulus. The default value of 1% (about 0.1 dB) is a good starting compromise for most level measurements.
<i>Floor</i>	The Floor value is used by the algorithms instead of the Tolerance value whenever the Floor value is larger. When the measurements values are greater than a few percent of full scale on the instrument range in use, the Tolerance value is normally the determining parameter. If the measurements are very near the bottom of the instruments dynamic range, use of only a Tolerance parameter could result in a hang up situation, since the percentage difference between two

adjacent values (quantization levels) at the bottom of a meters range is large. The Floor parameter thus serves as a safety valve, avoiding slowing or hang ups in the highly resolution-limited situations where the signal is near the bottom of a measurable range. The default values of Floor for each meter are chosen to be approximately the resolution of that meter on its most sensitive range. Since resolution varies with reading rate (slower reading rates give more resolution), it may be appropriate to change the default values when reading rate is fixed at a given value.

<i>FloorUnit</i>	String that designates the desired unit to be used with the Floor# Parameter. Refer to the reading to determine the appropriate unit selections.
<i>Points</i>	The value determines how many consecutive readings are examined by the Settling Algorithm to qualify a measurement to be returned for display.
<i>Delay</i>	The value determines how long APWIN software waits at each new step of a sweep before starting to examine measurements from the instrument. The Delay value is effective even when the Algorithm selection is None. The Delay time will be taken at the beginning of each nest of a nested sweep, including nested FFT measurements with the FFT at Source 1 and another parameter such as generator amplitude at Source 2. Acquisition of signal into any of the FFT programs will not begin until the Delay value (or 200 milliseconds, whichever is greater) has passed. For Time sweeps where it is desired to make as many measurements per second as possible, the Delay value should be set to zero in addition to selecting None for settling.
<i>Algorithm</i>	0 = None: no settling process takes place for that meter. However, the Delay value (see the Delay topic) is still implemented before each point is plotted even with None selected as the settling algorithm. Measurements such as

wow and flutter, phase jitter, and (with System Two Dual Domain) interface signal jitter are examples of cases where no settling should be used, since it is normally desired to see the extreme variations in measurements.

1 = Exponential: the newest reading (N) must agree with the immediately preceding reading (N-1) within the Tolerance value, with the reading before (N-2) that within twice the Tolerance value, with the reading before that (N-3) within four times the Tolerance value, etc. Exponential is the recommended settling algorithm for most audio applications, since typical device transients tend to die away in an exponential fashion. Exponential thus will usually provide repeatable results to the Tolerance acceptable to the user in the minimum length of time.

2 = Flat: the percentage difference between each set of two consecutive readings (N vs N-1, N-1 vs N-2, etc.) must be equal to or less than the specified Tolerance value, through the number of readings specified as the Points value. Illustrating the Flat algorithm for 1% Tolerance would result in an envelope bounded by two horizontal lines at the plus and minus 1% levels across the full number of Points. The Flat algorithm thus guarantees that the transients have been settled to the specified Tolerance for some time, which tends to take longer than the Exponential algorithm.

3 = Average: measurements are first discarded for the duration of the Delay interval, as with Exponential and Flat. At the conclusion of the Delay period, the number of consecutive readings specified in the Points field is accumulated and their average value computed and plotted. Tolerance and Floor values are ignored when Average is selected. The Average algorithm is particularly useful when the signal is fundamentally noisy and might never settle within a practical Tolerance.

User Notes

Appendix B Parameter ID# List

Using an ID# as the setting (*idnumber*) for the sweep Data 1-6, Source 1-2, and source 1 Min Level Source Selector (External Sweeps) commands is analogous to the selecting the Sweep panel Data 1 browser and choosing the desired instrument and parameter.

Example: To obtain the ID# in order to programmatically assigned a sweep parameter. Manually select the desired instrument and parameter from the desired sweep browser and note the text displayed in the selection box. Locate the text displayed in the selection box from the following list and use the associated value with the appropriate AP.Sweep.???.ID command designating the desired sweep parameter to be changed.

Sweep panel ID Text	Value
Anlr.Ampl	5906
Anlr.Bandpass	5907
Anlr.BandReject	5908
Anlr.BP Ampl	5917
Anlr.BP Pct	5918
Anlr.BPBR Freq	5155
Anlr.CCIF	5912
Anlr.DIM	5913
Anlr.Freq A	5901
Anlr.Freq A & Freq B	5920
Anlr.Freq B	5902
Anlr.Level A	5903
Anlr.Level A & Level B	5919
Anlr.Level B	5904
Anlr.Pct	5916
Anlr.Phase	5905
Anlr.SMPTE	5911
Anlr.THG Ampl	5909
Anlr.THG Pct	5910
Anlr.WF	5914
Aux.Reading 1 (Double)	6275
Aux.Reading 2 (Double)	6276
Aux.Reading 3 (Long)	6277

Sweep panel ID Text	Value
Aux.Reading 4 (Long)	6278
Aux.Setting 1 (Double)	6271
Aux.Setting 2 (Double)	6272
Aux.Setting 3 (Long)	6273
Aux.Setting 4 (Long)	6274
Bittest.Ch.A Errors	6051
Bittest.Ch.A Data	6049
Bittest.Ch.B Errors	6052
Bittest.Ch.B Data	6050
Dcx.DC Out 1	5258
Dcx.DC Out 2	5260
Dcx.Dig In	5953
Dcx.Dig Out	5265
Dcx.DMM Ohms	5952
Dcx.DMM Volts	5951
Dcx.Gate Delay	5271
Dcx.Port A	5268
Dcx.Port B	5269
Dcx.Port C	5270
Dcx.Port D	5272
DGen.Ampl A	5106
DGen.Ampl A & Ampl B	5121
DGen.Ampl B	5107
DGen.Ampl Ratio	5105
DGen.Burst Interval	5129
DGen.Burst Low Lvl	5130
DGen.Burst On	5128
DGen.Center Freq	5134
DGen.Ch. A Freq	5114
DGen.Ch. B Freq	5115
DGen.Freq	5102
DGen.Freq 2	5115
DGen.High Freq	5133
DGen.IM Freq	5104
DGen.Offset	5136
DGen.Phase	5131
DGen.Samples/Step	5135

Sweep panel ID Text	Value
Dio.Common Mode Ampl	5317
Dio.Common Mode Freq	5318
Dio.Delay from Output	6104
Dio.Input Resolution	5325
Dio.Input Sample Rate	6101
Dio.Input Voltage	6102
Dio.Interface Jitter	6105
Dio.Interfering Noise Ampl	5305
Dio.Jitter Ampl	5323
Dio.Jitter Freq	5322
Dio.Output Resolution	5326
Dio.Output Sample Rate	5301
Dio.Output Voltage	5304
Dio.Rise/Fall Time	5303
Distortion.Ch1 Fund Ampl	6125
Distortion.Ch1 Fund Freq	6127
Distortion.Ch1 Harm Sum1	6129
Distortion.Ch1 Harm Sum2	6130
Distortion.Ch2 Fund Ampl	6126
Distortion.Ch2 Fund Freq	6128
Distortion.Ch2 Harm Sum1	6131
Distortion.Ch2 Harm Sum2	6132
DSP Anlr.Ampl	6014
DSP Anlr.Bandpass	6019
DSP Adio Anlr.BP/BR Filter Freq	5542
DSP Anlr.Crosstalk	6016
DSP Anlr.Freq A	6009
DSP Anlr.Freq A & Freq B	5923
DSP Anlr.Freq B	6010
DSP Anlr.Level A	6005
DSP Anlr.Level A & Level B	5922
DSP Anlr.Level B	6006
DSP Anlr.Noise	6020
DSP Anlr.Ratio	6015
DSP Anlr.THD+N Ampl	6018
DSP Anlr.THD+N Ratio	6017
Fasttest.Ch. 1 Jitter	6063

Sweep panel ID Text	Value
Fasttest.Ch. 2 Jitter	6064
Fasttest.Ch.1 Ampl	6309
Fasttest.Ch.1 Phase	6033
Fasttest.Ch.2 Ampl	6312
Fasttest.Ch.2 Phase	6034
Fasttest.FFT Freq	5621
Fasttest.FFT Time	5620
Fasttest.Freq Resolution	5551
Fft.Ch.1 Ampl	6023
Fft.Ch.1 Phase	6065
Fft.Ch.2 Ampl	6026
Fft.Ch.2 Phase	6066
Fft.FFT Freq	5515
Fft.FFT Pre-Trig Time	5519
Fft.FFT Start Time	5518
Fft.FFT Time	5516
Gen.Ampl A	5052
Gen.Ampl B	5053
Gen.Ampl A & Ampl B	5076
Gen.Ampl Ratio	5086
Gen.Burst Interval	5069
Gen Burst On	5068
Gen.Center Freq	5088
Gen.Dual Ampl Ratio	5085
Gen.Freq	5051
Intervu.Amplitude	6053
Intervu.Freq	5613
Intervu.Jitter	6055
Intervu.Probability	6054
Intervu.Time	5612
Mls.Ch.1 Ampl	6325
Mls.Ch.1 Phase	6045
Mls.Ch.2 Ampl	6328
Mls.Ch.2 Phase	6046
Mls.DGen Ampl	5580
Mls.MLS Freq	5581
Mls.Ref Time	5579

Sweep panel ID Text	Value
None	5049
Swr.Ch. A Input	5201
Swr.Ch. A Input/Output	5206
Swr.Ch. A Output	5203
Swr.Ch. A+B Input	5208
Swr.Ch. A+B Input/Output	5210
Swr.Ch. A+B Output	5209
Swr.Ch. B Input	5202
Swr.Ch. B Input/Output	5207
Swr.Ch. B Output	5204
Sync/Ref.In from Ref In Delay	6103
Sync/Ref.Input Freq	6106
Time.External Sweep Time	6253
Time.Time Since Test Loaded	6251

User Notes

Appendix C FFT Window Descriptions

Window	Description
Hann	This window is a raised cosine window named after its inventor, Austrian meteorologist Julius von Hann. It provides good selectivity near the center frequency with no side lobes. Its skirts are not as steep as the Blackman-Harris window. The Hann window causes approximately a -1.5 dB maximum amplitude error due to window attenuation if the signal is at the extreme edge of the bin.
Flat-Top	This window is designed for the greatest amplitude measurement accuracy. It provides a maximum amplitude error due to window attenuation of less than 0.02 dB even if the signal is at the extreme end of the bin. However, its selectivity is poorer than either Hann or Blackman-Harris. The Flat-Top window is the appropriate window for accurate amplitude measurements (such as when measuring individual harmonics) except when signals are so closely spaced that its selectivity becomes a problem. For example, the 2.93 Hz bin width of a 16,384 sample FFT at the 48 kHz sample rate would permit accurate measurements of signals differing by nearly 90 dB in amplitude as long as they are at least 26.4 Hz (9 bins) apart
BH4	The Blackman-Harris 4-term minimum sidelobe window furnished as part of several Audio Precision FFT programs was developed by R.B. Blackman and F.J. Harris. Compared to the Hann window, it is not as selective near the nose but has steeper skirts below that point. The Blackman-Harris window has sidelobes below -92 dB (response fall-off is not monotonic). It has a reasonably flat top with a maximum amplitude error of about -0.8 dB if the signal is at the extreme edge of the bin.
Equiripple	The Equiripple window, developed at Audio Precision, is an approximation to the Dolph-Chebyshev window which provides the narrowest mainlobe width for a given maximum sidelobe depth. The mainlobe is approximately 12 bins wide; that is, the first null is about 6 bins away from the mainlobe center. The first sidelobe, which is also the highest sidelobe,

	is 147 dB down from the mainlobe. Maximum amplitude error across the bin is approximately 0.6 dB.
Hamming	The Hamming window has the sharpest nose selectivity of all the furnished windows. Adjacent bins average about 7 dB down and two bins away the response is about 40 dB down. Amplitude error is about -1.7 dB for a signal at the extreme edge of a bin. The Hamming window has side lobes (that is, response fall-off is not monotonic) starting only 40-50 dB below the center bin, or about 4 bins away from center. The cyan trace shows the Equiripple window, for reference.
Gaussian	The Gaussian window nose selectivity is only slightly wider than the Blackman Harris window and the near-by rejection is considerably better than Blackman Harris, reaching an average of 100 dB down in the fifth bin away from center. The side lobes are down more than 130 dB, compared to about -100 dB for Blackman Harris. Maximum amplitude error is about -0.7 dB for a signal at bin edge.
Rife-Vincent 4	Both Rife-Vincent windows have smooth, monotonically-falling responses with no sidelobes. The Rife-Vincent 4 window has about -0.6 dB maximum amplitude error, is down about -100 dB 7 bins off and about -150 dB 15 bins off. The Rife-Vincent 5 is slightly wider at the nose, with about -0.5 dB maximum amplitude error with a signal at bin edge. It has sharper skirts with attenuation reaching about 106 dB 7 bins off and about 150 dB 12 bins off.
Rife-Vincent 5	See Rife-Vincent 4 above.

Appendix D Extensions Error Codes

Errors

Codes	Description
8501	Undefined DSP error.
8503	Minimum generator amplitude attempted.
8504	Maximum generator amplitude attempted.
8505	Minimum generator frequency attempted.
8506	Maximum generator frequency attempted.
8523	Error reading waveform file.
8526	Error writing waveform file.
8544	Bad selection attempted.
8549	Burst on cycles greater than interval cycles.
8552	Maximum DC Volts output attempted.
8553	Minimum DC Volts output attempted.
8569	DSP is not returning readings.
8570	DSP Host vector %1 not available.
8571	DSP Transmit register not available.
8572	DSP Receive register not available.
8578	Error loading DSP program.
8581	Error loading DSP program.
8583	Conflict with minimum DSP setting value.
8584	Conflict with maximum DSP setting value.
8590	DSP not responding to reset.
8591	File specified not a valid DSP File.
8594	Waveform transfer not supported by this DSP program.
8596	DSP reading unit selected must have input source from Analyzer.
8597	Ratio unit not supported for DSP readings from ANLR-A or ANLR-B.
8599	Minimum BPBR frequency attempted.
8600	Maximum BPBR frequency attempted.
8611	DIO Host vector not available.
8612	DIO Transmit register not available.
8613	DIO Receive register not available.

Codes	Description
8614	Could not find or open the file - DIOBOOT.AZ2.
8615	Could not find or open the file - DIOXTRA.AZ2.
8616	Could not find or open the file - DSPBOOT.AZ2.
8617	DSP has not responded to the request to go to waveform download state.
8620	Burst Level greater than 100% attempted.
8621	Maximum AES Output Voltage attempted.
8622	Maximum AES Noise Voltage attempted.
8623	DSP Module Not Found
8624	Error When Reading Sample Rate From DSP Gen Waveform File
8625	DSP Gen Waveform File Sample Count Is Too Small - Less Than 2
8626	DSP Gen Waveform File Sample Count Is Too Large - More Than 8192
8627	DSP Gen Waveform Download - Timeout While Waiting for DSP Rest State
8628	DSP Gen Waveform Download - Timeout While Waiting for DSP Transfer State
8629	Cannot set Channel B Amplitude while B-Track-A is enabled
8630	The DSP cannot load a waveform while waiting for trigger, acquiring, or transforming
8631	The Sum of the Sinewave Amplitude and of the Constant Value Cannot Exceed Full Scale
8632	The Sample Rate is Not Indicated in the Incoming AES Status Bytes
8633	Error opening WaveForm file %1

General Errors

Codes	Description
9002	Could not open the file %1
9003	Cannot find the file %1 9004 %1
9005	The Increment must be greater than zero.
9006	%1 is a setting and currently cannot be swept as data.

Codes	Description
9007	%1 is a reading and currently cannot be swept as a source.
9008	%1 currently cannot be a sweep parameter.
9009	Source 1 and Source 2 cannot be the same: %1
9011	Cannot save to file: %1
9012	Incompatible file versions! %1
9013	Incompatible Minor file versions! %1
9014	Incompatible axes, load whole sweep setup?
9015	Incompatible sweep parameters for appending data. Would you like to destroy the present data and run the sweep anyway?
9016	This is not a valid APWIN Test.
9017	Unable to save this test. Make sure the disk is not full.
9018	This is an older version of an APWIN Test and cannot be opened.
9019	Cannot load this APWIN Test because it was saved with System One hardware and you are currently running System Two hardware.
9020	Cannot load this APWIN Test because it was saved with System Two hardware and you are currently running System One hardware.
9021	This file is not recognized as an APWIN Test.
9022	Procedure still active.
9023	Maximum Bar Graphs Exceeded (32).
9024	Invalid Chassis
9025	Invalid Instrument
9026	Invalid Element
9027	Invalid Unit Entered.
9028	Maximum data editors exceeded (8).
9029	The file does not exist: %1
9030	Insufficient access to open file. The file is either marked as Read-Only or is in use by another application.
9031	Cannot open file because the maximum number of files are already opened. Close some files and try again.
9032	Out of Memory.
9033	Invalid filename specified.
9034	Hardware failure during sweep.

Codes	Description
9035	Hardware still dead after attempted restoration.
9036	Error Creating panel. There are probably not enough system resources available. Try closing some panels before creating new ones.
9037	Could not find Logo.bmp.
9038	Could not load Logo.bmp
9039	Error creating the log file %1. Make sure the directory specified exists.
9040	Sweep cannot mix FFT and non-FFT Readings and Settings
9041	An unspecified file I/O error occurred.
9042	The file could not be located.
9043	All or part of the specified path is invalid.
9044	The permitted number of open files was exceeded.
9045	The file could not be accessed.
9046	Attempted to use an invalid file handle.
9047	Current working directory cannot be removed.
9048	There are no more directory entries.
9049	File error when trying to set the file pointer.
9050	There was a hardware error during file I/O.
9051	SHARE.EXE was not loaded or shared region was locked.
9052	There was an attempt to lock a region already locked during file I/O.
9053	The disk is full!
9054	The end of the file was reached.
9055	Import file is incompatible with current sweep setup.
9056	This DSP Program cannot load Generator Waveforms.
9057	Could not save file. File may be write protected, device or path may not exist, or file may be in use by another application.
9058	No default printer is installed in Windows. Please install one and try again.
9059	This is a newer version of an APWIN test and cannot be opened.
9060	APWIN is already running. Executing multiple instances is not allowed.
9061	Error creating map file.

Codes	Description
9062	During load test, an invalid unit was found in %1. Please check the value shown on the panel.
9063	During load test, an invalid choice was found in %1. Please check the value shown on the panel.
9064	Source unit (%1) is not compatible with Target Value unit (%2).
9065	Change %1 to %2 for stereo sweep?
9066	The horizontal value %1 is out of range of the data.
9067	Settings cannot be copied due to incompatible units.
9068	Incompatible unit for %1 limit comparisons on %2. Abort the sweep?
9069	This sweep cannot execute! The unit in the table (%1) is incompatible with the source 1 sweep setup.
9070	Incompatible unit for %1 limit comparisons on %2. Remove this limit choice?
9071	Nested stereo sweeps are not supported at this time.
9072	No compute performed.
9073	No data box checked.
9074	No points to compute.
9075	Compute center requires both Upper and Lower Limits.
9076	No compute performed — %1
9077	Invalid Delta file or column number.
9078	No data to compute for specified Data parameter.
9079	References to this file will be removed from the test. Reassign correct file names and save the test or move the files to the expected directory and reload the test.
9080	Resaving this test in its current state will remove the reference to the above mentioned file completely.
9081	S1.EXE tests can only be imported to APWIN with S1 hardware.
9082	The selected file does not have the correct source unit.
9083	The regulation target %1 is currently inactive.
9084	The settling floor is too big or the regulation tolerance too small.
9085	Invalid equalization curve file or column number.
9086	Table Sweep

Codes	Description
9087	Compute Delta
9088	The file %1 specified for the %2 cannot be found or is invalid.
9089	An EQ Curve has not been selected, would you like to choose one now?
9090	Error reading file %1.
9091	Can only import version 2.10 or 2.00 S1.EXE test files
9092	Unregulated after %1 iterations.
9093	Unregulated after %1 iterations. The minimum resolution of the setting being varied has been reached. Try increasing the initial Step Size.
9094	Unregulated after %1 iterations. The minimum resolution of the setting being varied has been reached.
9095	The minimum resolution of the setting being varied was reached after %1 iterations. The best result possible has been achieved.
9096	Incompatible units.
9097	Cannot import this S1.EXE test because of the following:
9098	Unsupported DSP program!
9099	Unregulated after %1 iterations. The low boundary of the setting being varied has been reached.
9100	Unregulated after %1 iterations. The high boundary of the setting being varied has been reached.
9101	The file %1 is an overlay file and is not supported by import test.
9102	No Bar Graph by this index.
9103	Error Creating Metafile. Operation aborted.
9104	Not a valid column number.
9105	Invalid Data Id number.
9106	Error - Export Graphic format not supported.
9107	APBasic not active.
9108	The append file data is incompatible with current sweep setup or graph mode.
9109	Unregulated after reaching the limit of %1 iterations.
9110	Unregulated after %1 iterations. The reading being regulated is not changing.

Codes	Description
9111	The current Sweep DataSet and the Appending Data columns are not equivalent in count or position.
9112	The current Sweep DataSet and the Appending Source columns are not equivalent in count.
9113	Invalid Trace in Sweep Data.
9114	The append file data units are incompatible with current sweep units.
9115	Sweep table empty, not loaded properly.
9116	Incompatible units with current sweep setup, or with system units.
9117	Cannot load this APWIN Test because it was saved with System Two hardware and you are currently running System Two Cascade hardware.
9118	Cannot load this APWIN Test because it was saved with System One hardware and you are currently running System Two Cascade hardware.

General Warnings

Codes	Description
10002	System One not properly installed or not powered on. Will run in DEMO mode.
10003	System Two not properly installed or not powered on. Will run in DEMO mode.
10004	Could not find apwaterm.bmp in resource file.
10005	Could not load apwaterm.bmp from resource file.
10006	Could not find an ISA-WIN or PCM-WIN APIB Interface Card. Will run in DEMO mode.
10007	Zoom will reprocess the last trace only and all previous data will be lost. Continue?
10008	Found a non-WIN APIB card. This program requires an ISA-WIN or PCM-WIN card to function. Will run in DEMO mode.
10009	Option Filter selected in Test File not found, changing to NONE.

Codes	Description
10010	APWIN must be restarted for the European Option change to take effect.
10011	This release of APWIN does not support System Two Hardware. "System One" is the only hardware choice allowed.
10012	A graph must be displayed on any page to Print, Print Preview or Export to Metafile.
10013	Select a graph panel to print or preview.
10014	There is no procedure to save.
10015	The changes made to %1 will have no effect if the file is not saved. Do you want to save the changes?
10016	The %1 Option Filter specified in the Test being loaded was not found. Filters will be set to NONE
10017	You are currently using the old Digital Domain Audio Analyzer and have requested to switch to the new Audio Analyzer. Do you want to map all of the settings from the old Analyzer to the new Analyzer?
10018	Overwrite existing file?
10019	Changing the Digital Analyzer is not allowed during a sweep. Stop the sweep and then change the Digital Analyzer.
10020	Changing this setting will cause the stereo checkbox on the sweep panel to be turned off. Please re-check this if you still want a stereo sweep.
10021	The test being loaded is an unreleased beta version (%1). This test may not load correctly or may not function as intended.
10022	The data editor must be displayed on any page to Print.
10023	This version of APWIN requires a newer version of the Windows system file COMCTL32.DLL than is currently installed on your computer.

Exception Errors

Codes	Description
11002	Expecting an array of Doubles.

Codes	Description
11003	Invalid Data ID
11004	Invalid Data Column
11005	Array too long.
11006	Could Not Load Procedure
11007	Element Not found - ID was not correct
11008	Element type not a Setting
11009	Element type not a Reading
11010	Data Type for Setting must be Integer, Long, Single or Double
11011	Element specified in argument not found - ID was not correct
11012	Unit String specified was not found. Setting was aborted.
11013	Element is presently not active
11014	Value is over list item max.
11015	Consumer AES element not active.
11016	Professional AES element not active.
11017	The Communications control (MSCOMM32.OCX) could not be loaded. AP.CommA and B will not be functional.
11018	Error setting %s %s to %s %s because %s %s is not active.
11019	Could not find an Element associated with the argument Id. The Id %d is invalid.
15004	%1DSP Waiting for trigger...
15005	%1DSP Acquiring Data...
15006	%1DSP Transforming Data...
15007	Ready
15008	Loading DSP Program
15009	Loading Waveform
15010	Saving Waveform
15011	Loading test...
15012	Loading data...
15013	Loading %1 chassis...
15014	Loading panels...
15015	Sweep Completed
15016	Auto detecting for installed filters...
15017	Loading computes...
15018	(Average #%5d)

DSP Errors

Codes	Description
18000	No DSP Program Specific Error
18001	No DSP Program Specific Error
18002	DSP Program requires DIO or MEM option
18003	Main DSP processor's stack overflow exception was raised
18004	DIO option not present — A/D or DGEN are the only valid input settings
18005	DIO option not present — D/A is the only valid output setting
18006	At least one input channel must be enabled in order to acquire
18008	DSP program does not support external sweeps except for Time
18009	Trigger and Frequency Correction modes require Channel 1 and Channel 2 generator waveforms
18010	Transform size setting out of bounds
18011	Waveform file is not an mls impulse response
18012	Waveform file is not the proper type for selected buffer
18013	Frequency Resolution may only be a sweep source-2 selection
18014	Generator Amplitude may only be a sweep source-2 selection
18015	Generator Frequency may only be a sweep source-2 selection
18016	FFT Start Time may only be a sweep source-2 selection
18017	FFT Pretrigger may only be a sweep source-2 selection
18018	Reference Time may only be a sweep source-2 selection
18020	Channel 1 de-emphasis overload detected
18021	Channel 2 de-emphasis overload detected
18022	Excessive tones in waveform for proper operation
18023	Waveform load not of valid length
18024	Channel 1 & Channel 2 generator waveforms not of equal length
18025	Generator waveform frequencies too close for triggering or frequency correction 18026 Frequency correction data overrun
18027	Frequency resolution setting conflicts with requested frequency
18028	Maximum BP/BR filter frequency exceeded

Codes	Description
18029	Narrow bandpass filter only available at 48 kHz sample rate
18030	Sweep Data incompatible with Sweep Source, Select Data = Probability or change sweep source
18031	Sweep Data = Eye incompatible with Sweep Source = Frequency
18032	If a Sweep Data is set to Eye Opening, other Sweep Data must be set to an Eye Opening.
18033	FFT trigger delay time exceeds the acquire buffer size. Consider using a shorter trigger delay time or a larger acquire size
18034	FFT Start Time is specified beyond the end of acquired data. Consider using a shorter FFT Start Time or a larger acquire size
18035	The sum of FFT Start Time and transform length will exceed the acquired data size
18036	FFT length is larger than acquired data size. Select a smaller FFT length or re-acquire data with a larger acquisition size
18037	Crosstalk information cannot be displayed. The generator waveform does not contain crosstalk tones for channel 1.
18038	Crosstalk information cannot be displayed. The generator waveform does not contain crosstalk tones for channel 2.
18039	Frequency to be corrected is too large.
18040	Frequency to be corrected is too small.
18042	FFT Start Time must be equal to or greater than trigger delay time.
18043	Main DSP processor's reset exception was raised
18044	Main DSP processor's illegal instruction exception was raised
18045	Either channel A or B is off.
18046	Cannot retransform (F6) if averaging in the frequency domain, because the acquisition buffer has only the latest acquisition.
18050	Too many filters turned on for function meter.
18051	Transform size must be 8192 or less when using synchronous averaging.
18052	Memory limitations prevent synchronous averaging before frequency correction when transform length is 8192. Consider a smaller transform size or frequency correct before synchronous averaging.

Codes	Description
18053	The DC component of the Ch 1 signal is greater in magnitude than any AC component and so the sync to sine process was halted. Choosing the subtract 1/2 pk-pk or subtract avg options on the panel may allow you to use the sync to sine feature.
18054	The DC component of the Ch 2 signal is greater in magnitude than any AC component and so the sync to sine process was halted. Choosing the subtract 1/2 pk-pk or subtract avg options on the panel may allow you to use the sync to sine feature.
18055	Channel 1 and Channel 2 generator waveforms have not been loaded
18056	The greatest magnitude tone for Ch 1 is too low in frequency to use sync to sine. To calculate the minimum frequency for which sync to sine will work use the following formula: $\text{sample frequency in Hz} / \text{FFT length} = \text{size of each FFT bin in Hz}$. $6 * \text{size of FFT bin in Hz} = \text{minimum frequency}$. To use sync to sine, the greatest magnitude tone must have a frequency greater than this minimum frequency.
18057	The greatest magnitude tone for Ch 2 is too low in frequency to use sync to sine. To calculate the minimum frequency for which sync to sine will work use the following formula: $\text{sample frequency in Hz} / \text{FFT length} = \text{size of each FFT bin in Hz}$. $6 * \text{size of FFT bin in Hz} = \text{minimum frequency}$. To use sync to sine, the greatest magnitude tone must have a frequency greater than this minimum frequency.
18058	Resampling process aborted. No zero crossings found. Check level meters to verify signal is present.

DSP Warnings

Codes	Description
19000	No DSP Program Specific Warning
19001	Waveform load overrun — file is longer than selected buffer'
19002	Waveform load underrun — file is shorter than selected buffer

Codes	Description
19003	Channel 1 generator waveform should be loaded before channel 2
19004	Channel 1 time frame not set — must do a Time sweep before Frequency sweep
19005	Channel 2 time frame not set — must do a Time sweep before Frequency sweep
19006	Ch1 & Ch2 time frames not set - must do a Time sweep before Frequency sweep
19007	Not enough tones in waveform for reliable triggering
19008	Lobe width even, 0 or 1
19009	Frequency correction out of range
19010	Main filter over-ranged
19011	Sample rate may be insufficient
19012	Filtered level ranged
19013	RMS Filter 1 overload occurred
19014	RMS Filter 2 overload occurred
19015	Trigger check failed when trying to frequency correct downloaded waveform - correction aborted
19016	Not enough samples downloaded for frequency error correction
19017	Frequency correction skipped - waveform has already been corrected
19018	Trigger check failed while trying to frequency correct acquired waveform - correction aborted
19019	Crosstalk mode requires at least one generator tone different in each of the two channels
19020	Current generator waveform has no crosstalk tones for channel 1
19021	Current generator waveform has no crosstalk tones for channel 2

User Notes

Appendix E Analog Filter ID# List

To obtain the ID# in order to programmatically assigned a filter locate the filter text displayed in the following list and use the associated value with the `AP.Anlr.FuncFilter` command.

Numerical Listing

ID #	Filter Text
12000	Empty Slot
12017	A-Weighting
12018	CCIR 468-3
12019	CCITT P.53
12020	C-Message
12021	C-Weighting
12033	50 μ s de-emph
12034	50 μ s de-emph + 15.6kHz notch
12035	50 μ s de-emph + 19kHz notch
12037	75 μ s de-emph
12038	75 μ s de-emph + 15.7kHz notch
12039	75 μ s de-emph + 19kHz notch
12041	75 μ s de-emph + A-wtg
12049	20-15kHz +15.6kHz notch
12050	200-15kHz + 19kHz notch
12051	Video notch
12052	19kHz FM Pilot notch
12053	FIL-VOX
12080	200Hz lowpass
12081	300Hz lowpass
12082	400Hz lowpass
12083	500Hz lowpass
12086	1kHz lowpass
12088	3.4kHz lowpass
12089	3kHz lowpass
12090	4kHz lowpass
12091	8kHz lowpass

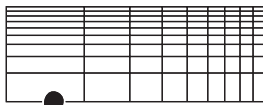
ID #	Filter Text
12092	5kHz lowpass
12093	12.7kHz 6dB/oct lowpass
12094	50kHz lowpass
12097	10kHz lowpass
12098	12.7kHz lowpass
12099	15kHz lowpass
12100	18kHz lowpass
12101	19kHz lowpass
12102	20kHz lowpass
12103	22kHz lowpass
12106	15kHz LP + 15.6kHz notch
12107	15kHz LP + 19kHz notch
12113	30Hz highpass
12117	70Hz highpass
12120	400Hz highpass
12123	2kHz highpass
12126	22kHz highpass
12129	100Hz bandpass
12130	120Hz bandpass
12131	180Hz bandpass
12132	250Hz bandpass
12133	300Hz bandpass
12134	400Hz bandpass
12135	500Hz bandpass
12136	600Hz bandpass
12137	666Hz bandpass
12138	800Hz bandpass
12139	945Hz bandpass
12140	315Hz bandpass
12141	333Hz bandpass
12145	1kHz bandpass
12146	1.2kHz bandpass
12147	1.5kHz bandpass
12148	2kHz bandpass
12149	2.04kHz bandpass
12150	3kHz bandpass

ID #	Filter Text
12151	3.15kHz bandpass
12152	3.4kHz bandpass
12153	4kHz bandpass
12154	4.5kHz bandpass
12155	5kHz bandpass
12156	6kHz bandpass
12157	8kHz bandpass
12158	6.3kHz bandpass
12161	10kHz bandpass
12162	12.5kHz bandpass
12163	15kHz bandpass
12164	16kHz bandpass
12165	20kHz bandpass
12166	22kHz bandpass
12167	24kHz bandpass
12168	25kHz bandpass
12169	30kHz bandpass
12170	12.7kHz bandpass
12177	100Hz hi-Q bandpass
12178	500Hz hi-Q bandpass
12179	1kHz hi-Q bandpass
12180	3kHz hi-Q bandpass
12181	12.5kHz hi-Q bandpass
12182	400Hz hi-Q bandpass
12193	100-5kHz bandpass
12194	100-22kHz bandpass
12209	20kHz Apogee lowpass
12240	External

User Notes



Audio
precision®



Audio Precision, Inc.
PO Box 2209
Beaverton, Oregon 97075-2209
U.S. Toll Free: 1-800-231-7350
Tel: (503) 627-0832 Fax: (503) 641-8906
email: techsupport@audioprecision.com
Web: www.audioprecision.com



Audio Precision
PO Box 2209
Beaverton, Oregon 97075-2209
Tel 503 627-0832 Fax 503 641-8906
Us Toll Free 1-800-231-7350
email techsupport@audioprecision.com
Web www.audioprecision.com