

APWIN 1.52 BASIC ADDENDUM



APWIN VERSION 1.52 BASIC ADDENDUM
TO APWIN BASIC USER'S MANUAL AND
PROGRAMMER'S REFERENCE

APWIN BASIC User's Manual and Programmer's Reference



Addendum for version 1.52

Version 1.52
December, 1998

Copyright © 1998 Audio Precision, Inc.

All rights reserved

Version 1.52 September, 1998

APWIN Basic and APWIN Basic Editor
Copyright 1995 Audio Precision Incorporated
Copyright 1993-1995 Polar Engineering and Consulting
All rights reserved.

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 trademarks or 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 # 8211-0055

Introduction

This manual is designed to be used in conjunction with the APWIN BASIC User's Manual and Programmers Reference Version 1.5. Described in this addendum are additions and changes to Volume 1 through Volume 3 of the APWIN BASIC User's Guide and Programming Reference manuals Version 1.5

In the Volume 3 section of this addendum an entire Group has been added for the System Two Digital Data Analyzer. This information is shown as it will appear in a later printing of the APWIN BASIC User's Manual and Programmers Reference Volume 3.

Note:

Minor changes have been made to many areas to the APWIN BASIC User's Manual and Programmers Reference Volume 1 through Volume 3. Minor changes are not reflected in this manual. Refer to the On Line Help system to confirm information if an error is suspected.

Language Commands

Listed on page 7-1 under Groups/Declaration is Class Modulal and Object Modulal. Class Modulal and Object Modulal is not available in APWIN Version 1.52.

The Declatation Group has also had the **#Reference** and **WithEvents** commands added as shown below.

Declaration **#Reference**, #Uses, Attribute, Class Module, Code Module, Const, Declare, Deftype, Dim, Enum...End Enum, Function...End Function, Object Module, Option, Private, Property...End Property, Public, ReDim, Static, Sub...End Sub, Type...End Type, **WithEvents**.

The String Group has also had the **InStrRev** and **Replace** commands added as shown below.

String Asc, AscB, AscW, Chr, ChrB, ChrW, Format, Hex, InStr, InStrB, **InStrRev**, LCase, Left, LeftB, Len, LenB, LTrim, Mid, MidB, Oct, **Replace**, Right, RightB, RTrim, Space, String, Str, StrComp, StrConv, Trim, UCase.

The Math Group commands **dBToPowerRatio**, **dBToVoltageRatio**, **PowerRatioTodB**, and **VoltageRatioTodB** are included in the APWIN BASIC User's Manual and Programmers Reference Version 1.5 but have be omitted from the APWIN Basic Language On Line Help in version 1.50A

Math Abs, Atn, Cos, **dBToPowerRatio**, **dBToVoltageRatio**, Exp, Exp10, Fix, Int, Log, Log10, Pow, **PowerRatioTodB**, Randomize, Rnd, Sgn, Sin, Sqr, Tan, **VoltageRatioTodB**.

Reference

Comment

Syntax

```
`#Reference
{uuid}#vermajor.verminor#lcid#[path[#name]]
```

Description

The Reference comment indicates that the current macro/module references the type library identified. Reference comment lines must be the first lines in the macro/module (following the global Attributes). Reference comments are in reverse priority (from lowest to highest). The IDE does not display the reference comments.

Parameters

Name	Description
<i>uuid</i>	Type library's universally unique identifier.
<i>vermajor</i>	Type library's major version number.
<i>verminor</i>	Type library's minor version number.
<i>lcid</i>	Type library's locale identifier.
<i>path</i>	Type library's path.
<i>name</i>	Type library's name.

Example

```
`#Reference
{00025E01-0000-0000-C000-000000000046}#4.0#0#C: _
\PROGRAM FILES\COMMON FILES\MICROSOFT SHARED\DAO\_
DAO350.DLL#Microsoft DAO 3.5 Object Library
```

WithEvents

Definition

Syntax

```
[Dim | Private | Public] _
WithEvents name As objtype[, ...]
```

Description

Dimensioning a module level variable WithEvents allows the macro to implement event handling Subs. The variable's As type must be a type from a referenced type library (or language extension) which implements events.

Remarks

This keyword is supported by the single DLL IDE/interpreter (aka the Enterprise edition). It is not supported by the interpreter implemented in WW_CU516.DLL or WW_CU532.DLL.

See Also Dim, Private, Public.

Example

```
Dim WithEvents X As Thing
Sub Main
    Set X = New Thing
    X.DoIt ' DoIt method raises DoingIt event
End Sub

Private Sub X_DoingIt
    Debug.Print "X.DoingIt event"
End Sub
```

InStrRev

Function

Syntax `InStrRev(S1$, S2$[, Index])`

Description Return the index where S2\$ last matches S1\$. If no match is found return 0.

Parameters	Name	Description
	<i>S1\$</i>	Search for S2\$ in this string value. If this value is Null then Null is returned.
	<i>S2\$</i>	Search S1\$ for this string value. If this value is Null then Null is returned.
	<i>Index</i>	Start searching for S2\$ ending at this index in S1\$. If this is omitted then start searching from the end of S1\$.

See Also Left\$(), Len(), Mid\$(), Replace\$(), Right\$().

Example

```
Sub Main
    Debug.Print InStrRev("Hello","l") ' 4
End Sub
```

Replace

Function

Syntax `Replace[$](S, Pat, Rep, [Index], [Count])`

Description Replace Pat with Rep in S.

Parameters	Name	Description
	<i>S</i>	This string value is searched. Replacements are made in the string returned by Replace.
	<i>Pat</i>	This string value is the pattern to look for.
	<i>Rep</i>	This string value is the replacement.
	<i>Index</i>	This numeric value is the starting index in S. Replace(S,Pat,Rep,N) is equivalent to Replace(Mid(S,N),Pat,Rep). If this is omitted use 1.
	<i>Count</i>	This numeric value is the maximum number of replacements that will be done. If this is omitted use -1 (which means replace all occurrences).

See Also

InStr(), InStrRev(), Left\$(), Len(), Mid\$(), Right\$().

Example

```
Sub Main
    Debug.Print Replace$("abcabc","b","B")      ` "aBcaBc"
    Debug.Print Replace$("abcabc","b","B",,1)   ` "aBcabc"
    Debug.Print Replace$("abcabc","b","B",3)    ` "caBc"
    Debug.Print Replace$("abcabc","b","B",9)    ` ""
End Sub
```


Volume 2

Extensions Commands

The **Application** Group has had the following command added.

AP.Application.SuppressErrorMessages

①② Method

Syntax `AP.Application.MacroEditorVisible`

Data Type Boolean

True Error messages not displayed.

False (Default) Display Error messages.

Result Void

Description This command enables or disables display of Error Messages. Note: This command affects APWIN globally. If set to True APWIN will not display any Error messages under manual operation or programatic control. When APWIN is loaded this command defaults to False so that all Error messages are displayed.

Example

```
Sub Main
  With AP.App
    .SuppressErrorMessages = False 'Display Errors
    AP.File.OpenTest("xt5alk.at2") 'Errors displayed
    .SuppressErrorMessages = True 'Don't Display Errors
    AP.File.OpenTest("xt5alk.at2") 'Errors NOT displayed
    .SuppressErrorMessages = False
  End Sub
```

The following command is not implemented at this time.

AP.Bits.ChAXmitStatus

①② Property

Syntax `AP.Bits.ChAXmitStatus(string)`

Data Type Integer

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

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

See Also `AP.Bits.ChAStatusXferToString`

Example

```

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

```

The **Print** Group has had the following command added.

AP.Print.Data

 **Method**

Syntax `AP.Print.Data`

Parameters None

Result Void

Description This command prints the data displayed in the Data Editor in tabular format. The Data Editor must be displayed on at least one of the APWIN Pages. The printer is defined by the settings on the File, Print Setup menu.

See Also `AP.Print.Graph`

Example

```
Sub Main
    System = AP.Application.SysType
    If System = 1 Then
        `Load test containing measurement data
            AP.File.OpenTest("GRAPH.AT1")
    Else
        `Load test containing measurement data
            AP.File.OpenTest("GRAPH.AT2")
    End If
    AP.Application.PanelOpen apbPanelDataEditor
    AP.Sweep.Start
    AP.Print.Data `Print Data in tabular form
End Sub
```

Volume 3

Extensions Commands

The **System Two Digital Input/Output** Group has had the following commands added.

AP.S2Dio.FlagChAInvalidRdg

② Property

Syntax AP.S2Dio.FlagChAInvalidRdg

Data Type Boolean

True Error
False 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.S2Dio.OutSendInvalid,
 AP.S2Dio.FlagChBInvalidRdg,
 AP.S2Dio.FlagInvalidRdg.

Example

```
Sub Main
  AP.Application.NewTest
  AP.Application.PanelOpen apbPanelDigIOSmall
  AP.S2Dio.InFormat = 3
  If AP.S2Dio.FlagInvalidRdg = True Then _
    Debug.Print "Invalid Error"
  AP.Application.PanelOpen apbPanelDigIOLarge
  AP.S2Dio.OutSendInvalid = True
  Wait 1
  If AP.S2Dio.FlagConfidenceRdg = True Then _
    Debug.Print "Confidence Error"
```

```

If AP.S2Dio.FlagLockRdg = True Then _
    Debug.Print "Lock Error"
If AP.S2Dio.FlagCodingRdg = True Then _
    Debug.Print "Coding Error"
If AP.S2Dio.FlagParityRdg = True Then _
    Debug.Print "Parity Error"
If AP.S2Dio.FlagChAInvalidRdg = True Then _
    Debug.Print "ChA Invalid Error"
If AP.S2Dio.FlagChBInvalidRdg = True Then _
    Debug.Print "Ch Invalid Error"
End Sub

```

AP.S2Dio.FlagChBInvalidRdg

 **Property**

Syntax AP.S2Dio.FlagChBInvalidRdg

Data Type Boolean

<i>True</i>	Error
<i>False</i>	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.S2Dio.OutSendInvalid,
AP.S2Dio.FlagChAInvalidRdg,
AP.S2Dio.FlagInvalidRdg.

Example See example for AP.S2Dio.FlagChAInvalidRdg.

AP.S2Dio.FlagCodingRdg

 **Property**

Syntax AP.S2Dio.FlagCodingRdg

Data Type Boolean

<i>True</i>	Error
-------------	-------

False Proper operation

Description This reading returns the state of the channel B Validity bit. The Coding reading 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.S2Dio.FlagChAInvalidRdg`.

AP.S2Dio.FlagConfidenceRdg

② Property

Syntax `AP.S2Dio.FlagConfidenceRdg`

Data Type 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.S2Dio.FlagChAInvalidRdg`.

AP.S2Dio.FlagInvalidRdg

② Property**Syntax** `AP.S2Dio.FlagInvalidRdg`**Data Type** 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.S2Dio.OutSendInvalid`,
`AP.S2Dio.FlagChAInvalidRdg`,
`AP.S2Dio.FlagChBInvalidRdg`.**Example** See example for `AP.S2Dio.FlagChAInvalidRdg`.

AP.S2Dio.FlagLockRdg

② Property**Syntax** `AP.S2Dio.FlagLockRdg`**Data Type** 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.S2Dio.FlagChAInvalidRdg`.

AP.S2Dio.FlagParityRdg

② Property**Syntax** `AP.S2Dio.FlagParityRdg`**Data Type** Boolean*True* Error

False 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.S2Dio.FlagChAInvalidRdg`.

Digital Data Analyzer

AP.S2DSP.Bittest.ChADataRdg

ⓘ Property

Syntax `AP.S1DSP.Bittest.ChADataRdg(unit$)`

Data Type Variant

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 an unsettled reading for the Digital Data Analyzer channel A Data meter and zeros the ready count.

See Also

AP.S2DSP.Bittest.ChADataReady,
AP.S2DSP.Bittest.ChADataTrig

Example

```
Sub Main
  AP.Application.NewTest
  AP.S2Dsp.Program = 6
  AP.DGen.Wfm 4, 5
  AP.DGen.ChAAmpl("dBFS") = -3
  AP.DGen.OutDitherType = 3
  AP.DGen.Output = True
  AP.S2Dio.InFormat = 3
  AP.S2Dsp.BitTest.ChADataTrig 'Trigger a new reading
  Do
    Ready = AP.S2Dsp.BitTest.ChADataReady
  Loop Until Ready > 0          'Wait for new reading
  Reading1 = AP.S2Dsp.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.S2DSP.Bittest.ChADataReady

Property**Syntax** `AP.S2DSP.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.S2DSP.Bittest.ChADataRdg` or `AP.S2DSP.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.S2DSP.Bittest.ChADataRd`,
`AP.S2DSP.Bittest.ChADataTrig`**Example Output** See example for `AP.S2DSP.Bittest.ChADataRdg`.

AP.S2DSP.Bittest.ChADataTrig

Method**Syntax** `AP.S2DSP.Bittest.ChADataTrig`**Description** Causes a restart of the reading cycle and zeros the ready count for the `AP.S2DSP.Bittest.ChADataRdg` command. The reading in progress is aborted.**See Also** `AP.S2DSP.Bittest.ChADataRdg`,
`AP.S2DSP.Bittest.ChADataReady`**Example Output** See example for `AP.S2DSP.Bittest.ChADataRdg`.

AP.S2DSP.Bittest.ChAErrRdg

② Property

Syntax `AP.S2DSP.Bittest.ChAErrRdg(unit$)`

Data Type Variant

Part	Description
<i>unit</i> \$	String that designates the desired unit. The following units are valid for this command: dec.

Description This command returns an unsettled reading for the Digital Data Analyzer channel A Errors meter and zeros the ready count.

See Also `AP.S2DSP.Bittest.ChAErrReady`,
`AP.S2DSP.Bittest.ChAErrTrig`

Example

```

Sub Main
  AP.Application.NewTest
  AP.S2Dsp.Program = 6
  AP.DGen.Wfm 4, 5
  AP.DGen.ChAAmpl("dBFS") = -3
  AP.DGen.OutDitherType = 0
  AP.DGen.Output = True
  AP.S2Dio.InFormat = 3
  With AP.S2Dsp.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
  Do
    Ready = .ChAErrReady
  Loop Until Ready > 0      'Wait for new reading
  Reading1 = .ChAErrRdg("dec") 'Get new reading
End With
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.S2DSP.Bittest.ChAErrReady**② Property****Syntax** `AP.S2DSP.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.S2DSP.Bittest.ChAErrRdg` or `AP.S2DSP.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.S2DSP.Bittest.ChAErrRdg`,
`AP.S2DSP.Bittest.ChAErrTrig`**Example** See example for `AP.S2DSP.Bittest.ChAErrRdg`.**AP.S2DSP.Bittest.ChAErrTrig****② Method****Syntax** `AP.S2DSP.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.S2DSP.Bittest.ChAErrRdg`,
`AP.S2DSP.Bittest.ChAErrReady`**Example** See example for `AP.S2DSP.Bittest.ChAErrRdg`.

AP.S2DSP.Bittest.ChBDataRdg**② Property**

Syntax `AP.S2DSP.Bittest.ChBDataRdg(unit$)`

Data Type Variant

Part	Description
<i>unit\$</i>	String that designates the desired unit. The following units are valid for this command: dec.

Description This command returns an unsettled reading for the Digital Data Analyzer channel B Data meter and zeros the ready count.

See Also `AP.S2DSP.Bittest.ChBDataReady`,
`AP.S2DSP.Bittest.ChBDataTrig`

Example

```

Sub Main
  AP.Application.NewTest
  AP.S2Dsp.Program = 6
  AP.DGen.Wfm 4, 5
  AP.DGen.ChAAmpl("dBFS") = -3
  AP.DGen.OutDitherType = 3
  AP.DGen.Output = True
  AP.S2Dio.InFormat = 3
  AP.S2Dsp.BitTest.ChBDataTrig 'Trigger a new reading
  Do
    Ready = AP.S2Dsp.BitTest.ChBDataReady
  Loop Until Ready > 0 'Wait for new reading
  Reading1 = AP.S2Dsp.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.S2DSP.Bittest.ChBDataReady**② Property****Syntax** `AP.S2DSP.Bittest.ChBDataReady`**Data Type** Integer

0 Reading not ready.
 >0 Reading ready.

Description This command returns the Digital Data Analyzer channel B 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.S2DSP.Bittest.ChBDataRdg` or `AP.S2DSP.Bittest.ChBDataTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2DSP.Bittest.ChBDataRdg` command will be guaranteed to return quickly.

See Also `AP.S2DSP.Bittest.ChBDataRdg`,
`AP.S2DSP.Bittest.ChBDataTrig`**Example** See example for `AP.S2DSP.Bittest.ChBDataRdg`.**AP.S2DSP.Bittest.ChBDataTrig****② Method****Syntax** `AP.S2DSP.Bittest.ChBDataTrig`**Description** Causes a restart of the reading cycle and zeros the ready count for the `AP.S1DSP.Bittest.ChBDataRdg` comand. The reading in progress is aborted.**See Also** `AP.S2DSP.Bittest.ChBDataRdg`,
`AP.S2DSP.Bittest.ChBDataReady`**Example** See example for `AP.S2DSP.Bittest.ChBDataRdg`.

AP.S2DSP.Bittest.ChBErrRdg

② Property

Syntax `AP.S2DSP.Bittest.ChBErrRdg(unit$)`

Data Type Variant

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 an unsettled reading for the Digital Data Analyzer channel B Errors meter and zeros the ready count.

See Also `AP.S2DSP.Bittest.ChBErrReady`,
`AP.S2DSP.Bittest.ChBErrTrig`

Example

```

Sub Main
    AP.Application.NewTest
    AP.S2Dsp.Program = 6
    AP.DGen.Wfm 4, 5
    AP.DGen.ChAAmpl("dBFS") = -3
    AP.DGen.OutDitherType = 0
    AP.DGen.Output = True
    AP.S2Dio.InFormat = 3
    AP.S2Dsp.BitTest.ChBErrTrig    'Trigger a new reading
    Do
        Ready = AP.S2Dsp.BitTest.ChBErrReady
    Loop Until Ready > 0          'Wait for new reading
    Reading1 = AP.S2Dsp.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.S2DSP.Bittest.ChBErrReady**② Property****Syntax** `AP.S2DSP.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.S2DSP.Bittest.ChBErrRdg` or `AP.S2DSP.Bittest.ChBErrTrig` commands will zero the ready count.

If the reading is found to be ready, a call to the `AP.S2DSP.Bittest.ChBErrRdg` command will be guaranteed to return quickly.

See Also `AP.S2DSP.Bittest.ChBErrRdg`,
`AP.S2DSP.Bittest.ChBErrTrig`**Example** See example for `AP.S2DSP.Bittest.ChBErrRdg`.**AP.S2DSP.Bittest.ChBErrTrig****② Method****Syntax** `AP.S2DSP.Bittest.ChBErrTTrig`**Description** Causes a restart of the reading cycle and zeros the ready count for the `AP.S2DSP.Bittest.ChBErrRdg` command. The reading in progress is aborted.**See Also** `AP.S2DSP.Bittest.ChBErrTRdg`,
`AP.S2DSP.Bittest.ChBErrTReady`**Example** See example for `AP.S2DSP.Bittest.ChBErrRdg`.

AP.S2DSP.Bittest.DisplayError**Property****Syntax** `AP.S2DSP.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.S2Dio.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.S2DSP.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.S2DSP.Bittest.RdgRate`**Example** See example for `AP.S2DSP.Bittest.ChAErrRdg`.

AP.S2DSP.Bittest.FreezeOnError

② Property**Syntax** `AP.S2DSP.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.S2DSP.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.S2DSP.Bittest.FreezeOnError` is set to (False), the Data fields will continue updating independent of any errors detected.

See Also `AP.S2DSP.Bittest.RdgRate`**Example** See example for `AP.S2DSP.Bittest.ChAErrRdg`.

AP.S2DSP.Bittest.RdgRate

② Property**Syntax** `AP.S2DSP.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 at which the Data (and Errors) readings are updated.**Example** See example for `AP.S2DSP.Bittest.ChAErrRdg`.

AP.S2DSP.BitTest.Wfm**② Property****Syntax** `AP.S2DSP.BitTest.Wfm`**Data Type** Integer

<i>0</i>	Constant
<i>1</i>	Random.
<i>2</i>	Walking-1.
<i>3</i>	Walking-0.
<i>4</i>	Sine.

Description This command selects the Digital Data Analyzer Waveform pattern to analyze.**Example** See example for AP.S2DSP.BitTest.ChAErrRdg.

User Notes

User Notes

User Notes

User Notes

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