APWIN 2.20 ADDENDUM

APWIN VERSION 2.20 ADDENDUM FOR
SYSTEM ONE, SYSTEM TWO & SYSTEM TWO CASCADE
2.0 USER'S & BASIC EXTENSION REFERENCE MANUALS

APWIN Version 2.2 Addendum
to Version 2 User's Manuals

…including new features for
System Two Cascade *Plus*

**This document
is designed to be used in conjunction with
and as a supplement to the existing
version 2 manual sets**.

September 2001

Contents

Chapter 1: S2C+

New Features Available Only in
■ *System Two Cascade* Plus

Audio Precision® APWIN™ version 2.2 is being released concurrent with new instrument hardware, the Audio Precision System Two Cascade *Plus*. Most of the new features in APWIN version 2.2 (features added since APWIN version 2.14) apply to new capabilities in the System Two Cascade *Plus* hardware and will not function without that instrument.

See Chapters 2, 3 and 4 for documentation of other changes and improvements for the System One, System Two and System Two Cascade family since APWIN version 2.0.

New Serial Interface Adapter (SIA) Panel   V 2.2

APWIN version 2.2 includes a new Serial Interface Adapter panel to control a new auxiliary unit, the PSIA-2522 Programmable Serial Interface Adapter.

Background: Serial Digital Interface

The entire Audio Precision line of digital-equipped instruments accommodate the standard serial digital interface signals defined in AES3 and IEC60958, which include the signals referred to as AES3, AES/EBU, AES3id, IEC60958, S/PDIF-EIAJ and so on.

Work with other serial interface signals such as encountered in telephone communications and chip-level converter design must be approached differently.

In the past, Audio Precision has manufactured an auxiliary unit called the SIA-2322 Serial Interface Adapter, which addressed these needs by adapting the System Two Cascade parallel digital inputs and outputs to interface with a variety of serial digital signals.

With System Two Cascade *Plus* comes support for the new Audio Precision PSIA-2522 Programmable Serial Interface Adapter. The many settings necessary to configure a custom serial interface can now be easily accomplished in software rather than by rows of switches; what's more, converter-specific setups can be saved, reloaded and even downloaded from the Web.

The SIA panel
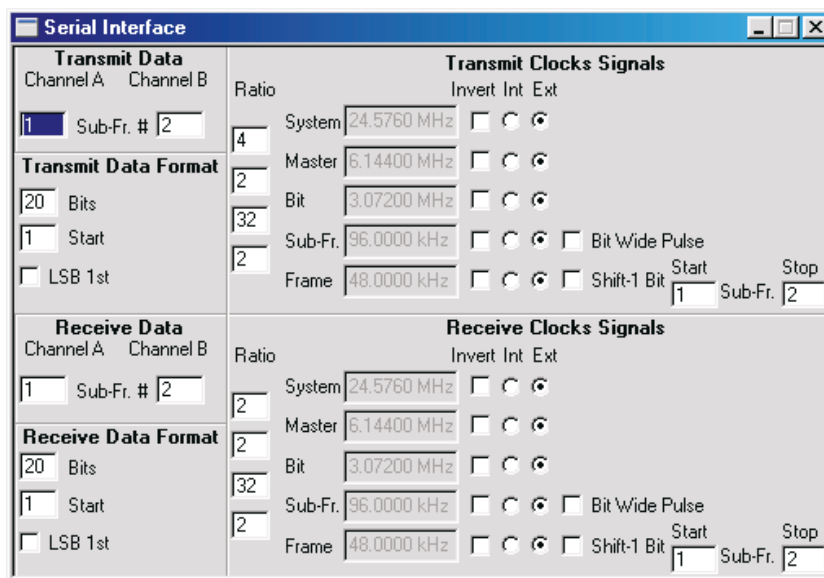
Figure 1. The SIA panel.

The APWIN SIA panel is available by selecting **Panels > Serial Interface** from the Menu bar or by clicking the **SIA** button on the toolbar; you can also launch the SIA panel using the keyboard combination **CTRL+E.**

The top half of the panel controls the transmitted data and clock signals; the bottom half controls the reception of data and clock signals.

You can specify the frame, word length, start bit, and MSB/LSB order for both the transmit and the receive data; using either System Two Cascade *Plus*'s system clock or an external clock, you can derive the master clock rate, the bit rate, the sub-frame and frame rates for the transmit and receive signals.

See the documentation provided with PSIA-2522 for specific information on the interconnection and use of the Programmable Serial Interface Adapter.

FFT Spectrum Analyzer

APWIN version 2.2 brings two improvements to the Digital Analyzer's FFT Spectrum Analyzer instrument for System Two Cascade *Plus*.

4M FFT Acquisition Memory   V 2.2

The System Two Cascade *Plus* hardware provides significantly more memory for FFT signal acquisition, raising the maximum from 256 kBytes to 4 MBytes.

Figure 2. The FFT Acquisition memory list.
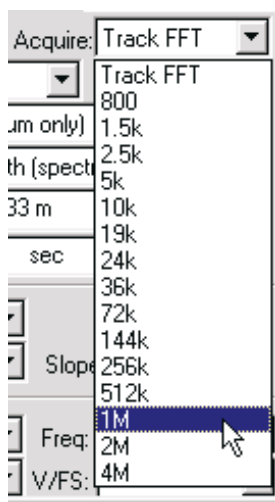
To access the additional memory, drop the Acquire list on the FFT Analyzer panel. Choose one of the following new options from the end of the list:

- **512k**
- **1M**
- **2M**
- **4M**

FFT Smoothing   V 2.2

APWIN version 2.2 for System Two Cascade *Plus* adds a new display method to the FFT Analyzer, *FFT octave smoothing*.

Previous versions of APWIN offered the FFT display options **Interpolate, Display Samples, Peak Values** and **Absolute Values** in

the **Wave Display** field. These display methods only affect the time-domain, "oscilloscope" waveform view of the acquired data.

Figure 3. The FFT Wave Display list.
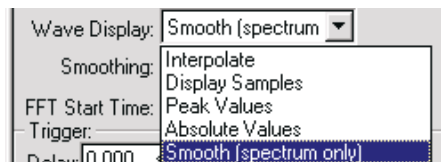
Now you can also select FFT octave smoothing by clicking the new fifth option in the **Wave Display** list, **Smooth (spectrum only)**. The **Smooth** display method only affects the frequency-domain, spectrum analyzer view of the acquired data.

Unlike FFT power averaging, which takes the average of a number of measurements, FFT smoothing is a display method that shows the results of one measurement as modified by a smoothing algorithm.

When **Smooth** is chosen the **Smoothing** setting field becomes available just under the **Wave Display** field. Specify the degree of smoothing you desire by entering a value between 0 to 2.64 octaves.

Octave smoothing is a common technique in loudspeaker response measurement, useful in revealing trends by smoothing out anomalies in the response curve. The APWIN implementation uses a hybrid FFT bin averaging and interpolation technique to achieve smooth results even at very low bin densities. Smoothing, which only affects frequency-domain displays, effectively passes the raw response data through multiple constant-Q bandpass filters, one filter centered on each frequency requested from the Sweep panel. The bandwidth of these filters, in octaves, is specified in the **Smoothing** field.

Chapter 2: S2C, S2C+

Changes Affecting
■ *System Two Cascade*
■ *System Two Cascade* Plus

See also Chapter 3 and Chapter 4 for other changes in the operation of System Two Cascade running under APWIN version 2.11 and later.

Digital I/O Panel

There are several changes to the Digital Input/Output panel which affect System Two Cascade. Be sure to see **Digital I/O Panel: Rate Ref** (page 20) for an update on the **Rate Ref** ranges available to System Two as well. Changes that affect only System Two Cascade and System Two Cascade *Plus* follow here.

Digital Input Sample Rate Scaling   V 2.11

The choices available for the **Scale Freq. by** field on the Input side of the Digital I/O panel have changed.

Measured Rate ▼
Output Rate
Measured Rate
Status Bits A
DIO Rate Ref

Old Names

Meas Input Rate ▼
Output Rate
Meas Input Rate
Status Bits A
DIO Rate Ref
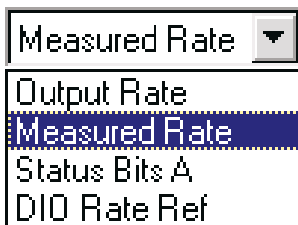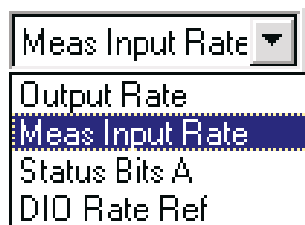
New Names

The function for each choice is the same as in earlier versions of
APWIN. The list choice names have been changed to avoid confusion with
choices on the new output **Scale Freq. by** list.

V 2.11   Digital Output Sample Rate Scaling

Previously, the frequency of the audio signal embedded in the digital
output was scaled based solely on the setting in the **Sample Rate-OSR**
field. A new **Scale Freq. by** field has been added to the Output side of
the Digital I/O panel with these four options for digital output audio
scaling:

```
Output Rate      ▼
Output Rate
Meas Input Rate
Meas Output Rate
DIO Rate Ref
```
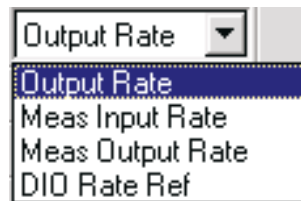
- **Output Rate**, the default, selects the value entered in the **Sample
  Rate-OSR** field higher on the panel. This provides the same
  operation as in previous versions of APWIN.

  > *The Parallel Digital Output was an exception to this rule in
  > previous software versions. In APWIN versions 1.60 through
  > 1.62, the embedded audio frequency was scaled by the value
  > in the Sample Rate-OSR field; in APWIN versions 2.00
  > through 2.03 it was scaled by DIO Rate Ref.*

- **Meas Input Rate** selects the value measured at the digital input
  port. This is the same value selected by the **Meas Input Rate** choice
  on the Input **Scale Freq. by** field.

- **Meas Output Rate** selects the value measured at the parallel output
  port when **Output Format** is set to **Parallel**. This value is the
  sample rate read from the DUT. **Meas Output Rate** defaults to
  Sample Rate-OSR (the same as the **Output Rate** choice) if an
  output other than parallel is used.

- **DIO Rate Ref** selects the value entered in the **Rate Ref** field,
  located on the Input side of the panel.
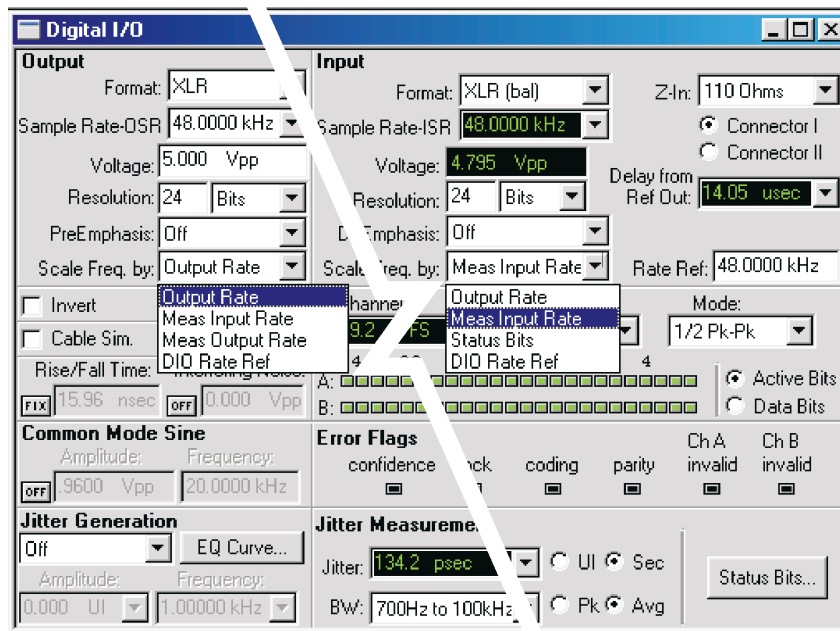
*Figure 4. New **Scale Freq by** features in Digital I/O Panel Output and Input sections.*

Parallel Input Sample Rate

V 2.11

Starting with APWIN 2.11, the parallel digital input sample rate is now correctly measured. Prior to version 2.11, the measured **Sample Rate-ISR** reading displayed incorrect values when the input **Format** was set to **Parallel**. The valid range for the measured parallel input sample rate is greater than 7001 Hz and less than 216 kHz.

µ-Law/A-Law

V 2.11

The International Telecommunication Union (ITU) specifies in its standard G.711 two similar approaches for reducing the bit rate in digital voice telephony. Called *µ-Law* and *A-Law,* these techniques have been widely used in digital telecommunications for a number of years.

µ-Law and A-Law are known as *companders*, converting 14-bit linear PCM samples (µ-Law) or 13-bit linear PCM samples (A-Law) to an 8-bit pseudo floating-point representation via *compression* at the encoder and *expansion* at the decoder.

A new feature as of version 2.11 enables APWIN to both transmit and receive µ-Law and A-Law encoded digital signals. Although most µ-Law and A-Law testing will be done via the parallel output and input ports in conjunction with a Serial Interface Adapter (Audio Precision PSIA-2722 or

SIA-2322), these features support the AES3, S/PDIF and optical inputs and outputs as well, when the sample rate specified is above the minimum for these ports.
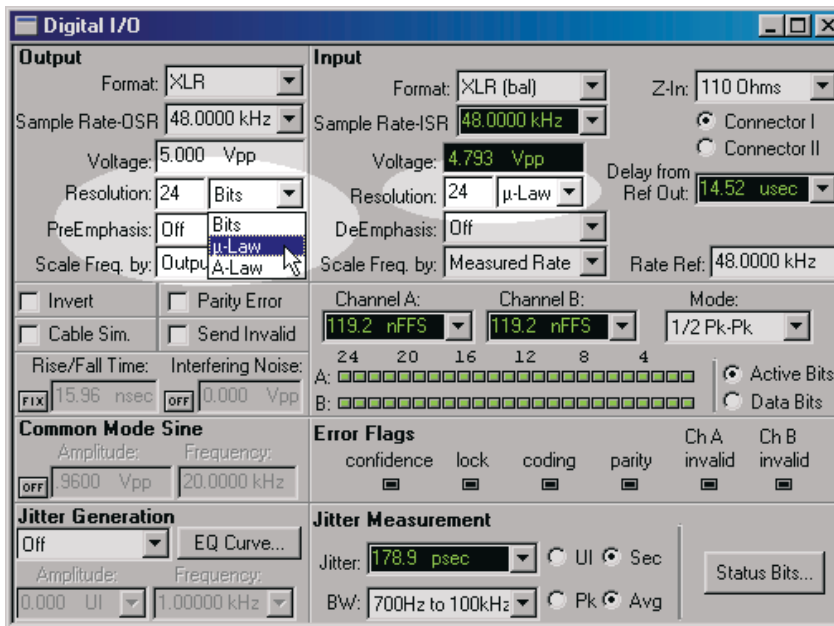
Figure 5. Digital I/O Panel showing µ-Law/A-Law settings.

    µ-Law and A-Law can be selected by new control fields on the Digital I/O panel. The µ-Law/A-Law encoders are found on the Output side of the panel; the decoders are on the Input side. In either section, go just to the right of the **Resolution** field and click on the list arrow next to **Bits** setting. Then simply select **µ-Law** or **A-Law** from the list.

> *The Digital Generator outputs a 14-bit or 13-bit signal when the **Resolution** is set to **µ-Law** or **A-Law**. Dither, if enabled, is properly scaled to these truncated word lengths.*

V 2.11   User Downloadable Filters

    APWIN has a number of built-in filters implemented in software for the Digital Analyzer program, including two low-pass filters, three high-pass filters and six weighting filters. The use of these filters is discussed in detail in the *APWIN User's Manual for System Two Cascade Version 2* in pages 11-12 through 11-16.

APWIN version 2.11 for System Two Cascade added a powerful new feature to the Digital Analyzer: the ability to use custom-designed software filters in any of the three filter groups. These are called User Downloadable Filters.

Creating User Downloadable Filters

An APWIN User Downloadable Filter consists of a properly formatted text file containing information defining the characteristics of the filter. The filter must be named with one of these file name extensions:

- for a low-pass filter, *.afl;

- for a high-pass filter, *.afh;

- for a weighting filter, *.afw.

As long as the file meets the specifications set out below, you can create the filter by any means, including the use of a text editor or third-party software that has digital filter design capabilities.

User Downloadable Filter Design Constraints:

- Each file specifies a single infinite impulse response (IIR) filter at one or more sample rates.

Each IIR filter can be implemented by cascading one or more second-order sections, shown here:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \, .$$

First-order sections are not allowed; a first-order section can be implemented with the second-order section by simply setting b_2 and a_2 to zero.

- A low-pass filter may not have more than 6 poles or 6 zeros; that is, a low-pass filter can have at most three second-order sections.

- A high-pass filter may not have more than 4 poles or 4 zeros; that is, a high-pass filter can have at most two second-order sections.

- A weighting filter may not have more than 8 poles or 8 zeros; that is, a weighting filter can have at most four second-order sections.

- Unstable filters are not allowed. An unstable filter is a filter with poles on or outside the unit circle on the z-plane.

- Filters with a gain of zero are not allowed. Such filters would result in all zero outputs.

- No coefficient may be greater than 2.0 or less than –2.0.

A simple text file format is used for each user-designed filter. The file format is specified by the following rules:

- All lines starting with # are comment lines.

- Blank lines, that is, lines consisting of zero or more spaces and tabs, are ignored.

- The parameters for filters are specified in a line-oriented fashion, with one entity specified per line.

- Each line has three components: a predefined *keyword*, followed by a *colon*, followed by the *keyword-dependent data*. Any white space surrounding these components is ignored. Three keywords are currently defined: **info**, **sample_rate**, and **biquad**.

- The **info** keyword specifies a text string of printable ASCII characters to be displayed on the APWIN User Downloadable Filters panel when the "**Filter Info**" box is clicked. The **info** comment must be fewer than 1024 characters in length. Only the first **info** keyword found is used; any others are ignored.

- A filter at a given sample rate is specified by the **sample_rate** keyword line followed by one or more **biquad** keyword lines.

- The **sample_rate** keyword is followed by a colon and a floating-point number expressing the sample rate in hertz. There is no limit on the number of sample rates. APWIN's built-in filters are provided at these sample rates:

8000 Hz	11025 Hz	12000 Hz	16000 Hz
22050 Hz	24000 Hz	32000 Hz	44100 Hz
48000 Hz	65536 Hz	88200 Hz	96000 Hz
131072 Hz	176400 Hz	192000 Hz	262144 Hz

Table 1. Sample rates for APWIN built-in filters.

- The **biquad** keyword is followed by a colon and five floating-point coefficients a_1, a_2, b_1, b_2, and b_0 in that order, as described in the equation above.

A valid sample file is shown here:

```
info: 100 Hz Butterworth high-pass filter

sample_rate: 8000.0
biquad: -1.8590763 0.8648249 -1.8048889 0.9024445 0.9024445
biquad: -1.9357148 0.9417005 -2.0000000 1.0000000 1.0000000

sample_rate: 16000.0
biquad: -1.9285085 0.9299964 -1.8999636 0.9499818 0.9499818
biquad: -1.9688775 0.9703966 -2.0000000 1.0000000 1.0000000
```

As you can see, a digital filter can be specified by $(5 \cdot N + 1)$ numbers, where N is the number of second-order sections, and 1 is the sample rate.

User Downloadable Filters can also be created by specialized third-party software such as MATLAB, available from The Mathworks, Inc. See Appendix B, "MATLAB Functions" (Page) for four MATLAB functions useful in working with downloadable filters and Audio Precision waveform files.

Audio Precision has also included with APWIN version 2.11 and later a DSP filter creation program from Momentum Data Systems called Filter Design Package for Audio Precision (FDP). See the separate *Filter Design Package User's Manual* ('Filter_Design_Package.pdf' installed in \Apwin\Documentation\UserManuals\) for instruction in the use of this program.

Using User Downloadable Filters in APWIN

You can choose a User Downloadable filter for the Digital Analyzer in the same way that you might choose one of the provided DSP filters.

With the Digital Analyzer display set to its large form (double-click on the panel Title Bar) the low-pass and high-pass filters are selected in the two fields to the right of the **BW** (bandwidth) designation. The weighting filters are selected in the **Fltr** (filter) field.

> *The DSP filters are not available in every measurement function of the Digital Analyzer. Set **Measurement Function** to **Amplitude**, **2-Channel Ratio** or one of the two **THD+N** modes to choose DSP filters.*

Figure 7. User Downloadable Filter Choices

Click the arrow to drop down the list of filters for any of these fields. The last filter option on all lists selects the User Downloadable Filter appropriate for that setting: **User HP**, **User LP** or **User Weighting Filter.**

Although you can save many different user filter files for use in APWIN, you can only select one user file for any of the three filter positions at any one time. When APWIN is launched, by default no user filter files are selected. To choose your filters, click the ellipsis box to the right of the **Fltr** field. A file browser window will appear that will allow you to select filter files for all three the filter types: low-pass, high-pass and weighting. You can also view the filter **Info** string for any of the three selected filters. The default folder for user filter files is C:\Apwin\S2Cascade\Dspfilters.
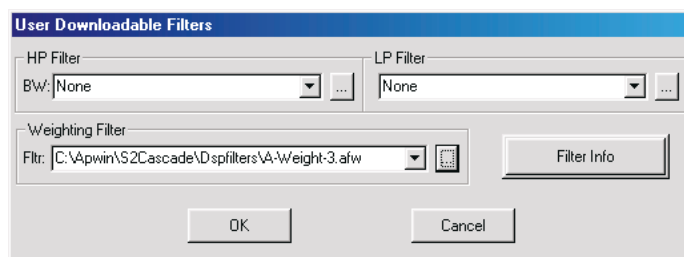
Figure 6. User Downloadable Filter File Browsers.

When you have chosen your user filter files and closed the browser window, notice that the **BW** and **Fltr** fields have been automatically set to

User to reflect your choices. As long as you do not exit APWIN or load a test file, these user filter files will remain attached to their respective filter lists. You still have the option, however, of choosing other APWIN-provided filters (or **None**) from the lists without losing the link to the file you have selected.

A test saved with an attached user filter will re-attach the filter to the Digital Analyzer when the test is loaded.

If you select **User** from any of the three filter lists and a user filter file has not been previously selected for that setting, the user filter file browser will appear and prompt you to choose a filter file.

Quasi-Anechoic Acoustical Tester (MLS)

Two new features have been added to the Quasi-Anechoic Acoustical Tester, also called the Maximum Length Sequence (MLS) tester. These additions have resulted in a rearrangement of the settings fields at the bottom of the Digital Analyzer panel when MLS is selected.

MLS Averaging   V 2.11

When measuring a coherent signal in the presence of uncorrelated noise, synchronous averaging of many measurements will reduce the noise reading and allow the coherent signal to be recovered more effectively. MLS averaging is done synchronously in the time domain. To enable MLS averaging, click the arrow by the new **Averages** field and select the number of readings to be averaged from the list.

*Figure 8. MLS
Averaging
choices.*
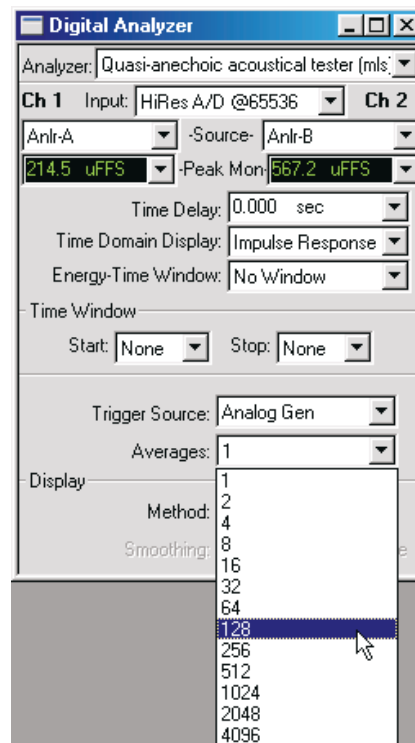
V 2.11   MLS Octave Smoothing

Unlike MLS Averaging, which takes the average of a number of
measurements, MLS Octave Smoothing is a display option that shows the
results of one measurement as modified by a smoothing algorithm.

In previous versions of APWIN, the MLS display options **Interpolate**,
Display Samples and **Peak Values** were available in the **Wave
Display** field. These same display choices are still available in the
Method field in the new Display area of the panel. You can select Octave
Smoothing by clicking the new fourth option, **Smooth**.

Figure 9. Octave Smoothing Option.
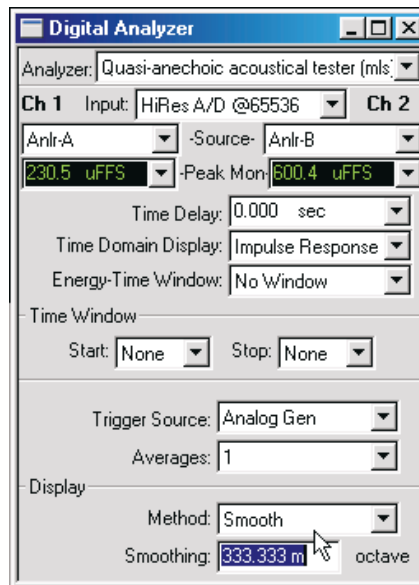
When **Smooth** is chosen the **Smoothing** setting field becomes available. Here you can specify the degree of smoothing by entering values from 0 to 2.64 octaves.

Octave smoothing is a common technique in loudspeaker response measurement, useful in revealing trends by smoothing out anomalies in the response curve. The APWIN implementation uses a hybrid FFT bin averaging and interpolation technique to achieve smooth results even at very low bin densities. Smoothing, which only affects frequency-domain displays, effectively passes the raw response data through multiple constant-Q bandpass filters, one filter centered on each frequency requested from the Sweep panel. The bandwidth of these filters, in octaves, is specified in the **Smoothing** field.

Default Test Settings for Jitter Settling V 2.11

Audio Precision provides many sample test files with APWIN, as well as a default test file which defines the base parameters for each new test you begin. The initial conditions for all these tests depend on the default settings saved in each test file.

As of APWIN version 2.11, the default test file settings for interface jitter settling have been made more discriminating to offer better performance. The new settings will more consistently recognize settled jitter amplitude readings during sweeps.
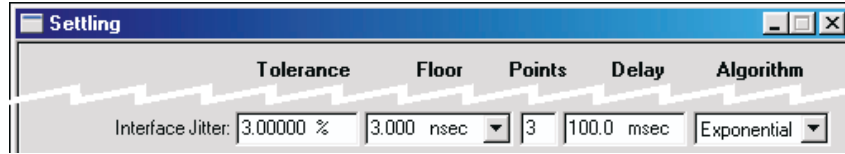
*Figure 12. **Interface Jitter** settings on the DIO section of the Settling panel. Shown above are the defaults used before APWIN version 2.11.*
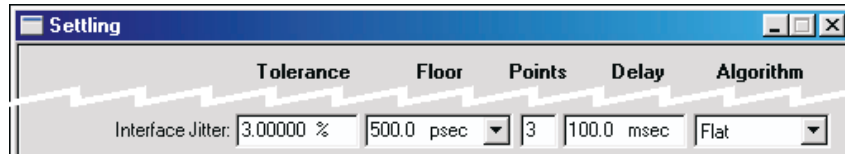
Figure 10. Jitter settling, new defaults.
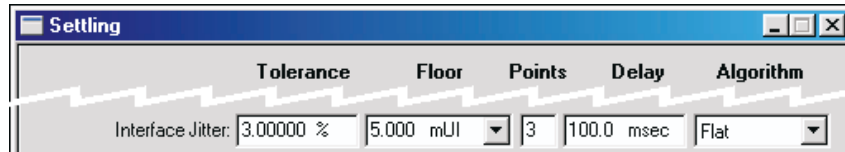
Figure 11. Jitter settling, new defaults in UI.

Using the previous defaults, the occurrence of several closely-spaced jitter amplitude readings in the sweep between desired data points would sometimes satisfy **Settling**; the last value of the series would then be reported as a settled reading. The Settling panel **Floor** default setting has been 3 ns, which is a rather broad range for many jitter amplitude readings. In the test files provided with APWIN version 2.11 and later, the **Floor** setting has been reduced to 500 ps; additionally, **Algorithm** has been set to **Flat**. Figures 12, 10 and 11 compare the old defaults to the new.

> *If you choose to view jitter in UI rather than in seconds, the default **Floor** setting is 5 mUI, which is equivalent to about 800 ps at the default sampling rate of 48 kHz. However, if you should set a different sample rate, the value of the **Floor** setting in seconds will change, and with it the discrimination of settled jitter readings.*

This change is only a default setting for new tests. Older versions of tests will retain the settings incorporated in them.

Of course, you may change and save new settings at any time.

Chapter 3: S2, S2C, S2C+

Changes Affecting
■ *System Two*
■ *System Two Cascade*
■ *System Two Cascade* Plus

Also *see* Chapter 4 for other changes in the operation of System Two, System Two Cascade and System Two Cascade *Plus* running under APWIN version 2.11 and later.

Analog Analyzer: AES17 Low-Pass Filter   V 2.2

Figure 13. AES17 Low-Pass Filter Selection and Channel A routing.
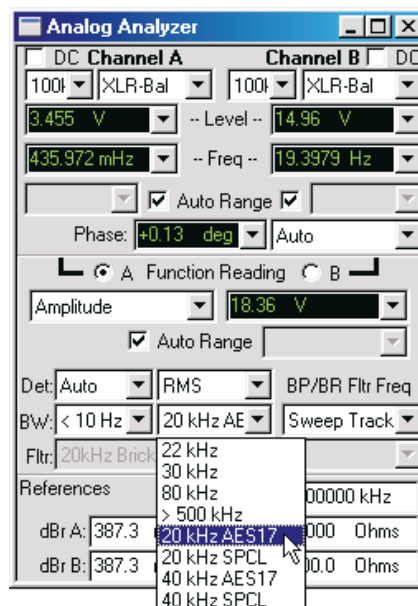
AES17 specifies a standard low-pass filter for THD+N measurements of digital-to-analog converters (DACs) which exhibit high-level out-of-band noise.

Audio Precision now manufactures a new hardware filter for optional installation in System Two, System Two Cascade or System Two Cascade *Plus* which satisfies the AES17 specification. APWIN 2.2 and later versions support the use of this filter.

This new filter is designated the S-AES17 Low-Pass Filter Option, and replaces the earlier S2-AES17LP filter released in March of 2000. The new S-AES17 option offers improved performance and features compared to the earlier S2-AES17LP.

Contact your Audio Precision representative for information about ordering the S-AES17 option. The option consists of a dual-frequency pre-analyzer filter module, the SLPX; two additional analyzer option filters, FLP-B20K and FLP-B40K, and installation software and instructions.

An essential feature of the S-AES17 filter is its location *previous* to the Analog Analyzer, where it attenuates the out-of-band noise components before they can overload the circuitry and make distortion and other low-level measurements difficult. The additional option filters complete the job to satisfy the AES17 recommendation.

When installed, the AES17 filter is enabled by choosing one of the four new choices available on the low-pass **BW** (bandwidth) filter list.

*The four AES17 filter selections will not appear on the **BW** list until the filter installation software (which is provided with the filter kit) is run.*
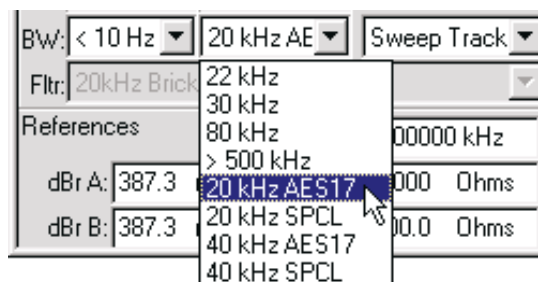
The new selections are:

- **20 kHz AES17**
  This choice selects both the 20 kHz pre-analyzer filter and the 20 kHz brick-wall option filter. Selection of other option filters is disabled. If the required option filter is not installed, a warning will appear and the selection will default to **20 kHz SPCL**.

- **20 kHz SPCL**

   This choice selects only the 20 kHz pre-analyzer filter. You may choose any option filter, or **None**.

- **40 kHz AES17**

   This choice selects both the 40 kHz pre-analyzer filter and the 40 kHz brick-wall option filter. Selection of other option filters is disabled. If the required option filter is not installed, a warning will appear and the selection will default to **40 kHz SPCL**.

- **40 kHz SPCL**

   This choice selects only the 40 kHz pre-analyzer filter. You may choose any option filter, or **None**.

The pre-analyzer section of the filter operates on only one channel at a time and follows the **A** or **B** channel selection made when setting **Function Reading**, as shown in Figures 13 and 14.
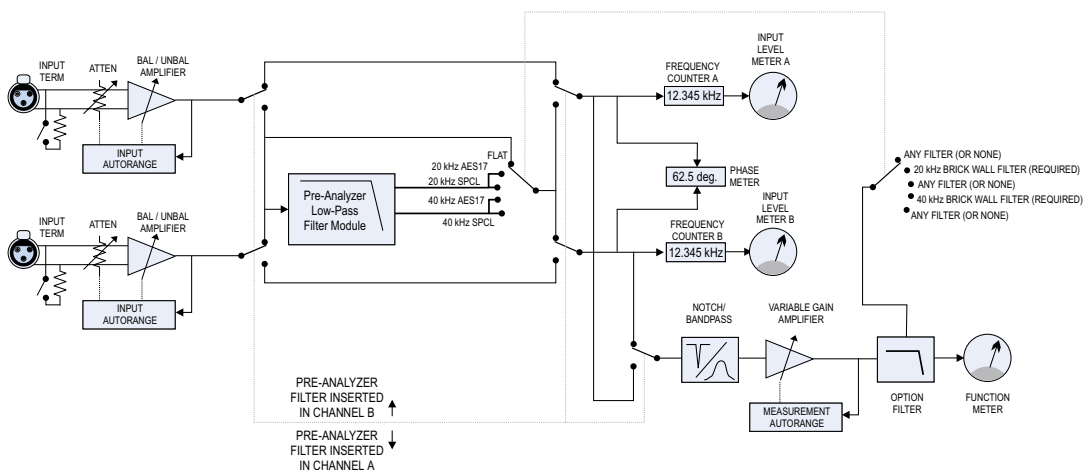
Figure 14. Block Diagram of AES17 Low-Pass Filter implementation in System Two, System Two Cascade or System Two Cascade Plus. Note the location of the pre-analyzer section of the filter, and the linked impletmentation of the post-analyzer option filters.

In the 20 kHz AES17 mode, the combination of pre-analyzer filter and the option filter is designed to provide a flat passband (± 0.1 dB) through 20 kHz with a stopband attenuation of 60 dB or better above 24 kHz, satisfying the AES17-1998 Section 4.2.1 specification for a "standard low-pass filter."

In the 40 kHz AES17 mode, the combination of pre-analyzer filter and the option filter is designed to provide a flat passband (± 0.1 dB) through 40 kHz with a stopband attenuation of 60 dB or better above 48 kHz.

> *This higher-bandwidth mode can be used with the high-bandwidth DACs typically used to convert direct streaming digital (DSD) audio analog audio. DSD is the key audio format used in making super audio compact discs (SACD).*

In either case, this filter will attenuate out-of-band noise typically created by oversampled converters and provide meaningful THD+N measurements of audio signals accompanied by high out-of-band noise. For detailed information see Appendix A, "AES17 Filter" (Page 45) and the documentation provided with the filter.

V 2.11 — Digital I/O Panel: Rate Ref

The **Rate Ref** entry field on the Digital I/O panel will now accept a wider range of entry settings. The previous range was 8 kHz–192 kHz; in APWIN version 2.11 and later versions, any frequency value between 6.75 kHz and 216 kHz can be entered as the **Rate Ref**.
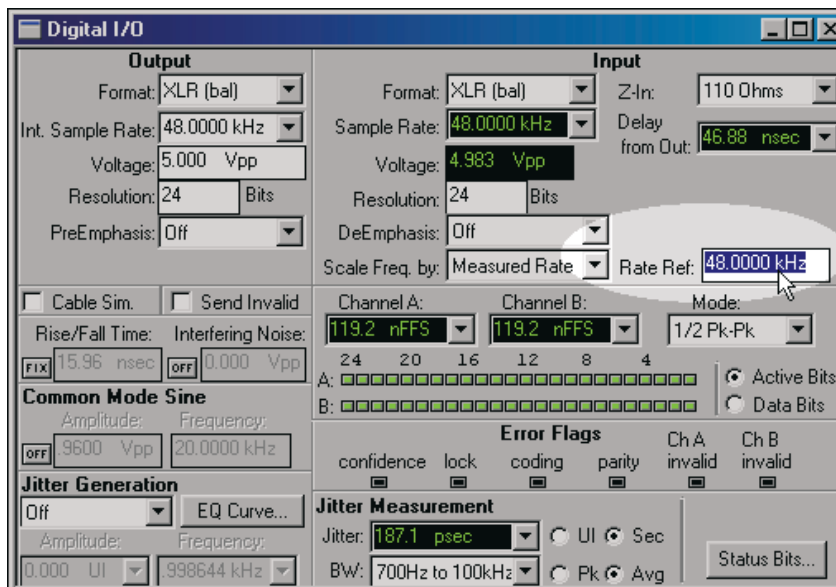
*Figure 15. Digital I/O Panel **Rate Ref** setting, set at 48.0000 kHz (System Two shown; System Two Cascade has similar appearance).*

The **Rate Ref** entry field has two uses, one interface-related and one digital audio-related. For digital audio measurements, this field serves as an absolutely stable value for the nominal sample rate when **DIO Rate Ref** is selected in the **Scale Freq. by** field. For serial digital interface parameter measurements, the **Rate Ref** value is the reference for all

relative frequency units selectable in the **Sample Rate** (System Two Cascade: **Sample Rate-ISR**) display field. Thus, if it is desired to display measured sample rate in terms of PPM deviation from the nominal rate, enter the nominal rate into the **Rate Ref** field and select PPM units for the **Sample Rate** (System Two Cascade: **Sample Rate-ISR**) display.

> *For System Two, **Scale Freq. by** is a digital input setting only, and when **Rate Ref** is selected in this field it serves as a reference for the input sample rate. For System Two Cascade under APWIN 2.11 and later, **Scale Freq. by** is also a digital output setting, and **Rate Ref** can be used as a reference for either (or both) input or output sample rate scaling. See **Digital Output Sample Rate Scaling** (Page 6).*

For more information about using **Rate Ref** settings, see Page 7-5 in the *APWIN User's Manual for System Two Version 2* or Page 7-6 in the *APWIN User's Manual for System Two Cascade Version 2*.

Shaped Burst Performance Improved   V 2.14

Previous limitations to the Digital Generator **Shaped Burst** duration have been raised from 2^{16} samples (65,536) to ($2^{23} - 1$) samples (8,388,607) in APWIN version 2.14. This allows much longer shaped bursts to be specified.

Chapter 4: S1, S2, S2C, S2C+

Changes Affecting
■ System One
■ System Two
■ System Two Cascade
■ System Two Cascade Plus

Generator Auto ON/OFF Option   `V 2.2`

*Figure 16. **Auto On** checkboxes on Analog and Digital Generator panels.*
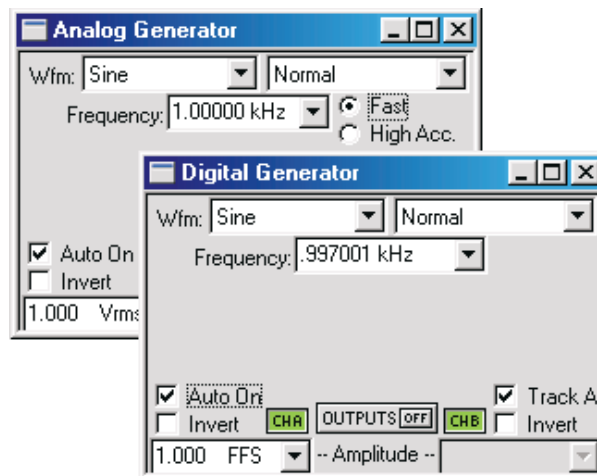
A new feature for all systems is the generator **Auto On** function. For both the Analog and Digital Generators, **Auto On** automatically switches the generator **ON** when a sweep starts, and **OFF** when the sweep terminates.

This feature will be of particular interest for those involved in power amplifier or loudspeaker testing. With the generator set to **OFF** and **Auto**

On enabled, signal will only be applied to the DUT while the sweep is actually running.

The initial transient created when a generator is turned on requires a delay before measurement to allow the generator, DUT, ranging and analyzer circuits to stabilize. When **Auto On** is enabled, the **ON** command is applied to the generator at the moment the sweep starts. To avoid the effects of the switching transient, set the **Pre-Sweep Delay** on the expanded Sweep panel to a sufficient time, in many cases **.5 s** to **1 s** or more.

> *Tests saved with the generator set to **ON** will open under the same condition, with the generator **ON**. **Auto On** only affects the generator state during a sweep.*

V 2.11 APWIN Support for New PCI Interface Card

Audio Precision has made the APIB interface available on both ISA and PCMCIA cards for a number of years. As a third option, you can now connect your PC to System One, System Two or System Two Cascade using a PCI APIB interface card. Software drivers for the PCI interface were included as part of APWIN version 2.11.

Your PC operating system must be either Microsoft® Windows 98®, Windows 2000®, Windows ME® or Windows NT® 4.0 to be compatible with the PCI interface. S1.EXE running under DOS or APWIN running under Windows 95® are not compatible with the PCI interface.

> *The APWIN PCI interface is recommended for use with System One analog domain (SYS-22a) and with all configurations of System Two and System Two Cascade.*

To install a PCI-WIN interface card, first install the APWIN software on your computer. Then shut down the computer, disconnect its power, remove the cover and install the card in an unused PCI slot. Install the hold-down screw on the card bracket and replace the cover.

When the computer is turned on, the software should detect the PCI-WIN card and install the proper drivers.

For more information about installing APIB cards, see *APWIN Installation & Getting Started* for your System and APWIN version.

With version 2.14, the ISA, PCI and PCMCIA interface drivers have been further updated to improve performance and to broaden the combinations of interface cards and operating systems available.

See the latest APIB interface driver compatibility chart at audioprecision.com/techsupport/compatibility.htm.

Interface Drivers Folder **V 2.14**

When APWIN is installed prior to the hardware installation of an ISA, PCI or PCMCIA APIB interface card, the drivers are normally installed by the Windows Plug and Play feature. If Windows prompts you to locate a driver manually, go to the Drivers folder (new in APWIN version 2.14), found in the APWIN CD-ROM root folder.

Changes to External Sweeps Operation **V 2.11**

Background: Internal and External sweeps

Many of the sweeps performed in APWIN involve one of the internal generators, with the sweep following controlled parameters such as frequency or amplitude. The sweep and the generator share the control settings, and there is no ambiguity about sweep direction, or sweep **Start** or **Stop** values, for example.

However, it is often necessary to have APWIN track a sweep source that is not under internal control, such as a CD alignment disc, a recorder alignment tape, or a remote sweep generator. This is accomplished in the external sweep mode. When performing an external sweep, APWIN monitors the input signal and extracts sweep controls from the signal itself.

This information is always less certain than the mutually shared settings of an internal sweep, and APWIN requires some hints: the range and direction of the sweep, under what conditions to graph a point, when to end the sweep, and so on. With real-world signals, satisfying these criteria can be difficult, and external sweeps can sometimes fail to start or stop, or gather too many or too few points, or graph spurious points. As of APWIN version 2.11, several changes have been made to improve the performance of external sweeps.

> *If you have performed external sweeps with earlier versions of APWIN, read* **"Start On" Rules** *(Page 28) carefully to determine which external sweep rule sets the behavior you prefer. The default rule is* **Within Start Tolerance,** *which corresponds to the external sweep behavior in APWIN versions 2.01 through 2.10. The Start On rule can be changed with an OLE command, as explained on Pages 28 and 29.*

New External Sweep Operation

External sweeps are driven by actual measurements. The **Source 1** field on the Sweep Panel must be configured as a meter reading rather than a generator setting, and the reading must satisfy the requirements

entered in the Settling Panel for that meter. For a frequency sweep, the meter will be one of the frequency meters. For amplitude-controlled sweeps a level or amplitude meter is normally used.
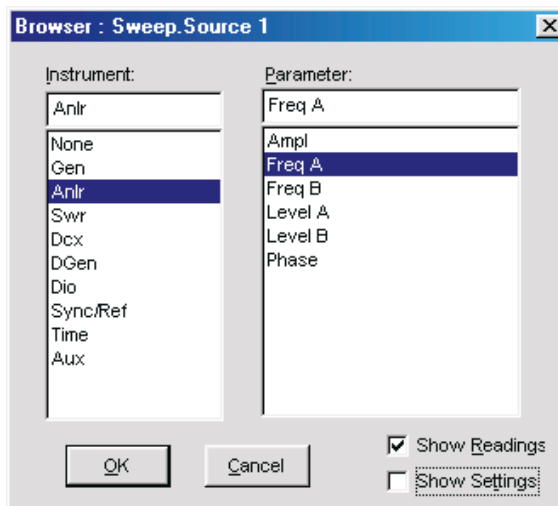
Figure 17. Sweep Source 1 Browser, set to Freq A meter readings.

*Other meters may be selected for special or difficult measurements. For example, the DSP analyzer in **Bandpass** mode can be used to control an amplitude sweep with Analog Analyzer measurements. The narrow bandpass filter in the DSP analyzer allows measurements with much-reduced sensitivity to noise.*

Start and Stop

The extent and direction of the sweep are set by the **Start** and **Stop** fields. If the value of **Start** is less than **Stop**, the sweep will proceed upward, in the direction of lesser to greater. If **Start** is greater than **Stop**, the sweep will proceed downward.

Spacing

The **Spacing** field helps to determine the step size within the sweep. **Spacing** is entered as a percentage, and the range for each new data point is the value of the last point plotted plus or minus the **Spacing** percentage. At the beginning of a sweep, the Start On rules (below) determine whether or not **Spacing** affects the acquisition of the first plotted point.

Figure 18. Sweep Panel in External Sweep Mode.
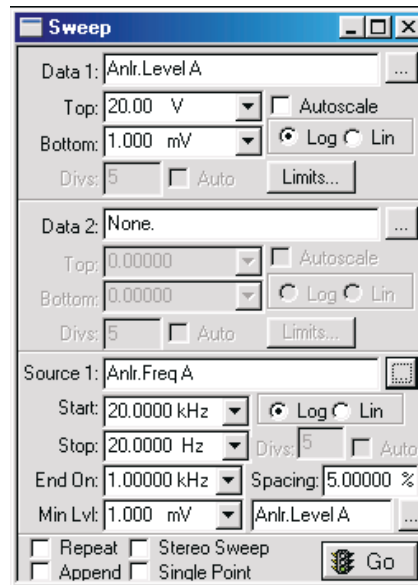
End On

The **End On** field in the Source 1 section of the Sweep panel allows you to set the conditions under which an External Sweep test will terminate. Of course, an External Sweep test can be manually terminated by pressing the **Esc** key or clicking on the **Stop** icon when it is apparent that the sequence of test tones has ended, but **End On** (or the equivalent OLE command, AP.Sweep.Source1.EndOn, if in a procedure) can terminate the sweep automatically.

End On now follows this logic:

- if the value set in **End On** equals the **Stop** value (plus or minus the **Spacing** percentage), the sweep will terminate when the **End On** value occurs;

- if the value set in **End On** does *not* equal **Stop** (±**Spacing**) and is outside the range set by **Start** and **Stop**, the sweep will terminate when the **End On** value occurs;

- if the value set in **End On** does not equal **Stop** (±**Spacing**) and is *within* the range set by **Start** and **Stop**, the sweep will terminate on a second occurrence of the **End On** value.

This produces several different behaviors:

- The first case, where **End On** equals **Stop** (±**Spacing**), allows you to set **End On** at the end of your desired sweep. The last value acquired (**Stop** ±**Spacing**) will be graphed.

■ The second case, where **End On** does not equal **Stop** (±**Spacing**) and is outside the **Start–Stop** range, allows you to set **End On** beyond your desired sweep. You can, for example, place an out-of-band cue tone after the swept tones to terminate the sweep.

■ The third case, where **End On** does not equal **Stop** (±**Spacing**) and is within the **Start–Stop** range, allows you to set **End On** to a value in the mid-range of your desired sweep. This accommodates, for example, a frequency test which sweeps across the desired spectrum and then returns to a mid-spectrum tone. If **End On** is set to that mid-spectrum tone, it will note the first occurrence of that tone as the sweep passes through it, and at the return the second occurrence of the tone will terminate the sweep. Many pre-recorded test tapes and CDs are made just this way, with, for example, a 20 Hz to 20 kHz sweep followed by a 1 kHz reference tone. When performing split-site measurements with an Audio Precision System One or System Two, the remote generator can be set up so that the generator dwells at its Generator panel setting (which can be a mid-band reference frequency) before and after a Source 1 frequency sweep.

Min Lvl

The value set in the **Min Lvl** (minimum level) field allows you to exclude any measurements below a preset level, as measured by the meter specified in the field just to the right of **Min Lvl**. This acts as a "noise gate," preventing noise or low-level signals from interfering with the process. When **Min Lvl** is set very low, all signals will exceed the setting and will be plotted; when **Min Lvl** is set very high, no signals will exceed the setting and nothing will be plotted.

"Start On" Rules

An APWIN sweep is initiated by pressing **F9** or clicking **GO**. In an internal sweep, two things happen in response to this: the generator you have selected begins its sweep, and the acquisition of sweep measurements is enabled. In an external sweep, of course, APWIN does not start a generator, but **F9** (or **GO**) enables the acquisition of sweep measurements in the same way. However, no data will be acquired until the externally-generated signal satisfies the conditions set by the Start On rule which is in effect. There are three Start On rules. The default is

■ **Within Start Tolerance.** (OLE command control 0). The sweep starts on a settled reading that is above the **Min Lvl** value, and that is also within the tolerance set by applying the **Spacing** percentage to the **Start** value. For example, if **Start** is 20 Hz and **Spacing** is 5 %, a reading between 19 and 21 Hz will

start the sweep. Or, if **Start** is 1 V rms and **Spacing** is 10 %, an amplitude between 0.9 and 1.1 V rms will start the sweep. This rule was new with APWIN version 2.01, and is the default behavior for APWIN version 2.11 and later versions.

This rule governs the normal behavior of external sweeps. By using OLE commands (a feature which was new with APWIN version 2.11), the advanced user can set one of two other Start On rules.

- **Beyond Start Value.** (OLE command control 1).
  The sweep starts on a settled reading that is above the **Min Lvl** value and exceeds the **Start** value in the sweep direction (up or down, as determined by the relative values of the **Start** and **End** settings). No data is collected until the **Source1** meter reading passes (or equals) the **Start** value. This rule was new with APWIN version 2.11.

- **Any Settled Reading**. (OLE command control 2).
  The sweep starts on a settled reading that is above the **Min Lvl** value, without regard to the value in the **Start** field. Sweep measurements are collected in the direction set by the **Start** and **Stop** field values. In a sweep with **Start** at 20 Hz and **Stop** at 20 kHz, a 1 kHz measurement will not accept the next value until 1 kHz +**Spacing** is satisfied. Under this rule, the data collected can include values above or below the sweep graph limits set in **Start** and **Stop**. This rule was the default behavior in APWIN version 2.00 and earlier.

External Sweep OLE Commands

If you would like to read other meters in APWIN while the sweep is running, use:

```
AP.Sweep.StartNoWait
```

To select one of the three Start On rules listed above, use:

```
AP.Sweep.External.StartOnRule
```

If you are sitting at the keyboard, you can manually start and stop the sweep. However, if the external sweep is within a procedure, it may hang the program while you are away from your desk. These commands can be used to escape a running sweep:

```
AP.Sweep.AbortTime
AP.Application.SetWatchDogTimer1
AP.Application.SetWatchDogTimer2
```

Some Hints

In a glide sweep, the signal is gliding continuously from one value to another and will never settle as it will in a step sweep. In an external frequency glide sweep it is essential that the frequency **Settling Algorithm** be set to **None**, the **Points** to **1**, and **Delay** to **0**.

Also, check your **Min Lvl** settings. Acting rather like a noise gate, **Min Lvl** sets an amplitude threshold below which data will not be acquired; this is useful in directing the sweep to ignore the silence between tracks on a test CD, for example. The default meter assigned to make the **Min Lvl** reading is **Anlr A**, but a browser can offer you a selection of sources. Be sure the meter that you have selected is the appropriate meter for your test, that you have considered the effect of any filters you may have attached to the meter, and that the **Min Lvl** threshold setting is at the correct level.

V 2.11   Installation Program Warning

The APWIN Installation Program now detects previous installations of APWIN and displays a warning message. You are then given the choice to continue, to choose a different folder for installation, or to abort the installation altogether.
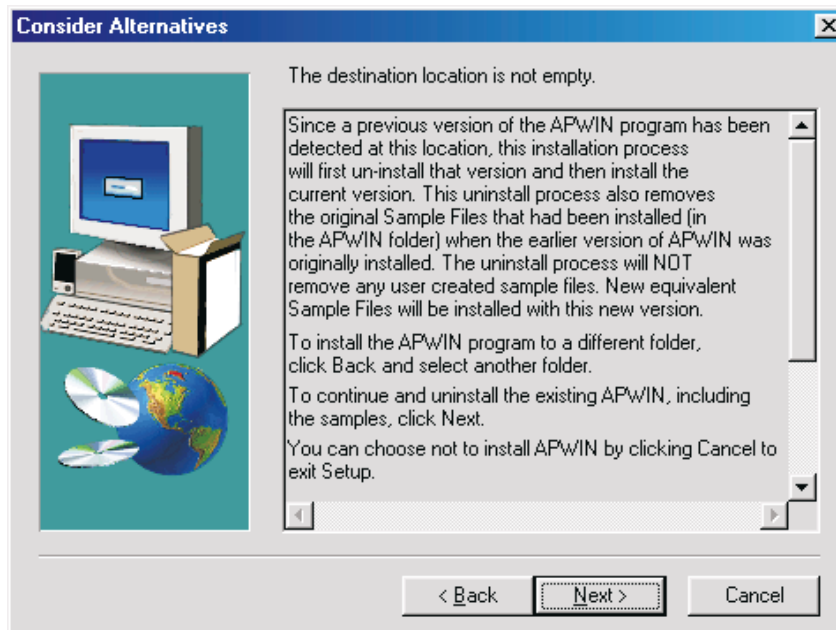
Figure 19. Installation warning

See *APWIN Installation & Getting Started* for your platform and version for detailed information about installing APWIN.

Japanese Language O.S. Support   V 2.14

Conflicts in character code usage caused APWIN to operate improperly under the Japanese language version of Microsoft Windows 2000. This has been fixed in APWIN version 2.14.

New Audio Track List   V 2.14

The APWIN version 2.14 CD-ROM audio test recording tracks have undergone some modifications and deletions, with the result that the CD track numbering has changed. Check the track list in the CD-ROM booklet.

New Sample Files   V 2.14

With APWIN 2.14, New sample files for External Sweeps ("X-files") have been added. Check the file list in the CD-ROM booklet.

Chapter 5: AP Basic

AP Basic Extensions

This chapter features both OLE commands which have changed and new OLE commands which have been added to APWIN Basic Extensions to provide programmatic control for the new features first introduced in APWIN 2.11.

OLE Commands for System Two Cascade

S2C: User Downloadable Filters

Three new OLE commands for programming with User Downloadable Filters follow. See **User Downloadable Filters**, Page 8.

AP.S2CDsp.Analyzer.FuncFilterHPUserDefined   Method

Syntax	`AP.S2CDsp.Analyzer.FuncFilterHPUserDefined(`ByVal *FileName* As String`)`

Parameters	Name	Description
	FileName	Long Path and File Names permitted up to 128 characters. The file must be an APWIN high-pass filter file (.afh). Enter "None" for the file name to remove the User-Defined Filter.

Result	Boolean	
	True	File attachment successful.

False	File attachment failed.

Description   This command attaches a User-Defined Filter for use
with the DSP Audio Analyzer High Pass filter selection.
When the filter is attached, the design is tested to
determine that there are not more than 2 second-order
sections used to create the filter and that the filter is
stable. If either of these conditions is not met then this
command returns False. To select the attached
User-Defined Filter select "User Defined" with the
AP.S2CDsp.Analyzer.FuncFilterHP command.

See Also   `AP.S2CDsp.Analyzer.FuncFilterHP,`
`AP.S2CDsp.Analyzer.FuncFilterLPUserDefi`
`ned,`
`AP.S2CDsp.Analyzer.FuncFilterWeightingU`
`serDefined`

Example   ```
Sub Main
 AP.S2CDsp.Analyzer.FuncFilterHPUserDef
ined _
 ("My_HP-1.afh")
End Sub
```

## AP.S2CDsp.Analyzer.FuncFilterLPUserDefined        Method

**Syntax**      **`AP.S2CDsp.Analyzer.FuncFilterLPUserDef`**
**`ined(`**`ByVal FileName As String`**`)`**

**Parameters**

| Name | Description |
|------|-------------|
| *FileName* | Long Path and File Names permitted up to 128 characters. The file must be an APWIN low-pass filter file (.afl). Enter "None" for the file name to remove the User-Defined Filter. |

**Result**      Boolean

| | |
|---|---|
| *True* | File attachment successful. |
| *False* | File attachment failed. |

**Description**    This command attaches a User-Defined Filter for use with the DSP Audio Analyzer Low Pass filter selection. When the filter is attached, the design is tested to determine that there are not more than 3 second-order sections used to create the filter and that the filter is stable. If either of these conditions is not met then this command returns False. To select the attached User-Defined Filter select "User   Defined" with the AP.S2CDsp.Analyzer.FuncFilterLP command.

**See Also**    
```
AP.S2CDsp.Analyzer.FuncFilterLP,
AP.S2CDsp.Analyzer.FuncFilterHPUserDef
ined,
AP.S2CDsp.Analyzer.FuncFilterWeighting
UserDefined
```

**Example**    
```
Sub Main
 AP.S2CDsp.Analyzer.FuncFilterLPUserDe
fined _
 ("My_LP-1.afl")
End Sub
```

## AP.S2CDsp.Analyzer.FuncFilterWeightingUserDefined   Method

**Syntax**    
```
AP.S2CDsp.Analyzer.FuncFilterWeighting
UserDefined(ByVal FileName As String)
```

**Parameters**

| Name | Description |
|---|---|
| *FileName* | Long Path and File Names permitted up to 128 characters. The file must be an APWIN weighting filter file (.afw). Enter "None" for the file name to remove the User-Defined Filter. |

**Result**    Boolean

| | |
|---|---|
| *True* | File attachment successful. |
| *False* | File attachment failed. |

**Description**    This command attaches a User-Defined Filter for use with the DSP Audio Analyzer Weighting filter selection. When the filter is attached, the design is tested to determine that there are not more than 4 second-order sections used to create the filter and that the filter is stable. If either of these conditions is not met then this command returns False. To select the attached User-Defined Filter select "User Defined" with the AP.S2CDsp.Analyzer.FuncFilter command.

**See Also**
```
AP.S2CDsp.Analyzer.FuncFilter,
AP.S2CDsp.Analyzer.FuncFilterHPUserDef
ined,
AP.S2CDsp.Analyzer.FuncFilterLPUserDef
ined
```

**Example**
```
Sub Main
 AP.S2CDsp.Analyzer.FuncFilterWeightin
gUser _ Defined("My WT-1.afw")
End Sub
```

### S2C: μ-Law and A-Law

Two new OLE commands for programming with μ-Law and A-Law companding follow. See **μ-Law and A-Law**, Page 7.

## AP.S2CDio.InDecode                                    **Property**

**Syntax**       `AP.S2CDio.InDecode`

**Data Type**    Integer

| | |
|---|---|
| *0* | No data compression applied. |
| *1* | Apply μ-Law decoding to data signal. |
| *2* | Apply A-Law decoding to data signal. |

**Description**  This command selects the Digital Interface Receive Data Format Expansion.

**See Also**
```
AP.S2CDio.OutEncode,
AP.SI.RecData.Format.Decode,
AP.SI.TransData.Format.Encode
```

**Example**          Sub Main
               AP.S2CDio.InDecode = 1    'μ-Law
             decode
             End Sub

## AP.S2CDio.OutEncode                                **Property**

**Syntax**          `AP.S2CDio.OutEncode`

**Data Type**       Integer

| | |
|---|---|
| *0* | No data compression applied. |
| *1* | Apply μ-Law encoding to data signal. |
| *2* | Apply A-Law encoding to data signal. |

**Description**     This command selects the Digital Interface Transmit
                    Data Format Compression.

**See Also**        AP.S2CDio.InDecode,
                    AP.SI.RecData.Format.Decode,
                    AP.SI.TransData.Format.Encode

**Example**         Sub Main
                     AP.S2CDio.OutEncode = 2    'A-Law
                    Encode
                    End Sub

### S2C: MLS Averaging and Smoothing

Two new OLE commands for programming with
MLS Averaging and MLS Octave Smoothing follow.
See **Quasi-Anechoic Acoustical Tester (MLS)**,
Page 13.

## AP.S2CDsp.Mls.Averages                             **Property**

**Syntax**          `AP.S2CDsp.Mls.Averages`

**Data Type**       Integer

| | |
|---|---|
| *0* | 1 |
| *1* | 2 |

| | |
|---|---|
| *2* | 4 |
| *3* | 8 |
| *4* | 16 |
| *5* | 32 |
| *6* | 64 |
| *7* | 128 |
| *8* | 256 |
| *9* | 512 |
| *10* | 1024 |
| *11* | 2048 |
| *12* | 4096 |

**Description**   This command sets the number of acquisitions for the averaging function of the Quasi-Anechoic Acoustical Tester (MLS).
When measuring a coherent signal in the presence of uncorrelated noise, synchronous averaging of many measurements will reduce the noise reading and allow the coherent signal to be recovered more effectively. MLS averaging is done synchronously in the time domain.

**Example**
```
Sub Main
 AP.S2CDsp.Mls.Averages = 9
 'set MLS Av. to 512
End Sub
```

## AP.S2CDsp.Mls.Smoothing                                          Property

**Syntax**        `AP.S2CDsp.Mls.Smoothing`

**Data Type**     Double          Range of Values: 0 to 2.64 octaves

**Description**     This command controls the width of the MLS Smoothing algorithm in octave units.

Octave smoothing is a common technique in loudspeaker response measurement, useful in revealing trends by smoothing out anomalies in the response curve. The APWIN implementation uses a hybrid FFT bin averaging and interpolation technique to achieve smooth results even at very low bin densities. Smoothing, which only affects frequency-domain displays, effectively passes the raw response data through multiple constant-Q bandpass filters, one filter centered on each frequency requested from the Sweep panel.

**Example**
```
Sub Main
 AP.S2CDsp.Mls.Smoothing = .3333
  'set MLS smoothing to 1/3 octave
End Sub
```

### S2C: Digital Output Sample Rate Scaling

A new OLE command for programming with Digital Output Sample Rate Scaling follows. See **Digital Output Sample Rate Scaling**, Page 6.

## AP.S2CDIO.OutScaleFreq                        Property

**Syntax**          `AP.S2CDIO.OutScaleFreq`

**Data Type**       Integer

| | |
|---|---|
| *0* | **Output Rate**: the value in the **Sample Rate-OSR** field near the top of the Output section of the DIO panel. |
| *1* | **Meas Input Rate**: the measured value in the **Sample Rate-ISR** field. |
| *2* | **Meas Output Rate**: the measured value at the parallel output port when **Output Format** is set to **Parallel**; otherwise, the value in the **Sample Rate-OSR** field. |
| *3* | **DIO Rate Ref**: the value defined by the `AP.S2CDio.RefRate` command. |

**Description**        This command sets the source for scaling for the audio
                       frequency embedded in the digital output signal.

**See Also**           `AP.S2CDIO.InScaleFreq,`
                       `AP.S2CDIO.RefRate`

**Example**            `Sub Main`
                       ` AP.S2CDIO.OutScaleFreq = 1`
                       ` 'use measured input rate`
                       `End Sub`

## S2C: Digital Input Sample Rate Scaling

The existing OLE command for programming with
Digital Input Sample Rate Scaling has been changed,
as follows. See **Digital Input Sample Rate
Scaling**, Page 5.

## AP.S2CDIO.InScaleFreq                                         Property

**Syntax**             `AP.S2CDIO.OutScaleFreq`

**Data Type**          Integer

*0*                    **Output Rate**: the value in the Sample
                       Rate-OSR field near the top of the Output
                       section of the DIO panel.

*1*                    **Meas Input Rate**: the measured value in
                       the **Sample Rate-ISR** field.

*2*                    **Status Bits A**: the value of sample
                       frequency encoded into the received
                       channel A status bits.

*3*                    **DIO Rate Ref**: the 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
                       embedded 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.OutScaleFreq,`
                       `AP.S2CDIO.RefRate`

**Example**        
```
Sub Main
 AP.S2CDIO.InScaleFreq = 1
 'use measured input rate
End Sub
```

# OLE Commands for System Two

## S2: Digital Output Sample Rate Scaling

A new OLE command for programming with Digital Output Sample Rate Scaling follows.

---

## AP.S2DIO.OutScaleFreq                    Property

**Syntax**         `AP.S2DIO.OutScaleFreq`

**Data Type**      Integer

| | |
|---|---|
| *0* | **Output Rate**: the value in the **Int. Sample Rate** field near the top of the Output section of the DIO panel. |
| *1* | **Meas Input Rate**: the measured value in the **Sample Rate** field near the top of the Input section of the DIO panel. |
| *2* | **DIO Rate Ref**: the value defined by the `AP.S2Dio.RefRate` command. |

**Description**    This command sets the source for scaling for the audio frequency embedded in the digital output signal.

**See Also**       `AP.S2DIO.InScaleFreq, AP.S2DIO.RefRate`

**Example**        
```
Sub Main
 AP.S2DIO.OutScaleFreq = 1
  'use measured input rate
End Sub
```

## S2: Digital Input Sample Rate Scaling

The existing OLE command for programming with Digital Input Sample Rate Scaling has been changed, as follows.

---

## AP.S2DIO.InScaleFreq                     Property

**Syntax**         `AP.S2DIO.OutScaleFreq`

**Data Type**      Integer

| | | |
|---|---|---|
| *0* | | **Output Rate**: the value in the **Int. Sample Rate** field near the top of the Output section of the DIO panel. |
| *1* | | **Meas Input Rate**: the measured value in the **Sample Rate** field near the top of the Input section of the DIO panel. |
| *2* | | **Status Bits A**: the value of sample frequency encoded into the received channel A status bits. |
| *3* | | **DIO Rate Ref**: the value defined by the `AP.S2Dio.RefRate` command. |

**Description**   This command selects a source from which the digital audio sample rate is determined. The frequency of embedded 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.S2DIO.OutScaleFreq,`
`AP.S2DIO.RefRate`

**Example**
```
Sub Main
 AP.S2DIO.InScaleFreq = 1
  'use measured input rate
End Sub
```

## OLE Commands for All Systems

### S1, S2, S2C: External Sweep Start On Rules

A new OLE command for programming with the external sweep Start On rules follows.

## AP.Sweep.External.StartOnRule                    Property

**Syntax**   `AP.Sweep.External.StartOnRule`

**Data Type**   Long

| | |
|---|---|
| *0* | **Within Start Tolerance:** (Default) Data collection begins as soon as a settled reading is detected that is within the **Start** value plus or minus the **Spacing** tolerance as specified by the `AP.Sweep.Source1.Start,` and `AP.Sweep.Source1.Spacing` commands.<br>**Note:** this sweep Start On behavior was exhibited in APWIN versions 2.01 through 2.1. |
| *1* | **Beyond Start Value:** Data collection begins as soon as a settled reading is detected that is within the **Start** and **Stop** values as specified by the `AP.Sweep.Source1.Start` and `AP.Sweep.Source1.Stop` commands. Additional readings are retained as long as the readings satisfy the sweep direction and **Spacing** requirement as specified by the `AP.Sweep.Source1.Spacing` command. |
| *2* | **Any Settled Reading:** Data collection begins as soon as a settled reading is detected. Additional readings are retained as long as the readings satisfy the sweep direction and **Spacing** requirements as specified by the `AP.Sweep.Source1.Start,` `AP.Sweep.Source1.Stop,` and `AP.Sweep.Source1.Spacing` commands.<br>**Note:** this sweep Start On behavior was exhibited in APWIN versions 1.0 through 2.0. |

**Description**    This command selects the external sweep Start On rule.

**See Also**
```
AP.Sweep.Source1.Start,
AP.Sweep.Source1.Stop,
AP.Sweep.Source1.Spacing
```

**Example**
```
Sub Main
 AP.Sweep.External.StartOnRule = 1
 'use Beyond Start Value rule
End Sub
```

# Appendix A

## *AES17 Low-Pass Filter*

The Audio Precision S-AES17 Low-Pass Filter Option provides a sharp roll-off beyond 20 kHz early in the measurement path to insure that measurements accurately reflect the true performance within the audio band. As an additional feature, the S-AES17 filter has a switchable mode which offers a broader passband, setting the sharp roll-off beyond 40 kHz.

This filter is a hardware option which meets the requirements of AES17-1998 and can be added to System Two, System Two Cascade or System Two Cascade *Plus*. It is unlike other optional filters available from Audio Precision in that it uses a larger module, and also in that this module is inserted in a different location in the measurement circuit, just after the input conditioning amplifier and prior to any measurement circuits. The S-AES17 filter option also contains two option filters, FLP-B20K and FLP-B40K, which must be used in conjunction with the pre-analyzer low-pass module for proper operation.

The Audio Precision S-AES17 Low-Pass Filter Option improves upon the capabilities of the earlier S2-AES17LP filter, which is no longer available.

## Introduction

Digital-to-analog converters frequently use delta-sigma modulation and oversampling techniques to achieve high performance at an affordable cost. The noise inherent in this type of conversion is pushed up in frequency out of the audio band by means of high-order noise shaping. The resultant signal is quiet within the audio range but carries large amounts of out-of-band noise.

Measuring the audio performance of such converters can be a challenge for most contemporary wide-band audio distortion and noise analyzers. These instruments are designed to characterize classic analog audio

devices, which typically exhibit a noise floor spread evenly over the full spectrum of the analyzer, usually diminishing with increased frequency. Out-of-band noise and interference are expected to be low in amplitude in relation to the signal, so band-limiting or noise-weighting filters, if needed, are inserted at the end of the measurement path, following several gain stages. When measuring classic analog audio devices, this approach yields accurate and repeatable THD+N results.



*Figure 20. Conceptual signal path diagram showing how the AES17 Low-Pass Filter attenuates out-of-band noise and allows accurate measurement of audio-band distortion.*

The spectrum of the noise floor of a noise-shaped delta-sigma D-to-A converter, however, shows a steeply rising energy characteristic beyond the 20 kHz upper limit of the audio band. When measuring low-level signals, the energy contribution of this ultrasonic noise can be substantial. In many situations it can overload instrument gain stages or throw off ranging circuits and cause grossly inaccurate measurements. Conventional band-limiting and noise-weighting filters cannot solve the problem because they are located too late in the measurement chain—the damage has already been done.

## The AES17 Low-Pass Filter Specification

The Audio Engineering Society has defined the techniques for measuring digital audio equipment—that is, systems that contain digital-to-analog converters—in its standard, AES17. In section 4.2.1 AES17-1998 specifies the use of a "standard low-pass filter" that has a sharp roll-off above the audio band to attenuate out-of-band noise.

The AES17 standard specifies a stop-band attenuation of 60 dB or better above 24 kHz, quite a steep slope. The filter must be inserted early in the measurement path in order to remove the out-of-band noise before

the measurement notch filter and its subsequent gain. This will insure that the "+N" part of THD+N contains only the in-band noise and distortion. Without the filter, the automatic gain ranging that normally follows the THD+N notch filter can behave incorrectly and the resulting measurement will be in error.



*Figure 21. S-AES17 Low-Pass Filter Option passband and stopband response with 20 kHz setting, used with FLP-B20K option filter.*

Figure 20 graphs the performance of the Audio Precision S-AES17 Low-Pass Filter Option at the **20 kHz** setting, used with the option filter FLP-B20K. This filter option satisfies the AES17 standard.

*SPECIFICATIONS*

*The S-AES17 Low-Pass Filter Option is intended for measuring the THD+N of D-to-A converters in accordance with AES17-1998 (section 4.2.1.1). It replaces the earlier Audio Precision S2-AES17LP filter.*

*The filter option consists of a dual-frequency pre-analyzer low-pass module, the SLPX; two additional analyzer option filters, FLP-B20K and FLP-B40K, and installation software and instructions.*

*An essential feature of the pre-analyzer low-pass module is its location previous to the Analog Analyzer, where it attenuates the out-of-band noise components before they can overload the circuitry and make distortion and other low-level measurements difficult. The additional option filters complete the job to satisfy the AES17 recommendation.*

*For specified performance, the S-AES17 option must be used with both the pre-analyzer low-pass module and the option filter FLP-B20K when set for 20 kHz, and with option filter FLP-B40K when set for 40 kHz.*

| S-AES17 Specifications | |
|---|---|
| Passband Response, 20 kHz setting: | ± 0.10 dB, 10 Hz–20.0 kHz |
| Passband Response, 40 kHz setting: | ± 0.10 dB, 10 Hz–40.0 kHz |
| Stopband Attenuation, 20 kHz setting: | $^3$ 60 dB, 24.0 kHz–200 kHz |
| Stopband Attenuation, 40 kHz setting: | $^3$ 60 dB, 48.0 kHz–200 kHz |
| Residual THD+N (1 kHz), 20 kHz AES mode:<br>Residual THD+N (1 kHz), 40 kHz AES mode: | £ (0.0003% [–110.5 dB] +1.0 µV)<br>£ (0.0004% [–108 dB] +1.4 µV) |

# Appendix B

## *MATLAB Functions*

MATLAB is a technical computing environment used in DSP design and analysis. MATLAB is a product of The Mathworks, Inc., who can be reached at 508-647-7000 or on the Web at http://www.mathworks.com.

The following MATLAB functions can be found in four MATLAB files (each with the filename extension *.m), which have been installed in the folder C:\Apwin\Matlab. The files are provided with APWIN version 2.11 and later versions, as an aid to our customers who also use MATLAB.

These functions are useful in creating and modifying downloadable filters and Audio Precision waveform files for use with APWIN and System Two Cascade.

## Downloadable Filter Support

Two MATLAB functions supporting Audio Precision downloadable filters are provided. `ap_write_filter` generates filter files from within MATLAB, and `ap_read_filter` imports such files into MATLAB for further manipulation.

### ap_write_filter

*Syntax*

```
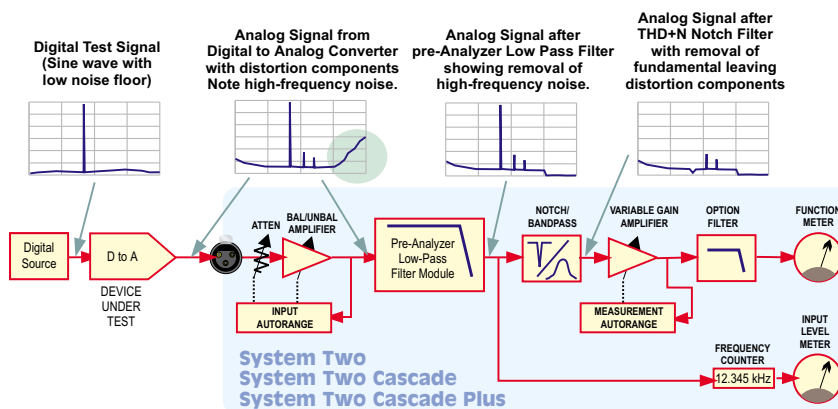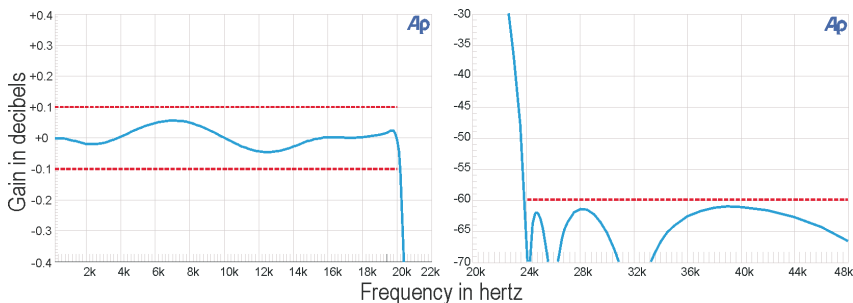size = ap_write_filter(filename,filter_type,info,sample_rate,sos)
size = ap_write_filter(filename,filter_type,info,sample_rate,b,a)
size = ap_write_filter(filename,filter_type,info,sample_rate,z,p,k)
size = ap_write_filter(filename,filter_type,info,sample_rate,a,b,c,d)
size = ap_write_filter(fid,filter_type,info,sample_rate,sos)
size = ap_write_filter(fid,filter_type,info,sample_rate,b,a)
size = ap_write_filter(fid,filter_type,info,sample_rate,z,p,k)
size = ap_write_filter(fid,filter_type,info,sample_rate,a,b,c,d)
```

Description

ap_write_filter generates a text file in the format recognized by the downloadable filters feature of APWIN version 2.11 and later. Filters in any of the standard MATLAB forms (second-order section, transfer function, zero-pole-gain, and state-space) can be written to the file, along with the sample rate and textual information.

The AP downloadable filter format consists of one or more sample rates, with accompanying filter coefficients in the form of second-order sections. ap_write_filter converts the filter into second-order section form before writing it to the file. This will require functions found in the MATLAB signal processing toolbox if the filter is not already in second-order section form.

ap_write_filter accepts both filenames and file handles. If a filename is supplied, any extension is replaced with .afl, .afh, or .afw, depending on the filter type. If the filename ends in '+', the file is appended to, or created if non-existent. ap_write_filter closes the file when the function exits. If a file handle is supplied, ap_write_filter appends to the file, leaving it open on exit.

The arguments filter_type, info, and sample_rate are required. filter_type is the string 'l', 'h', or 'w', for low-pass, high-pass, and weighting filters respectively. ap_write_filter uses this information to determine the filename extension, and to restrict the number of second-order sections to the maximum allowed in the Audio Precision hardware for each filter type. info is a character string that can be viewed inside APWIN by clicking the **Filter Info** button after loading the filter. This provides filter information to the APWIN user. sample_rate, a scalar, is the sample rate in hertz for which the filter was designed. See **Multiple Sample Rates** for more information.

ap_write_filter uses the number of remaining arguments to determine whether the filter is supplied in second-order section form, in

transfer function form, in zero-pole-gain form, or in state-space form. The transfer function form is not recommended for large filters because factorization into second-order sections may cause the poles and zeros to move slightly, deforming the filter response.

The total number of bytes written to the file is returned in `size`.

Multiple Sample Rates

A digital filter will have the intended frequency response only at the sample rate for which it was designed. Thus, a unique filter must be designed for each sample rate that will be encountered. If a filter file contains filters for multiple sample rates, APWIN chooses the filter coefficients corresponding to the sample rate closest to the current hardware sample rate.

An Audio Precision filter file can contain a filter defined at any number of sample rates. Either a file handle or the file append mode can be used to add filters for other sample rates to an existing file. `ap_write_filter` writes an **info** string for each sample rate; however, APWIN will only report the first of these strings.

Filter Restrictions

The number of filter orders is limited to 6 for low-pass filters, 4 for high-pass filters, and 8 for weighting filters. The sample rate must be between 6750 Hz and 262144 Hz. All zeros must be on or inside the unit circle, and all poles must be inside the unit circle. All coefficients must be in the range [-2,2].

Examples

1. Compute a Chebyshev high-pass filter with a corner frequency of 200 Hz and a passband ripple of 0.1 dB for three common sample rates.

```
for sr=[32000 44100 48000]
  [z,p,k]=cheby1(4,0.1,200/(sr/2),'high');
  ap_write_filter('c:\filters\cheby200+','h','200 Hz Cheby highpass',sr,z,p,k)
end
```

We use the zero-pole-gain form for accuracy, and the append mode to attach the three sample rates to the same file.

2. Compute a voice-band boosting filter at two sample rates. Use the file handle method to write to the output file.

```
ofp=fopen('c:\filters\voice.afw','wt');
fprintf(ofp,'# Voice band filter designed in MATLAB by yulewalk\n\n');
for sr=[48000 65536]
  [b,a]=yulewalk(8,[0 500 1000 4000 8000 sr/2]/(sr/2),[0.5 0.5 1 1 0.25 0.25]);
  ap_write_filter(ofp,'w','Voice band filter',sr,b,a)
end
fclose(ofp);
```

Here we have the flexibility to write comments to the output file (preceded by the '#' identifier) which will not appear inside APWIN. Note that it is the user's responsibility to create a file with the appropriate extension, and to properly close the file.

See also `ap_read_filter`.

ap_read_filter

Syntax

```
[filter_array,info] = ap_read_filter(filename)
[filter_array,info] = ap_read_filter(fid)
```

Description

`ap_read_filter` imports an Audio Precision downloadable filter text file into MATLAB. It parses the file and returns a structure array of filters, one structure for each sample rate contained in the file. It also returns the **info** comment strings. The file is closed on exit.

The AP downloadable filter format consists of one or more sample rates, with accompanying filter coefficients in the form of second-order sections. `ap_read_filter` creates one structure in the output array for each sample rate in the file. Each structure contains the sample rate and a matrix consisting of the filter in second-order section form. If other forms are required, such as state-space, the signal processing toolbox conversion functions should be used.

Comments in the file are discarded, but **info** strings are concatenated into the returned `info` variable. The **info** strings are used by APWIN to provide information to the user when the **Filter Info** button is clicked.

`ap_read_filter` attempts to deal with syntax errors in the file by issuing a warning of the line number of the error.

Example

To read the filter file generated by the first example in `ap_read_filter`:

```
[filters,info] = ap_read_filter('c:\filters\cheby200.afw');
```

This returns a 1×3 structure array. Typing `filters(1)` at the prompt returns

```
ans = sample_rate: 32000
              sos: [2x6 double]
```

The second-order sections can be examined by typing `filters(1).sos` at the prompt.

See also `ap_write_filter`.

AP Waveform File Support

Two MATLAB functions supporting Audio Precision formatted wave files are provided. `ap_write_wave` generates AP wave files from within MATLAB, and `ap_read_wave` imports such files into MATLAB for further manipulation.

ap_write_wave

Syntax

```
size = ap_write_wave(filename,sample_rate,normalize,data)
size = ap_write_wave(fid,sample_rate,normalize,data)
```

Description

`ap_write_wave` generates a binary waveform file that can be loaded into Audio Precision hardware. Files with extensions .agm (mono generator waveform) and .ags (stereo generator waveform) are supported.

The AP waveform file format consists of a 256-byte header containing sample rate and other information, and a $3n$-byte payload, where n is the number of samples in the waveform. Each sample is quantized to 24 bits and is in the range [-1, $1-2^{-23}$].

`ap_write_wave` accepts both filenames and file handles. If a filename is supplied, any extension is replaced with .agm or .ags, depending on the size of the data matrix. `ap_write_wave` closes the file when the function exits. If a file handle is supplied, `ap_write_wave` appends to the file, leaving it open on exit.

`sample_rate`, a scalar, is the sample rate of the waveform in hertz.

If `normalize` is 1, the data is scaled so that the peak of the waveform reaches full scale. The data is also dithered before reducing the word size to 24 bits. This preserves low-level information and eliminates harmonic distortion due to quantization. This mode is recommended unless multiple waveforms are being generated whose amplitude relative to one another is important.

If `normalize` is 0, the data is written to the file unscaled. If any samples are outside the allowable range, they are clipped and a warning is issued.

`data` is a matrix containing the waveform. If the smaller dimension is 1, a mono waveform file is generated. If it is 2, a stereo waveform file is generated. Any other size will generate an error.

Example

At a sample rate of 40 kHz, create a sine wave of approximately 1 kHz in channel A and noise in channel B. The length is 2048 samples.

```
len=2048;
samp_num=0:len-1;
samp_rate=40000;
ch_A=sin(samp_num/len*2*pi*round(1000/samp_rate*len));
ch_B=randn(1,len);
ap_write_wave('c:\waveforms\sin_noise',40000,1,[ch_A; ch_B])
```

This generates a file sin_noise.ags. The actual frequency of the sine wave is 996.094 Hz. This is a frequency that is synchronous with the sample rate and the waveform length. That is, when the waveform is repeated, there are no discontinuities.

See also `ap_read_wave`.

ap_read_wave

Syntax

```
waves = ap_read_wave(filename)
waves = ap_read_wave(fid)
```

Description

`ap_read_wave` imports an Audio Precision waveform file into a MATLAB structure array. One structure is created for each waveform in the file. Files with extensions .aam (mono acquired waveform), .aas (stereo

acquired waveform), .agm (mono generator waveform), and .ags (stereo generator waveform) are supported. The file is closed on exit.

Each structure consists of a sample rate, a trigger point, a time vector, and the waveform vector. The sample rate and trigger point are determined from the header information in the waveform file.

For acquired waveforms, the trigger point is the sample at which the trigger occurred during acquisition. For generated waveforms (such as arbitrary waveforms created by the Multitone Waveform utility in APWIN), the trigger point is usually zero. The time vector is constructed from the trigger point, the total number of points in the waveform, and the sample rate. The data vector consists of the file payload converted to `double`.

If `ap_read_wave` determines that the file contains data acquired from an analog source, it attempts to scale the data so that the value of a sample represents the analog voltage at that time. Because some gain constants are known only to the hardware on which the waveform was acquired, however, this will only be approximate. Data originally from a digital source is not scaled.

Example

Read the .ags file generated by the `ap_write_wave` example.

```
waves=ap_read_wave('c:\waveforms\sin_noise.ags');
```

This returns a 1×2 structure array. Typing `waves(1)` at the prompt returns

```
ans = sample_rate: 40000
    trigger_point: 0
             time: [2048x1 double]
             data: [2048x1 double]
```

The data can now be examined:

```
figure(1), plot(waves(1).time,waves(1).data)
figure(2), plot(waves(2).time,waves(2).data)
```

See also `ap_write_wave`.