

AFGROW Workshop 2019

# AFGROW and Spectrum Manager COM API

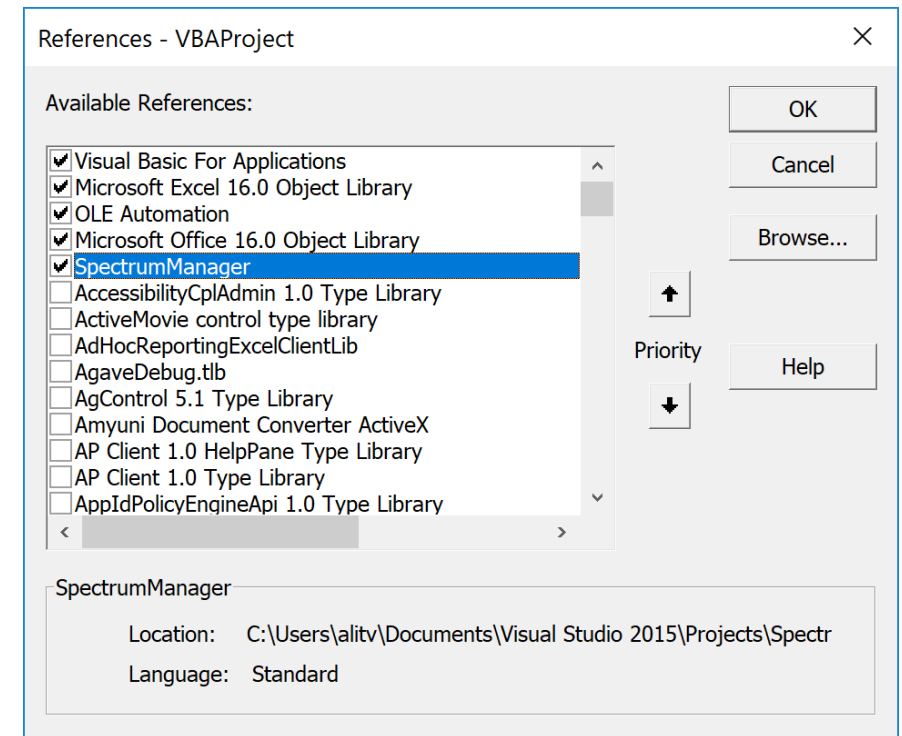
Alex Litvinov, James Lambert  
LexTech, Inc .

# Outline

- Spectrum Manager COM Interface Overview
- Spectrum Manager *Application* Object
- Spectrum Manager *SpectrumStats* Object
- Spectrum Manager *SubSpectrumBlock* and *SubSpectrumBlockCollection* Objects
- Examples of Using AFGROW and Spectrum Manager interface together
- Questions

# Spectrum Manager COM Interface overview

- SpectrumManager version 1.2 implements an OLE server application as an in-process server — a DLL running in the process space of the container application. It is different from AFGROW, which is a local server — an EXE running in its own process space.
- In SpectrumManager COM communication between container and server is simplified because communication between the two can take the form of normal function calls. That allows to save time and memory on transfer large amounts of data, that are necessary to create or edit spectrum data.
- The *Application* object that provides access to all SpectrumManager functionality
- Three supplementary objects (*SpectrumStats*, *SubSpectrumBlock* and *SubSpectrumBlockCollection*) are used to optimize data transfer



# Spectrum Manager *Application* Object

- Represents the entire *SpectrumManager* application
- Contains functions that return top-level objects such as *SpectrumStats*, *SubSpectrumBlockCollection*, and so on.
- Still a work in process
- No events or properties
- Composed of methods that can be logically divided into four groups
  1. Data Serialization/Deserialization
  2. Spectrum creation
  3. Reporting
  4. Spectrum Manipulation

## Members of 'Application'

- AddLevelToSubSpectrumByIndex
- AddLevelToSubSpectrumByName
- AddSubSpectrumToSequence
- CheckSequence
- ClearSequence
- Clip
- CloseSpectrum
- CreateSpectrum
- CreateSubSpectrum
- DeleteSubSpectrum
- GetAllSubSpectra
- GetStats
- GetSubSpectrumStats
- ImportSubSpectrum
- IsObfuscated
- OpenFile
- RemoveLevelFromSubSpectrum
- RemoveSubSpectrumFromSequence
- ReverseSpectrum
- SaveSpectrumAsSP3
- SaveSpectrumAsSPX
- SaveSubSpectrumAsSUX
- Truncate

# Application Object composition

## Data Serialization/Deserialization

- Sub **ImportSubSpectrum**(path As String, generateName As Boolean)
- Sub **OpenFile**(path As String)
- Sub **SaveSpectrumAsSP3**(path As String)
- Sub **SaveSpectrumAsSPX**(path As String)
- Sub **SaveSubSpectrumAsSUX**(subSpectrumName As String, path As String)

## Spectrum creation

- Sub **AddLevelToSubSpectrumByIndex**(subSpectrumIndex As Long, min As Double, max As Double, cycles As Long, [minBending As Double = -1.#IND], [maxBending As Double = -1.#IND], [minBearing As Double = -1.#IND], [maxBearing As Double = -1.#IND])
- Sub **AddLevelToSubSpectrumByName**(subSpectrumName As String, min As Double, max As Double, cycles As Long, [minBending As Double = -1.#IND], [maxBending As Double = -1.#IND], [minBearing As Double = -1.#IND], [maxBearing As Double = -1.#IND])
- Sub **AddSubSpectrumToSequence**(subSpectrumName As String)
- Sub **CloseSpectrum**()
- Sub **CreateSpectrum**(title As String, [subSpectrumLabel As String = "flight"], [description As String], [multiChannel As Boolean = False], [damageTag As Boolean = False], [timeDependant As Boolean = False], [enviormentTags As Boolean = False])
- Sub **CreateSubSpectrum**(name As String, [description As String])
- Sub **DeleteSubSpectrum**(subSpectrumName As String)

# *Application* Object composition continued

## Reporting

- Function **GetAllSubSpectra()** As SubSpectrumBlockCollection
- Function **GetStats()** As SpectrumStats
- Function **GetSubSpectrumStats**(subSpectrumName As String) As SpectrumStats

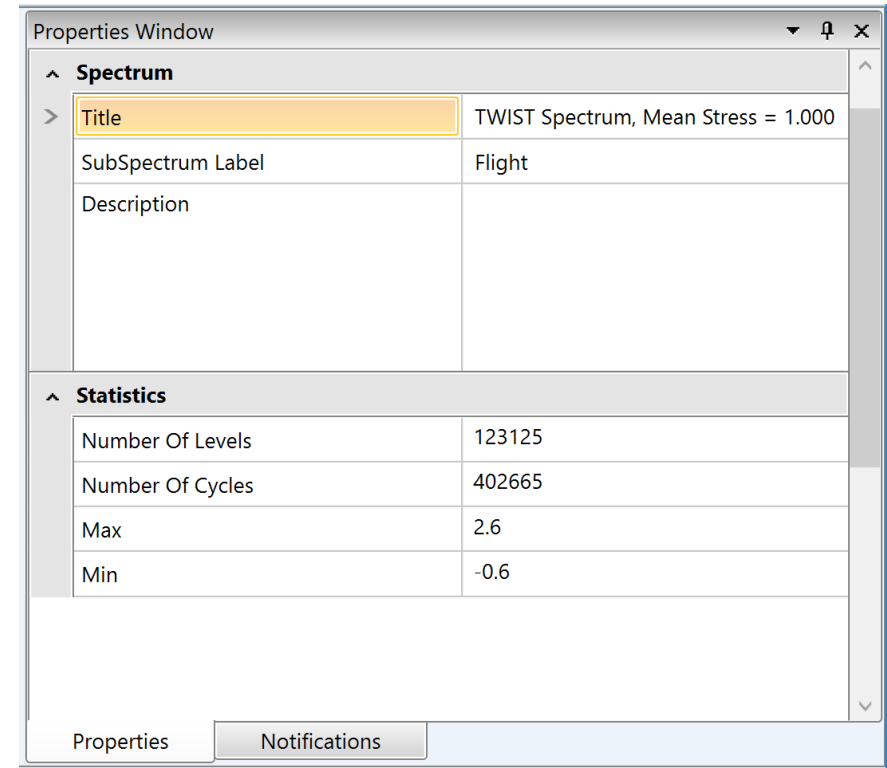
## Spectrum Manipulation

- Function **CheckSequence()** As String
- Sub **ClearSequence()**
- Sub **Clip**(upperBound As Double, [subSpectrumName As String])
- Sub **RemoveLevelFromSubSpectrum**(subSpectrumName As String, index As Long)
- Sub **RemoveSubSpectrumFromSequence**(subSpectrumName As String)
- Sub **ReverseSpectrum()**
- Sub **Truncate**(delta As Double, [subSpectrumName As String])

# *SpectrumStats* Object

Object that provides access to the spectrum/sub-spectrum statistics. It reports the same data that are available in the property Spectrum Manager window.

- double MaxMax
- double MaxMin
- double MinMax
- double MinMin
- int NumberOfLevels
- double Time
- int Cycles
- double MinMinBending
- double MaxMaxBending
- double MinMaxBending
- double MaxMinBending
- double MinMinBearing
- double MaxMaxBearing
- double MinMaxBearing
- double MaxMinBearing



The screenshot shows a 'Properties Window' with two main sections: 'Spectrum' and 'Statistics'. The 'Spectrum' section is expanded, showing a table with the following data:

Spectrum	
Title	TWIST Spectrum, Mean Stress = 1.000
SubSpectrum Label	Flight
Description	

The 'Statistics' section is also expanded, showing a table with the following data:

Statistics	
Number Of Levels	123125
Number Of Cycles	402665
Max	2.6
Min	-0.6

At the bottom of the window, there are two tabs: 'Properties' and 'Notifications'.

# *SubSpectrumBlock* and *SubSpectrumBlockCollection* Objects

*SubSpectrumBlock* is the object that represents a spectrum level. It has the following properties:

- double Min
- double Max
- String Name
- int Cycles
- double MinBending
- double MaxBending
- double MinBearing
- double MaxBearing



# Simple Spectrum Manager COM interface example

```
Sub Button1_Click()
```

```
Dim sm As SpectrumManager.Application  
Dim s As SpectrumManager.SpectrumStats  
Dim m As Double
```

```
Set sm = CreateObject("SpectrumManager.Application")
```

```
sm.OpenFile (Sheet1.Cells(4, 4))  
Set s = sm.GetStats
```

```
m = s.MaxMax  
sm.SaveSpectrumAsSPX Sheet1.Cells(5, 4)
```

```
End Sub
```

Declaration

Instantiation

- Open file
- Get statistical information
- Save file in the different format

# Using AFGROW and Spectrum Manager interface together Example



	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2					Randomize		Run Afgrow						
3													
4					Save sp3		Save sp3, Run Afgrow						
5		Inputs:											
6		Number Of Rows	10		Row #	Cycles	Max	Min	MaxBend	MinBend	MaxBear	MinBear	Block Label
7		MultiChannel	0		1	70	0.766712	0.289759					Flight1
8		Generate Block Label	1		2	28	0.650974	0.38737					
9		# of Flights	3		3	1	0.880362	0.407245					
10		Block Label	Flight		4	70	0.522676	0.207016					Flight2
11		Spectrum Title	ExcelSpectrum		5	86	0.89524	0.186768					
12		File Name	ExcelSpectrum.sp3		6	96	0.935723	0.028118					
13					7	94	0.682009	0.262434					Flight3
14					8	76	0.526752	0.296229					
15					9	46	0.649083	0.311348					
16					10	64	0.631896	0.139671					
17													
18		Afgrow Inputs:			Cycles	Final C	Final K						
19		SMF	14										
20													

# Spectrum Manager Part



```

Dim sm As SpectrumManager.Application
Set sm = CreateObject("SpectrumManager.Application")
Dim currentSub As String
Dim multiChannel As Boolean
Dim i As Integer
Dim count As Integer
count = Sheet1.Cells(6, 3)
multiChannel = Sheet1.Cells(7, 3)
sm.CreateSpectrum Sheet1.Cells(11, 3), Sheet1.Cells(10, 3), "", multiChannel
For i = 1 To count
  If Not IsEmpty(Sheet1.Cells(7 + (i - 1), 13)) Then
    currentSub = Sheet1.Cells(7 + (i - 1), 13)
    sm.CreateSubSpectrum currentSub
    sm.AddSubSpectrumToSequence currentSub
  End If
  sm.AddLevelToSubSpectrumByName currentSub, Sheet1.Cells(7 + (i - 1), 8), Sheet1.Cells(7 + (i - 1), 7), Sheet1.Cells(7 + (i - 1), 6)
Next i
sm.SaveSpectrumAsSP3 ("C:\Users\Developer\Documents\ProducedByExcel\" + Cells(12, 3))
Set sm = Nothing
  
```

Declaration  
And  
Instantiation

Spectrum Created

Sub-Spectra Created  
and added to  
Spectrum

Spectrum File Saved

Levels added to Sub-  
spectra

# AFGROW Part



```

Dim Af As Afgrow.Application
Set Af = CreateObject("Afgrow.Application")
Af.Visible = True
Af.OpenSpectrumFile ("C:\Users\Developer\Documents\ProducedByExcel\" + Cells(12, 3))
Af.SMF = Cells(19, 3)
Dim a As Double, b As Double, c As Double, d As Double, e As Double, f As Double, g As Double, h As Double
Dim res As Integer
res = Af.RunFrozPredict(a, b, c, d, e, f, g)
    
```

Declaration  
And  
Instantiation

Open Spectrum file  
and setup predicion

Run prediction

# Short Term Plans

- Add support for spectrum tags and time-dependent spectra
- Add support for spectrum generation from exceedance curve
- Expand the capabilities of the *Application* interface

# Questions