Audio Precision Technical Training

API Training

Using APx500 v 4.4

Agenda

- **Introduction**
- **Development Strategies**
- **Development Resources**
- **API Review**
- **Programming Language Support**
  - Multiple Language Sample Code Review
- **LabVIEW & DAQ Driver support**
  - LabVIEW Driver
  - DAQ Driver

Introduction

- **What's is an API?**
  - Application Programming Interface

- **Who can benefit from the APx500 API?**
  - Engineering
  - Manufacturing
  - Quality Control
  - Others…

- **What are the benefits?**
  - Simplification of complex automation
    - Lower skill level required
  - Speed
  - Repeatable results
  - ?

Development Strategies…

- **Issues to take into account when making this decision.**
  - Complex UI?
  - Required Software platform?
  - Data collection requirements?
  - Other requirements?
    - Statistics?

- **What other APx500 features may be helpful**
  - Sequence Mode
    - Navigator
    - Production Test Mode
    - Measurement automation (Sub Steps)
    - Automatic Measurement Data Reporting

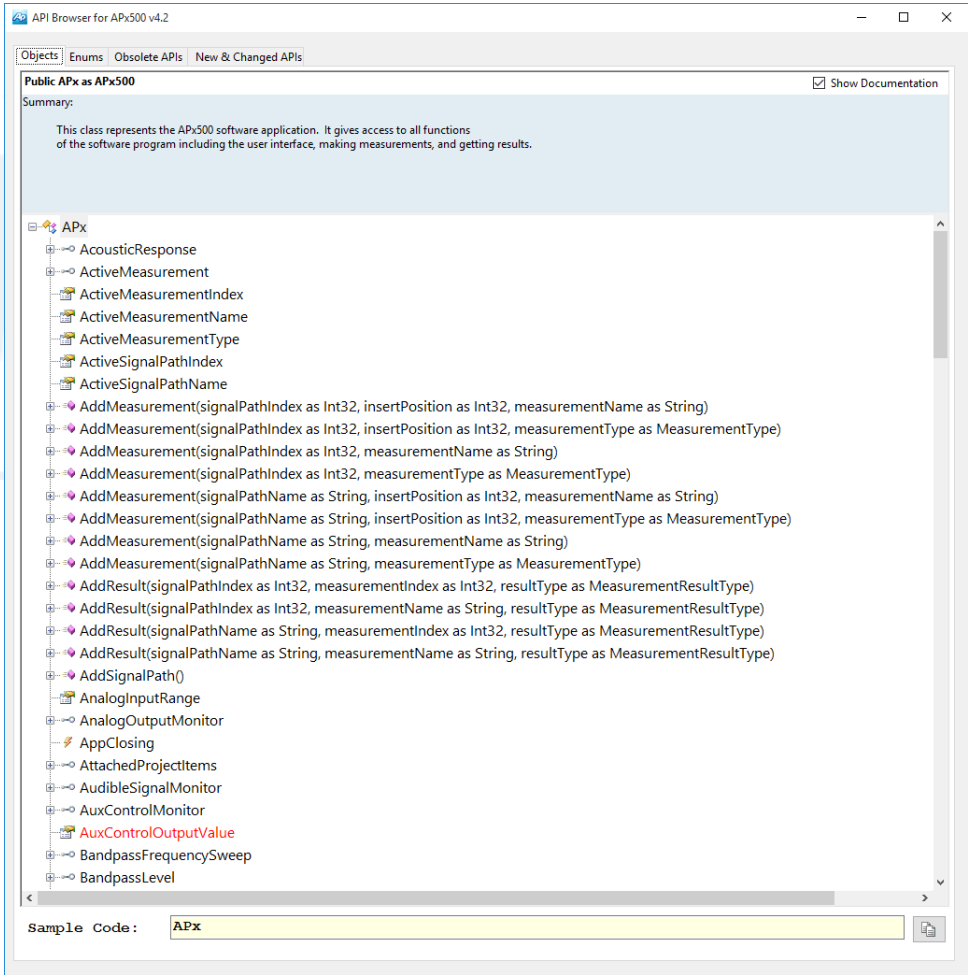
Development Strategies

- **Audio testing is a large part of the overall solution**
  - Using the APx500 sequencer
  - Using the Production Test features

- **Audio testing is a small part of the overall solution**
  - Using the APx500 sequencer combined with an API application
  - Using only an API application

Development Resources…

- **Provided via APx500 installation…**
  - API Browser Utility
    - Added to Start menu during installation
    - Multiple instances permitted
    - Hierarchical Tree View
    - Tabs
      - Objects
      - Enums
      - Obsolete APIs
      - New & Changed APIs
    - F1 launches Programmer's Reference Guide Help for selected command
    - **More on this later**



Development Resources…

- **Provided via APx500 installation**
  - Programmer's Reference Guide (APx500_API_PRG.chm)
    - UI to API Graphics
    - AudioPrecision.API
    - Namespace
      - Classes
        - » Methods
        - » Properties
      - Enumerations
      - Delegates
        - » ApplicationClosingEventHandler
        - » MeterValuesChangedEventHandler
    - Provides search capabilities
    - Provides Inheritance information

Development Resources…

- **Provided via AP.com web-site…**
  - APx API Developer Tools (multiple versions)
    - http://www.ap.com/?s=Developer+Tools&submit=Search
      - Documentation Tools (also provided via installation)
      - Examples
      - Project Templates
      - About the APx500 API (API)
      - About the APx500 API Wrapper (API2)
      - Introduction to Programming the APx500 API
  - Controlling APx500 with Python article
    - http://www.ap.com/technical-library/controlling-apx500-with-python
      - *Sample code*
  - APx LabVIEW .NET Driver (multiple versions)
    - http://www.ap.com/?s=APx+LabVIEW+.NET+Driver&submit=Search
      - APx LabVIEW .NET Driver Getting Started (.pdf)
  - APx LabVIEW DAQ Driver (for APx515 only)
    - http://www.ap.com/?s=APx+LabVIEW+DAQ+Driver+(for+APx515+only)&submit=Search
      - APx LabVIEW DAQ Driver Getting Started (.pdf)

Development Resources…

- **Provided via AP.com web-site**
  - Matlab Toolbox
    - http://www.ap.com/?s=Matlab+Toolbox&submit=Search
    - Functions for legacy and AP2700 products
    - Functions for APx series analyzers
      - apx_read_wave.m
        - » Read waveforms saved by APx500, WfmGenerator.exe, and apx_write_wave.m
      - apx_write_wave.m
        - » Write Apx-compatible generator waveforms

API Review

- **Use API Browser to view API hierarchy.**
  - Provided via APx500 software installation and downloadable from AP.com

- **Object Oriented**
  - The API uses an object oriented design which allows common functionality and provides a consistent programming interface across many parts of the software
  - API has many “levels” where desired functionality may be deep down inside an object

Properties and Methods

- **Objects expose Properties which get and set single setting values**
  - `APx.SignalPathSetup.OutputConnector.Type = OutputConnectorType.AnalogBalanced`
- **Objects expose Methods which are function calls which have an effect on the system**
  - `APx.FrequencyResponse.Start()`
  - `APx.FrequencyResponse.SequenceMeasurement.Run()`
  - `APx.FrequencyResponse.ClearData()`
  - APx.LevelAndGain.ExportData("c:\temp\LevelAndGain.csv")

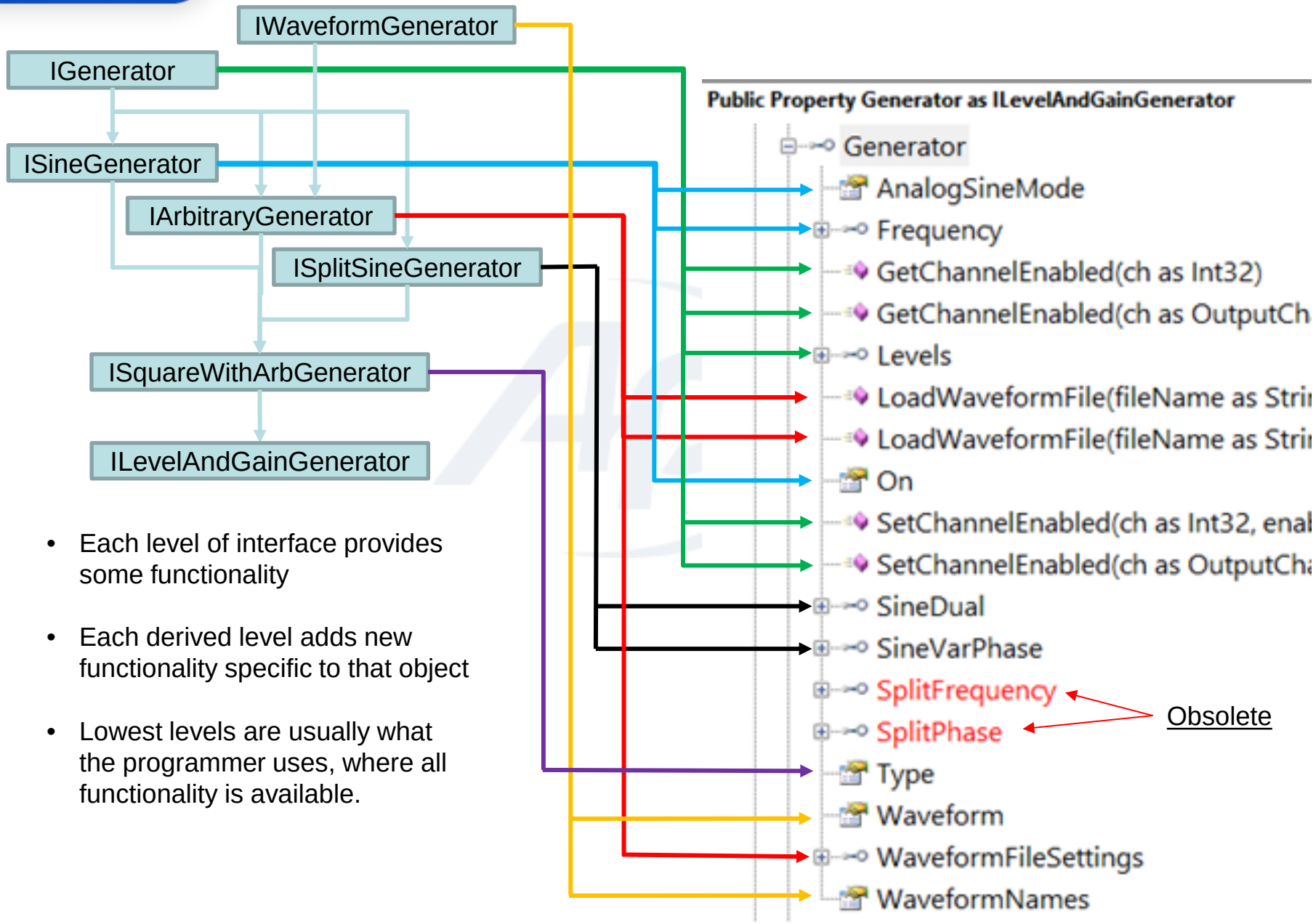
Interfaces and Inheritance…

- **Interfaces define entry points into the API**

- **All entry points into the API (with the exception of the APx500 object) are interfaces and the objects type names start with a capital "I". Example: "IFrequencyResponseMeasurement"**

- **Inheritance allows a "base" interface to provide common functions that are shared across multiple objects**
  - Example: IGenerator has GetChannelEnabled and SetChannelEnabled methods
  - ISineGenerator has a Frequency property
  - ILevelAndGainGenerator inherits from both IGenerator and ISineGenerator and therefore has GetChannelEnabled, SetChannelEnabled methods, and the Frequency property

Interfaces and Inheritance

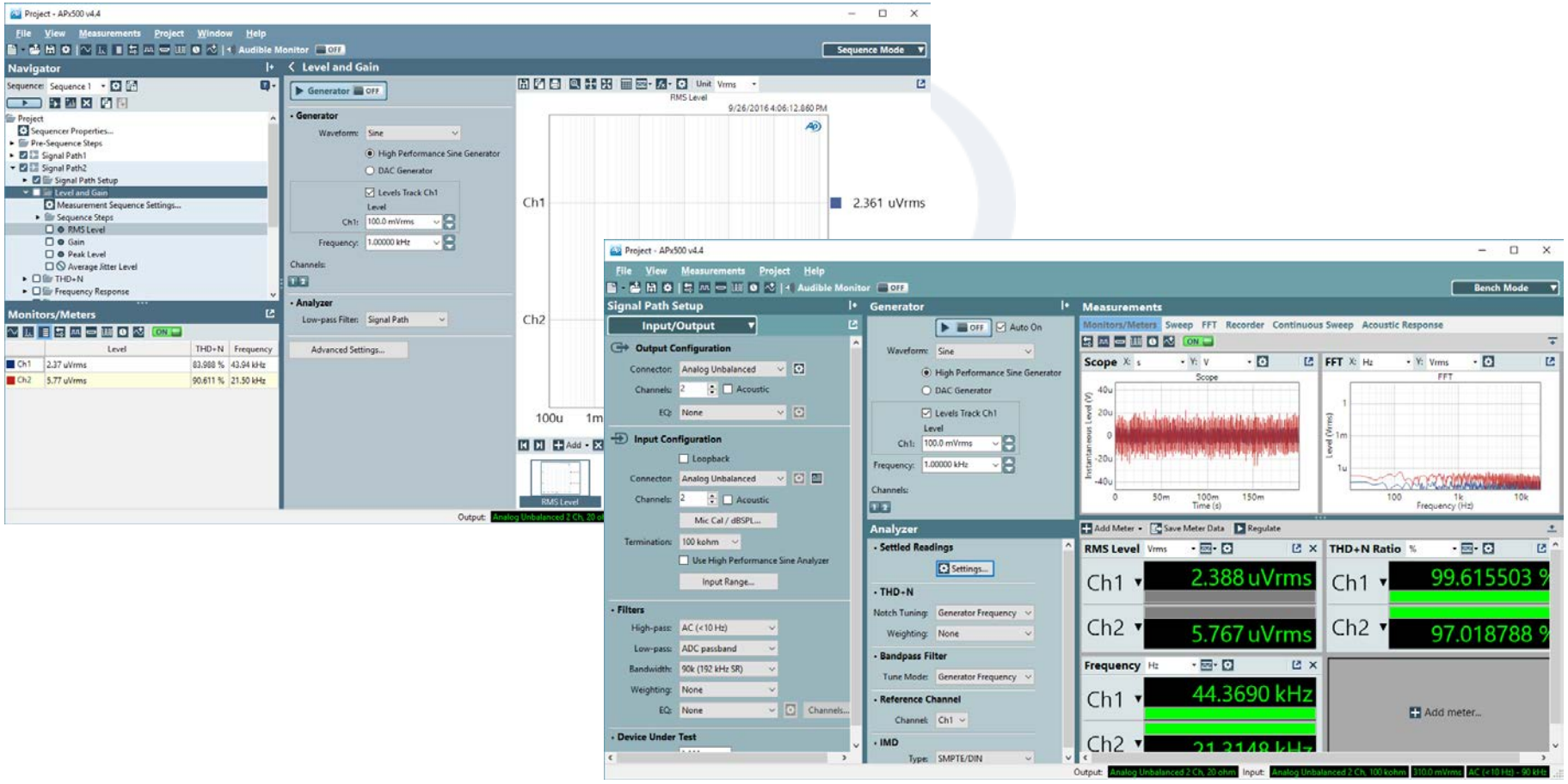


- Each level of interface provides some functionality
- Each derived level adds new functionality specific to that object
- Lowest levels are usually what the programmer uses, where all functionality is available.

APx500 API Architecture

APx500 has two operating modes

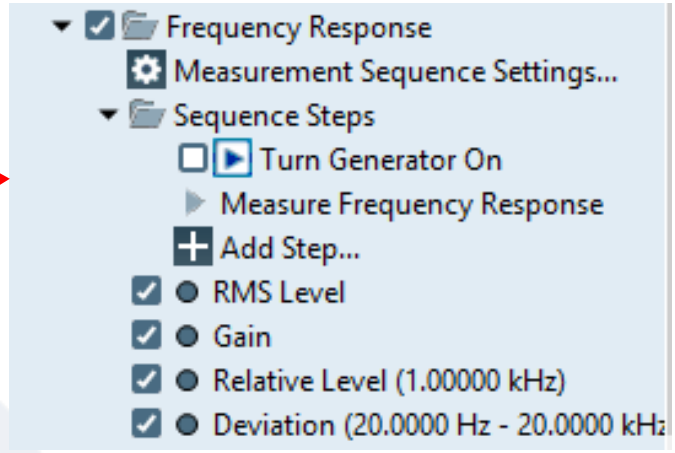
- **Sequence Mode: All API commands start with APx.**
- **Bench Mode: All API commands start with APx.Benchmode**

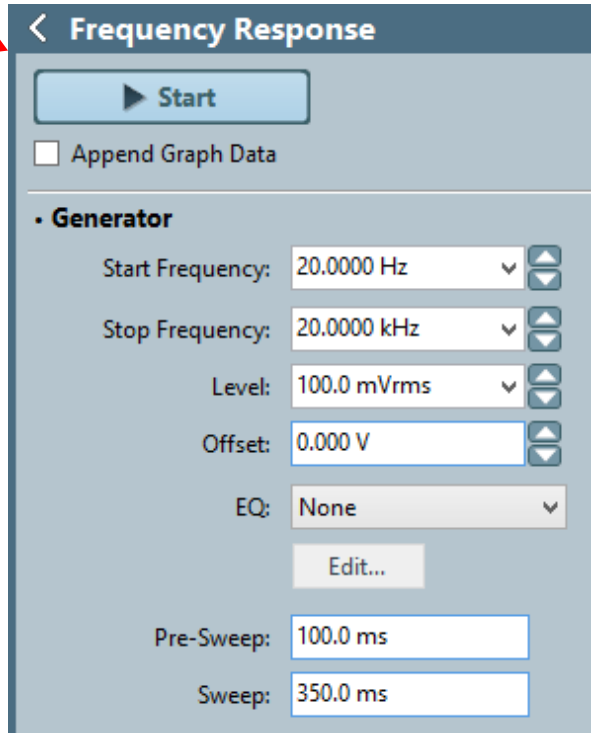


Sequence Mode API…

- **The Sequence Mode API is generally split into three parts**
  - Navigator
  - Active measurement
  - Other…

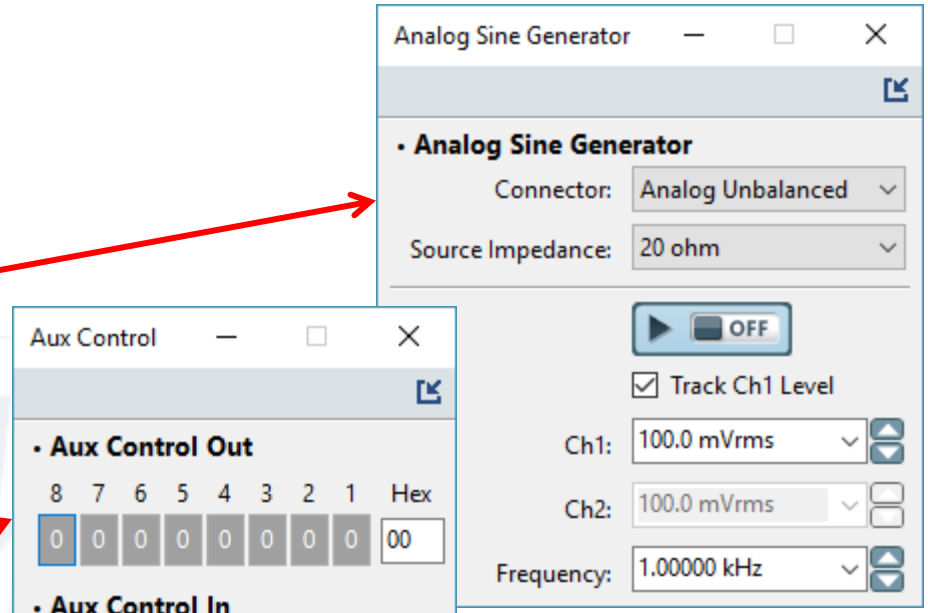
- **In the APx500 application, you can select measurements, results, and sequence steps from the Navigator, but you can only change the measurement settings by making the measurement "active".**

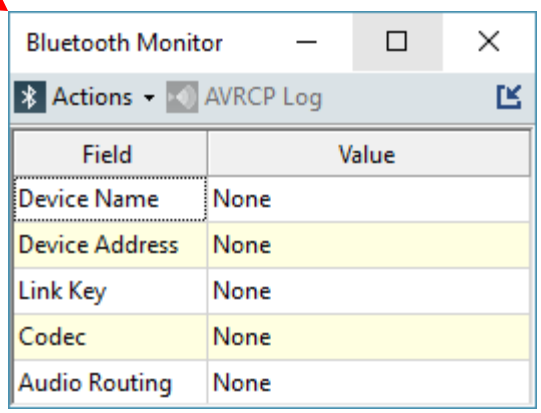




Sequence Mode API…

- ❖ **APx.ActiveMeasurement**
  (More about this on the next slide)
- ➢ **APx.AddMeasurement**
- ▪ **APx.AnalogOutputMonitor**
- ▪ **APx.APx1701**
- ▪ **APx.AttachedProjectItems**
- ▪ **APx.AudibleSignalMonitor**
- ▪ **APx.AuxControlMonitor**
- ▪ **APx.BluetoothMonitor**
- ➢ **APx.Delete…**
  - Measurement
  - Result
  - SignalPath
  - UncheckedItems

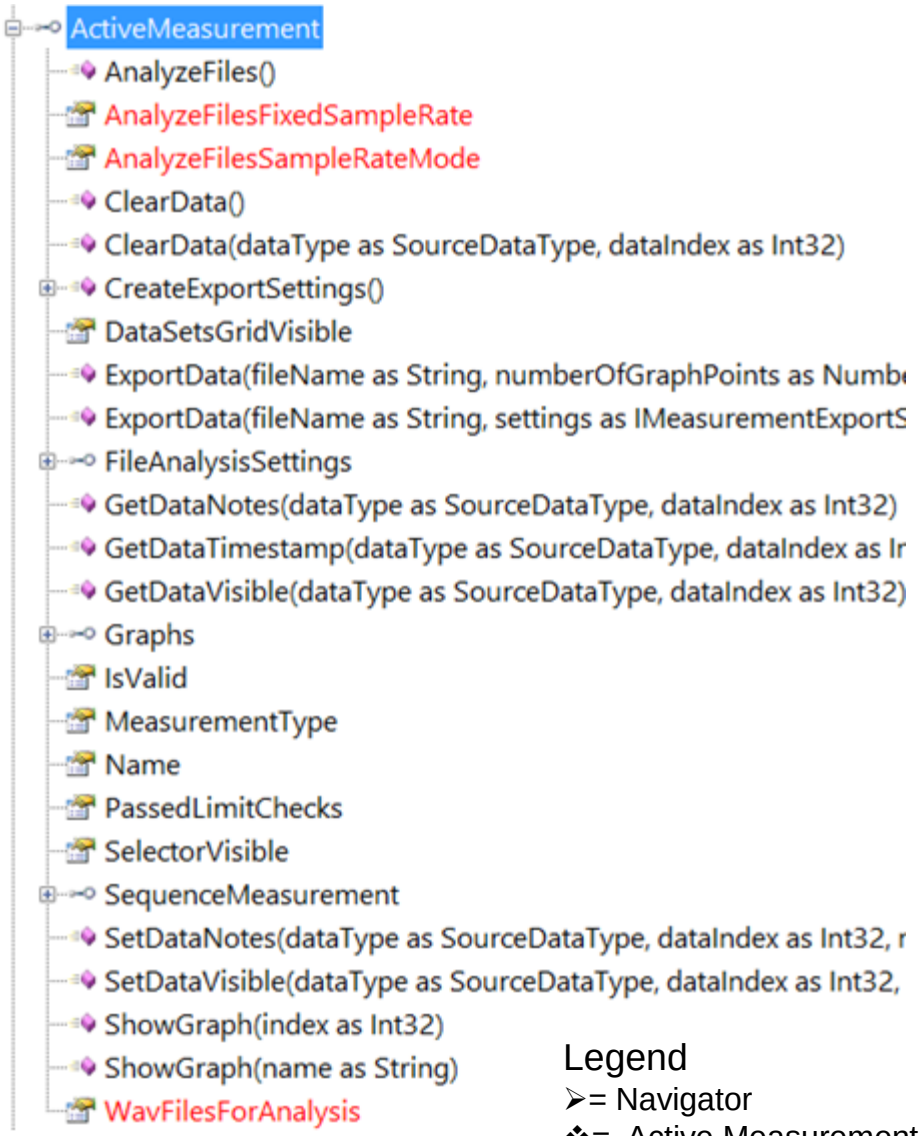




Legend
➢= Navigator
❖= Active Measurement
▪ = Other

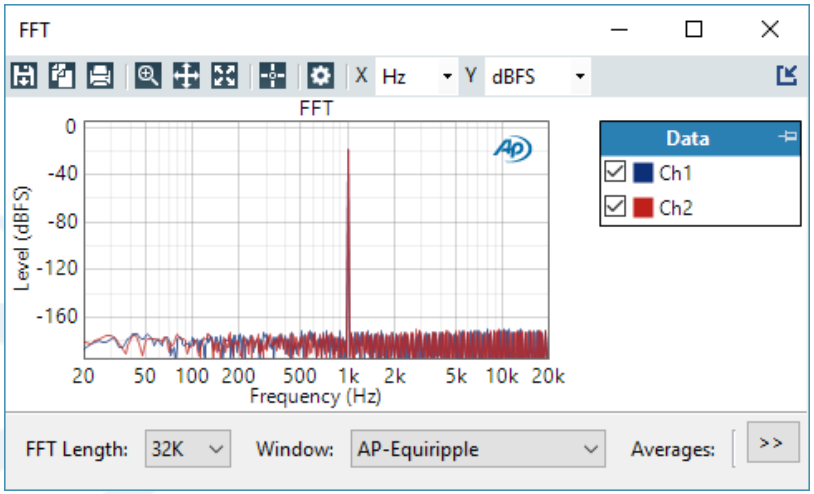
❖ APx.ActiveMeasurement

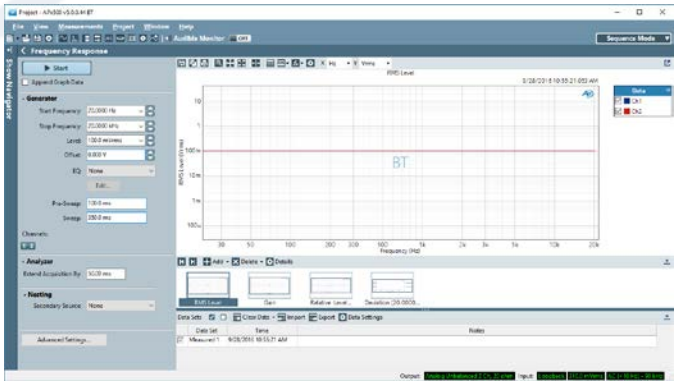
- **Common subset of controls for all measurements.**
- **Use APx.ShowMeasurement to select an active measurement.**
- **Can Run a measurement including Sub Steps using APx.ActiveMeasurement.SequenceMeasurement.Run command.**
- **Can access results using APx.ActiveMeasurement.SequenceMeasurement.SequenceResults commands.**



Sequence Mode API…

- **APx.Exit**
- **APx.FftSpectrumSignalMonitor**
- **APx.Height / Width / Top / Left**
- **APx.Is…**
  - DemoMode
  - ProcessingSuspended
  - ProjectLocked
  - ProjectModified
- **APx.Maximize / Minimize / Restore**
- ➢ **APx.NavigatorVisible**
- **APx.OpenProject / SaveProject**
- **APx.OperatingMode**

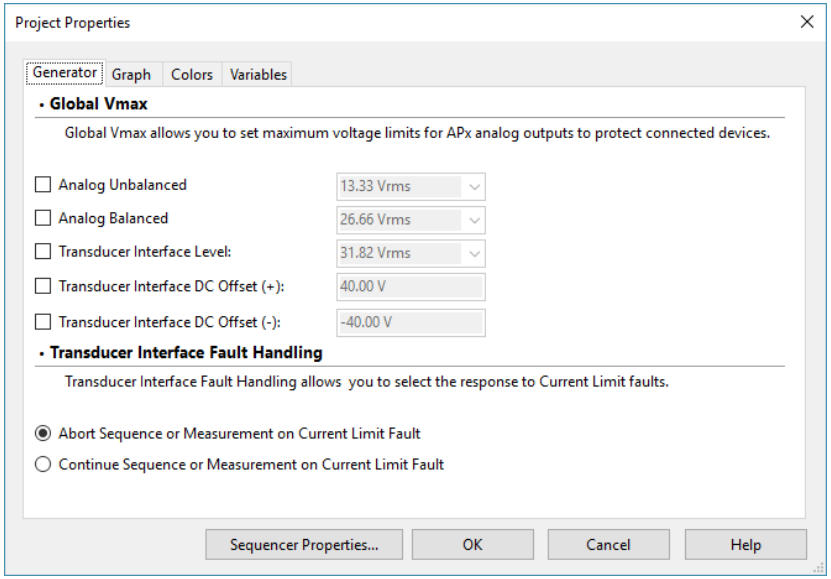




Legend
➢= Navigator
❖= Active Measurement
▪ = Other

Sequence Mode API…

- **APx.ProjectSettings**



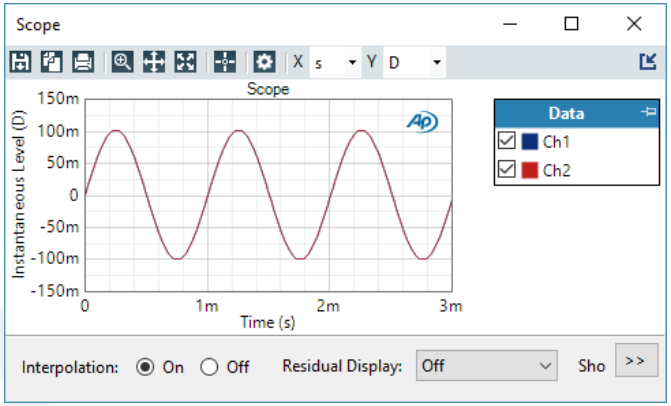
- ProjectSettings
  - ExportTraceCycleStyles(fileName as String)
  - GetLineStyle(cycleIndex as Int32, styleIndex as Int32)
  - GetLineWidth(cycleIndex as Int32, styleIndex as Int32)
  - GetTraceColor(cycleIndex as Int32, styleIndex as Int32)
  - GlobalVmax
  - ImportTraceCycleStyles(fileName as String)
  - ResetStyles(cycleIndex as Int32)
  - SaveGraphData
  - SetLineStyle(cycleIndex as Int32, styleIndex as Int32, lineStyle as DashStyle)
  - SetLineWidth(cycleIndex as Int32, styleIndex as Int32, lineWidth as Int32)
  - SetTraceColor(cycleIndex as Int32, styleIndex as Int32, color as Color)
  - ShowMeasuredTime
  - TraceStyleCycleCount
  - TransducerCurrentLimitFault

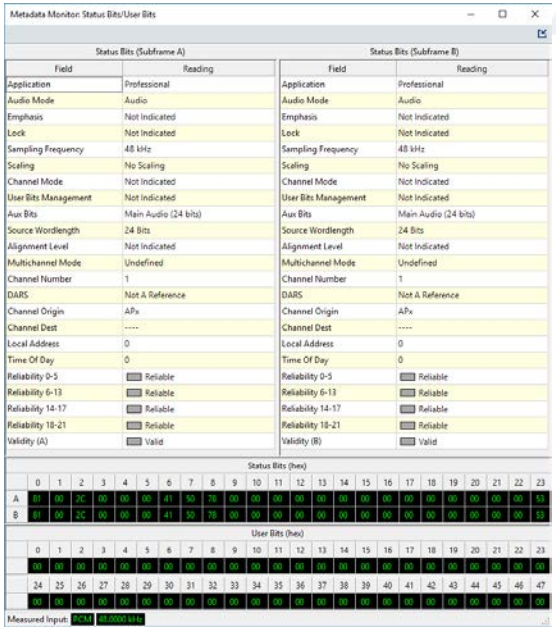
Legend
➢= Navigator
❖= Active Measurement
▪ = Other

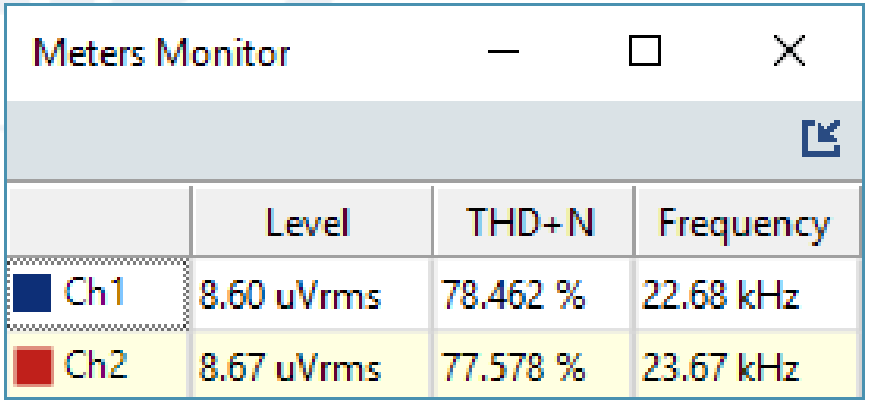
Sequence Mode API…

- APx.ResumeProcessing / SuspendProcessing
- APx.ScopeSignalMonitor
- APx.Sequence

  (More about this on the next slide)
- APx.ShowMeasurement
- APx.SignalMeters
- APx.StatusBitsMonitor



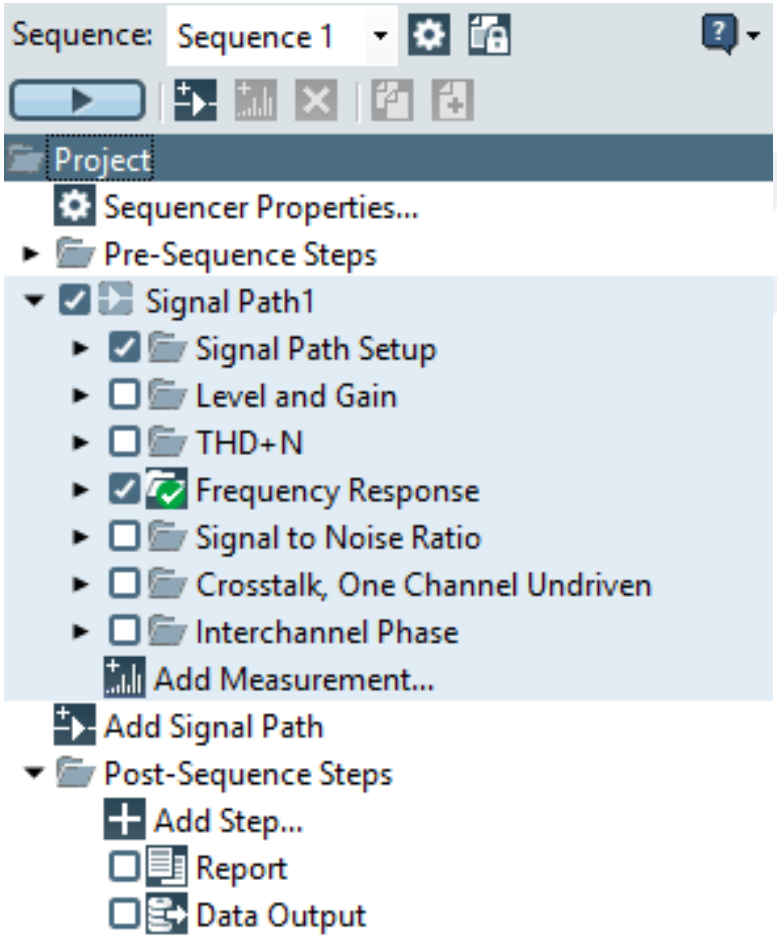




Legend
➢= Navigator
❖= Active Measurement
▪ = Other

➢ APx.Sequence…

Gets the sequence of measurements displayed in the Measurement Navigator. The sequence allows automatic execution of a series of measurements and collects all of the results.

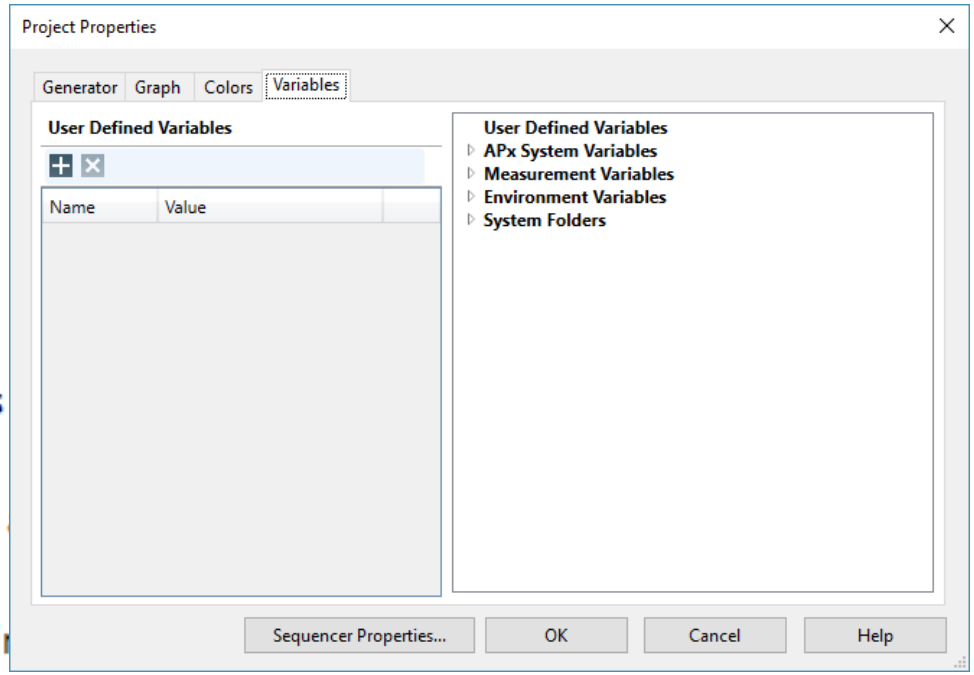


- **Access to all Navigator settings**
- **Access to all Sequence results for measurements that have been run either as a part of a Sequence or individually.**

Legend
➢= Navigator
❖= Active Measurement
▪ = Other

Sequence Mode API…

- **APx.Variables**
- **APx.Version**
- **APx.Visible**



- Variables
  - DeleteUserDefinedVariable(varName as
  - ExpandVariableString(input as String)
  - GetAPxMeasurementVariable(varName
  - GetAPxMeasurementVariables()
  - GetAPxSystemVariable(varName as Stri
  - GetAPxSystemVariables()
  - GetEnvironmentVariable(varName as String)
  - GetEnvironmentVariables()
  - GetSystemFolder(folder as SpecialFolder)
  - GetUserDefinedVariable(varName as String)
  - GetUserDefinedVariables()
  - SetUserDefinedVariable(varName as String, value as String)

Legend
➢= Navigator
❖= Active Measurement
▪ = Other

Sequence Mode Measurement API…

- **Measurement Objects**
  - APx.AcousticResponse (More about this measurement on the next slide)
  - APx.BandpassFrequencySweep
  - APx.BandpassLevel
  - APx.BandpassLevelSweep
  - APx.Cmrr
  - APx.CmrrIec
  - APx.CompareEncodedBitstream
  - APx.ContinuousSweep
  - APx.CrosstalkCustom
  - APx.CrosstalkFrequencySweepCustom
  - APx.CrosstalkFrequencySweepOneChannelDriven
  - APx.CrosstalkFrequencySweepOneChannelUndriven
  - APx.CrosstalkOneChannelDriven
  - APx.CrosstalkOneChannelUndriven
  - APx.DCLevel
  - …

Legend
➢= Navigator
❖= Active Measurement
▪ = Other

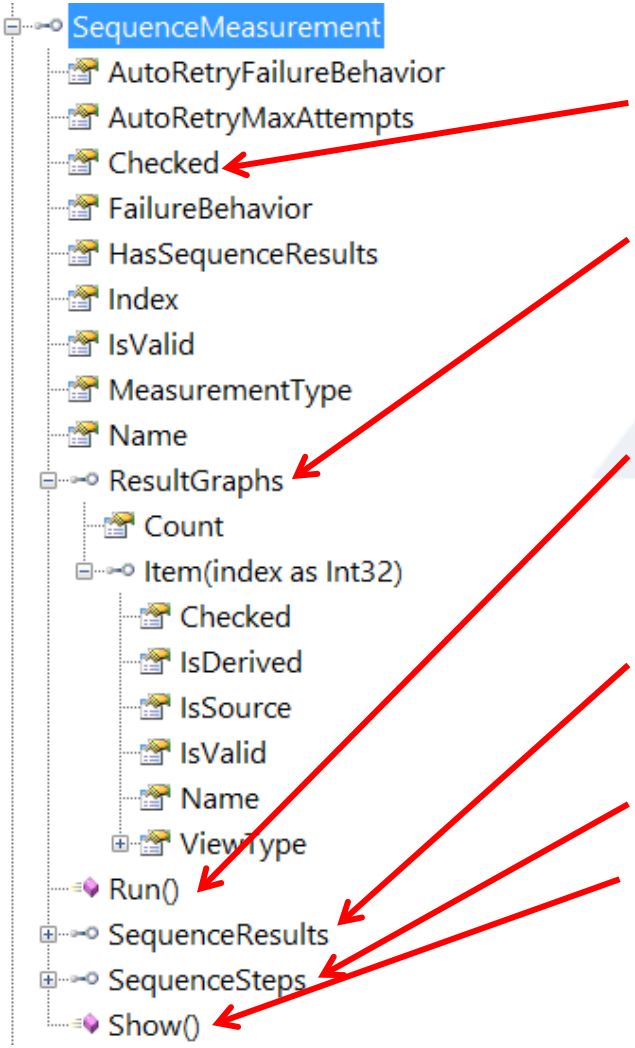
Sequence Mode Measurement API…

APx.AcousticResponse
Access to individual measurement settings and results.

- AcousticResponse
  - AcquiredWaveform
  - AdditionalAcquisitionTime
  - AnalogInputRanges
  - AnalyzeFiles()
  - AnalyzeFilesFixedSampleRate
  - AnalyzeFilesSampleRateMode
  - Append
  - Averages
  - ClearData()
  - ClearData(dataType as SourceDataType, dataIndex as Int32)
  - CreateExportSettings()
  - DataSetsGridVisible
  - Delay
  - Deviation
  - DistortionProductLevel
  - DistortionProductRatio
  - EnergyTimeCurve
  - ExportData(fileName as String, numberOfGraphPoints as NumberOfGr
  - ExportData(fileName as String, settings as IMeasurementExportSetting
  - FileAnalysisSettings
  - Generator
  - GetDataNotes(dataType as SourceDataType, dataIndex as Int32)
  - GetDataTimestamp(dataType as SourceDataType, dataIndex as Int32)
  - GetDataVisible(dataType as SourceDataType, dataIndex as Int32)
  - Graphs
  - GroupDelay
  - ImpulseResponse
  - IsValid
  - Level
  - LevelAndDistortion
  - MeasurementType
  - Name
  - NestedSweep
  - PassedLimitChecks
  - Phase
  - RelativeLevel
  - RubAndBuzz
  - RubAndBuzzCrestFactor
  - RubAndBuzzPeakRatio
  - SelectorVisible
  - SequenceMeasurement
  - SetDataNotes(dataType as SourceDataType, datal
  - SetDataVisible(dataType as SourceDataType, data
  - ShowGraph(index as Int32)
  - ShowGraph(name as String)
  - Start()
  - ThdLevel
  - ThdRatio
  - TimeWindowAutoStart
  - WavFilesForAnalysis

➢ Sequence Mode Measurement Navigator API

APx.AcousticResponse.SequenceMeasurement
Gets the sequencer settings and results for this measurement.

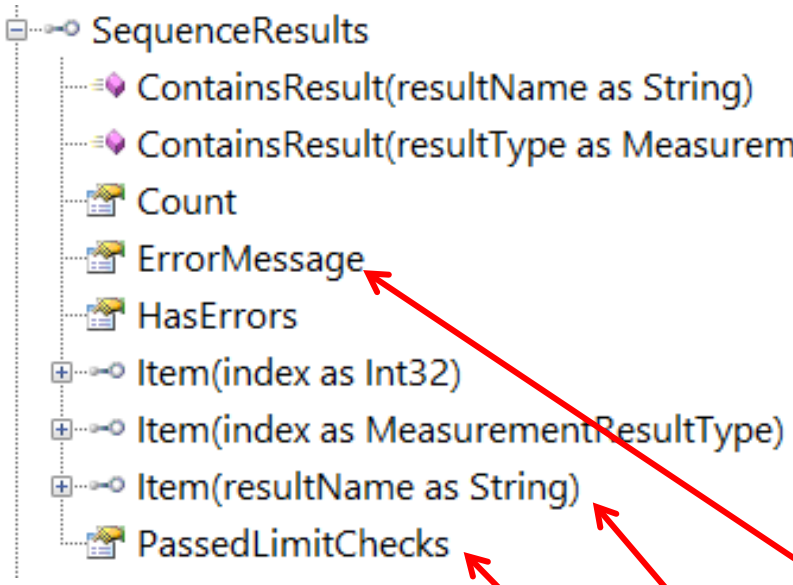


- **Access to sequence properties like Checked**
- **Access to Result Graphs but not their settings like Checked**
- **Run just this measurement to get settled readings and results with the Run command**
- **Sequence results (More about this on the next slide)**
- **Access to sequence steps**
- **Make the measurement active with the Show command**

Legend
➢= Navigator
❖= Active Measurement
▪ = Other

Navigator API – Sequence Results

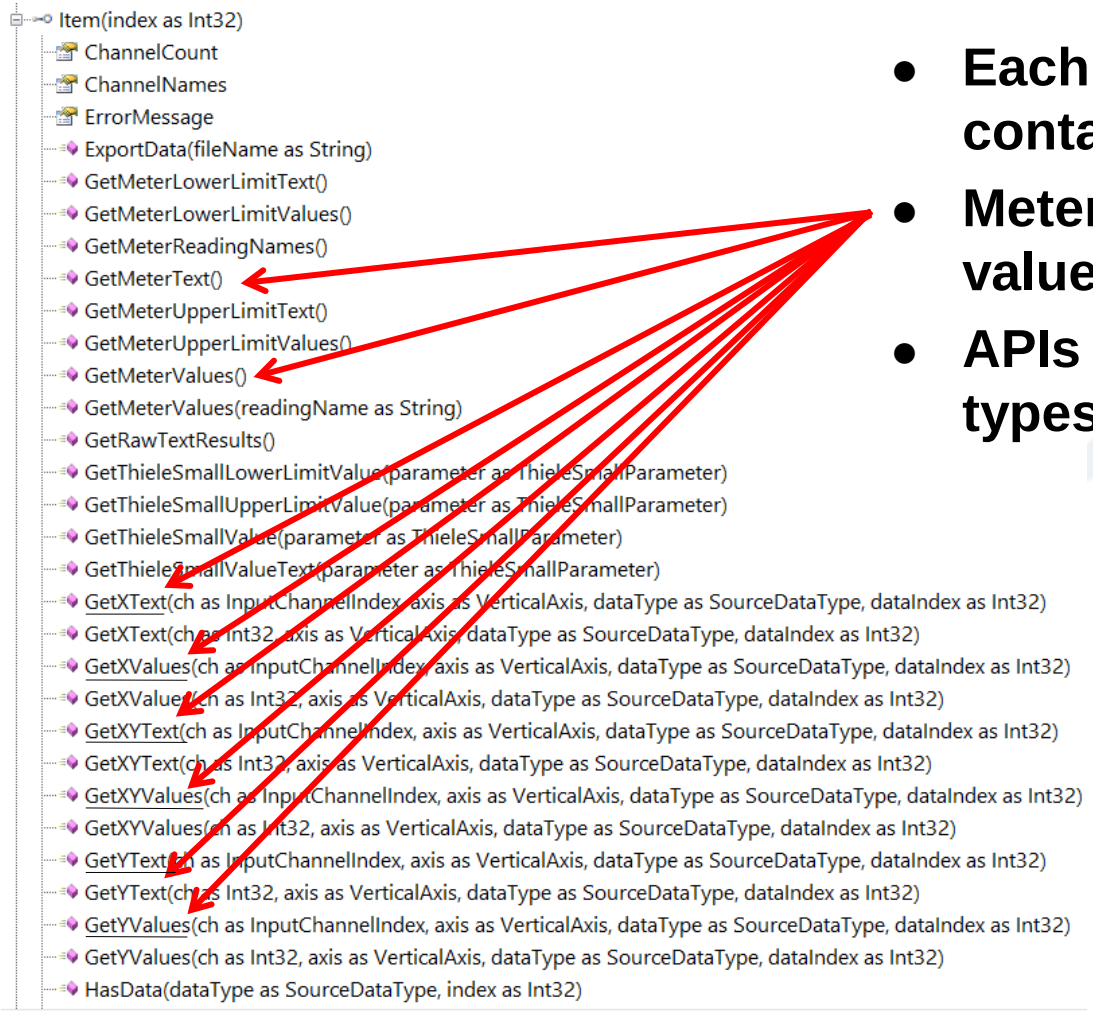
APx.AcousticResponse.SequenceMeasurement.SequenceResults



- **Sequence results object is a collection containing 0 or more results.**
- **One entry for each checked graph**
- **If the measurement failed due to an error, the ErrorMessage property indicates the error**
- **Item functions to get individual results**
- **PassedLimitChecks indicates if all results were successful**

Navigator API – Extracting result data

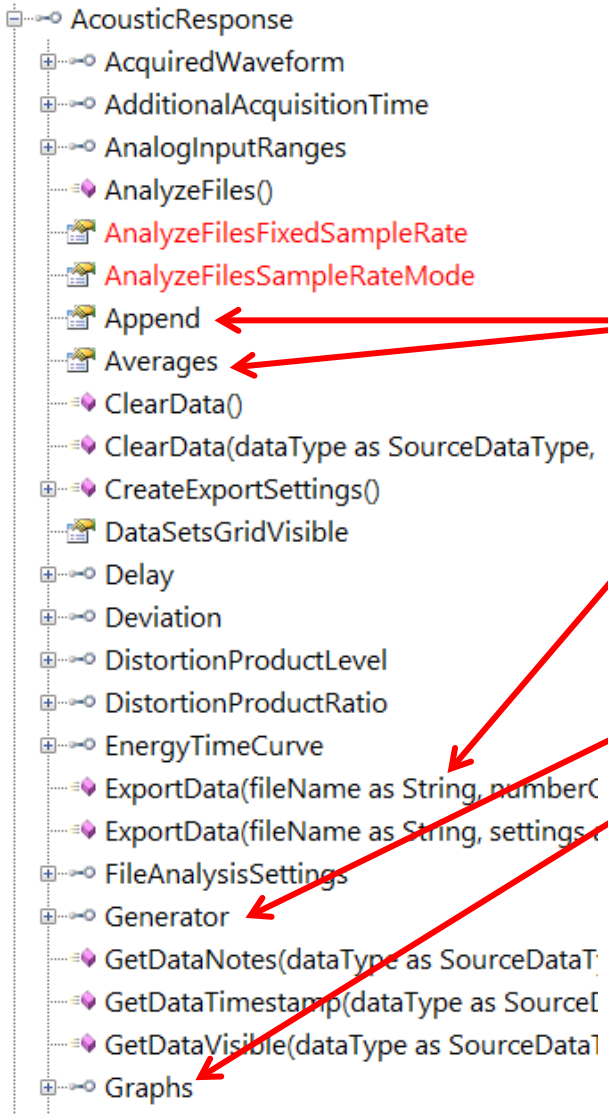
APx.AcousticResponse.SequenceMeasurement.SequenceResults.Item
or
APx.Sequence(index)(index).SequenceResults



- **Each sequence result can contain one type of data**
- **Meter readings, XY data, text values, etc.**
- **APIs available to extract all types of data from each result**

Measurement API Settings

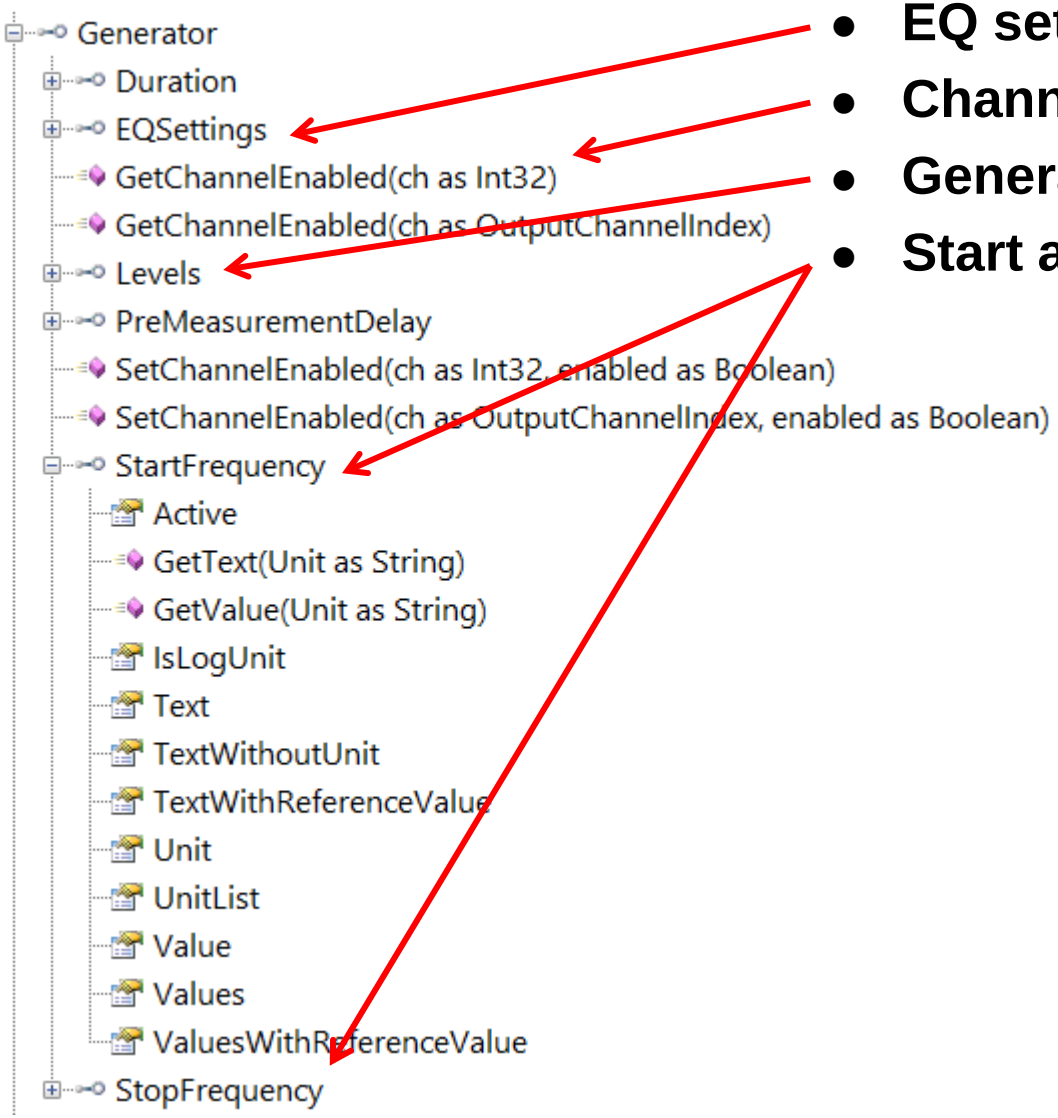
APx.AcousticResponse



- **Access to measurement settings**
- **Ability to import or export data if applicable**
- **Access to generator settings**
- **Access to graphs and their appearance and data settings**
- **Access to the Sequence API with the SequenceMeasurement property (not shown in image)**

Measurement API – Generator

APx.AcousticResponse.Generator

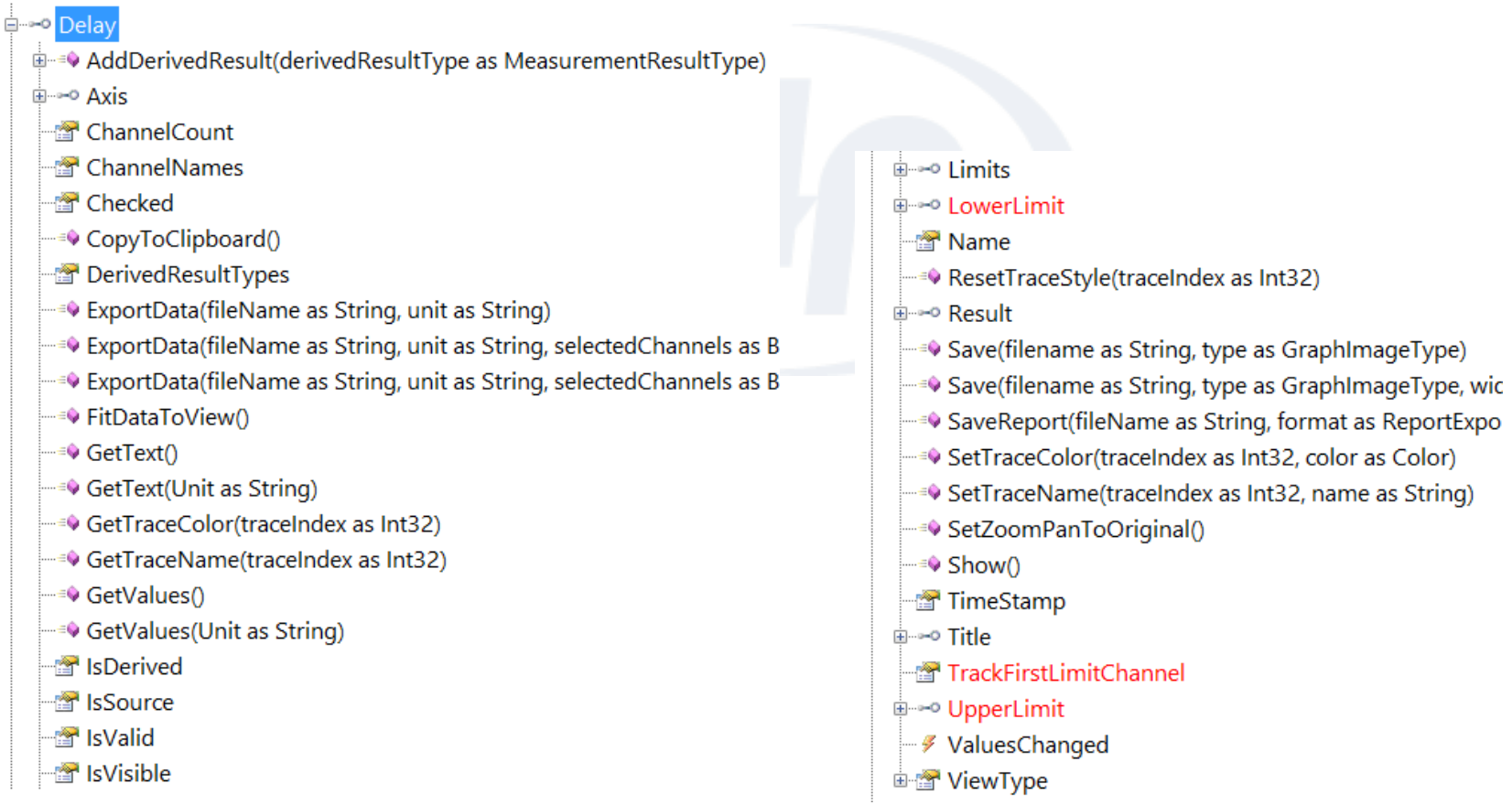


- **EQ setting if applicable**
- **Channel settings**
- **Generator levels**
- **Start and Stop Frequency settings**

Measurement API – Individual Results

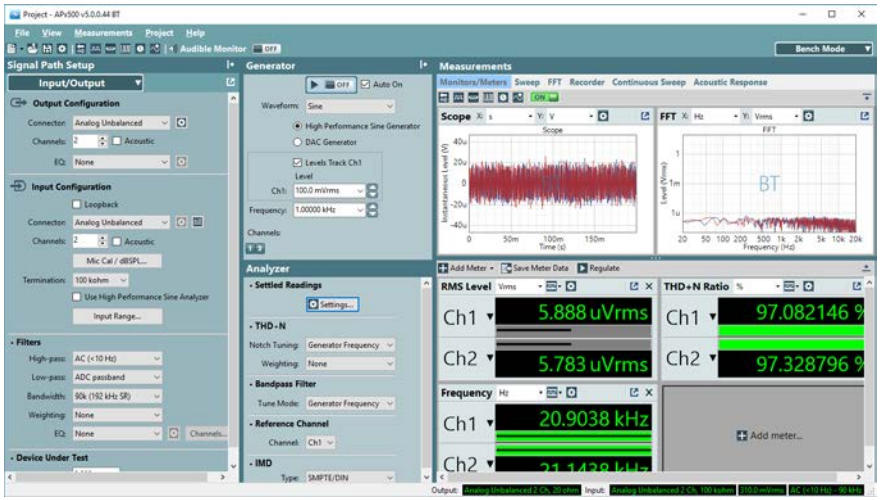
APx.AcousticResponse.Delay
Access to all result Settings and Data

Access to results when using the APx.AcousticResponse.Start or APx.AcousticResponse.SequenceMeasurement.Run() commands to run the measurement



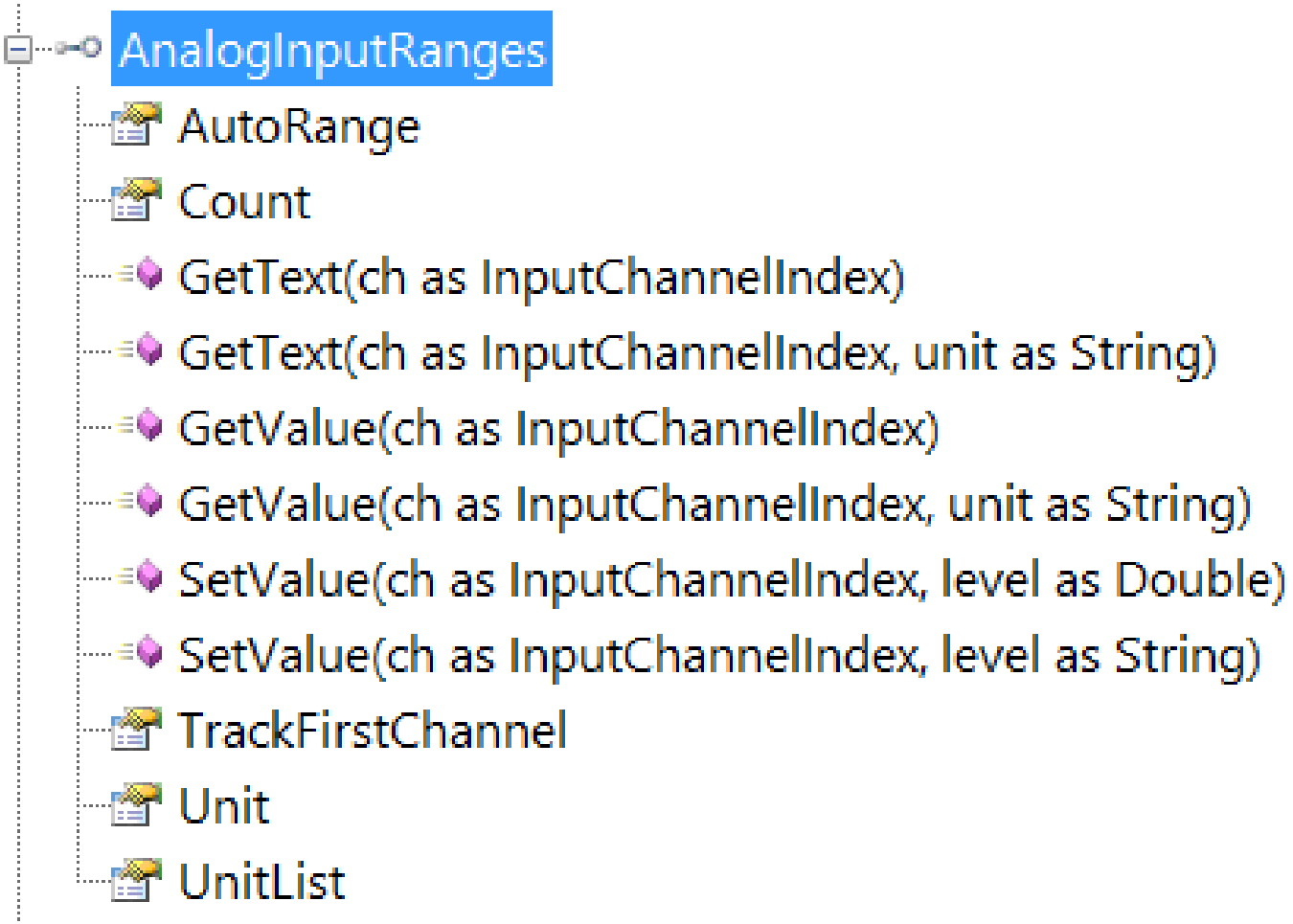
Bench Mode API...

- BenchMode
  - ActiveMeasurement
  - AnalogInputRanges
  - Analyzer
  - Generator
  - GeneratorAnalyzerSettingsVisible
  - GetSettledMeterReadings
  - Measurements
  - Meters
  - Monitors
  - Setup
  - SignalPathSettingsVisible



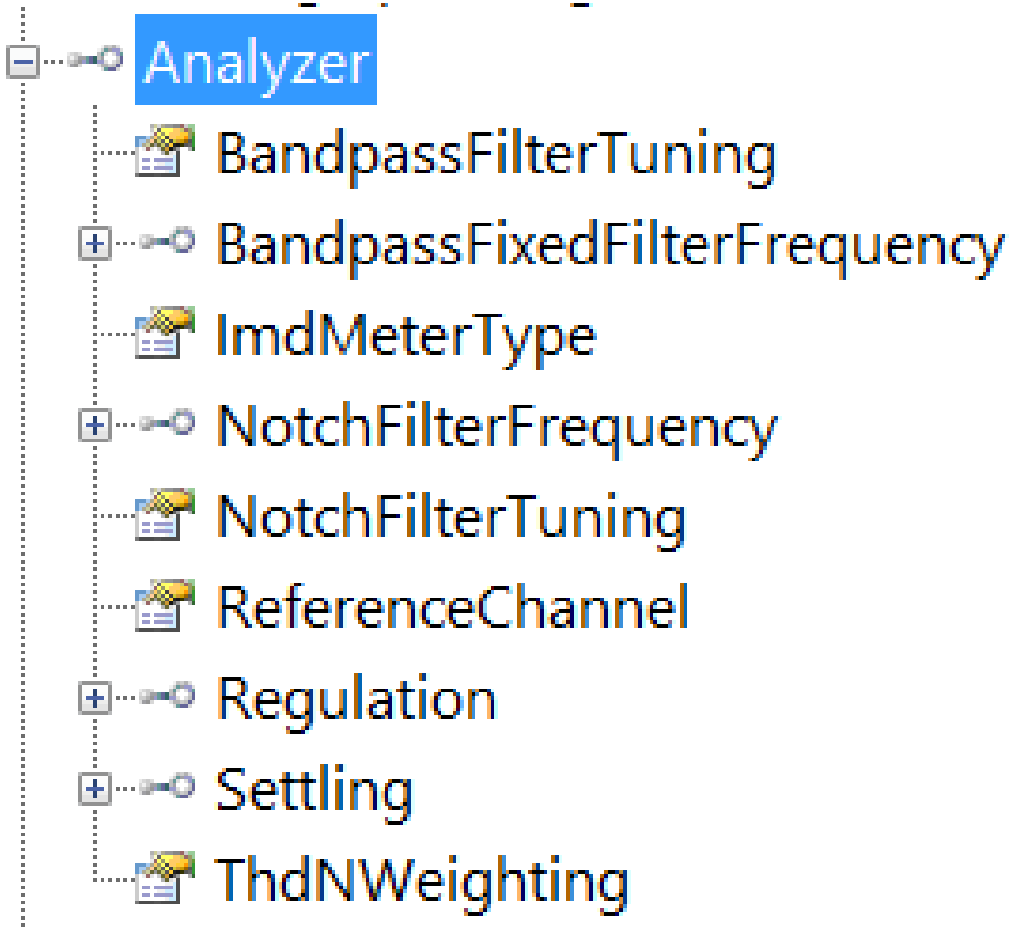
Bench Mode API…

APx.BenchMode.AnalogInputRanges



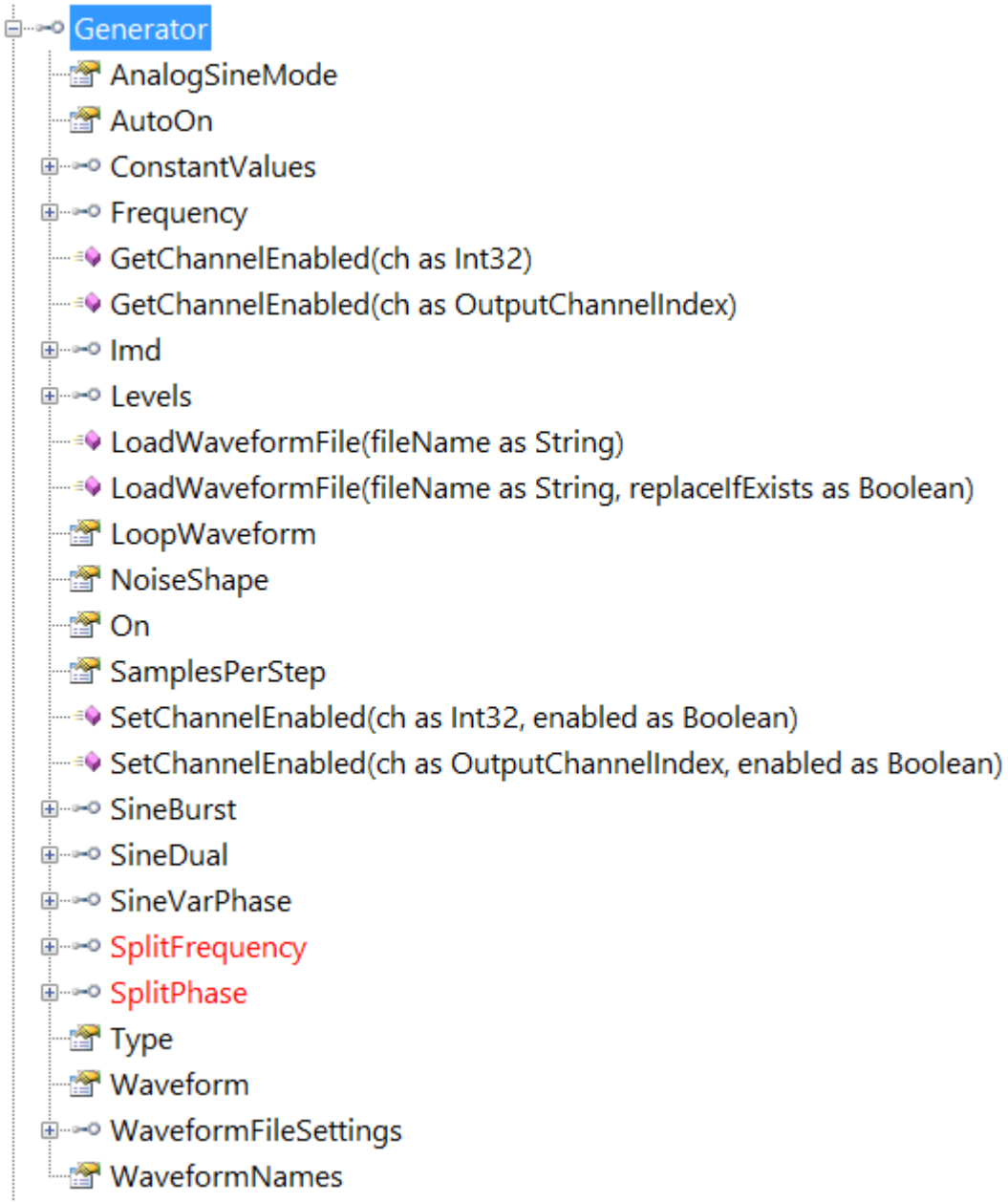
Bench Mode API…

APx.BenchMode.Analyzer



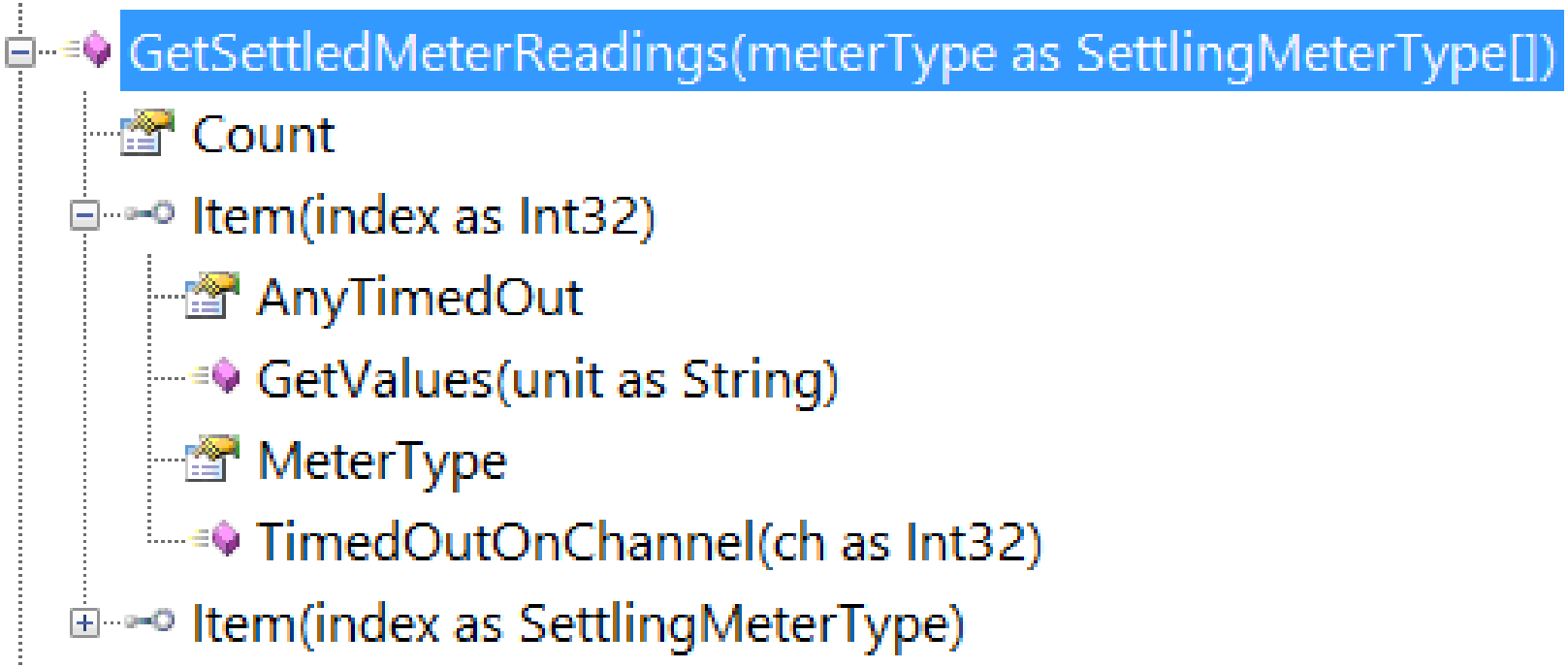
Bench Mode API…

APx.BenchMode.Generator



Bench Mode API…

APx.BenchMode.GetSettledMeterReadings(meterType)



API Programming

Determining Object Types – Visual Studio

- **Property**

```
Dim measurement As IAcousticResponseMeasurement
measurement = APx.AcousticResponse
```

ReadOnly Property APx500.AcousticResponse As IAcousticResponseMeasurement
Gets the currently active Acoustic Response Measurement.

```
Dim generator As IContinuousSweepGenerator
generator = measurement.Generator
generator.Duration.Value = 2.0
```

Property IDoubleSetting.Value As Double
Gets or sets the value, in the unit specified by the Unit property.

- **Method/Function**

```
measurement.st
```

Start
TimeWindowAutoStart
ToString

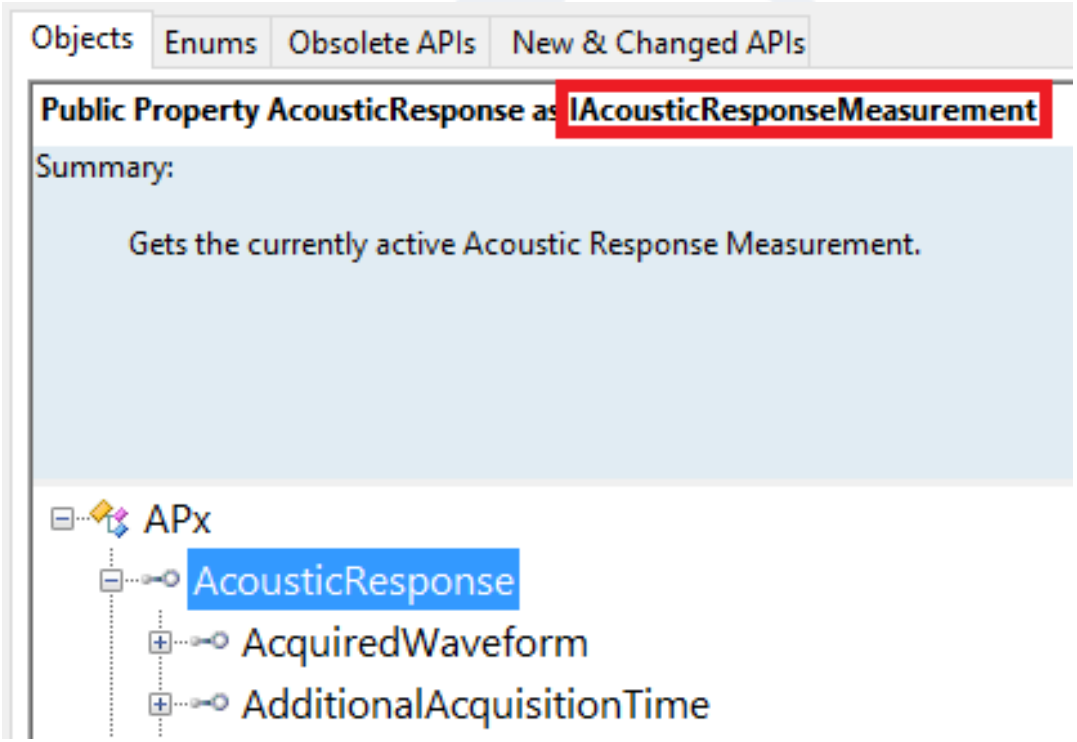
Function IBatchMeasurement.Start() As Boolean
The measurement runs synchronously and does not return control to the user until the measurement has completed.

- **Microsoft Visual Studio uses IntelliSense to help**
- **Hover over Objects or type the path to an API and help automatically appears**

Determining Object Types – API Browser

- **Like Visual Studio, the API Browser shows the return type of every property and method**

- **Navigate to a property or method and observe the highlighted area in the API Browser window**

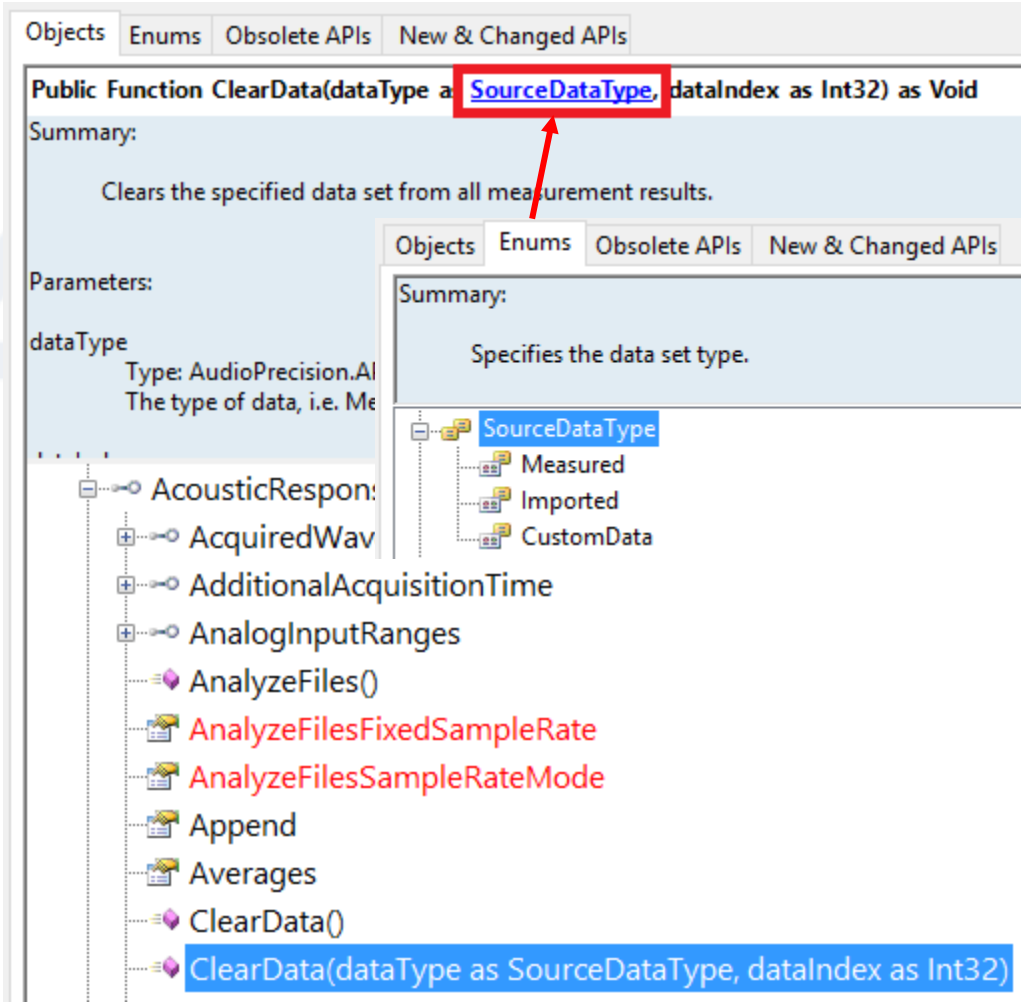


API Browser – Enums

- **Enumerated Constants**
  - Provided for version compatibility

- **Working With Enums**
  - Identifying
    - Link shown in help for API …
  - Selection
    - Select link then appropriate Enums are displayed in the Enums Tab

      Note: Some Enum lists are very long. Select an Enum with a name that matches what you are trying to control.



API Browser
Obsolete APIs Tab

- **Filter setting**
  - Objects, Enums, All
- **Version**
  - The version the API Method, Property, or Enum was removed in.
- **Object**
  - The base object containing the API Method, Property, or Enum
- **Old API**
  - The API Method, Property, or Enum
- **Suggestion**
  - Help text intended to provide information needed to update existing code.

API Browser
New & Changed APIs Tab

- **Filter setting**
  - Objects, Enums, All
- **Object**
  - The base object containing the API Method, Property, or Enum
- **API**
  - The API Method, Property, or Enum
- **Description**
  - API Parameters
- **Example**
  - API usage

Sample Code Review

- **What languages are officially supported by Audio Precision**
  - VB.net
  - C#
  - C++
  - Python
  - Matlab
  - LabVIEW

- **The following code review is done using Visual Basic**

VB.net

- **Advantages:**
  - Dot Net Compatible
  - Easy User Interface development
  - IntelliSense
  - Compiled application
  - FREE Express versions available
  - VS Template provided

- **Disadvantages:**
  - Cost
  - Not ideal for some programming tasks

- **Note:**
  - Uses API.dll

- **Advantages:**
  - Dot Net Compatible
  - API native language
  - Easy User Interface development
  - Compiled application
  - IntelliSense
  - FREE Express versions available
  - VS Template provided

- **Disadvantages:**
  - Cost

- **Note:**
  - Uses API.dll

C++

- **Advantages:**
  - Dot Net Compatible
  - VS Template provided
- **Disadvantages:**
  - Difficult User Interface development
  - Cost
- **Note:**
  - Uses API.dll

Python…

- **Advantages:**
  - FREE

- **Disadvantages:**
  - Can’t control APx500 because it is NOT Dot Net Compatible

IronPython…

- **Advantages:**
  - Dot Net Compatible
  - Code is compact and more readable
  - Conforms to standards including zero-based indexing and usage of square brackets
  - FREE

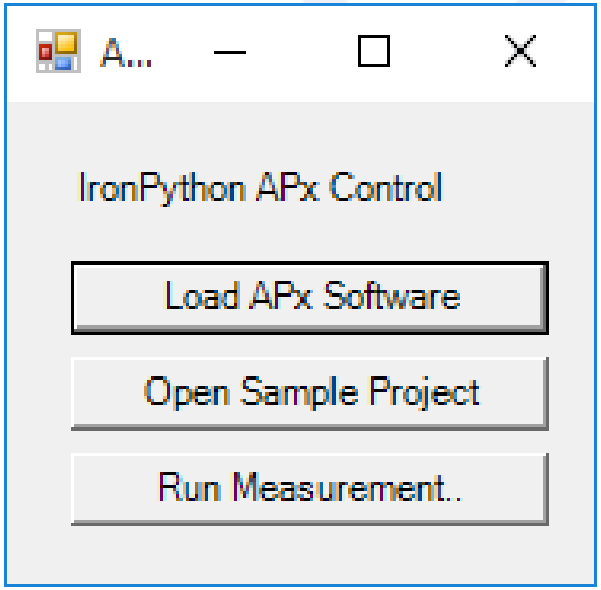
- **Disadvantages:**
  - Not compliable (less security)

- **Note:**
  - Uses API.dll and API2.dll

IronPython…

- **Using the Console/Interpreter**
  - Create a file association to the IronPython interpreter (IPY.exe) for .py files
  - Double click on .py file to run

- **Sample Code**
  - Knowledge Base article with sample code
    - http://www.ap.com/technical-library/controlling-apx500-with-python



IronPython

- **Running Python code from Visual Studio**
  - Download Tools for Visual Studio from http://microsoft.github.io/PTVS
    - Select the PTVS 2.2 Downloads selection
    - Select the VS 2013 or 2015 version

- **PTVS provides**
  - Editing with IntelliSense
  - Interactive debugging
  - Profiling
  - Cross-platform and cross-language debugging support,

Matlab

- **Advantages:**
  - Dot Net Compatible
  - Built-in graphics
  - Large library of tools

- **Disadvantages:**
  - Cost
  - Object-oriented programming scheme is complex and confusing.
  - Not compliable (less security)

- **Note:**
  - Uses API.dll and API2.dll

Visual Basic Sample Code Review…

- **User Interface Review**
  - Launch APx500 (UI Button 1)
  - Load and Run APx project (UI Button 2)
    - Get results from Level & Gain measurement
      - RMS Level
      - Gain
      - Peak Level
    - Get results from Stepped Frequency Sweep measurement
      - Level vs Frequency
      - RMS Level Smoothed
      - Deviation
      - Get results using For Each statement
    - Get results from Signal Analyzer measurement
      - FFT Spectrum
      - Scope

Visual Basic Sample Code Review

- Configure and Run Bench Mode measurements (UI Button 3)
  - Configure Signal Path Setup Input/Output
  - Configure Signal Path Setup Filters
  - Configure Generator
  - Configure Analyzer
  - Monitors/Meters
    - Delete current Monitors/Meters
    - Add desired Monitors/Meters
    - Configure settling For desired Monitors/Meters
    - Get Settled readings from meters all at the same time.
    - Get Settled readings from meters one at a time.
  - Configure Sweep measurement
    - Set Right axis data To THD+N Ratio
    - Run the measurement and wait for Nested Sweep to finish
    - Get X values from sweep 1
    - Get Y values from sweep 1 & 2
  - Configure FFT measurement
    - Get all FFT data
    - Get all Scope data

LabVIEW and DAQ Driver Review

END