# Table of Contents

## Chapter 1: About Automation

**Overview of Automation** .. 1-3
- Standards .. 1-3
- Compatibility with Other LeCroy Scopes .. 1-3
- Automation and IEEE 488.2 Remote Control – How Do They Compare? .. 1-3
- General Characteristics .. 1-4

**Introduction to the X-Stream Browser** .. 1-5

**Step-by-Step Introduction to Automation Using VBScript** .. 1-6

**Where is Automation Used?** .. 1-8

**Setups (Panel Files) are Programs!** .. 1-9

**Custom Math and Measurements** .. 1-11
- CustomDSO .. 1-11
- Control from External Applications .. 1-12
  - From Visual Basic .. 1-12
  - From MATLAB .. 1-13
  - From MS Office (Excel) .. 1-14

**Control by Distributed COM (DCOM)** .. 1-15
- Security Settings on the Instrument (Server): User Level .. 1-15
- Security Settings on the Instrument (Server): Share Level .. 1-15
- Initialize the Controlling PC (Client) .. 1-15
- Connecting to a Remote Instrument .. 1-15

**Control Variables Explained** .. 1-17

**Accessing Waveform/Measurement Results** .. 1-19
- Waveforms .. 1-19
- Measurements .. 1-20
- Result Status .. 1-21

**Synchronization** .. 1-22

**Good Practices** .. 1-24

**Examples** .. 1-24
- Example 1: Excel Macro to Perform FFT of C1 .. 1-24
- Example 2: VBScript Program to Perform FFT of C1 and Store Results in Text File .. 1-25
- Example 3: Script to Measure the Rise Time of the Signal on C1 and Display It in a Popup Window .. 1-26

**Early and Late Binding** .. 1-27

**VBS Remote Command** .. 1-28

**X-Stream DSO Objects** .. 1-29
CHAPTER 2: CONTROL REFERENCE

LECROY.XSTREAMDSO.1 APP........................................................................................................ 2-2
AddZoomTrace Action .................................................................................................................. 2-2
AutoSetup Action ....................................................................................................................... 2-3
ClearSweeps Action .................................................................................................................... 2-3
DoPrint Action .......................................................................................................................... 2-3
ExitWithoutConfirm Action ....................................................................................................... 2-4
FirmwareVersion String.............................................................................................................. 2-4
Height Property .......................................................................................................................... 2-4
HideClock Bool .......................................................................................................................... 2-4
InstrumentID String ................................................................................................................... 2-5
InstrumentModel String ............................................................................................................. 2-5
Left Property ............................................................................................................................... 2-5
Minimize Action .......................................................................................................................... 2-6
Quit() Method ............................................................................................................................. 2-6
ResetPreferences Action ............................................................................................................ 2-6
SetToDefaultSetup Action ......................................................................................................... 2-7
Shutdown Action .......................................................................................................................... 2-7
Sleep([in] double timeoutMilliseconds) Method ...................................................................... 2-7
Top Property ............................................................................................................................... 2-8
TouchScreenEnable Bool ............................................................................................................ 2-8
WaitUntilIdle([in] double timeoutSeconds) Method .................................................................. 2-8
Width Property ........................................................................................................................... 2-9
Windowed Action ....................................................................................................................... 2-9
WindowState Property ................................................................................................................. 2-9

ACQUISITION APP.ACQUISITION ......................................................................................... 2-10
Acquire([in] double timeoutSeconds, [in] long Method................................................................ 2-10
Calibrate Action .......................................................................................................................... 2-10
ClearSweeps Action.................................................................................................................... 2-11
TriggerMode Enum ..................................................................................................................... 2-11

AUXOUTPUT APP.ACQUISITION.AUXOUTPUT .................................................................. 2-13
Amplitude Double...................................................................................................................... 2-13
AuxInCoupling Enum.................................................................................................................. 2-14
Frequency DoubleLockstep ....................................................................................................... 2-14
Mode Enum ................................................................................................................................. 2-14
PulseWidth Double..................................................................................................................... 2-15
SetToTTL Bool ............................................................................................................................ 2-15

CHANNELS APP.ACQUISITION.CHANNELES ...................................................................... 2-16
CX app.Acquisition.Cx ............................................................................................................... 2-16
AverageSweeps Integer .............................................................................................................. 2-17
AxisXRotation Integer ............................................................................................................... 2-17
AxisYRotation Integer .............................................................................................................. 2-18
BandwidthLimit Enum ............................................................................................................... 2-18
ClearSweeps Action .................................................................................................................... 2-18
Coupling Enum .......................................................................................................................... 2-19
Deskew Double .......................................................................................................................... 2-19
InterpolateType Enum............................................................................................................... 2-19
Invert Bool .................................................................................................................................. 2-20
LabelsPosition String................................................................................................................. 2-20
LabelsText String....................................................................................................................... 2-20
Persist3DQuality Enum............................................................................................................. 2-21
Persisted Bool............................................................................................................................. 2-21
Persistence3d Bool ................................................................. 2-21
PersistenceMonoChrome Bool .................................................. 2-22
PersistenceSaturation Integer .................................................. 2-22
PersistenceTime Enum .......................................................... 2-22
ProbeAttenuation Double ....................................................... 2-23
ShowLastTrace Bool ............................................................. 2-23
UseDotJoin Bool .................................................................... 2-23
UseGrid String ....................................................................... 2-23
VerOffset Double .................................................................... 2-24
VerScale DoubleLockstep .......................................................... 2-24
VerScaleVariable Bool ............................................................ 2-25
View Bool .............................................................................. 2-25
ViewLabels Bool ..................................................................... 2-25
RESULT app.Acquisition.Cx.Out.Result .................................... 2-26
DataArray Property ................................................................ 2-26
FirstEventTime Property ....................................................... 2-27
HorizontalFrameStart Property .............................................. 2-27
HorizontalFrameStop Property .............................................. 2-27
HorizontalOffset Property .................................................... 2-28
HorizontalPerStep Property .................................................. 2-28
HorizontalResolution Property .............................................. 2-28
HorizontalUnits Property ...................................................... 2-29
IndexOfFirstSampleInFrame Property .................................... 2-29
LastEventTime Property ....................................................... 2-29
NumFrameDimensions Property ............................................. 2-30
NumSamplesInFrame Property ................................................ 2-30
Samples Property .................................................................. 2-30
Status Property ..................................................................... 2-31
StatusDescription Property ................................................... 2-31
Sweeps Property ................................................................... 2-31
VerticalFrameStart Property .................................................. 2-31
VerticalFrameStop Property ................................................... 2-32
VerticalMaxPossible Property ................................................. 2-32
VerticalMinPossible Property ............................................... 2-32
VerticalOffset Property ........................................................ 2-32
VerticalPerStep Property ....................................................... 2-33
VerticalResolution Property .................................................. 2-33
VerticalUnits Property .......................................................... 2-33

**HORIZONTAL app.ACQUISITION.HORIZONTAL** ........................................ 2-35
AcquisitionDuration Double ................................................. 2-35
ActiveChannels Enum ............................................................. 2-36
HorOffset Double ................................................................. 2-36
HorOffsetControl Enum ....................................................... 2-36
HorOffsetOrigin Double ....................................................... 2-37
HorScale DoubleLockstep ...................................................... 2-37
HorUnits String ................................................................... 2-37
MaxSamples DoubleLockstep ................................................... 2-38
NumPoints Integer ............................................................... 2-38
NumSegments Integer ............................................................ 2-38
ReferenceClock Enum ........................................................... 2-39
SampleClock Enum ............................................................... 2-39
SampleMode Enum ............................................................... 2-39
SampleRate DoubleLockstep .................................................... 2-40
SamplingRate Double ............................................................ 2-40
SequenceTimeout Double ........................................................ 2-40
## TRIGGER APP.ACQUISITION.TRIGGER

### Parameters
- **DropoutTime** `Double`  
- **Glitch** `Enum`  
- **GlitchHigh** `Double`  
- **GlitchLow** `Double`  
- **HoldoffEvents** `Integer`  
- **HoldoffTime** `Double`  
- **HoldoffType** `Enum`  
- **Interval** `Enum`  
- **IntervalDelta** `Double`  
- **IntervalHigh** `Double`  
- **IntervalLow** `Double`  
- **IntervalNominal** `Double`  
- **IntervalRange** `Enum`  
- **PatternType** `Enum`  
- **QualEvents** `Integer`  
- **QualFirst** `Bool`  
- **QualState** `Enum`  
- **QualTime** `Double`  
- **QualWait** `Enum`  
- **Source** `Enum`  
- **TrigLevel** `Double`  
- **Type** `Enum`  
- **ValidateSource** `Enum`  
- **Width** `Enum`  
- **WidthDelta** `Double`  
- **WidthNominal** `Double`  
- **WidthRange** `Enum`  
- **ZeroDelay** `Action`  
- **ZeroLevel** `Action`  

## CX APP.ACQUISITION.TRIGGER.CX

### Parameters
- **InputImpedance** `Enum`  
- **Level** `Double`  
- **PatternState** `Enum`  
- **Slope** `Enum`  

## EXT APP.ACQUISITION.TRIGGER.EXT

### Parameters
- **Coupling** `Enum`  
- **InputImpedance** `Enum`  
- **Level** `Double`  
- **PatternState** `Enum`  
- **Slope** `Enum`  

## LINE APP.ACQUISITION.TRIGGER.LINE

### Parameters
- **Coupling** `Enum`  
- **InputImpedance** `Enum`  
- **Level** `Double`  
- **PatternState** `Enum`  
- **Slope** `Enum`  

## SOURCES APP.ACQUISITION.TRIGGER.SOURCES

### Parameters
- **Pos1** `Double`  
- **Pos2** `Double`  
- **Readout** `Enum`  
- **Track** `Bool`  
- **Type** `Enum`  

---

*SequenceTimeoutEnable* `Bool`  
*SmartMemory* `Enum`  
*TimePerPoint* `Double`  
*ZeroDelay* `Action`  

**Manual**

*ISSUED: July 2003*
# Table of Contents

<table>
<thead>
<tr>
<th>View</th>
<th>Bool</th>
<th>2-60</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPos1</td>
<td>Double</td>
<td>2-60</td>
</tr>
<tr>
<td>XPos2</td>
<td>Double</td>
<td>2-60</td>
</tr>
<tr>
<td>YPos1</td>
<td>Double</td>
<td>2-60</td>
</tr>
<tr>
<td>YPos2</td>
<td>Double</td>
<td>2-61</td>
</tr>
</tbody>
</table>

**CUSTOMDSO**

<table>
<thead>
<tr>
<th>ACTION</th>
<th>Enable1</th>
<th>Bool</th>
<th>2-62</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION</td>
<td>Enable2</td>
<td>Bool</td>
<td>2-62</td>
</tr>
<tr>
<td>ACTION</td>
<td>Enable3</td>
<td>Bool</td>
<td>2-63</td>
</tr>
<tr>
<td>ACTION</td>
<td>Enable4</td>
<td>Bool</td>
<td>2-63</td>
</tr>
<tr>
<td>ACTION</td>
<td>Enable5</td>
<td>Bool</td>
<td>2-63</td>
</tr>
<tr>
<td>ACTION</td>
<td>Enable6</td>
<td>Bool</td>
<td>2-63</td>
</tr>
<tr>
<td>ACTION</td>
<td>Enable7</td>
<td>Bool</td>
<td>2-63</td>
</tr>
<tr>
<td>ACTION</td>
<td>Enable8</td>
<td>Bool</td>
<td>2-63</td>
</tr>
<tr>
<td>ACTION</td>
<td>Script1</td>
<td>FileName</td>
<td>2-63</td>
</tr>
<tr>
<td>ACTION</td>
<td>Script2</td>
<td>FileName</td>
<td>2-63</td>
</tr>
<tr>
<td>ACTION</td>
<td>Script3</td>
<td>FileName</td>
<td>2-64</td>
</tr>
<tr>
<td>ACTION</td>
<td>Script4</td>
<td>FileName</td>
<td>2-64</td>
</tr>
<tr>
<td>ACTION</td>
<td>Script5</td>
<td>FileName</td>
<td>2-64</td>
</tr>
<tr>
<td>ACTION</td>
<td>Script6</td>
<td>FileName</td>
<td>2-64</td>
</tr>
<tr>
<td>ACTION</td>
<td>Script7</td>
<td>FileName</td>
<td>2-64</td>
</tr>
<tr>
<td>ACTION</td>
<td>Script8</td>
<td>FileName</td>
<td>2-64</td>
</tr>
<tr>
<td>MODE</td>
<td>Enum</td>
<td>2-64</td>
<td></td>
</tr>
<tr>
<td>PlugIn1Install</td>
<td>Action</td>
<td>2-65</td>
<td></td>
</tr>
<tr>
<td>PlugIn1ProgId</td>
<td>String</td>
<td>2-65</td>
<td></td>
</tr>
<tr>
<td>PlugIn1Remove</td>
<td>Action</td>
<td>2-65</td>
<td></td>
</tr>
<tr>
<td>PresentAtPowerUp</td>
<td>Bool</td>
<td>2-65</td>
<td></td>
</tr>
</tbody>
</table>

**DISPLAY**

<table>
<thead>
<tr>
<th>ACTION</th>
<th>AxisLabels</th>
<th>Bool</th>
<th>2-67</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION</td>
<td>AxisXRotation</td>
<td>Integer</td>
<td>2-67</td>
</tr>
<tr>
<td>ACTION</td>
<td>AxisYRotation</td>
<td>Integer</td>
<td>2-67</td>
</tr>
<tr>
<td>ACTION</td>
<td>C1Color</td>
<td>Color</td>
<td>2-68</td>
</tr>
<tr>
<td>ACTION</td>
<td>C1PrintColor</td>
<td>Color</td>
<td>2-68</td>
</tr>
<tr>
<td>ACTION</td>
<td>C2Color</td>
<td>Color</td>
<td>2-69</td>
</tr>
<tr>
<td>ACTION</td>
<td>C2PrintColor</td>
<td>Color</td>
<td>2-69</td>
</tr>
<tr>
<td>ACTION</td>
<td>C3Color</td>
<td>Color</td>
<td>2-69</td>
</tr>
<tr>
<td>ACTION</td>
<td>C3PrintColor</td>
<td>Color</td>
<td>2-69</td>
</tr>
<tr>
<td>ACTION</td>
<td>C4Color</td>
<td>Color</td>
<td>2-69</td>
</tr>
<tr>
<td>ACTION</td>
<td>C4PrintColor</td>
<td>Color</td>
<td>2-69</td>
</tr>
<tr>
<td>ACTION</td>
<td>ClearSweeps</td>
<td>Action</td>
<td>2-69</td>
</tr>
<tr>
<td>ACTION</td>
<td>DisplayMode</td>
<td>Enum</td>
<td>2-70</td>
</tr>
<tr>
<td>ACTION</td>
<td>F1Color</td>
<td>Color</td>
<td>2-70</td>
</tr>
<tr>
<td>ACTION</td>
<td>F1PrintColor</td>
<td>Color</td>
<td>2-70</td>
</tr>
<tr>
<td>ACTION</td>
<td>F2Color</td>
<td>Color</td>
<td>2-70</td>
</tr>
<tr>
<td>ACTION</td>
<td>F2PrintColor</td>
<td>Color</td>
<td>2-70</td>
</tr>
<tr>
<td>ACTION</td>
<td>F3Color</td>
<td>Color</td>
<td>2-70</td>
</tr>
<tr>
<td>ACTION</td>
<td>F3PrintColor</td>
<td>Color</td>
<td>2-70</td>
</tr>
<tr>
<td>ACTION</td>
<td>F4Color</td>
<td>Color</td>
<td>2-71</td>
</tr>
<tr>
<td>ACTION</td>
<td>F4PrintColor</td>
<td>Color</td>
<td>2-71</td>
</tr>
<tr>
<td>ACTION</td>
<td>F5Color</td>
<td>Color</td>
<td>2-71</td>
</tr>
<tr>
<td>ACTION</td>
<td>F5PrintColor</td>
<td>Color</td>
<td>2-71</td>
</tr>
<tr>
<td>ACTION</td>
<td>F6Color</td>
<td>Color</td>
<td>2-71</td>
</tr>
<tr>
<td>ACTION</td>
<td>F6PrintColor</td>
<td>Color</td>
<td>2-71</td>
</tr>
<tr>
<td>ACTION</td>
<td>F7Color</td>
<td>Color</td>
<td>2-71</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>F7PrintColor</td>
<td>Color</td>
<td>2-71</td>
<td></td>
</tr>
<tr>
<td>F8PrintColor</td>
<td>Color</td>
<td>2-72</td>
<td></td>
</tr>
<tr>
<td>M1Color</td>
<td>Color</td>
<td>2-73</td>
<td></td>
</tr>
<tr>
<td>M1PrintColor</td>
<td>Color</td>
<td>2-73</td>
<td></td>
</tr>
<tr>
<td>M2Color</td>
<td>Color</td>
<td>2-73</td>
<td></td>
</tr>
<tr>
<td>M2PrintColor</td>
<td>Color</td>
<td>2-74</td>
<td></td>
</tr>
<tr>
<td>M3Color</td>
<td>Color</td>
<td>2-74</td>
<td></td>
</tr>
<tr>
<td>M3PrintColor</td>
<td>Color</td>
<td>2-74</td>
<td></td>
</tr>
<tr>
<td>M4Color</td>
<td>Color</td>
<td>2-74</td>
<td></td>
</tr>
<tr>
<td>M4PrintColor</td>
<td>Color</td>
<td>2-74</td>
<td></td>
</tr>
<tr>
<td>FactoryDefault</td>
<td>Action</td>
<td>2-72</td>
<td></td>
</tr>
<tr>
<td>GridIntensity</td>
<td>Integer</td>
<td>2-72</td>
<td></td>
</tr>
<tr>
<td>GridMode</td>
<td>Enum</td>
<td>2-72</td>
<td></td>
</tr>
<tr>
<td>GridOnTop</td>
<td>Bool</td>
<td>2-73</td>
<td></td>
</tr>
<tr>
<td>LockPersistence</td>
<td>Enum</td>
<td>2-73</td>
<td></td>
</tr>
<tr>
<td>M1PrintColor</td>
<td>Color</td>
<td>2-73</td>
<td></td>
</tr>
<tr>
<td>M1PrintColor</td>
<td>Color</td>
<td>2-73</td>
<td></td>
</tr>
<tr>
<td>M2PrintColor</td>
<td>Color</td>
<td>2-73</td>
<td></td>
</tr>
<tr>
<td>M2PrintColor</td>
<td>Color</td>
<td>2-74</td>
<td></td>
</tr>
<tr>
<td>M3PrintColor</td>
<td>Color</td>
<td>2-74</td>
<td></td>
</tr>
<tr>
<td>M3PrintColor</td>
<td>Color</td>
<td>2-74</td>
<td></td>
</tr>
<tr>
<td>M4PrintColor</td>
<td>Color</td>
<td>2-74</td>
<td></td>
</tr>
<tr>
<td>M4PrintColor</td>
<td>Color</td>
<td>2-74</td>
<td></td>
</tr>
<tr>
<td>M4PrintColor</td>
<td>Color</td>
<td>2-74</td>
<td></td>
</tr>
<tr>
<td>NumSegmentsDisplayed</td>
<td>Integer</td>
<td>2-74</td>
<td></td>
</tr>
<tr>
<td>Persist3DQuality</td>
<td>Enum</td>
<td>2-74</td>
<td></td>
</tr>
<tr>
<td>Persisted</td>
<td>Bool</td>
<td>2-75</td>
<td></td>
</tr>
<tr>
<td>Persistence3D</td>
<td>Bool</td>
<td>2-75</td>
<td></td>
</tr>
<tr>
<td>PersistenceLastTrace</td>
<td>Bool</td>
<td>2-75</td>
<td></td>
</tr>
<tr>
<td>PersistenceMonoChrome</td>
<td>Bool</td>
<td>2-76</td>
<td></td>
</tr>
<tr>
<td>PersistenceSaturation</td>
<td>Integer</td>
<td>2-76</td>
<td></td>
</tr>
<tr>
<td>PersistenceStyle</td>
<td>Enum</td>
<td>2-76</td>
<td></td>
</tr>
<tr>
<td>PersistenceTime</td>
<td>Enum</td>
<td>2-76</td>
<td></td>
</tr>
<tr>
<td>PreviewPrintColors</td>
<td>Action</td>
<td>2-77</td>
<td></td>
</tr>
<tr>
<td>ResetAll</td>
<td>Action</td>
<td>2-77</td>
<td></td>
</tr>
<tr>
<td>SegmentMode</td>
<td>Enum</td>
<td>2-77</td>
<td></td>
</tr>
<tr>
<td>StartSegment</td>
<td>Integer</td>
<td>2-78</td>
<td></td>
</tr>
<tr>
<td>TraceStyle</td>
<td>Enum</td>
<td>2-78</td>
<td></td>
</tr>
<tr>
<td>Destination</td>
<td>Enum</td>
<td>2-79</td>
<td></td>
</tr>
<tr>
<td>Directory</td>
<td>FileName</td>
<td>2-79</td>
<td></td>
</tr>
<tr>
<td>EMailMessage</td>
<td>String</td>
<td>2-80</td>
<td></td>
</tr>
<tr>
<td>GridAreaOnly</td>
<td>Bool</td>
<td>2-80</td>
<td></td>
</tr>
<tr>
<td>HardcopyArea</td>
<td>Enum</td>
<td>2-80</td>
<td></td>
</tr>
<tr>
<td>ImageFileFormat</td>
<td>Enum</td>
<td>2-81</td>
<td></td>
</tr>
<tr>
<td>Orientation</td>
<td>Enum</td>
<td>2-81</td>
<td></td>
</tr>
<tr>
<td>PreferredFilename</td>
<td>String</td>
<td>2-81</td>
<td></td>
</tr>
<tr>
<td>Print</td>
<td>Action</td>
<td>2-82</td>
<td></td>
</tr>
<tr>
<td>PromptForMessage</td>
<td>Bool</td>
<td>2-82</td>
<td></td>
</tr>
<tr>
<td>SelectedPrinter</td>
<td>Enum</td>
<td>2-82</td>
<td></td>
</tr>
<tr>
<td>StripChart</td>
<td>Bool</td>
<td>2-82</td>
<td></td>
</tr>
<tr>
<td>StripChartFactor</td>
<td>Enum</td>
<td>2-82</td>
<td></td>
</tr>
<tr>
<td>UsePrintPalette</td>
<td>Bool</td>
<td>2-83</td>
<td></td>
</tr>
<tr>
<td>ClearSweeps</td>
<td>Action</td>
<td>2-84</td>
<td></td>
</tr>
<tr>
<td>ResetZoom</td>
<td>Action</td>
<td>2-84</td>
<td></td>
</tr>
<tr>
<td>FX</td>
<td>app.Math.Fx</td>
<td>2-85</td>
<td></td>
</tr>
<tr>
<td>AxisXRotation</td>
<td>Integer</td>
<td>2-85</td>
<td></td>
</tr>
<tr>
<td>AxisYRotation</td>
<td>Integer</td>
<td>2-86</td>
<td></td>
</tr>
<tr>
<td>ClearSweeps</td>
<td>Action</td>
<td>2-86</td>
<td></td>
</tr>
<tr>
<td>DoResetZoom</td>
<td>Action</td>
<td>2-86</td>
<td></td>
</tr>
</tbody>
</table>
# Table of Contents

- DoStoreToMemoryTrace Action ............................................................... 2-86
- Equation String.................................................................................. 2-87
- GraphOp Enum.................................................................................... 2-87
- LabelsPosition String......................................................................... 2-87
- LabelsText String................................................................................ 2-87
- MathMode Enum.................................................................................. 2-88
- MeasureOp Enum................................................................................ 2-88
- Operator1 Enum.................................................................................. 2-90
- Operator2 Enum.................................................................................. 2-92
- Persist3DQuality Enum....................................................................... 2-93
- Persisted Bool..................................................................................... 2-93
- Persistence3d Bool ............................................................................. 2-94
- PersistenceMonoChrome Bool............................................................. 2-94
- PersistenceSaturation Integer.............................................................. 2-94
- PersistenceTime Enum........................................................................ 2-94
- ShowLastTraceBool .......................................................................... 2-95
- Source1 Enum..................................................................................... 2-95
- Source2 Enum..................................................................................... 2-96
- Source3 Enum..................................................................................... 2-97
- UseDotJoin Bool................................................................................ 2-97
- UseGrid String.................................................................................... 2-98
- View Bool ......................................................................................... 2-98
- ViewLabels Bool .............................................................................. 2-98

**OPERATOR SETUP** APP.MATH.FX.OPERATOR SETUP ........................................... 2-99

**RESULT** app.Math.Fx.Out.Result ............................................................ 2-100

**ZOOM** APP.MATH.FX.ZOOM ................................................................. 2-101
- HorPos Double.................................................................................... 2-101
- HorZoom Double................................................................................ 2-101
- ResetZoom Action.............................................................................. 2-101
- VariableHorZoom Bool....................................................................... 2-102
- VariableVerZoom Bool....................................................................... 2-102
- VerPos Double.................................................................................... 2-102
- VerZoom Double................................................................................ 2-102

**XY** APP.MATH.XY ............................................................................. 2-103
- AxisXRotation Integer........................................................................ 2-103
- AxisYRotation Integer........................................................................ 2-103
- ClearSweeps Action........................................................................... 2-104
- InputX Enum...................................................................................... 2-104
- InputY Enum...................................................................................... 2-104
- Persist3DQuality Enum...................................................................... 2-105
- Persisted Bool.................................................................................... 2-105
- Persistence3d Bool ............................................................................ 2-105
- PersistenceMonoChrome Bool............................................................ 2-105
- PersistenceSaturation Integer............................................................ 2-105
- PersistenceTime Enum....................................................................... 2-105
- ShowLastTraceBool ......................................................................... 2-106
- UseDotJoin Bool................................................................................ 2-106

**RESULT** app.Math.XY.Out.Result ........................................................ 2-107
- DataArray Property........................................................................... 2-107
- FirstEventTime Property.................................................................. 2-107
- HorizontalFrameStart Property.......................................................... 2-108
- HorizontalFrameStop Property.............................................................. 2-108
- HorizontalOffset Property................................................................. 2-108
- HorizontalPerStep Property................................................................. 2-108
AUTOMATION MANUAL

<table>
<thead>
<tr>
<th>Property</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>HorizontalResolution Property</td>
<td>2-109</td>
</tr>
<tr>
<td>HorizontalUnits Property</td>
<td>2-109</td>
</tr>
<tr>
<td>LastEventTime Property</td>
<td>2-109</td>
</tr>
<tr>
<td>NumFrameDimensions Property</td>
<td>2-110</td>
</tr>
<tr>
<td>Samples Property</td>
<td>2-110</td>
</tr>
<tr>
<td>Sweeps Property</td>
<td>2-110</td>
</tr>
<tr>
<td>XFrameStart Property</td>
<td>2-110</td>
</tr>
<tr>
<td>XFrameStop Property</td>
<td>2-111</td>
</tr>
<tr>
<td>XMaxPossible Property</td>
<td>2-111</td>
</tr>
<tr>
<td>XMinPossible Property</td>
<td>2-111</td>
</tr>
<tr>
<td>XOffset Property</td>
<td>2-111</td>
</tr>
<tr>
<td>XPerStep Property</td>
<td>2-111</td>
</tr>
<tr>
<td>XResolution Property</td>
<td>2-112</td>
</tr>
<tr>
<td>XUnits Property</td>
<td>2-112</td>
</tr>
<tr>
<td>YFrameStart Property</td>
<td>2-112</td>
</tr>
<tr>
<td>YFrameStop Property</td>
<td>2-113</td>
</tr>
<tr>
<td>YMaxPossible Property</td>
<td>2-113</td>
</tr>
<tr>
<td>YMinPossible Property</td>
<td>2-113</td>
</tr>
<tr>
<td>YOffset Property</td>
<td>2-113</td>
</tr>
<tr>
<td>YPerStep Property</td>
<td>2-113</td>
</tr>
<tr>
<td>YResolution Property</td>
<td>2-114</td>
</tr>
<tr>
<td>YUnits Property</td>
<td>2-114</td>
</tr>
</tbody>
</table>

MEASURE APP.MEASURE................................................................. 2-115

<table>
<thead>
<tr>
<th>Action</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClearAll</td>
<td>2-115</td>
</tr>
<tr>
<td>ClearSweeps</td>
<td>2-115</td>
</tr>
<tr>
<td>HelpMarkers</td>
<td>2-116</td>
</tr>
<tr>
<td>HistoOn</td>
<td>2-116</td>
</tr>
<tr>
<td>MeasureMode</td>
<td>2-116</td>
</tr>
<tr>
<td>SetGateToDefault</td>
<td>2-116</td>
</tr>
<tr>
<td>StatsOn</td>
<td>2-116</td>
</tr>
<tr>
<td>StdGateStart</td>
<td>2-116</td>
</tr>
<tr>
<td>StdGateStop</td>
<td>2-117</td>
</tr>
<tr>
<td>StdSource</td>
<td>2-117</td>
</tr>
</tbody>
</table>

MEASURE APP.MEASURE.MEASURE....................................................... 2-118

<table>
<thead>
<tr>
<th>Property</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PX app.Measure.Px</td>
<td>2-118</td>
</tr>
<tr>
<td>ArithEngine</td>
<td>2-118</td>
</tr>
<tr>
<td>Equation</td>
<td>2-119</td>
</tr>
<tr>
<td>GateStart</td>
<td>2-119</td>
</tr>
<tr>
<td>GateStop</td>
<td>2-119</td>
</tr>
<tr>
<td>MeasurementType</td>
<td>2-119</td>
</tr>
<tr>
<td>ParamEngine</td>
<td>2-120</td>
</tr>
<tr>
<td>PSource1</td>
<td>2-121</td>
</tr>
<tr>
<td>PSource2</td>
<td>2-122</td>
</tr>
<tr>
<td>Source1</td>
<td>2-122</td>
</tr>
<tr>
<td>Source2</td>
<td>2-122</td>
</tr>
<tr>
<td>View</td>
<td>2-123</td>
</tr>
</tbody>
</table>

OPERATOR APP.MEASURE.PX.OPERATOR.................................................. 2-124

RESULT APP.MEASURE.PX.OUT.RESULT.................................................... 2-125

<table>
<thead>
<tr>
<th>Property</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>HorizontalResolution Property</td>
<td>2-125</td>
</tr>
<tr>
<td>HorizontalUnits Property</td>
<td>2-125</td>
</tr>
<tr>
<td>Value Property</td>
<td>2-125</td>
</tr>
<tr>
<td>VerticalResolution Property</td>
<td>2-126</td>
</tr>
<tr>
<td>VerticalUnits Property</td>
<td>2-126</td>
</tr>
</tbody>
</table>

ISSUED: July 2003
STATISTICS  APP.MEASURE.PX.STATISTICS................................................................. 2-127
RESULT  app.Measure.Px.Statistics("histo").Result.................................................. 2-127
Base  Property........................................................................................................ 2-127
BinPopulations  Property.......................................................................................... 2-128
Bins  Property.......................................................................................................... 2-128
BinWidth  Property................................................................................................... 2-128
FirstPopulatedBin  Property...................................................................................... 2-129
HorizontalFrameStart  Property............................................................................... 2-129
HorizontalFrameStop  Property.................................................................................. 2-129
HorizontalUnits  Property........................................................................................ 2-129
LastPopulatedBin  Property...................................................................................... 2-130
Max  Property........................................................................................................... 2-130
MaxPopulation  Property.......................................................................................... 2-130
MaxPopulationBin  Property....................................................................................... 2-130
Mean  Property......................................................................................................... 2-131
Min  Property............................................................................................................ 2-131
OffsetAtLeftEdge  Property....................................................................................... 2-131
PeakInfo  Property................................................................................................... 2-131
Peaks  Property........................................................................................................ 2-132
PopulationInside  Property....................................................................................... 2-132
PopulationOver  Property.......................................................................................... 2-132
PopulationUnder  Property......................................................................................... 2-133
Rms  Property............................................................................................................ 2-133
Sdev  Property.......................................................................................................... 2-133
Sweeps  Property...................................................................................................... 2-133
Top  Property............................................................................................................ 2-134
VerticalFrameStart  Property..................................................................................... 2-134
VerticalFrameStop  Property...................................................................................... 2-134
VerticalUnits  Property............................................................................................. 2-134
RESULT  app.Measure.Px.Statistics("last").Result...................................................... 2-135
Value  Property......................................................................................................... 2-135
ValueArray  Property................................................................................................ 2-135
VerticalResolution  Property..................................................................................... 2-135
VerticalUnits  Property............................................................................................. 2-135
RESULT  app.Measure.Px.Statistics("max").Result...................................................... 2-136
HorizontalResolution  Property................................................................................. 2-136
HorizontalUnits  Property......................................................................................... 2-136
Value  Property......................................................................................................... 2-136
VerticalResolution  Property..................................................................................... 2-137
VerticalUnits  Property............................................................................................. 2-137
RESULT  APP.MEASURE.PX.STATISTICS("MEAN").RESULT ...................................... 2-138
HorizontalResolution  Property................................................................................. 2-138
HorizontalUnits  Property......................................................................................... 2-138
Value  Property......................................................................................................... 2-138
VerticalResolution  Property..................................................................................... 2-139
VerticalUnits  Property............................................................................................. 2-139
RESULT  app.Measure.Px.Statistics("min").Result...................................................... 2-140
HorizontalResolution  Property................................................................................. 2-140
HorizontalUnits  Property......................................................................................... 2-140
Value  Property......................................................................................................... 2-140
VerticalResolution  Property..................................................................................... 2-141
VerticalUnits  Property............................................................................................. 2-141
RESULT  app.Measure.Px.Statistics("num").Result..................................................... 2-142
HorizontalResolution  Property................................................................................. 2-142
HorizontalUnits  Property......................................................................................... 2-142
## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Bool</td>
<td>2-157</td>
</tr>
<tr>
<td>RESULT app.PassFail.Qx.Out.Result</td>
<td>2-158</td>
</tr>
<tr>
<td>HorizontalResolution Property</td>
<td>2-158</td>
</tr>
<tr>
<td>HorizontalUnits Property</td>
<td>2-158</td>
</tr>
<tr>
<td>Status Property</td>
<td>2-158</td>
</tr>
<tr>
<td>StatusDescription Property</td>
<td>2-158</td>
</tr>
<tr>
<td>Value Property</td>
<td>2-158</td>
</tr>
<tr>
<td>ValueArray Property</td>
<td>2-158</td>
</tr>
<tr>
<td>PREFERENCES APP.PREFERENCES</td>
<td>2-159</td>
</tr>
<tr>
<td>AudibleFeedback Bool</td>
<td>2-159</td>
</tr>
<tr>
<td>OffsetControl Enum</td>
<td>2-159</td>
</tr>
<tr>
<td>Performance Enum</td>
<td>2-159</td>
</tr>
<tr>
<td>EMAIL APP.PREFERENCES.EMAIL</td>
<td>2-160</td>
</tr>
<tr>
<td>DefaultRecipient String</td>
<td>2-160</td>
</tr>
<tr>
<td>Mode Enum</td>
<td>2-160</td>
</tr>
<tr>
<td>OriginatorAddress String</td>
<td>2-160</td>
</tr>
<tr>
<td>SendTestMail Action</td>
<td>2-160</td>
</tr>
<tr>
<td>SMTPServer String</td>
<td>2-160</td>
</tr>
<tr>
<td>SAVERECALL APP.SAVERECALL</td>
<td>2-162</td>
</tr>
<tr>
<td>SETUP APP.SAVERECALL.SETUP</td>
<td>2-163</td>
</tr>
<tr>
<td>DoRecallDefaultNvIPanel Action</td>
<td>2-163</td>
</tr>
<tr>
<td>DoRecallDefaultPanel Action</td>
<td>2-164</td>
</tr>
<tr>
<td>DoRecallPanel Action</td>
<td>2-164</td>
</tr>
<tr>
<td>DoSavePanel Action</td>
<td>2-164</td>
</tr>
<tr>
<td>InternalName1 String</td>
<td>2-165</td>
</tr>
<tr>
<td>InternalName2 String</td>
<td>2-165</td>
</tr>
<tr>
<td>InternalName3 String</td>
<td>2-165</td>
</tr>
<tr>
<td>InternalName4 String</td>
<td>2-165</td>
</tr>
<tr>
<td>InternalName5 String</td>
<td>2-165</td>
</tr>
<tr>
<td>InternalName6 String</td>
<td>2-165</td>
</tr>
<tr>
<td>PanelFilename FileName</td>
<td>2-165</td>
</tr>
<tr>
<td>RecallInternal1 Action</td>
<td>2-165</td>
</tr>
<tr>
<td>RecallInternal2 Action</td>
<td>2-166</td>
</tr>
<tr>
<td>RecallInternal3 Action</td>
<td>2-166</td>
</tr>
<tr>
<td>RecallInternal4 Action</td>
<td>2-166</td>
</tr>
<tr>
<td>RecallInternal5 Action</td>
<td>2-166</td>
</tr>
<tr>
<td>RecallInternal6 Action</td>
<td>2-166</td>
</tr>
<tr>
<td>SavelInternal1 Action</td>
<td>2-166</td>
</tr>
<tr>
<td>SavelInternal2 Action</td>
<td>2-166</td>
</tr>
<tr>
<td>SavelInternal3 Action</td>
<td>2-166</td>
</tr>
<tr>
<td>SavelInternal4 Action</td>
<td>2-167</td>
</tr>
<tr>
<td>SavelInternal5 Action</td>
<td>2-167</td>
</tr>
<tr>
<td>SavelInternal6 Action</td>
<td>2-167</td>
</tr>
<tr>
<td>UTILITIES APP.SAVERECALL.UTILITIES</td>
<td>2-168</td>
</tr>
<tr>
<td>CreateDir Action</td>
<td>2-168</td>
</tr>
<tr>
<td>DeleteAll Action</td>
<td>2-168</td>
</tr>
<tr>
<td>DeleteAllWithPrompt Action</td>
<td>2-168</td>
</tr>
<tr>
<td>DeleteFile Action</td>
<td>2-169</td>
</tr>
<tr>
<td>WAVEFORM APP.SAVERECALL.WAVEFORM</td>
<td>2-170</td>
</tr>
<tr>
<td>Delimiter Enum</td>
<td>2-171</td>
</tr>
<tr>
<td>DoRecall Action</td>
<td>2-171</td>
</tr>
<tr>
<td>DoSave Action</td>
<td>2-171</td>
</tr>
</tbody>
</table>
### AUTOMATION MANUAL

RecallDestination *Enum* ................................................................. 2-171
RecallFrom *Enum* ........................................................................... 2-171
RecallSource *Enum* ......................................................................... 2-171
SaveDestination *Enum* ................................................................. 2-172
SaveSource *Enum* ........................................................................... 2-172
SaveTo *Enum* ................................................................................... 2-173
SubFormat *Enum* ............................................................................. 2-173
TraceTitle *String* ........................................................................... 2-173
WaveFormat *Enum* ......................................................................... 2-174
WaveformDir *FileName* ................................................................. 2-174

**SDA** APP.SDA .................................................................................. 2-175

BERParamsOn *Bool* .......................................................................... 2-175
CalcType *Enum* ............................................................................... 2-175
ClockSource *Enum* .......................................................................... 2-176
DarkCalLevel *Double* ..................................................................... 2-176
ErrorMapOn *Bool* ............................................................................ 2-177
EyeMode *Enum* ............................................................................... 2-178
FindFrequency *Action* ................................................................. 2-178
FindScale *Action* ............................................................................ 2-178
HiPassFreq *Double* ......................................................................... 2-178
LowPassFreqDouble .......................................................................... 2-178
MaskFailTraceOn *Bool* ................................................................. 2-178
MaskFileName *String* ....................................................................... 2-179
NumPatternBits *Integer* ............................................................... 2-179
PLLON *Bool* .................................................................................... 2-179
RefReceiver *Bool* ........................................................................... 2-180
SDAMode *Enum* .............................................................................. 2-180
ShowFailLocation *Bool* ............................................................... 2-180
SignalFrequency *Double* ............................................................. 2-181
SignalMode *Enum* .......................................................................... 2-181
SignalType *Enum* ........................................................................... 2-181
StartN *Integer* ............................................................................... 2-182
StopN *Integer* ................................................................................ 2-182
Units *Enum* ................................................................................... 2-183
UserSignal *Enum* .......................................................................... 2-183
Ver_autoFit *Bool* ............................................................................ 2-184

**BADBITS** APP.SDA.BADBITS ....................................................... 2-185

AbsLevel *Double* ............................................................................ 2-186
AxisXRotation *Integer* ................................................................. 2-186
AxisYRotation *Integer* ................................................................. 2-186
BitsInLocator *Integer* ................................................................. 2-186
C1ReceiverStandard *Enum* ......................................................... 2-186
C2ReceiverStandard *Enum* ......................................................... 2-186
C3ReceiverStandard *Enum* ......................................................... 2-187
C4ReceiverStandard *Enum* ......................................................... 2-187
ClearSweeps *Action* ................................................................. 2-187
ClockTIESlope *Enum* ..................................................................... 2-187
DarkCal *Action* ............................................................................. 2-188
DarkCalLevel *Double* ................................................................... 2-188
DataSource *Enum* ......................................................................... 2-188
EyeMode *Enum* .............................................................................. 2-188
EyeThresholdType *Enum* ............................................................ 2-188
FailCuratorsOn *Bool* ................................................................. 2-189
Table of Contents

FailedList Enum................................................................. 2-189
FailedSymbolsFilter Enum.................................................. 2-189
LabelsPosition String........................................................ 2-189
LabelsText String............................................................... 2-189
MaskFailTraceOn Bool....................................................... 2-189
MaskFailX Double.............................................................. 2-189
MaskFailY Double.............................................................. 2-189
MaskTypeEnum Enum.......................................................... 2-189
MaxFailures Integer.......................................................... 2-190
MeasurementMode Enum..................................................... 2-190
MonochromeEye Enum....................................................... 2-190
PercentLevel Integer........................................................ 2-190
Persist3DQuality Enum..................................................... 2-190
Persisted Bool................................................................. 2-190
Persistence3DBool............................................................ 2-190
PersistenceMonoChrome Bool............................................... 2-190
PersistenceSaturation Integer............................................. 2-190
PersistenceTime Enum...................................................... 2-190
RefReceiver Bool............................................................. 2-191
SDAMode Enum................................................................. 2-191
ShowFailLocation Bool...................................................... 2-191
ShowLastTrace Bool........................................................ 2-191
SignalFrequencyDouble.................................................... 2-191
SignalTypeEnum.................................................................. 2-191
SliceWidth Integer............................................................. 2-192
Stop Bool............................................................................ 2-192
TrackMaskFail Action......................................................... 2-192
UseDotJoin Bool................................................................. 2-192
UseGrid String..................................................................... 2-193
UserSignal Enum............................................................... 2-193
VerAutoFit Bool................................................................. 2-193
View Bool............................................................................ 2-193
ViewLabels Bool................................................................. 2-193
XMargin Integer................................................................... 2-193
YMargin Integer................................................................... 2-194
RESULT app.SDA.BadBits.Out.Result........................................ 2-195
DataArray Property............................................................. 2-195
ExtendedStatus Property..................................................... 2-195
FirstEventTime Property..................................................... 2-195
HorizontalFrameStart Property.......................................... 2-195
HorizontalFrameStop Property............................................. 2-196
HorizontalOffset Property................................................ 2-196
HorizontalPerStep Property................................................ 2-196
HorizontalResolution Property.......................................... 2-196
HorizontalUnits Property.................................................. 2-196
HorizontalVarianceArray Property...................................... 2-196
HorizontalVariances Property............................................. 2-196
IndexOfFirstSampleInFrame Property............................... 2-196
LastEventTime Property...................................................... 2-196
NumFrameDimensions Property........................................... 2-196
NumSamplesInFrame Property.............................................. 2-196
Samples Property.............................................................. 2-196
Status Property............................................................... 2-197
StatusDescription Property................................................. 2-197
Sweeps Property............................................................... 2-197
UpdateTime Property........................................................ 2-197
AUTOMATION MANUAL

VerticalFrameStart Property ................................................................. 2-197
VerticalFrameStop Property ................................................................. 2-197
VerticalMaxPossible Property ............................................................... 2-197
VerticalMinPossible Property ............................................................... 2-197
VerticalOffset Property ...................................................................... 2-197
VerticalPerStep Property .................................................................. 2-197
VerticalResolution Property ................................................................. 2-198
VerticalUnits Property ....................................................................... 2-198

BITS_APP.SDA.BITS .............................................................................. 2-199
AxisXRotation Integer ....................................................................... 2-199
AxisYRotation Integer ....................................................................... 2-199
ClearSweeps Action ......................................................................... 2-199
EyeMode Enum .................................................................................. 2-199
LabelsPosition String ....................................................................... 2-199
LabelsText String .............................................................................. 2-200
MaskType Enum ................................................................................ 2-200
Persist3DQuality Enum .................................................................. 2-200
Persisted Bool .................................................................................. 2-200
Persistence3d Bool ............................................................................ 2-200
PersistenceMonoChrome Bool ........................................................... 2-200
PersistenceSaturation Integer ............................................................ 2-200
PersistenceTime Enum .................................................................... 2-200
ShowLastTrace Bool ......................................................................... 2-200
SignalFrequency Double ................................................................. 2-201
SignalType Enum .............................................................................. 2-201
UseDotJoin Bool ............................................................................... 2-202
UseGrid String ................................................................................ 2-202
VerAutoFit Bool ................................................................................ 2-202
View Bool ........................................................................................ 2-202
ViewLabels Bool .............................................................................. 2-202

RESULT_APP.SDA.BITS.Out.Result ....................................................... 2-203
DataArray Property .......................................................................... 2-203
ExtendedStatus Property ................................................................ 2-203
FirstEventTime Property ................................................................ 2-204
HorizontalFrameStart Property ......................................................... 2-204
HorizontalFrameStop Property ......................................................... 2-204
HorizontalOffset Property ................................................................. 2-204
HorizontalPerStep Property ............................................................... 2-204
HorizontalResolution Property ........................................................ 2-204
HorizontalUnits Property ................................................................. 2-204
HorizontalVarianceArray Property ................................................... 2-204
HorizontalVariances Property .......................................................... 2-204
IndexOfFirstSampleInFrame Property ............................................. 2-204
LastEventTime Property .................................................................. 2-204
NumFrameDimensions Property ......................................................... 2-204
NumSamplesInFrame Property .......................................................... 2-205
Samples Property .............................................................................. 2-205
Status Property ................................................................................ 2-205
StatusDescription Property ............................................................... 2-205
Sweeps Property .............................................................................. 2-205
UpdateTime Property ........................................................................ 2-205
VerticalFrameStart Property ............................................................... 2-205
VerticalFrameStop Property ............................................................... 2-205
VerticalMaxPossible Property .......................................................... 2-205
VerticalMinPossible Property ........................................................... 2-205
## Table of Contents

<table>
<thead>
<tr>
<th>Method/Property</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>VerticalOffset Property</td>
<td>2-205</td>
</tr>
<tr>
<td>VerticalPerStep Property</td>
<td>2-206</td>
</tr>
<tr>
<td>VerticalResolution Property</td>
<td>2-206</td>
</tr>
<tr>
<td>VerticalUnits Property</td>
<td>2-206</td>
</tr>
<tr>
<td>AxisXRotation Integer</td>
<td>2-207</td>
</tr>
<tr>
<td>AxisYRotation Integer</td>
<td>2-207</td>
</tr>
<tr>
<td>ClearSweeps Action</td>
<td>2-207</td>
</tr>
<tr>
<td>DataSource Enum</td>
<td>2-207</td>
</tr>
<tr>
<td>EyeMode Enum</td>
<td>2-208</td>
</tr>
<tr>
<td>LabelsPosition String</td>
<td>2-208</td>
</tr>
<tr>
<td>MaskType Enum</td>
<td>2-208</td>
</tr>
<tr>
<td>Persist3DQuality Enum</td>
<td>2-209</td>
</tr>
<tr>
<td>Persisted Bool</td>
<td>2-209</td>
</tr>
<tr>
<td>Persistence3dBool</td>
<td>2-209</td>
</tr>
<tr>
<td>PersistenceMonoChrome Bool</td>
<td>2-209</td>
</tr>
<tr>
<td>PersistenceSaturation Integer</td>
<td>2-209</td>
</tr>
<tr>
<td>PersistenceTime Enum</td>
<td>2-209</td>
</tr>
<tr>
<td>ShowLastTrace Bool</td>
<td>2-209</td>
</tr>
<tr>
<td>SignalFrequency Double</td>
<td>2-209</td>
</tr>
<tr>
<td>SignalType Enum</td>
<td>2-210</td>
</tr>
<tr>
<td>UseDotJoin Bool</td>
<td>2-210</td>
</tr>
<tr>
<td>UseGrid String</td>
<td>2-210</td>
</tr>
<tr>
<td>UseMonochrome Bool</td>
<td>2-210</td>
</tr>
<tr>
<td>VerAutoFit Bool</td>
<td>2-211</td>
</tr>
<tr>
<td>View Bool</td>
<td>2-211</td>
</tr>
<tr>
<td>ViewLabels Bool</td>
<td>2-211</td>
</tr>
<tr>
<td>RESULT app.SDA.Eye.Out.Result</td>
<td>2-212</td>
</tr>
<tr>
<td>Columns Property</td>
<td>2-212</td>
</tr>
<tr>
<td>DataArray Property</td>
<td>2-212</td>
</tr>
<tr>
<td>FirstEventTime Property</td>
<td>2-212</td>
</tr>
<tr>
<td>HorizontalFrameStart Property</td>
<td>2-213</td>
</tr>
<tr>
<td>HorizontalFrameStop Property</td>
<td>2-213</td>
</tr>
<tr>
<td>HorizontalOffset Property</td>
<td>2-213</td>
</tr>
<tr>
<td>HorizontalPerColumn Property</td>
<td>2-213</td>
</tr>
<tr>
<td>HorizontalUnits Property</td>
<td>2-213</td>
</tr>
<tr>
<td>LastEventTime Property</td>
<td>2-213</td>
</tr>
<tr>
<td>MaxPopulationInRectangle Property</td>
<td>2-213</td>
</tr>
<tr>
<td>NumFrameDimensions Property</td>
<td>2-214</td>
</tr>
<tr>
<td>PopulationOfRectangle Property</td>
<td>2-214</td>
</tr>
<tr>
<td>Rows Property</td>
<td>2-214</td>
</tr>
<tr>
<td>Sweeps Property</td>
<td>2-214</td>
</tr>
<tr>
<td>UpdateTime Property</td>
<td>2-214</td>
</tr>
<tr>
<td>VerticalFrameStart Property</td>
<td>2-214</td>
</tr>
<tr>
<td>VerticalFrameStop Property</td>
<td>2-214</td>
</tr>
<tr>
<td>VerticalOffset Property</td>
<td>2-215</td>
</tr>
<tr>
<td>VerticalPerRow Property</td>
<td>2-215</td>
</tr>
<tr>
<td>VerticalUnits Property</td>
<td>2-215</td>
</tr>
<tr>
<td>AbsLevel Double</td>
<td>2-216</td>
</tr>
<tr>
<td>EyeThresholdType Enum</td>
<td>2-216</td>
</tr>
<tr>
<td>FindFrequency Action</td>
<td>2-216</td>
</tr>
<tr>
<td>PercentLevel Integer</td>
<td>2-216</td>
</tr>
</tbody>
</table>
PLLOn  Bool................................................................. 2-217
SignalFrequency  Double............................................ 2-217
Slope  Enum ............................................................... 2-217
View  Bool................................................................. 2-217
RESULT.app.SDA.TIE.Out.Result ...................................... 2-218
   ExtendedStatus  Property........................................ 2-218
   FirstEventTime  Property........................................ 2-218
   HorizontalResolution  Property............................... 2-218
   HorizontalUnits  Property ..................................... 2-218
   LastEventTime  Property ....................................... 2-218
   NumFrameDimensions  Property................................ 2-218
   Status  Property................................................... 2-218
   StatusDescription  Property.................................... 2-218
   UpdateTime  Property ........................................... 2-219
   Value  Property................................................... 2-219
   ValueArray  Property............................................. 2-219
   VerticalResolution  Property................................... 2-219
   VerticalUnits  Property ....................................... 2-219
DATETIMESETUP  APP.UTILITY.DATETIMESETUP ...................... 2-220
   CurrentDateAndTime  String .................................. 2-220
   Day  Integer......................................................... 2-220
   Hour  Integer....................................................... 2-221
   Minute  Integer.................................................... 2-221
   Month  Integer..................................................... 2-221
   Second  Integer.................................................... 2-222
   SetFromSNTP  Action.............................................. 2-222
   Validate  Action.................................................... 2-222
   Year  Integer....................................................... 2-223
OPTIONS  APP.UTILITY.OPTIONS........................................ 2-224
   InstalledHWOptions  String.................................... 2-224
   InstalledSWOptions  String..................................... 2-224
   ScopeID  String................................................... 2-224
REMOTE  APP.UTILITY.REMOTE......................................... 2-225
   AllowControlFrom  String........................................ 2-225
   Assistant  Enum.................................................... 2-225
   GpibAddress  Integer............................................. 2-225
   Interface  Enum.................................................... 2-225
   RestrictControl  Enum.......................................... 2-225
   SetToErrorsOnlyAndClearAtStartup  Bool........................ 2-226
WEBEDITOR  APP.WEBEDITOR........................................... 2-227
   AddConnection([in] VARIANT destProcessor, [in] VARIANT Method
destInputPin, [in] VARIANT sourceProcessor, [in] VARIANT
sourceOutputPin)....................................................... 2-228
   AddPreview([in] VARIANT sourceProcessor, [in] VARIANT Method
sourcePin, [in] BSTR previewName, [in] double xPosition, [in] double xPosition) ........... 2-228
   AddProcessor([in] VARIANT processorOrClassId, [in] BSTR Method
requestedName, [in] double xPosition, [in] double yPosition)........................................ 2-228
   ClearSweeps  Action............................................... 2-228
   GetProcessor([in] VARIANT processor) Method................ 2-228
   RemoveAll()  Method............................................... 2-229
   RemoveConnection([in] VARIANT destProcessor, [in] VARIANT destInputPin) Method .... 2-229
   RemovePreview([in] VARIANT processor) Method............. 2-229
   RemoveProcessor([in] VARIANT processor) Method............ 2-229
CHAPTER 3: MATH/MEASURE CONTROL

REFERENCE

AVERAGE APP.MATH.FX.OPERATORYSETUP (OPERATOR = "AVERAGE") .......................................................... 3-2
AVERAGE APP.MATH.FX.OPERATORYSETUP (OPERATOR = "AVERAGE") .......................................................... 3-3
  AverageType Enum ................................................................. 3-3
  ClearSweeps Action ......................................................... 3-3
  Sweeps Integer ............................................................... 3-4
BOXCAR APP.MATH.FX.OPERATORYSETUP (OPERATOR = "BOXCAR") ................................................... 3-5
  Length Integer ............................................................... 3-5
CORRELATION APP.MATH.FX.OPERATORYSETUP (OPERATOR = "CORRELATION") ........................................ 3-6
  CorrLength Double ......................................................... 3-6
  CorrStart Double ............................................................ 3-6
DERIVATIVE APP.MATH.FX.OPERATORYSETUP (OPERATOR = "DERIVATIVE") ................................................. 3-7
  EnableAutoScale Bool ...................................................... 3-7
  FindScale Action ........................................................... 3-7
  VerOffset Double ........................................................... 3-7
  VerScale DoubleLockstep .................................................. 3-7
DESKEW APP.MATH.FX.OPERATORYSETUP (OPERATOR = "DESKEW") ........................................................ 3-8
  WaveDeskew Double ....................................................... 3-8
ENVELOPE APP.MATH.FX.OPERATORYSETUP (OPERATOR = "ENVELOPE") .................................................. 3-9
  ClearSweeps Action ......................................................... 3-9
  Sweeps Integer ............................................................... 3-9
ERES APP.MATH.FX.OPERATORYSETUP (OPERATOR = "ENHANCEDRESOLUTION") ........................................ 3-10
  Bits Enum ................................................................. 3-10
EXCELMATH APP.MATH.FX.OPERATORYSETUP (OPERATOR = "EXCELMATH") ..................................................... 3-11
  AddChart Action ........................................................... 3-11
  AddLabels Action .......................................................... 3-11
  Advanced Bool ........................................................... 3-12
  ClearSheet Action ......................................................... 3-12
  CreateDemoSheet Action ............................................... 3-12
  FindScale Action .......................................................... 3-12
  NewSheet Bool ............................................................ 3-13
  OutputCell String ......................................................... 3-13
  OutputEnable Bool ....................................................... 3-13
  OutputHeaderCell String ............................................... 3-13
  Scaling Enum ............................................................... 3-14
  Source1Cell String ....................................................... 3-14
  Source1Enable Bool ...................................................... 3-14
Source1HeaderCell String .................................................................................. 3-14
Source2Cell String .......................................................................................... 3-14
Source2Enable Bool ......................................................................................... 3-14
Source2HeaderCell String .............................................................................. 3-15
SpreadsheetFilename Filename ..................................................................... 3-15
Status String ................................................................................................... 3-15
WithHeader Bool ............................................................................................. 3-16

FFT  APP.MATH.FX.OPERATORYSETUP (OPERATOR = "FFT") ................. 3-17
Algorithm Enum ............................................................................................. 3-17
FillType Enum ................................................................................................. 3-17
SuppressDC Bool ............................................................................................ 3-18
Type Enum ....................................................................................................... 3-18
Window Enum .................................................................................................. 3-18

FILTER APP.MATH.FX.OPERATORYSETUP (OPERATOR = "FILTER") .... 3-19
AutoLength Bool ............................................................................................. 3-19
CosineBeta Double .......................................................................................... 3-19
CustomFilename Filename ............................................................................. 3-20
FilterKind Enum .............................................................................................. 3-20
FilterType Enum .............................................................................................. 3-20
FirOrIir Enum .................................................................................................. 3-20
GaussianBT Double .......................................................................................... 3-21
HighFreqPass Double ..................................................................................... 3-21
LowFreqPass Double ....................................................................................... 3-21
NumberOfTaps Integer .................................................................................. 3-21
PassBandAttenuation Double ........................................................................ 3-21
PassBandRipple Double .................................................................................. 3-21
Rolloff Double .................................................................................................. 3-22
StopBandAttenuation Double .......................................................................... 3-22
TransitionWidth Double ................................................................................ 3-22
Window Enum .................................................................................................. 3-22

FLOOR  APP.MATH.FX.OPERATORYSETUP (OPERATOR = "FLOOR") ...... 3-23
ClearSweeps Action ......................................................................................... 3-23
Sweeps Integer ................................................................................................. 3-23

HISTOGRAM APP.MATH.FX.OPERATORYSETUP (OPERATOR =
"HISTOGRAM") .............................................................................................. 3-24
AutoFindScale Bool .......................................................................................... 3-24
Bins DoubleLockstep ....................................................................................... 3-24
Center Double .................................................................................................. 3-25
ClearSweeps Action ......................................................................................... 3-25
FindScale Action .............................................................................................. 3-25
HorScale DoubleLockstep ............................................................................... 3-25
Values Integer DoubleLockstep ....................................................................... 3-25
VerScaleType Enum .......................................................................................... 3-25

INTEGRAL APP.MATH.FX.OPERATORYSETUP (OPERATOR =
"INTEGRAL") ................................................................................................. 3-26
Adder Double .................................................................................................... 3-26
FindScale Action .............................................................................................. 3-26
Multiplier Double ............................................................................................ 3-26
VerOffset Double ............................................................................................. 3-27
VerScale DoubleLockstep ................................................................................ 3-27
| APP.MATH.FX.OPERATORYSETUP (OPERATOR = |
| "INTERPOLATE") | 3-28 |
| INTERPOLATE | 3-28 |
| Expand Double | Lockstep | 3-28 |
| InterpolateType Enum | 3-28 |
| MATH SCRIPT APP.MATH.FX.OPERATORYSETUP (OPERATOR = |
| "WAVESCRIPT") | 3-29 |
| Code String | 3-29 |
| Language Enum | 3-30 |
| Status String | 3-30 |
| MATHECADMATH APP.MATH.FX.OPERATORYSETUP (OPERATOR = |
| "MATHECADMATH") | 3-31 |
| Advanced Bool | 3-31 |
| FindScale Action | 3-31 |
| NewSheet Bool | 3-32 |
| OutputEnable Bool | 3-32 |
| OutputHeaderVar String | 3-32 |
| OutputVar String | 3-33 |
| Reload Action | 3-33 |
| Scaling Enum | 3-33 |
| Source1Enable Bool | 3-33 |
| Source1HeaderVar String | 3-34 |
| Source1Var String | 3-34 |
| Source2Enable Bool | 3-34 |
| Source2HeaderVar String | 3-34 |
| Source2Var String | 3-34 |
| Status String | 3-34 |
| WithHeader Bool | 3-34 |
| WorksheetFilename FileName | 3-34 |
| MATLAB MATH APP.MATH.FX.OPERATORYSETUP (OPERATOR = |
| "MATLABWAVEFORM") | 3-35 |
| MATLABCode String | 3-35 |
| MATLABPlot Bool | 3-35 |
| MATLABScalePerDiv Double | 3-35 |
| MATLABZeroOffset Double | 3-36 |
| PHISTOGRAM APP.MATH.FX.OPERATORYSETUP (OPERATOR = |
| "PERSISTENCEHISTOGRAM") | 3-37 |
| CenterCursor Action | 3-37 |
| ClearSweeps Action | 3-37 |
| CutDirection Enum | 3-37 |
| HorCutCenter Double | 3-38 |
| HorCutWidth Double | 3-38 |
| PctCutWidth Double | 3-38 |
| VerCutCenter Double | 3-38 |
| VerCutWidth Double | 3-38 |
| PTRACE MEAN APP.MATH.FX.OPERATORYSETUP (OPERATOR = |
| "PERSISTENCETRACEMEAN") | 3-39 |
| ClearSweeps Action | 3-39 |
| PTRACE RANGE APP.MATH.FX.OPERATORYSETUP (OPERATOR = |
| "PERSISTENCETRACERANGE") | 3-40 |
| ClearSweeps Action | 3-40 |
AUTOMATION MANUAL

PCTPOPULATION Double .......................... 3-40

PTRACESIGMA APP.MATH.FX.OPERATORYSETUP (OPERATOR = "PTANCESIGMA") ................................................................. 3-41
  ClearSweeps Action .......................................................... 3-41
  Sigma Double .................................................................. 3-41

RESCALE APP.MATH.FX.OPERATORYSETUP (OPERATOR = "RESCALE") ..... 3-42
  Adder Double .................................................................. 3-42
  CustomUnit Bool ................................................................ 3-42
  Multiplier Double ............................................................ 3-42
  Unit String ................................................................. 3-43

ROOF APP.MATH.FX.OPERATORYSETUP (OPERATOR = "ROOF") ........ 3-44
  ClearSweeps Action .......................................................... 3-44
  Sweeps Integer .............................................................. 3-44

SEGMENT APP.MATH.FX.OPERATORYSETUP (OPERATOR = "SEGMENTSELECT") ................................................................. 3-45
  SelectedSegment Integer .................................................. 3-45

SPARSE APP.MATH.FX.OPERATORYSETUP (OPERATOR = "SPARSE") ......... 3-46
  SparsingFactor Integer .................................................... 3-46
  SparsingPhase Integer ..................................................... 3-46

TRACK APP.MATH.FX.OPERATORYSETUP (OPERATOR = "TRACK") ......... 3-47
  AutoFindScale Bool.......................................................... 3-47
  Center Double ............................................................... 3-47
  FindScale Action ............................................................ 3-47
  VerScale DoubleLockstep .................................................. 3-47

TREND APP.MATH.FX.OPERATORYSETUP (OPERATOR = "TREND") ........ 3-48
  AutoFindScale Bool.......................................................... 3-48
  Center Double ............................................................... 3-48
  ClearSweeps Action .......................................................... 3-48
  FindScale Action ............................................................ 3-48
  Values Integer .............................................................. 3-48
  VerScale DoubleLockstep .................................................. 3-48

MATHCADPARAMARITH APP.MEASURE.FX.OPERATOR (ARITHENGINE = "MATHCADPARAMARITH") .............................................. 3-49
  Advanced Bool .............................................................. 3-49
  NewSheet Bool .............................................................. 3-49
  OutputEnable Bool .......................................................... 3-50
  OutputHeaderVar String .................................................. 3-50
  OutputVar String ........................................................... 3-50
  Reload Action ............................................................... 3-51
  Source1Enable Bool ......................................................... 3-51
  Source1HeaderVar String .................................................. 3-51
  Source1Var String .......................................................... 3-51
  Source2Enable Bool ......................................................... 3-52
  Source2HeaderVar String .................................................. 3-52
  Source2Var String .......................................................... 3-52
  Status String .............................................................. 3-53
  WithHeader Bool ............................................................ 3-53
  WorksheetFilename FileName ............................................. 3-54
<table>
<thead>
<tr>
<th>Table of Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>P INVERT APP.MEASURE.PX.OPERATOR (ARITHENGINE = &quot;PARAMINVERT&quot;) .........................................................</td>
</tr>
<tr>
<td>CycleForTimeUnits Bool .................................................</td>
</tr>
<tr>
<td>P SCRIPT APP.MEASURE.PX.OPERATOR (ARITHENGINE = &quot;PARAMSCRIPT&quot;) ..........................................................</td>
</tr>
<tr>
<td>Code String .....................................................................</td>
</tr>
<tr>
<td>Language Enum ...........................................................</td>
</tr>
<tr>
<td>Status String ...................................................................</td>
</tr>
<tr>
<td>Timeout Double ..................................................................</td>
</tr>
<tr>
<td>Q APP.MEASURE.PX.OPERATOR (PARAMENGINE = &quot;EYEQ&quot;) .................................................................................</td>
</tr>
<tr>
<td>PctCutWidth Double .....................................................</td>
</tr>
<tr>
<td>AREA APP.MEASURE.PX.OPERATOR (PARAMENGINE = &quot;AREA&quot;) ............................................................................</td>
</tr>
<tr>
<td>Cyclic Bool .........................................................................</td>
</tr>
<tr>
<td>AVG POWER APP.MEASURE.PX.OPERATOR (PARAMENGINE = &quot;EYEAVGPOWER&quot;) ..................................................</td>
</tr>
<tr>
<td>PctCutWidth Double .....................................................</td>
</tr>
<tr>
<td>DPERIOD@LEVEL APP.MEASURE.PX.OPERATOR (PARAMENGINE = &quot;DELTAPERIODATLEVEL&quot;) ..................................</td>
</tr>
<tr>
<td>AbsLevel Double ..................................................................</td>
</tr>
<tr>
<td>BaseFrequency Double .....................................................</td>
</tr>
<tr>
<td>FindBaseFrequency Action ...............................................</td>
</tr>
<tr>
<td>FindLevel Action ............................................................</td>
</tr>
<tr>
<td>GroupSize Integer ..........................................................</td>
</tr>
<tr>
<td>Hysteresis Double ..........................................................</td>
</tr>
<tr>
<td>LevelType Enum ..................................................................</td>
</tr>
<tr>
<td>PercentLevel Double ......................................................</td>
</tr>
<tr>
<td>SignalType Enum ..................................................................</td>
</tr>
<tr>
<td>Slope Enum ...........................................................................</td>
</tr>
<tr>
<td>StartCycle Integer ..........................................................</td>
</tr>
<tr>
<td>StdBaseFrequency Enum ....................................................</td>
</tr>
<tr>
<td>UseBaseFrequency Enum ....................................................</td>
</tr>
<tr>
<td>DTIME@LEVEL APP.MEASURE.PX.OPERATOR (PARAMENGINE = &quot;DELTATIMEATLEVEL&quot;) ....................................</td>
</tr>
<tr>
<td>AbsLevel1 Double .............................................................</td>
</tr>
<tr>
<td>AbsLevel2 Double .............................................................</td>
</tr>
<tr>
<td>FindLevel1 Action .............................................................</td>
</tr>
<tr>
<td>FindLevel2 Action .............................................................</td>
</tr>
<tr>
<td>Hysteresis1 Double ...........................................................</td>
</tr>
<tr>
<td>Hysteresis2 Double ...........................................................</td>
</tr>
<tr>
<td>LevelType1 Enum ..............................................................</td>
</tr>
<tr>
<td>LevelType2 Enum ..............................................................</td>
</tr>
<tr>
<td>PercentLevel1 Double .......................................................</td>
</tr>
<tr>
<td>PercentLevel2 Double .......................................................</td>
</tr>
<tr>
<td>Slope1 Enum .................................................................</td>
</tr>
<tr>
<td>Slope2 Enum .................................................................</td>
</tr>
<tr>
<td>DUTY@LEVEL APP.MEASURE.PX.OPERATOR (PARAMENGINE = &quot;DUTYATLEVEL&quot;) ...............................................</td>
</tr>
<tr>
<td>AbsLevel Double .............................................................</td>
</tr>
<tr>
<td>FindLevel Action .............................................................</td>
</tr>
<tr>
<td>Hysteresis Double ...........................................................</td>
</tr>
</tbody>
</table>
EYE AMPLITUDE APP.MEASURE.PX.OPERATOR (PARAMENGINE = "EYEAAMPLITUDE") .......................................................... 3-84
PctCutWidth Double .......................................................... 3-84

EYE BER APP.MEASURE.PX.OPERATOR (PARAMENGINE = "EYE BER") .......................................................... 3-85
PctCutWidth Double .......................................................... 3-85
## Table of Contents

**Eye Height APP.MEASURE.PX.OPERATOR (PARAMENGINE = "EYEHEIGHT")** ........................................................................................................... 3-86
- CalcUnits Enum ................................................................................. 3-86
- PctCutWidth Double ......................................................................... 3-86

**FALL@LEVEL APP.MEASURE.PX.OPERATOR (PARAMENGINE = "FALLATLEVEL")** .............................................................................. 3-87
- HighAbs Double ................................................................................ 3-87
- HighPct Double ................................................................................ 3-87
- LevelsAre Enum ................................................................................ 3-88
- LowAbs Double ................................................................................ 3-88
- LowPct Double ................................................................................ 3-88

**FREQ@LEVEL APP.MEASURE.PX.OPERATOR (PARAMENGINE = "FREQUENCYATLEVEL")** ........................................................................ 3-89
- AbsLevel Double .............................................................................. 3-89
- BaseFrequency Double .................................................................... 3-89
- FindBaseFrequency Action .................................................................. 3-90
- FindLevel Action .............................................................................. 3-90
- Hysteresis Double ............................................................................ 3-90
- LevelType Enum ............................................................................... 3-90
- PercentLevel Double ....................................................................... 3-90
- SignalType Enum ............................................................................. 3-91
- Slope Enum ........................................................................................ 3-91
- StdBaseFrequency Enum ................................................................... 3-91
- UseBaseFrequency Enum .................................................................. 3-92

**FWXX APP.MEASURE.PX.OPERATOR (PARAMENGINE = "FULLWIDTHATXX")** .................................................................................. 3-93
- HFractionHt Double .......................................................................... 3-93

**Half Period APP.MEASURE.PX.OPERATOR (PARAMENGINE = "HALFPERIOD")** .............................................................................. 3-94
- AbsLevel Double .............................................................................. 3-94
- FindLevel Action ............................................................................... 3-94
- Hysteresis Double ............................................................................ 3-94
- LevelType Enum ............................................................................... 3-94
- PercentLevel Double ....................................................................... 3-94
- Slope Enum ........................................................................................ 3-94

**HOLD TIME APP.MEASURE.PX.OPERATOR (PARAMENGINE = "HOLDTIME")** ...................................................................................... 3-96
- ClockAbsLevel Double ..................................................................... 3-96
- ClockFindLevel Action ..................................................................... 3-96
- ClockHysteresis Double ................................................................... 3-97
- ClockLevels Enum ........................................................................... 3-97
- ClockPctLevel Double ..................................................................... 3-98
- ClockSlope Enum ............................................................................. 3-98
- CursorDisplay Enum ........................................................................ 3-98
- DataAbsLevel Double ...................................................................... 3-99
- DataFindLevel Action ...................................................................... 3-99
- DataHysteresis Double ................................................................... 3-100
- DataLevels Enum ............................................................................ 3-100
- DataPctLevel Double ..................................................................... 3-100
- DataSlope Enum ............................................................................. 3-101
AUTOMATION MANUAL

LEVEL@X_APP.MEASURE.PX.OPERATOR (PARAMENGINE = "LEVELATX") ......................................................................................................................... 3-102
  CursorShape Enum ................................................................. 3-102
  HorValue Double .................................................................. 3-102
  LevelCursor Bool ................................................................. 3-103
  Marker Bool ......................................................................... 3-103
  PinToData Bool .................................................................... 3-103
  TimeCursor Bool .................................................................. 3-104

MATHCADPARAM_APP.MEASURE.PX.OPERATOR (PARAMENGINE = "MATHCADPARAM") ........................................................................................................... 3-105
  Advanced Bool ..................................................................... 3-105
  NewSheet Bool ..................................................................... 3-105
  OutputEnable Bool ............................................................... 3-105
  OutputHeaderVar String ....................................................... 3-105
  OutputVar String ................................................................. 3-105
  Reload Action ..................................................................... 3-106
  Source1Enable Bool .............................................................. 3-106
  Source1HeaderVar String ..................................................... 3-106
  Source1Var String ............................................................... 3-106
  Source2Enable Bool .............................................................. 3-106
  Source2HeaderVar String ..................................................... 3-106
  Source2Var String ............................................................... 3-106
  Status String ..................................................................... 3-107
  WithHeader Bool .................................................................. 3-107
  WorksheetFilename FileName ................................................... 3-107

MATLABPARAM_APP.MEASURE.PX.OPERATOR (PARAMENGINE = "MATLABPARAM") ........................................................................................................... 3-108
  MATLABCode String ............................................................ 3-108
  MATLABPlot Bool ............................................................... 3-108
  MATLABScalePerDiv Double ................................................... 3-108
  MATLABZeroOffset Double .................................................... 3-108

MAXIMUM_APP.MEASURE.PX.OPERATOR (PARAMENGINE = "MAXIMUM") .................................................................................................................... 3-109

MEAN_APP.MEASURE.PX.OPERATOR (PARAMENGINE = "MEAN") .......................................................................................................................... 3-110
  Cyclic Bool ..................................................................... 3-110

MEDIAN_APP.MEASURE.PX.OPERATOR (PARAMENGINE = "MEDIAN") .................................................................................................................... 3-111
  Cyclic Bool ..................................................................... 3-111

MINIMUM_APP.MEASURE.PX.OPERATOR (PARAMENGINE = "MINIMUM") .................................................................................................................... 3-112

NB PHASE_APP.MEASURE.PX.OPERATOR (PARAMENGINE = "NARROWBANDPHASE") .................................................................................................... 3-113
  Frequency Double ............................................................. 3-113

NB POWER_APP.MEASURE.PX.OPERATOR (PARAMENGINE = "NARROWBANDPOWER") .............................................................................................. 3-114
  Frequency Double ............................................................. 3-114

NPTS_APP.MEASURE.PX.OPERATOR (PARAMENGINE = "NPOINTS") ......................................................................................................................... 3-115
  UsePointsInFrame Bool ...................................................... 3-115
ONE LEVEL  APP.MEASURE.PX.OPERATOR (PARAMENGINE = "EYONELEVEL") .................................................................................................................. 3-116
PctCutWidth Double .................................................................................................................. 3-116

PARAM SCRIPT  APP.MEASURE.PX.OPERATOR (PARAMENGINE = "PARAMSCRIPT") .................................................................................................................. 3-117
Code String .................................................................................................................. 3-117
Language Enum .............................................................................................................. 3-118
Status String .................................................................................................................. 3-118
Timeout Double .............................................................................................................. 3-118

PERCENTILE  APP.MEASURE.PX.OPERATOR (PARAMENGINE = "PERCENTILE") .................................................................................................................. 3-119
HPctPop Double .................................................................................................................. 3-119

PERIOD@LEVEL  APP.MEASURE.PX.OPERATOR (PARAMENGINE = "PERIODATLEVEL") .................................................................................................................. 3-120
AbsLevel Double .............................................................................................................. 3-120
BaseFrequency Double .................................................................................................. 3-120
FindBaseFrequency Action .......................................................................................... 3-120
FindLevel Action .......................................................................................................... 3-121
Hysteresis Double ........................................................................................................... 3-121
LevelType Enum ............................................................................................................ 3-121
PercentLevel Double .................................................................................................... 3-121
SignalType Enum ........................................................................................................... 3-121
Slope Enum ................................................................................................................... 3-122
StdBaseFrequency Enum .............................................................................................. 3-122
UseBaseFrequency Enum ............................................................................................ 3-122

PHASE  APP.MEASURE.PX.OPERATOR (PARAMENGINE = "PHASE") .................................................................................................................. 3-123
OutputType Enum ............................................................................................................ 3-123
RefAbsLevel Double ...................................................................................................... 3-123
RefFindLevel Action ...................................................................................................... 3-124
RefHysteresis Double ...................................................................................................... 3-124
RefLevelType Enum ......................................................................................................... 3-124
RefPercentLevel Double ................................................................................................. 3-125
RefSlope Enum .............................................................................................................. 3-125
SigAbsLevel Double ...................................................................................................... 3-125
SigFindLevel Action ...................................................................................................... 3-125
SigHysteresis Double ...................................................................................................... 3-125
SigLevelType Enum ......................................................................................................... 3-126
SigPercentLevel Double ................................................................................................. 3-126
SigSlope Enum .............................................................................................................. 3-126

RISE@LEVEL  APP.MEASURE.PX.OPERATOR (PARAMENGINE = "RISEATLEVEL") .................................................................................................................. 3-127
HighAbs Double .............................................................................................................. 3-127
HighPct Double .............................................................................................................. 3-127
LevelsAre Enum .............................................................................................................. 3-127
LowAbs Double .............................................................................................................. 3-127
LowPct Double .............................................................................................................. 3-128

RMS  APP.MEASURE.PX.OPERATOR (PARAMENGINE = "ROOTMEAN_SQUARE") .................................................................................................................. 3-129
Cyclic Bool ................................................................................................................... 3-129
**Automation Manual**

**Setup**  
APP.MEASURE.PX.OPERATOR (PARAMENGINE = "SETUP") ..........3-130

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClockAbsLevel</td>
<td>Double</td>
<td>Represents the absolute level of the clock.</td>
</tr>
<tr>
<td>ClockFindLevel</td>
<td>Action</td>
<td>Determines when to find the level.</td>
</tr>
<tr>
<td>ClockHysteresis</td>
<td>Double</td>
<td>Specifies the hysteresis for the clock.</td>
</tr>
<tr>
<td>ClockLevels</td>
<td>Enum</td>
<td>Defines the number of levels for the clock.</td>
</tr>
<tr>
<td>ClockPctLevel</td>
<td>Double</td>
<td>Indicates the percentage level of the clock.</td>
</tr>
<tr>
<td>ClockSlope</td>
<td>Enum</td>
<td>Specifies the slope for the clock.</td>
</tr>
<tr>
<td>CursorDisplay</td>
<td>Enum</td>
<td>Controls the display of the cursor.</td>
</tr>
<tr>
<td>DataAbsLevel</td>
<td>Double</td>
<td>Represents the absolute level of the data.</td>
</tr>
<tr>
<td>DataFindLevel</td>
<td>Action</td>
<td>Determines when to find the data level.</td>
</tr>
<tr>
<td>DataHysteresis</td>
<td>Double</td>
<td>Specifies the hysteresis for the data.</td>
</tr>
<tr>
<td>DataLevels</td>
<td>Enum</td>
<td>Defines the number of levels for the data.</td>
</tr>
<tr>
<td>DataPctLevel</td>
<td>Double</td>
<td>Indicates the percentage level of the data.</td>
</tr>
<tr>
<td>DataSlope</td>
<td>Enum</td>
<td>Specifies the slope for the data.</td>
</tr>
<tr>
<td>Summary</td>
<td>String</td>
<td>Provides a summary of the measurement.</td>
</tr>
</tbody>
</table>

**Skew**  
APP.MEASURE.PX.OPERATOR (PARAMENGINE = "SKEW") ..........3-133

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock1AbsLevel</td>
<td>Double</td>
<td>Represents the absolute level of the clock 1.</td>
</tr>
<tr>
<td>Clock1FindLevel</td>
<td>Action</td>
<td>Determines when to find the level of clock 1.</td>
</tr>
<tr>
<td>Clock1Hysteresis</td>
<td>Double</td>
<td>Specifies the hysteresis for clock 1.</td>
</tr>
<tr>
<td>Clock1Levels</td>
<td>Enum</td>
<td>Defines the number of levels for clock 1.</td>
</tr>
<tr>
<td>Clock1PctLevel</td>
<td>Double</td>
<td>Indicates the percentage level of clock 1.</td>
</tr>
<tr>
<td>Clock1Slope</td>
<td>Enum</td>
<td>Specifies the slope for clock 1.</td>
</tr>
<tr>
<td>Clock2AbsLevel</td>
<td>Double</td>
<td>Represents the absolute level of clock 2.</td>
</tr>
<tr>
<td>Clock2FindLevel</td>
<td>Action</td>
<td>Determines when to find the level of clock 2.</td>
</tr>
<tr>
<td>Clock2Hysteresis</td>
<td>Double</td>
<td>Specifies the hysteresis for clock 2.</td>
</tr>
<tr>
<td>Clock2Levels</td>
<td>Enum</td>
<td>Defines the number of levels for clock 2.</td>
</tr>
<tr>
<td>Clock2PctLevel</td>
<td>Double</td>
<td>Indicates the percentage level of clock 2.</td>
</tr>
<tr>
<td>Clock2Slope</td>
<td>Enum</td>
<td>Specifies the slope for clock 2.</td>
</tr>
</tbody>
</table>

**Std Dev**  
APP.MEASURE.PX.OPERATOR (PARAMENGINE = "STANDARDDEVIATION") ..........3-136

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td>Bool</td>
<td>Indicates if measurement is cyclic.</td>
</tr>
</tbody>
</table>

**Tie@Level**  
APP.MEASURE.PX.OPERATOR (PARAMENGINE = "TIE") ..........3-137

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsLevel</td>
<td>Double</td>
<td>Represents the absolute level.</td>
</tr>
<tr>
<td>BaseFrequency</td>
<td>Double</td>
<td>Specifies the base frequency.</td>
</tr>
<tr>
<td>CutOffDivisor</td>
<td>Double</td>
<td>Defines the cut-off divisor.</td>
</tr>
<tr>
<td>DataIsNRZ</td>
<td>Bool</td>
<td>Indicates if the data is Non Return to Zero.</td>
</tr>
<tr>
<td>FindBaseFrequency</td>
<td>Action</td>
<td>Determines when to find the base frequency.</td>
</tr>
<tr>
<td>LevelType</td>
<td>Enum</td>
<td>Specifies the level type.</td>
</tr>
<tr>
<td>PercentLevel</td>
<td>Double</td>
<td>Indicates the percentage level.</td>
</tr>
<tr>
<td>ResultScaling</td>
<td>Enum</td>
<td>Specifies the result scaling.</td>
</tr>
<tr>
<td>SignalType</td>
<td>Enum</td>
<td>Defines the signal type.</td>
</tr>
<tr>
<td>Slope</td>
<td>Enum</td>
<td>Specifies the slope.</td>
</tr>
<tr>
<td>StdBaseFrequency</td>
<td>Enum</td>
<td>Specifies the standard base frequency.</td>
</tr>
<tr>
<td>UseBaseFrequency</td>
<td>Enum</td>
<td>Indicates if the base frequency is used.</td>
</tr>
<tr>
<td>UseGoldenPLL</td>
<td>Bool</td>
<td>Indicates if Golden PLL is used.</td>
</tr>
</tbody>
</table>

**Time@Level**  
APP.MEASURE.PX.OPERATOR (PARAMENGINE = "TIMEATELEVEL") ..........3-141

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsLevel</td>
<td>Double</td>
<td>Represents the absolute level.</td>
</tr>
<tr>
<td>FindLevel</td>
<td>Action</td>
<td>Determines when to find the level.</td>
</tr>
<tr>
<td>Hysteresis</td>
<td>Double</td>
<td>Specifies the hysteresis.</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>HysteresisPct Double</td>
<td>3-141</td>
<td></td>
</tr>
<tr>
<td>HysteresisType Enum</td>
<td>3-141</td>
<td></td>
</tr>
<tr>
<td>LevelType Enum</td>
<td>3-141</td>
<td></td>
</tr>
<tr>
<td>PercentLevel Double</td>
<td>3-142</td>
<td></td>
</tr>
<tr>
<td>Slope Enum</td>
<td>3-142</td>
<td></td>
</tr>
<tr>
<td>WIDTH@LEVEL APP.MEASURE.PX.OPERATOR (PARAMENGINE = &quot;WIDTHATLEVEL&quot;)</td>
<td>3-143</td>
<td></td>
</tr>
<tr>
<td>Absolute Level Double</td>
<td>3-143</td>
<td></td>
</tr>
<tr>
<td>FindLevel Action</td>
<td>3-143</td>
<td></td>
</tr>
<tr>
<td>FirstWhenBoth Enum</td>
<td>3-143</td>
<td></td>
</tr>
<tr>
<td>Hysteresis Double</td>
<td>3-143</td>
<td></td>
</tr>
<tr>
<td>HysteresisPct Double</td>
<td>3-143</td>
<td></td>
</tr>
<tr>
<td>HysteresisType Enum</td>
<td>3-143</td>
<td></td>
</tr>
<tr>
<td>LevelType Enum</td>
<td>3-143</td>
<td></td>
</tr>
<tr>
<td>PercentLevel Double</td>
<td>3-143</td>
<td></td>
</tr>
<tr>
<td>Slope Enum</td>
<td>3-143</td>
<td></td>
</tr>
<tr>
<td>X AT PEAK APP.MEASURE.PX.OPERATOR (PARAMENGINE = &quot;XATPEAK&quot;)</td>
<td>3-145</td>
<td></td>
</tr>
<tr>
<td>PeakNumber Integer</td>
<td>3-145</td>
<td></td>
</tr>
<tr>
<td>ZERO LEVEL APP.MEASURE.PX.OPERATOR (PARAMENGINE = &quot;EYEZEROLEVEL&quot;)</td>
<td>3-146</td>
<td></td>
</tr>
<tr>
<td>PctCutWidth Double</td>
<td>3-146</td>
<td></td>
</tr>
</tbody>
</table>
About this Manual

This manual is a reference guide to the “Automation” capabilities of LeCroy’s XStream™ DSOs. The manual includes a complete list of all instrument controls that are available to a controlling application. In contrast to previously available remote control possibilities for LeCroy instruments, Automation enables the controlling application to run on the instrument itself.

Part One, “About Automation,” covers the principles of control via Automation and offers practical examples. It also presents the overall hierarchy of controls accessible via Automation.

Part Two, “Control Reference,” presents each of the controls in detail, including examples of use in many cases.

Part Three, “Math/Measure Control Reference,” presents controls available in each of the multitude of the Math/Measure ‘processors’ available in XStream DSOs.
ABOUT AUTOMATION

This section presents an overview of the Automation capabilities of X-Stream DSOs along with an overview of the technology itself.
Chapter One: Overview

In this chapter learn about

- Microsoft’s COM-based “OLE Automation”
- How to create simple applications in Visual Basic to control the instrument
- How to use the X-Stream Browser to interactively control the instrument
CHAPTER ONE

OVERVIEW OF AUTOMATION

In addition to supporting the familiar ASCII-based remote commands that have been used to control all LeCroy DSOs for many years, all of the Windows-based “X-Stream” instruments fully support control by Automation interfaces based on Microsoft’s Component Object Model (COM). Using COM, the controlling application can run directly on the instrument without requiring an external controller; or, alternatively, it can run using Microsoft’s distributed COM standard (DCOM) on a networked computer.

Standards

Automation is a Microsoft technology, formerly referred to as “OLE Automation,” that has primarily been used to enable cross-application macro programming. It is based upon the Component Object Model (COM), which is similar in nature to CORBA, more commonly found in the UNIX world.

An application that “exposes Automation Objects” is referred to as an “Automation Server.” Automation Objects expose “Automation Interfaces” to the controlling “Automation Client.” This manual describes these Automation objects and interfaces in detail.

It is important to note that Automation itself is not language dependent; it can be used from any programming language that supports COM. This manual, however, concentrates mainly on the use of Automation from the Visual Basic Script (VBScript) language for several reasons, including the fact that it is one of the easiest to use. Also, it is the language that X-Stream instruments use for setup files (more on this later). In addition, the VBScript interpreter is installed by default on all X-Stream instruments and, therefore, is available without your having to purchase any additional software.

Compatibility with Other LeCroy Scopes

Throughout LeCroy’s history, we have striven to maximize compatibility, and this policy remains in force. However, due to the fact that the underlying technologies used by Automation require the 32-bit Windows operating system, and that this system is available only on our X-Stream instruments, Automation is not available on the older LeCroy scope families.

Automation and IEEE 488.2 Remote Control – How Do They Compare?

Automation does not replace the IEEE 488.2-based remote command set, which is also supported by X-Stream instruments (and will continue to be). Instead, it augments it and allows a new class of application to be created that can run on the DSO itself.

Automation however can be considered as the “Native Language,” or “Mother Tongue” of X-Stream instruments. All of the instrument’s controls and features are available to the Automation Client.

Automation controls generally are more granular than 488.2 remote commands. That is, many 488.2 remote commands set more than one control at the same time; whereas, via Automation, this is not the case.
The following table summarizes the differences between the two remote control possibilities:

<table>
<thead>
<tr>
<th></th>
<th>IEEE 488.2 Control</th>
<th>Automation Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Transport</td>
<td>GPIB, TCP/IP over Ethernet</td>
<td>Inter-process using COM, inter-PC using DCOM (TCP/IP)</td>
</tr>
<tr>
<td>Textual parsing of instrument responses required</td>
<td>Yes. All instrument responses need ‘parsing’ to extract useful information.</td>
<td>No. Each element in the Automation hierarchy appears as a “variable” to the Automation client.</td>
</tr>
<tr>
<td>Compatibility with previous LeCroy DSOs</td>
<td>Very good. In most cases remote control applications written for older DSOs will work without modification.</td>
<td>None. Automation is a new standard first introduced with LeCroy’s X-Stream DSOs.</td>
</tr>
<tr>
<td>Ability to run controlling application “in the box”</td>
<td>Yes, by using the TCP/IP (VICP) protocol to talk to the “local host”</td>
<td>Yes, natively</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>Not trivial. It’s easier using a tool such as ActiveDSO that hides some of the complexities.</td>
<td>Very easy to use from scripting languages and office productivity tools</td>
</tr>
<tr>
<td>Format of Waveform results</td>
<td>Binary or ASCII. Both require parsing before use.</td>
<td>Waveforms are presented as arrays of floating point values.</td>
</tr>
<tr>
<td>Control from MS Office suite</td>
<td>Possible via ActiveDSO utility</td>
<td>Yes, natively (see examples later in this manual)</td>
</tr>
</tbody>
</table>

**General Characteristics**

- When an application is running locally on the instrument and requests a connection to the DSO via Automation (for example, by using `CreateObject(“LeCroy.XStreamDSO”)` from Visual Basic), one of two things will happen. If the X-Stream DSO application is already running, the object returned will be a “pointer” to the running application. If the DSO application is not running, it will be started. It is not possible to run two simultaneous instances of the X-Stream DSO application.

- More than one simultaneous connection to the instrument via Automation will be accepted, but simultaneous connections are not recommended.

- When the final client has been disconnected from the instrument (server), the X-Stream DSO application will remain running and will accept further client connections.

- Operations that cause Modal Dialogs to appear in the instrument’s display will, by default, disrupt access from Automation. This behavior can be changed using the `app.SystemControl.ModalDialogTimeout` and `app.SystemControl.EnableMessageBox` controls. Refer to the description of each of these controls in the reference section for more information.

- The instrument application can be minimized in order to allow the controlling application to take over the display and touch panel by means of the `app.Minimize` control. It can also be resized and repositioned on the display by means of the `app.Top`, `app.Left`, `app.Bottom`, `app.Right` controls.
Chapter One: Overview

INTRODUCTION TO THE X-STREAM BROWSER

The easiest way to get up and running with Automation, and also to visualize the “X-Stream Object Model” is to use the X-Stream browser tool, which is pre-installed on all instruments.

To launch the tool, first minimize the instrument application (File -> Minimize), then double-click on the X-Stream Browser icon on the desktop:

To connect to the running DSO application, click the Connect toolbar icon:

Upon connection, the root of the object hierarchy is shown in a layout similar to that presented when a file system is browsed using Windows Explorer.

As a quick demonstration of how the X-Stream Browser can be used, open the Acquisition folder, then click on the C1 folder. Find the VerScale (Volts/Div) control in the right-hand window and right-click on it, then select the Set Value menu option.

Enter a new value for VerScale and click the Set this value button. Restore the DSO’s X-Stream window and note that the V/Div value for C1 should have changed to reflect the entered value.
This section of the manual presents a walk-through of how to create a simple remote control application, which will run on the instrument, from scratch. It doesn’t rely upon any 3rd-party development tools, since it uses Windows’ built-in text editor (Notepad) and the Visual Basic Script interpreter (VBScript), which is also installed on all instruments.

1. Use the File? Windowed menu option to place the DSO application into windowed mode. This allows the windows start-bar to be accessed.

2. Open Windows Notepad via Start->Programs->Accessories->Notepad

3. Write the following text into the editor:

```vbs
Set app = CreateObject("LeCroy.XStreamDSO")
app.AutoSetup
app.Display.GridMode = "Quad"
myVerScale = app.Acquisition.C1.VerScale
MsgBox myVerScale
```

4. Save the file to drive D:\ and name it Exercise1.vbs. Leave Notepad open, we’ll need it again.


6. Navigate to drive D:\ and double-click on Exercise1.vbs.

7. That’s it. If these steps were followed correctly, you should hear relays clicking while the scope performs an auto-setup operation and enters its quad-grid display mode.

So, what did this “program” actually do?

- The `CreateObject` statement.

```vbs
Set app = CreateObject("LeCroy.XStreamDSO")
```

CreateObject is the Visual Basic function that creates an instance of a COM Server (a.k.a. ActiveX Control). The argument “LeCroy.XStreamDSO” refers to our DSO application. Once it has instantiated (connected to) our DSO application we need some kind of ‘handle’ (pointer) to it so that we can use it later to communicate with the instrument. **CreateObject** returns a handle to us, which we store in the **app** variable.

**NOTE:** Only a single instance of the X-Stream DSO software can run on a system at one time. If the DSO software is already running when CreateObject is called, a handle to that running instance is returned. If the DSO software is not running, it will be started.

- The `app.AutoSetup` statement.

```vbs
app.AutoSetup
```

Using the **app** handle, this line of code calls the **AutoSetup** method, which performs the same task as the front-panel Auto-Setup button. Documentation for this method can be found later in the reference section.

- The `app.Display.GridMode = "Quad"` statement.

```vbs
app.Display.GridMode = "Quad"
```
Using the **app** handle, this line of code sets the **GridMode** control of the **Display** system to the value “Quad”. It’s important to note that the controls are arranged in a hierarchy, with each ‘level’ of the hierarchy delimited with a decimal point (\.).

- The **myVerScale = app.Acquisition.C1.VerScale** statement.

  ```vb
  myVerScale = app.Acquisition.C1.VerScale
  ```

  Instead of setting the value of a control, this line of code retrieves the current value of a control, in this case the Vertical Scale (Volts/Div) of Channel 1. The value returned is stored within the variable **myVerScale**.

  **NOTE:** In Visual Basic Script it is not necessary to “Dimension” variables before using them (for example, using statements like “Dim myVerScale as Double”).

- The **MsgBox myVerScale** statement

  ```vb
  MsgBox myVerScale
  ```

  This line of code does not communicate with the scope at all, but calls the standard Visual Basic Script function **MsgBox**. This function displays a dialog containing the value of the variable following “MsgBox”. In our case the value of Channel 1’s vertical scale, and waits for the **OK** button to be clicked.

  Documentation about the **MsgBox** function can be found in Microsoft’s Visual Basic Scripting documentation at www.microsoft.com\scripting.

Another point that should be mentioned here is something that is used extensively in Setup files created by the instrument: the ability to use “abbreviations” to simplify programs. Following is an example in which a shorthand method is used to replace some rather long-winded code. It is also important to note that this also enhances performance. For example, the “lookup” of the object **app.Acquisition.C1** occurs only once in the modified code, but three times in the original code.

Instead of:

```vb
app.Acquisition.C1.VerScale = 0.5
app.Acquisition.C1.VerOffset = 0.1
app.Acquisition.C1.Coupling = “DC50”
```

The following may be used:

```vb
set myChannel1 = app.Acquisition.C1
myChannel1.VerScale = 0.5
myChannel1.VerOffset = 0.1
myChannel1.Coupling = “DC50”
```
Automation is used in several places in the X-Stream based instrument.

- Instrument Setups (Panel Files)
- Custom Math/Measurements
- CustomDSO, User Interface customization
- Control from external applications (COM/DCOM)

Each of these uses is described in more detail in the following sections.
SETUPS (PANEL FILES) ARE PROGRAMS!

Setup files, used to save and recall the state of the instrument between runs, are traditionally binary files whose internal structure is neither documented nor obvious to the user.

In X-Stream DSOs, however, this is no longer the case. Setups are ASCII text files that contain a complete Visual Basic Script “program” that, when “executed,” will restore the instrument to a previously recorded state. In effect, each time a panel is saved, the instrument effectively writes you a program that, when executed, returns the instrument to the saved state.

Due to the fact that these setups are already programs, they are a great way to get started quickly with Automation. As an example, try saving a setup into a file and examine it using a text-editor, as follows:

1. Touch **File** in the menu bar, then **Save Setup** from the drop-down menu.
2. Touch the **Browse** button and specify drive D: as the location to save the .lss (LeCroy setup script) file.
3. Touch **Save Now!**
4. Minimize the application using the **Minimize** option from the **File** menu
5. Open Microsoft’s Notepad application from the **Accessories** program folder (Start? Programs? Accessories).
6. Open the file saved above. You will see a Visual Basic Script program that begins like this:

   ```vbscript
   ' XStreamDSO ConfigurationVBScript ...
   ' LECROY,WM8300,WM000001,0.0.0
   ' Thursday, February 20, 2003 11:26:55 AM
   On Error Resume Next
   set XStreamDSO = CreateObject("LeCroy.XStreamDSO")
   XStreamDSO.RecallingSetup = True
   ' AladdinPersona ...
   XStreamDSO.HideClock = False
   XStreamDSO.TouchScreenEnable = True
   ...
   ```

Since the entire state of the instrument, including all controls for all installed software options, is saved, this panel may look fairly complex. But don’t let this fool you; the basic concept is, in fact, fairly simple.

As a quick example of how setups can be used as the starting point for controlling applications, scroll down to the end of the file and add the following code (shown in bold-type) to the file.

```vbscript
' Place any custom VBScript code after this point

' Perform an Auto Setup
XStreamDSO.AutoSetup
```

When this setup is recalled, the complete state of the instrument will be restored, followed by an Auto-Setup operation.

Obviously this is a fairly trivial “application,” but it is easy to imagine how automated testing could be performed with the introduction of loops and conditional execution.
NOTE: Setup files stored by the instrument have file extension “.lss” (LeCroy Setup Script). These files are syntactically identical to Microsoft Visual Basic Script (VBScript) files, which have a “.vbs” extension.

TIP: A simple alternative to recalling the panel into the instrument is to execute it, either by double-clicking on the .lss file in Windows Explorer, or by executing it from the command line.
CUSTOM MATH AND MEASUREMENTS

Custom Math and Measurements can be coded using VBScript or JavaScript in instruments equipped with the XDEV and/or XMAP software options. Using Automation control of the instrument, decisions can be made during custom processing that reconfigure the DSO.

When you are developing custom processing routines using the reference section of this manual, `app.Acquisition.Cx.Out.Result` may be used as a comprehensive reference to the Result Object, which is used to describe waveform data (InResult, InResult1, InResult2, OutResult).

For more detail about this capability, see the “Customization” section of the on-line Help Manual.

CustomDSO

CustomDSO enables customization of the instrument’s UI in instruments equipped with the XDEV and/or XMAP options. Two modes of operation are supported: Basic mode and Plug-in mode.

In Basic mode a Visual Basic Script (VBScript) program can be assigned to each of 8 buttons that can, optionally, appear at the bottom of the instrument’s display. By means of Automation, each of these may further reconfigure all 8 buttons, which would allow simple menu hierarchies to be generated.

In Plug-in mode an ActiveX control, created in any of a number of programming languages, can be inserted into the instrument’s menu system. Once “embedded,” this Plug-in can take full control of the DSO using Automation.

Full documentation on CustomDSO is available in the CustomDSO section of the on-line Help Manual.
PART ONE: ABOUT AUTOMATION

Control from External Applications

Control of an X-Stream based instrument by Automation is possible from most modern programming languages (interpreted and/or compiled), and also from the “macro” capability of office productivity suites such as Microsoft Office.
From Visual Basic

From Visual Basic the **CreateObject** method is used to create the connection to an instrument by Automation.

The following code example creates this connection and sets up some of the instrument’s controls:

```
' Connect to the X-Stream DSO
Dim app as Object
Set app = CreateObject("LeCroy.XStreamDSO")

' Setup Vertical and Horizontal settings
app.Acquisition.C1.VerScale = 0.5
app.Acquisition.C1.VerOffset = 0.25
app.Acquisition.Horizontal.HorScale = 0.000001

' Disconnect from the DSO
Set app = Nothing
```

From MATLAB

MATLAB uses the **actxserver** keyword to connect to the instrument.

The following code example creates this connection, enables variable vertical scale, reads the vertical scale value for C1, and increases it by a factor of 0.75.

```
DSO = actxserver('Lecroyxstreamdso')
set(DSO.Acquisition.C1.VerScaleVariable,'value','-1')
verscale = get(DSO.Acquisition.C1.VerScale,'value')
verscale = 0.75 * verscale
set(DSO.Acquisition.C1.VerScale,'value',verscale)
```

**NOTE:** Don’t confuse the control of the instrument from MATLAB (MATLAB “drives”) with the use of MATLAB from within a custom processing function (the instrument “drives”).
From MS Office (Excel)

Using Automation, control of the instrument from Microsoft Excel is very similar to control from Visual Basic. This is because the “macro language” used in the office suite is Visual Basic for Applications, a lightweight version of Visual Basic.

The following example shows how to add a button to an Excel spreadsheet that connects to, and controls, the instrument on which Excel is running. Note that this example was generated using Excel 2000, other versions of Excel support similar functionality, but the specific sequence of commands may be slightly different:

1. Enable the Forms toolbar using the **View** ? **Toolbars** ? **Forms** menu option.

2. Click the button icon on the toolbar and draw a button on the spreadsheet. The button will be labeled “Button 1” by default.

3. Right-click on the edge of the button and select Assign Macro from the drop-down menu.

4. Select the **New** button from the Assign Macro dialog; the macro editor will appear:

5. Type the following code into the subroutine:

   ```vba
   Set app = CreateObject("LeCroy.XStreamDSO")
   app.AutoSetup
   Set app = Nothing
   ```

   Clicking on the newly created button will now execute this code segment, which connects to the DSO and performs an Auto Setup.
Distributed COM (DCOM) can be used to control an X-Stream based instrument remotely over any network supporting TCP/IP (including the Internet).

For obvious security reasons all LeCroy instruments are shipped with this feature disabled, but the fairly simple procedure detailed below can be followed to enable DCOM and define network user accounts that have the access permissions to use it.

There are two basic ways to configure DCOM (User-Level or Share-Level), the choice of which really depends upon whether the client and server (controlling PC and instrument) are both members of the same NT Domain or not.

Security Settings on the Instrument (Server): User Level

This mode requires that the client and server be logged into the same NT Domain. Use the share-level configuration (below) if this is not the case.

1. Run DCOMCnfg.exe from either Start?Programs?Accessories?Command Prompt or from Start?Run…

2. Select LeCroyXStreamDSO from the list of applications.

3. Click the Properties button.

4. Click the General tab and set the Authentication Level control to Connect.

5. Click the Security tab. Select Use custom access permissions and add the list of users that are allowed to control this scope via DCOM. Select Everyone (All Users) if you wish everyone to have access.

6. Click the Identity tab and select The Interactive User.

Security Settings on the Instrument (Server): Share Level

1. Run DCOMCnfg.exe from either Start?Programs?Accessories?Command Prompt or from Start?Run…

2. Select LeCroyXStreamDSO from the list of applications.

3. Click the Properties button.

4. Click the General tab and set the Authentication Level control to (None).

5. Click the Security tab. Select Use custom access permissions and add Everyone to the list.

6. Click the Identity tab and select The interactive user.

Initialize the Controlling PC (Client)

1. Copy LeCroyXStreamDSO.exe from the C:\Program Files\LeCroy\XStream directory onto the controlling PC (location not important).

2. Execute LeCroyXStreamDSO.exe. Note that it will not run, but it will install enough information into the registry to allow remote control of other instruments.

Connecting to a Remote Instrument

1. Visual Basic: Add a second argument to the CreateObject method, which specifies the network location of the remote instrument:

   ```vba
   Set o = CreateObject("LeCroy.XStreamDSO.1", "wavemaster00121")
   ```
2. **MATLAB:** Add a second argument to the actxserver call:

   ```
   h = actxserver (progid [, MachineName])
   ```

Once connection is made using one of the above techniques, the remainder of the communication with the instrument is the same as it would be in the case where the application runs directly on the instrument.

Remember, however, that because the connection is via a network, the performance of DCOM control of an instrument will not equal that of a direct connection created by running the client application directly on the instrument.

**NOTE:** Use version 2.6.0.0 or later for DCOM remote operation.
CHAPTER ONE: Overview

CONTROL VARIABLES EXPLAINED

Traditionally, properties presented to an Automation Client are simple “variables” with types such as Integer (int), String (BSTR), Floating Point (single, double), etc.

Control variables in X-Stream are an extension of the traditional Automation pattern, without affecting how they appear to most Automation clients (see section below on early/late bound clients).

As an example of what this enables, consider the following:

Take a control such as VerScale (Volts/Div), this may be set and queried in Visual Basic as follows:

```vbnet
app.Acquisition.C1.VerScale = 0.5
CurrentValue = app.Acquisition.C1.VerScale
```

In addition, the following is supported:

```vbnet
minValue = app.Acquisition.C1.VerScale.GetMinValue
maxValue = app.Acquisition.C1.VerScale.GetMaxValue
```

This enables an Automation Client to not only set and query the current value of a control, but also to query its limits. This is useful in the generation of instrument-independent applications, or applications that present scope controls in a graphical user interface in which limits are required.

Various types of control variables are supported. The **Type** column in the X-Stream browser shows the control type:

![XStream Browser - Online, browsing LeCroy.XStreamDSO](image)

The type designations are also given in the reference section of this manual, and are defined as follows:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>32-bit Integer Value</td>
</tr>
<tr>
<td>Double</td>
<td>Double-precision floating point value</td>
</tr>
<tr>
<td>DoubleLockstep</td>
<td>Double-precision floating point value locked to a non-linear (e.g., 1, 2, 5) sequence.</td>
</tr>
<tr>
<td>Enum</td>
<td>List Value (e.g., “Orange,” “Apple,” “Pear”)</td>
</tr>
<tr>
<td>String</td>
<td>String value</td>
</tr>
<tr>
<td>Bool</td>
<td>Boolean Value { True, False }, { 0, -1 }</td>
</tr>
<tr>
<td>Action</td>
<td>Action (no arguments or value)</td>
</tr>
</tbody>
</table>
The properties and methods available for each control are type-specific. Listed below are the most commonly used:

<table>
<thead>
<tr>
<th>TYPE</th>
<th>PROPERTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>VARIANT Value&lt;br&gt;Value(VARIANT)&lt;br&gt;int GetAdaptedValue&lt;br&gt;SetRequestedValue(int)&lt;br&gt;int GetRequestedValue&lt;br&gt;int GetDefaultValue&lt;br&gt;int GetGrain&lt;br&gt;int GetMax&lt;br&gt;int GetMin&lt;br&gt;Increment(int)</td>
</tr>
<tr>
<td>Double</td>
<td>VARIANT Value&lt;br&gt;Value(VARIANT)&lt;br&gt;double GetAdaptedValue&lt;br&gt;SetRequestedValue(double)&lt;br&gt;double GetRequestedValue&lt;br&gt;double GetDefaultValue&lt;br&gt;double GetGrainValue&lt;br&gt;double getMaxValue&lt;br&gt;double GetMinValue&lt;br&gt;Increment(int)</td>
</tr>
<tr>
<td>DoubleLockstep</td>
<td>See Double</td>
</tr>
<tr>
<td>Enum</td>
<td>VARIANT Value&lt;br&gt;Value(VARIANT)&lt;br&gt;int GetAdaptedValue&lt;br&gt;SetRequestedValue(int)&lt;br&gt;int GetRequestedValue&lt;br&gt;int GetDefaultValue&lt;br&gt;int GetMax&lt;br&gt;int GetMin&lt;br&gt;int GetNumberOfValueStates&lt;br&gt;Increment(int)&lt;br&gt;BSTR GetRangeStringScreen&lt;br&gt;BSTR GetRangeStringRemote</td>
</tr>
<tr>
<td>String</td>
<td>VARIANT Value&lt;br&gt;Value(VARIANT)&lt;br&gt;int GetMaxLength&lt;br&gt;BSTR GetRequestedValue&lt;br&gt;BSTR GetAdaptedValue&lt;br&gt;SetRequestedValue(BSTR)</td>
</tr>
<tr>
<td>Bool</td>
<td>VARIANT Value&lt;br&gt;Value(VARIANT)&lt;br&gt;BOOL GetAdaptedValue&lt;br&gt;BOOL GetRequestedValue&lt;br&gt;BOOL GetDefaultValue&lt;br&gt;Set&lt;br&gt;Clear</td>
</tr>
<tr>
<td>Action</td>
<td>ActNow</td>
</tr>
</tbody>
</table>
Overview

ACCESSING WAVEFORM/MEASUREMENT RESULTS

Waveforms

Waveform data is exposed by a ‘Result’ object, which appears at various places in the object hierarchy depending upon which waveform is to be accessed. Some examples follow:


Waveform data is exposed as a simple array, no deciphering of proprietary binary formats is performed, as was necessary in the past. An example of how it is used follows.

The example is coded as an Excel macro, and should be assigned to a button as described earlier. The macro reads the number of samples in the waveform and places it in cell B1 of the Excel spreadsheet. It then reads all available sample data values and copies them into cells in the first column of the spreadsheet (A1...Axx).

    Sub Button1_Click()
        ' Connect to the DSO
        Set app = CreateObject("LeCroy.XStreamDSO")

        ' Query the number of samples in C1 and store in cell "B1"
        Cells(1, 2).Value = numSamples

        ' Access the waveform data array, and fill the first column
        ' of the spreadsheet with it
        For i = 0 To numSamples - 1
            Cells(i + 1, 1).Value = wave(i)
        Next i
    End Sub

NOTE: Ensure that the record length is < 32kSamples, since Excel has a limit on the number of rows in a spreadsheet. Ideally, you should start experimenting with short (500 point) records.
**Measurements**

Measurement results are read in the same way as Waveforms. The following example, when copied into an Excel macro, will enable Standard Vertical parameters. It will then transfer the eight parameter values into the spreadsheet (cells C1...C8):

```vba
Sub Button1_Click()
    ' Connect to the DSO
    Set app = CreateObject("LeCroy.XStreamDSO")

    ' Enable Standard Vertical Parameters
    app.Measure.MeasureMode = "StdVertical"

    ' Transfer the 8 parameter values into the spreadsheet

    Set app = Nothing
End Sub
```

Statistics are also available for each parameter:

```vba
```

In addition, the data used to display the Histicon is available using the “histo” statistic:

```vba
Sub Button1_Click()
    Set app = CreateObject("LeCroy.XStreamDSO")

    For i = 0 To numBins - 1
        Cells(i + 1, 4).Value = bins(i)
    Next I

    Set App = Nothing
End Sub
```
## Result Status

The waveform result object described above includes a status property (bit-field) that reflects the current status of the trace. This includes both ‘warning’ and ‘error’ conditions, as described below.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Bit #</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEC_Valid</td>
<td>0x0</td>
<td>N/A</td>
</tr>
<tr>
<td>LEC_Invalid</td>
<td>0x000000000001</td>
<td>0</td>
</tr>
<tr>
<td>LEC_Overflow</td>
<td>0x000000000002</td>
<td>1</td>
</tr>
<tr>
<td>LEC_Underflow</td>
<td>0x000000000004</td>
<td>2</td>
</tr>
<tr>
<td>LEC_ContainsUndefinedValues</td>
<td>0x000000000008</td>
<td>3</td>
</tr>
<tr>
<td>LEC_LessThan</td>
<td>0x000000000010</td>
<td>4</td>
</tr>
<tr>
<td>LEC_GreaterThan</td>
<td>0x000000000020</td>
<td>5</td>
</tr>
<tr>
<td>LEC_NotAPulse</td>
<td>0x000000000040</td>
<td>6</td>
</tr>
<tr>
<td>LEC_NotCyclic</td>
<td>0x000000000080</td>
<td>7</td>
</tr>
<tr>
<td>LEC_Averaged</td>
<td>0x000000000100</td>
<td>8</td>
</tr>
<tr>
<td>LEC_UnlockedPLL</td>
<td>0x000000000200</td>
<td>9</td>
</tr>
<tr>
<td>LEC_OtherError</td>
<td>0x000000000400</td>
<td>10</td>
</tr>
<tr>
<td>LEC_OtherWarning</td>
<td>0x000000000800</td>
<td>11</td>
</tr>
<tr>
<td>LEC_OtherInfo</td>
<td>0x000000001000</td>
<td>12</td>
</tr>
<tr>
<td>LEC_InputsIncompatible</td>
<td>0x000010000000</td>
<td>28</td>
</tr>
<tr>
<td>LEC_AlgorithmLimitsReached</td>
<td>0x000020000000</td>
<td>29</td>
</tr>
<tr>
<td>LEC_BadDefinition</td>
<td>0x000040000000</td>
<td>30</td>
</tr>
<tr>
<td>LEC_TooFewData</td>
<td>0x000080000000</td>
<td>31</td>
</tr>
<tr>
<td>LEC_TooManyData</td>
<td>0x000100000000</td>
<td>32</td>
</tr>
<tr>
<td>LEC_UniformHorizIntervalRequired</td>
<td>0x000200000000</td>
<td>33</td>
</tr>
<tr>
<td>LEC_BadUnits</td>
<td>0x000400000000</td>
<td>34</td>
</tr>
<tr>
<td>LEC_DataRangeTooLow</td>
<td>0x000800000000</td>
<td>35</td>
</tr>
<tr>
<td>LEC_DataUndersampled</td>
<td>0x001000000000</td>
<td>36</td>
</tr>
<tr>
<td>LEC_PoorStatistics</td>
<td>0x002000000000</td>
<td>37</td>
</tr>
<tr>
<td>LEC_SlowTransitionTime</td>
<td>0x004000000000</td>
<td>38</td>
</tr>
<tr>
<td>LEC_DataResampled</td>
<td>0x008000000000</td>
<td>39</td>
</tr>
<tr>
<td>LEC_DataInterpolated</td>
<td>0x010000000000</td>
<td>40</td>
</tr>
<tr>
<td>LEC_MeasurementScaleImprecise</td>
<td>0x020000000000</td>
<td>41</td>
</tr>
<tr>
<td>LEC_NoDataAvailable</td>
<td>0x040000000000</td>
<td>42</td>
</tr>
<tr>
<td>LEC_SomeCummulatedResultsInvalid</td>
<td>0x080000000000</td>
<td>43</td>
</tr>
<tr>
<td>LEC_InsufficientMemory</td>
<td>0x100000000000</td>
<td>44</td>
</tr>
<tr>
<td>LEC_ChannelNotActive</td>
<td>0x200000000000</td>
<td>45</td>
</tr>
<tr>
<td>LEC_UseStatusDescription</td>
<td>0x400000000000</td>
<td>46</td>
</tr>
</tbody>
</table>
Synchronization or, more specifically, knowing when to read results, is critical when working with a digital oscilloscope by remote control (it is just as important by IEEE488.2 control as by Automation). This is especially true when working with an oscilloscope that uses a multithreaded architecture.

A classic problem seen in the majority of custom applications that control LeCroy (or other) DSOs is that the scope is left to free-run in Auto-trigger mode while simultaneously (and asynchronously) results are queried.

While working with the instrument via the Automation interface, there are a few techniques that can be used to guarantee the synchronization and consistency of results, whether they be waveform or parameter measurements.

The following example demonstrates a useful technique for ensuring synchronization. This example runs as an Excel macro:

```vba
Sub Button1_Click()
    ' Connect to the DSO
    Set app = CreateObject("LeCroy.XStreamDSO")

    ' Enable Standard Vertical parameters
    app.Measure.MeasureMode = "StdVertical"

    ' Stop the free-running trigger and take a single acquisition
    ' Use a 10 second timeout in the case that a trigger is not detected
    app.Acquisition.TriggerMode = "Stopped"
    app.Acquisition.Acquire (10, True)

    ' Read the first parameter value and transfer into the spreadsheet

End Sub
```

The `Acquire` method arms the acquisition system and waits for a user-specified time for a trigger. The second argument, a Boolean, specifies whether or not to force a trigger before returning if a trigger doesn’t arrive within the allotted time period. The method also returns a Boolean value signifying whether or not a trigger arrived. See the reference section for more information on this useful method.

Another scenario where synchronization is necessary is between changing settings and reading results, even when no acquisition took place. For this the `WaitUntilIdle` method is used. This method is “blocking” and will not return control until the setup request has completed.

*Note that the `Acquire` method is equivalent to setting the Trigger Mode to “Single”, then executing `WaitUntilIdle`. 
An example of **WaitUntilIdle** usage follows:

```vba
Sub Button1_Click()
    ' Connect to the DSO
    Set app = CreateObject("LeCroy.XStreamDSO")

    ' Enable Standard Vertical parameters
    app.Measure.MeasureMode = "StdVertical"

    ' Wait for the change to take place for a max. of 5 seconds
    app.WaitUntilIdle(5)

    ' Read the value of measurement P1 (pkpk) and transfer into the spreadsheet

    ' Enable Standard Horizontal parameters
    app.Measure.MeasureMode = "StdHorizontal"

    ' Wait for the change to take place for a max. of 5 seconds
    app.WaitUntilIdle(5)

    ' Read the value of measurement P1 (rise time) and transfer into the spreadsheet

End Sub
```

**NOTE:** In almost all remote control applications, it is HIGHLY RECOMMENDED that you STOP acquisitions before accessing result data. Most remote control problems are caused by failure to follow this practice.
GOOD PRACTICES

- Using the `app.SetToDefaultSetup` action, restore the instrument to its default state before setting the controls required by an application. This eliminates any dependency on the previous configuration of the instrument. LeCroy strives to ensure that the default state of the instrument is constant from one software release to the next.

- Synchronization is an important concept that needs to be understood before you work with an X-Stream DSO via Automation. Attempting to read results while acquisitions are in progress could cause inconsistent results.

- Use the X-Stream Browser while developing Automation applications. This tool is guaranteed to show the up-to-date status of the Automation hierarchy since it retrieves it from a running instrument. It is also a very quick and easy way to exercise controls in real-time without your having to write a single line of code.

- When using a result object, verify that the status is valid to ensure that the acquisition and/or processing was valid.

EXAMPLES

Following are fairly complete examples of automating an X-Stream DSO, including configuration, acquisition, and reading results. Examples are given both as Excel macros, and as Visual Basic Scripts, which can run without Excel being loaded on the instrument.

**Example 1: Excel Macro to Perform FFT of C1**

```vba
Sub Button1_Click()
    ' Connect to the DSO
    Set app = CreateObject("LeCroy.XStreamDSO")

    ' Restore the instrument to its default state
    app.SetToDefaultSetup

    ' Stop acquisitions during setup
    app.Acquisition.TriggerMode = "Stopped"

    ' Turn C2 off (default state leaves C1 and C2 On)
    app.Acquisition.C2.View = False

    ' Configure F1=FFT(C1), using a Blackman-Harris filter
    app.Math.F1.View = True
    app.Math.F1.Source1 = "C1"
    app.Math.F1.Operator1 = "FFT"
    app.Math.F1.Operator1Setup.Window = "BlackmanHarris"

    ' Take a single acquisition, force after 2 seconds if it doesn't trigger
    app.Acquisition.Acquire 2, True

    ' Read out the FFT
    ' Query the number of samples in F1 and store in cell "B1"
    Cells(1, 2).Value = numSamples

    ' Access the waveform data array, and fill the first column
    ' of the spreadsheet with it
    For i = 0 To numSamples - 1
        Cells(i + 1, 3).Value = wave(i)
    Next
```

- Excel macros can be written on the instrument itself. Examples include: Create a macro, open Excel on the instrument, and run the macro.
Example 2: VBScript Program to Perform FFT of C1 and Store Results in Text File

This example requires no additional software to be installed on the instrument, since it relies upon the built-in Visual Basic Script interpreter. The example is very similar to the previous Excel example, the most notable difference being the use of a standard system ActiveX control, “Scripting.FileSystemObject”, to enable the creation of files containing waveform data in ASCII format.

' VBScript example
' Configure the DSO to perform an FFT on Channel 1 and store
' the resulting data in a text file in ASCII format

' Connect to the DSO
Set app = CreateObject("LeCroy.XStreamDSO")

' Restore the instrument to its default state
app.SetToDefaultSetup

' Stop acquisitions during setup
app.Acquisition.TriggerMode = "Stopped"

' Turn C2 off (default state leaves C1 and C2 On)
app.Acquisition.C2.View = False

' Configure F1=FFT(C1), using a Blackman-Harris filter
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.Operator1 = "FFT"
app.Math.F1.Operator1Setup.Window = "BlackmanHarris"

' Take a single acquisition, force after 2 seconds if it doesn't trigger
app.Acquisition.Acquire 2, True

' Readout the FFT
Set fso = CreateObject("Scripting.FileSystemObject")
Set MyFile= fso.CreateTextFile("c:\XStreamFFT.txt", True)

' Write the FFT power spectrum into the file, sample by sample
For i = 0 To numSamples - 1
    MyFile.WriteLine(wave(i))
Next

' Clean up
MyFile.Close
Set fso = Nothing
Set app = Nothing
Example 3: Script to Measure the Rise Time of the Signal on C1 and Display It in a Popup Window

This example configures the DSO to measure the rise time of the signal on C2, take a single acquisition, and present the results in a popup dialog. The example requires no additional software to be installed on the instrument, since it relies on the built-in Visual Basic Script interpreter.

' VBScript example
' Configure the DSO to measure the rise time of the signal
' on Channel 1 and display it in a popup message box.

' Connect to the DSO
Set app = CreateObject("LeCroy.XStreamDSO")

' Restore the instrument to its default state
app.SetToDefaultSetup

' Stop acquisitions during setup
app.Acquisition.TriggerMode = "Stopped"

' Turn C2 off (default state leaves C1 and C2 On)
app.Acquisition.C2.View = False

' Configure P1=rise(C1)
app.Measure.MeasureMode = "MyMeasure"
app.Measure.P1.View = True
app.Measure.P1.ParamEngine = "rise"

' Take a single acquisition, force after 2 seconds if it doesn't trigger
app.Acquisition.Acquire 2, True

' Present the rise time in a popup message box

' Clean up
Set app = Nothing
Chapter One: Early and Late Binding

The COM standard on which Automation is built supports two kinds of “binding” between client and server: early (static), and late (dynamic, dispatch). Static binding usually involves a type library and is used primarily by compiled languages such as C++. In this case, function entry points are resolved at compile time. Dynamic binding (also known as late binding) involves resolving method and property calls at run time, as opposed to compile time.

The Automation interfaces in X-Stream based DSOs use primarily the latter: Dynamic binding. From most programming languages (VB, VBScript, etc.) this is transparent. But when you are developing applications in C++, which doesn’t provide late-binding natively, the use of a “helper” class is required. This is demonstrated below:

```c++
#include "stdafx.h"
#include "AtlBase.h"
CComModule _Module;
#include "AtlCom.h"
CComPtr<IDispatch> spDso;
CComDispatchDriver ddDso;       // dispatch ptr. to root of object model (app)

int main(int argc, char* argv[]) {
    printf("Hello X-Stream World!\n");
    ::CoInitialize(NULL);
    HRESULT hr = spDso.CoCreateInstance(L"LeCroy.XStreamDSO");
    if(SUCCEEDED(hr))
    {
        ddDso = spDso;
        // perform an Auto-Setup (app.Autosetup)
        hr = ddDso.Invoke0(L"AutoSetup");
        // retrieve a Dispatch ptr. to the app.Display object
        CComVariant displayPtr;
        hr = ddDso.GetPropertyByName(L"Display", &displayPtr);
        CComDispatchDriver ddDisplay(displayPtr.pdispVal);
        // enter Dual-grid mode (app.Display.GridMode = "Dual")
        hr = ddDisplay.PutPropertyByName(L"GridMode", &CComVariant("Dual"));
    }
    return 0;
}
```
For users who wish to harness the power of Automation control of an instrument, but are currently using “traditional” remote commands via GPIB or the network (using the VICP protocol), there is a solution. This is primarily of interest in controlling the advanced features of X-Stream DSOs, which are not supported by a traditional remote command.

X-Stream instruments, in addition to supporting LeCroy’s standard remote command set, also support a new command/query called **VBS[?]**. This command may be used in traditional remote control applications to access Automation controls.

This example shows two methods for setting the V/Div of Channel 1, the former using a traditional remote command, **VDIV**, and the latter using an Automation control via the new remote command, **VBS**. These two commands are equivalent:

```
C1:VDIV 0.5
VBS 'app.Acquisition.C1.VerScale = 0.5'
```

In its query form the following are equivalent:

```
C1:VDIV?
VBS? 'return = app.Acquisition.C1.VerScale'
```

A couple of points to note here are that the **app** variable is pre-defined and refers to the root of the Automation hierarchy. Also note that for the query form the **return =** is important, it indicates which value you wish to return to the caller.

The **VBS[?]** Command/Query is documented in more detail in the *Remote Control Manual for X-Stream DSOs*. 
The object hierarchy exposed by X-Stream based instruments is rooted at the Application object. This is the object returned when the `CreateObject("LeCroy.XStreamDSO")` method is executed in Visual Basic. All major instrument subsystems are available from this object, and many of these subsystems themselves are broken down further. Note that to simplify this figure only the first and last of the collections of Channels, Memories, Math, and Measurements are shown.

The reference section of this manual describes the controls presented by each of these objects.
Control Reference

This section presents a reference guide for each object and control in the X-Stream object hierarchy.
This is the root of the automation hierarchy, all other nodes are accessed from this point.

```
<table>
<thead>
<tr>
<th>Method/Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddZoomTrace</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>AutoSetup</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>ClearSweeps</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>DoPrint</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>ExitWithoutConfirm</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>FirmwareVersion</td>
</tr>
<tr>
<td>String</td>
</tr>
<tr>
<td>Height</td>
</tr>
<tr>
<td>Property</td>
</tr>
<tr>
<td>HideClock</td>
</tr>
<tr>
<td>Bool</td>
</tr>
<tr>
<td>InstrumentID</td>
</tr>
<tr>
<td>String</td>
</tr>
<tr>
<td>InstrumentModel</td>
</tr>
<tr>
<td>String</td>
</tr>
<tr>
<td>Left</td>
</tr>
<tr>
<td>Property</td>
</tr>
<tr>
<td>Minimize</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>Quit()</td>
</tr>
<tr>
<td>Method</td>
</tr>
<tr>
<td>ResetPreferences</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>SetToDefaultSetup</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>Shutdown</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>Sleep([in] double timeoutMilliseconds)</td>
</tr>
<tr>
<td>Method</td>
</tr>
<tr>
<td>Top</td>
</tr>
<tr>
<td>Property</td>
</tr>
<tr>
<td>TouchScreenEnable</td>
</tr>
<tr>
<td>Bool</td>
</tr>
<tr>
<td>WaitUntilIdle([in] double timeoutSeconds)</td>
</tr>
<tr>
<td>Method</td>
</tr>
<tr>
<td>Width</td>
</tr>
<tr>
<td>Property</td>
</tr>
<tr>
<td>Windowed</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>WindowState</td>
</tr>
<tr>
<td>Property</td>
</tr>
</tbody>
</table>
```

Example

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Pop up a dialog containing the instrument model
MsgBox "Model is: " & app.InstrumentModel

**AddZoomTrace** *Action*

Description

Creates function traces, defined as zoom, for each visible channel trace.
Equivalent to the front-panel zoom button.

Example

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Create zooms of all visible channel traces
app.AddZoomTrace
AutoSetup

Description
Starts an AutoSetup operation. When input channels are visible, AutoSetup operates only on those visible channels. If no channels are visible, all channels are affected by AutoSetup. When more than one channel is visible, the first visible channel in numerical order (that has a detectable signal applied to it) is automatically set up for edge triggering.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Start an Auto-Setup process.
app.AutoSetup

ClearSweeps

Description
Clears all accumulated sweeps for all subsystems. These include Channel Pre-Processing, Math, Measure, and Display Persistence. Subsystem-specific clear sweeps controls are also available. For details, please refer to the ClearSweeps control for each subsystem.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Clear all accumulated sweeps for all subsystems.
app.ClearSweeps

DoPrint

Description
Executes a printout of the currently displayed screen image. The destination printer or file, and various other options, are defined in the Hardcopy subsystem (app.Hardcopy).

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Start a print of the screen to the pre-determined destination.
app.DoPrint
**ExitWithoutConfirm**

**Description**

Causes the instrument application to exit without prompting for a confirmation. Any acquisition in progress will be canceled.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Shut down the instrument application.
app.ExitWithoutConfirm
```

**FirmwareVersion**

**String**

**Range:** Any number of characters

**Description**

Queries the firmware version of the instrument. The response takes the form: "1.0.0 (build 12345)"

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Query the firmware version number of the instrument.
MsgBox "Firmware Version is: " + app.FirmwareVersion
```

**Height**

**Property**

**Description**

Sets/Queries the height in pixels of the instrument display on the PC screen.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the height of the instrument window to 400 pixels.
app.Height = 400
```

**HideClock**

**Bool**

**Description**

Hides/Shows the clock that resides in the lower-right corner of the display of the instrument.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Hide the clock for 3 seconds.
```
app.HideClock = True
app.Sleep(3000)
app.HideClock = False

**InstrumentID**

*String*

**Range:**      Any number of characters

**Description**

Reads the complete ID of the instrument in the format: "LECROY,WM8500,WM000001,0.0.0", which includes the maker, instrument model number, serial number, and version number.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Present the ID of the instrument.
MsgBox app.InstrumentID

**InstrumentModel**

*String*

**Range:**      Any number of characters

**Description**

Queries the model number of the instrument.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Present the model number of the instrument.
MsgBox app.InstrumentModel

**Left**

*Property*

**Description**

Sets/Queries the position in pixels of the left edge of the instrument display on the PC screen. The position is measured from the left edge of the screen to the left edge of the instrument window.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the position of the left edge of the instrument window to 100 pixels.
app.Left = 100
### Minimize

**Description**

Minimizes the instrument window to reveal the underlying desktop. It will display a small window in the bottom-right corner of the display, which, when clicked, will restore the window to full-screen mode. To programmatically restore the window, refer to the `app.WindowState` control.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Minimize the instrument display.
app.Minimize
```

### Quit()

**Description**

Closes the instrument application. The instrument will prompt you with an “Are you sure?” dialog before closing down. Until you respond to the dialog, control via Automation will be blocked.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Quit the instrument application with a confirmation prompt.
app.Quit
```

### ResetPreferences

**Description**

Resets all scope preferences to their default states. The set includes the current remote communications port, the color palette settings, etc., but does not include main DSO controls such as V/Div, T/Div, etc. These main instrument controls can be reset using the `SetToDefaultSetup` control.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Reset all instrument preferences.
app.ResetPreferences
```
SetToDefaultSetup

**Action**

**Description**

Restores the instrument setup to its default state. However, certain settings will not be restored to the default state. These are the user preferences such as current remote communications port and color settings, which can be reset, if required, using the ResetPreferences action.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Restore the instrument to its default state.
app.SetToDefaultSetup

Shutdown

**Action**

**Description**

Shuts down the instrument. It will prompt you with an “Are you sure?” dialog before shutting down. Until you respond to the dialog, control via Automation will be blocked.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Shut down the instrument with a confirmation prompt.
app.Shutdown

Sleep([in] double timeoutMilliseconds)

**Method**

**Description**

Causes the main execution thread of the instrument application to sleep for the specified time period, defined in milliseconds.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

MsgBox "Sleeping for 10 seconds..."
app.Sleep(10000)
MsgBox "Sleep finished"
**Top**

*Property*

**Description**

Sets/Queries the position in pixels of the top edge of the instrument display on the PC screen. The position is measured from the top of the screen to the top of the instrument window.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the position of the top edge of the instrument window to 100 pixels.
app.Top = 100
```

**TouchScreenEnable**

*Bool*

**Description**

Sets/Queries the state of the touch-screen enable control. This is the front-panel Touch Screen button.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Disable touch-screen if it is enabled.
if app.TouchScreenEnable = True then
    app.TouchScreenEnable = False
End if
```

**WaitUntilIdle([in] double timeoutSeconds)**

*Method*

**Description**

Waits until either the application is idle or the specified timeout (in seconds) expires. This evaluates to True if the application completes before the timeout expires, and to False if a timeout occurs. When the trigger mode is Auto or Run, the application is never Idle. In this case the call to WaitUntilIdle returns after the next acquisition and any configured processing.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Wait with a timeout of five seconds.
app.WaitUntilIdle(5)
```
Width

Description
Sets/Queries the width in pixels of the instrument display on the PC screen.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the width of the instrument window to 800 pixels.
app.Width = 800

Windowed

Action
Places the instrument application in windowed mode (as opposed to full-screen mode). Places the application in the upper part of the display screen with a sizable border.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the instrument display into the windowed mode.
app.Windowed

WindowState

Property
Description
Sets/Queries the state of the PC window used by the instrument display.

- 0  windowed
- 1  full screen
- 2  minimized

Trying to set values greater than 2 or less than 0 will result in the value

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the instrument window state to windowed.
app.WindowState = 0
This group of variables controls the input channels (C1, C2, C3, C), the timebase, the trigger, and the Aux Output. Names of the form app.Acquisition.Channels.xxxx are aliases for simpler names, which are described in this section of the manual. Examples of alias pairs are as follows:

- `app.Acquisition.Channels("Cx") = app.Acquisition.Cx`
- `app.Acquisition.Channels(1) = app.Acquisition.C1`

These longer names may be more suitable than the shorter ones for certain programming structures.

### Acquisition

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Calibrate</code></td>
<td>Action</td>
</tr>
<tr>
<td><code>ClearSweeps</code></td>
<td>Action</td>
</tr>
<tr>
<td><code>TriggerMode</code></td>
<td>Enum</td>
</tr>
</tbody>
</table>

#### Acquire([in] double timeoutSeconds, [in] long)

**Method**

**Description**

**Action/Query.** Takes a single acquisition. The first of the two arguments specifies a timeout; the second, which is optional, specifies whether or not to force a trigger when the timeout occurs. Evaluates to True if a trigger occurred, or False if a timeout occurred.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Start an acquisition, wait for up to 5 seconds for a trigger
' event, force a software trigger if a hardware trigger is not
' detected before the 5 second timeout expires.
triggerDetected = app.Acquisition.Acquire(5, true)
```

#### Calibrate

**Action**

**Description**

Initiates a full calibration of the acquisition system of the instrument.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Start a calibration.
app.Acquisition.Calibrate
```
ClearSweeps

Description

Resets any accumulated average data or persistence data for channel waveforms (C1–C4). Valid only when one or more channels have waveform averaging or persistence enabled in their preprocessing settings. An average can be reset on an individual basis using app.Acquisition.Cx.ClearSweeps control.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

app.Acquisition.ClearSweeps


TriggerMode

Description

Sets/Queries the trigger mode, using values from the following list:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td>After a timeout, if a real hardware trigger is not received, the instrument will force a trigger so that there are frequent automatic updates.</td>
</tr>
<tr>
<td>Normal</td>
<td>Accepts triggers as rapidly as the system permits, but will wait indefinitely for a trigger, without updating data.</td>
</tr>
<tr>
<td>Single</td>
<td>Armes the acquisition system to acquire once, and does not rearm automatically afterward. Once a trigger is received and the data is processed, the instrument enters the &quot;Stopped&quot; state.</td>
</tr>
<tr>
<td>Stop</td>
<td>Finishes the current acquisition and does not re-arm.</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

app.Acquisition.TriggerMode = "Stopped"
app.Acquisition.Acquire(5)
## Values

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td>Auto-trigger</td>
</tr>
<tr>
<td>Normal</td>
<td>Normal Trigger</td>
</tr>
<tr>
<td>Single</td>
<td>Single Trigger</td>
</tr>
<tr>
<td>Stopped</td>
<td>No trigger possible, Stopped</td>
</tr>
</tbody>
</table>
Controls for the Auxiliary output BNC can be programmed as a simple square-wave signal source, or as a pulse that is asserted when various events occur, including Trigger Enabled, Trigger Out, and Pass/Fail.

<table>
<thead>
<tr>
<th>Amplitude</th>
<th>Double</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuxInCoupling</td>
<td>Enum</td>
</tr>
<tr>
<td>Frequency</td>
<td>DoubleLockstep</td>
</tr>
<tr>
<td>Mode</td>
<td>Enum</td>
</tr>
<tr>
<td>PulseWidth</td>
<td>Double</td>
</tr>
<tr>
<td>SetToTTL</td>
<td>Bool</td>
</tr>
</tbody>
</table>

Example

' Microsoft Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Setup the Auxiliary output to be a square wave with an amplitude
' of 500mV a frequency of 5kHz
app.Acquisition.AuxOutput.Mode = "Square"
app.Acquisition.AuxOutput.Amplitude = 0.5

Amplitude
Range: From 0.005 to 1, step 0.001
Description
Sets/Queries the amplitude of the signal on the AUX OUT connector. This is the amplitude of the signal into a 1 Mohm load. Into 50 ohms the output voltage will be halved (since the source impedance is nominally 50 ohms). Units are Volts.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the amplitude of the signal from the AUX OUT connector
' to 0.6 V into 1 Mohm, or 0.3 V into 50 ohms.
app.Acquisition.AuxOutput.Amplitude = 0.6
**AuxInCoupling**

**Enum**

**Description**
Sets the input coupling for the Auxiliary input path.

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the coupling of the Auxiliary socket, when used as an input, to ground.
' In this condition, no input signal reaches the instrument.
app.Acquisition.AuxOutput.AuxInCoupling = "GND"
```

**Values**

<table>
<thead>
<tr>
<th>DC50</th>
<th>DC, 50 ohms coupling</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td>Grounded</td>
</tr>
</tbody>
</table>

**Frequency**

**DoubleLock step**

**Description**
Sets/Queries the auxiliary output frequency of the square wave. Units are Hertz. WaveMaster models (and derivatives) have a limit of 5 MHz. WavePro 7000 models (and derivatives) have a limit of 1 MHz. This control only has effect when the AuxOutput mode is "Square".

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the frequency of the signal from the AUX OUT connector to 1 MHz.
app.Acquisition.AuxOutput.Frequency = 1e6
```

**Mode**

**Enum**

**Description**
Sets/Queries the output mode of the AUX OUT connector.

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the output of the AUX OUT connector to output a pulse on a pre-determined Pass-Fail decision.
app.Acquisition.AuxOutput.Mode = "PassFail"
```
**Values**

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCLevel</td>
<td>Emit a DC level</td>
</tr>
<tr>
<td>Off</td>
<td>Output Disabled</td>
</tr>
<tr>
<td>PassFail</td>
<td>Pulse-out controlled by Pass/Fail system</td>
</tr>
<tr>
<td>Square</td>
<td>Square-wave signal generator</td>
</tr>
<tr>
<td>TriggerEnabled</td>
<td>Pulse-out when trigger is enabled</td>
</tr>
<tr>
<td>TriggerOut</td>
<td>Pulse-out when trigger occurs</td>
</tr>
</tbody>
</table>

**PulseWidth**

*Double*

**Range:** From 0.001 to 0.5, step 0.001

**Description**

This control has effect only if the Aux Output is in pulse mode (e.g., pass/fail, trigger out). It has no effect for modes that do not produce a pulse.

Sets the duration of the output pulse from the AUX OUT connector. Units are seconds.

**Example**

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the duration of the pulse from AUX OUT
to 15 ms.
app.Acquisition.AuxOutput.PulseWidth = 15e-3
```

**SetToTTL**

*Bool*

**Description**

Overrides the Amplitude setting when true. This control sets the Auxiliary output to produce TTL levels.

**Example**

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the levels of the signal from the AUX OUT connector to produce TTL levels.
appr.Acquisition.AuxOutput.SetToTTL = True
```
**Part Two: Reference**

**Channels**

This group of variables controls the acquisition channels C1, C2, C3, and C4.

Names of the form `app.Acquisition.Channels.xxx` are aliases for simpler names, which are described in the section of the manual devoted to `app.Acquisition`. Examples of alias pairs are as follows:

- `app.Acquisition.Channels("Cx") = app.Acquisition.Cx`
- `app.Acquisition.Channels(1) = app.Acquisition.C1`

**Channels Example**

```vba
Set app = CreateObject("LeCroy.XStreamDSO")
For X = 1 To 4
    app.Acquisition.Channels(X).VerScale = 0.2
Next
```

**CX**

This group of variables controls the input channels C1, C2, C3, and C4.

Names of the form `app.Acquisition.Channels.xxx` are aliases for simpler names, which are described in the section of the manual devoted to `app.Acquisition`. Examples of alias pairs are as follows:

- `app.Acquisition.Channels("Cx") = app.Acquisition.Cx`

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AverageSweeps</td>
<td>Integer</td>
</tr>
<tr>
<td>AxisXRotation</td>
<td>Integer</td>
</tr>
<tr>
<td>AxisYRotation</td>
<td>Integer</td>
</tr>
<tr>
<td>BandwidthLimit</td>
<td>Enum</td>
</tr>
<tr>
<td>ClearSweeps</td>
<td>Action</td>
</tr>
<tr>
<td>Coupling</td>
<td>Enum</td>
</tr>
<tr>
<td>Deskew</td>
<td>Double</td>
</tr>
<tr>
<td>InterpolateType</td>
<td>Enum</td>
</tr>
<tr>
<td>Invert</td>
<td>Bool</td>
</tr>
<tr>
<td>LabelsPosition</td>
<td>String</td>
</tr>
<tr>
<td>LabelsText</td>
<td>String</td>
</tr>
<tr>
<td>Persist3DQuality</td>
<td>Enum</td>
</tr>
<tr>
<td>Persisted</td>
<td>Bool</td>
</tr>
<tr>
<td>Persistence3d</td>
<td>Bool</td>
</tr>
<tr>
<td>PersistenceMonoChrome</td>
<td>Bool</td>
</tr>
<tr>
<td>PersistenceSaturation</td>
<td>Integer</td>
</tr>
<tr>
<td>PersistenceTime</td>
<td>Enum</td>
</tr>
<tr>
<td>ProbeAttenuation</td>
<td>Double</td>
</tr>
<tr>
<td>ShowLastTrace</td>
<td>Bool</td>
</tr>
<tr>
<td>UseDotJoin</td>
<td>Bool</td>
</tr>
<tr>
<td>UseGrid</td>
<td>String</td>
</tr>
<tr>
<td>VerOffset</td>
<td>Double</td>
</tr>
<tr>
<td>VerScale</td>
<td>DoubleLockstep</td>
</tr>
</tbody>
</table>
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Setup Channel C1
app.Acquisition.C1.VerScale = 0.5
app.Acquisition.C1.VerOffset = 0.0
app.Acquisition.C1.Coupling = "DC50"

' Setup Channel C2
app.Acquisition.C2.VerScale = 0.1
app.Acquisition.C2.VerOffset = 0.2
app.Acquisition.C2.Coupling = "DC50"

AverageSweeps  
**Integer**

Range: From 1 to 1000000, step 1

Description
Sets/Queries the number of averaging sweeps for input channel Cx. This is distinct from the math function app.Math.Fx. If the number of sweeps is 1 (the default value), the data will not be averaged.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the number of sweeps for channel C1 to 25.
app.Acquisition.C1.AverageSweeps = 25

AxisXRotation  
**Integer**

Range: From −90 to 90, step 1

Description
Sets/Queries the state of the X-axis rotation control, used only in 3-D persistence modes to control the viewing position.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the rotation about the X-axis to 35 degrees for trace C3.
app.Acquisition.C3.AxisXRotation = 35
### AxisYRotation

**Type:** Integer  

**Range:** From –90 to 90, step 1  

**Description:**  
Sets/Queries the state of the Y-axis rotation control, used only in 3-D persistence modes to control the viewing position.  

**Example:**  
`' Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO")  
  
' Set the rotation about the Y-axis to 25 degrees for trace C3.  
app.Acquisition.C3.AxisYRotation = 25`

### BandwidthLimit

**Type:** Enum  

**Description:**  
Sets/Queries the bandwidth limit for input channel Cx, in Hz. This control is an enum, and therefore requires a string value, and not a scalar value. Bandwidth limit choices vary between DSO models.  

**Example:**  
`' Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO")  
  
' Set the bandwidth limit for C2 to 20 MHz.  
app.Acquisition.C2.BandwidthLimit = "20MHz"

**Values:**  
- 1GHz  
- 200MHz  
- 20MHz  
- 3GHz  
- 4GHz  
- Full

### ClearSweeps

**Type:** Action  

**Description:**  
Clears all accumulated average data and persistence data for this channel. See `app.Acquisition.ClearSweeps` for a control that clears accumulated data for channels 1 to 4, or `app.ClearSweeps` for a control that clears accumulated data for all subsystems (including Math/Measure/Display, etc.)  

**Example:**  
`' Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO")  
  
' Reset channel C1  
Reset channels C1..C4

app.Acquisition.ClearSweeps

**Coupling**

**Description**
Sets/Queries the input coupling of input channel Cx. Coupling choices vary between instrument models. WavePro 7000 instruments, for example, support AC1M and DC1M modes in addition to DC50 and GND.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the input coupling for channel C2
CoupleC2 = app.Acquisition.C2.Coupling

' Set the coupling to DC, 50 ohms
app.Acquisition.C2.Coupling = "DC50"

**Values**

- DC50
- Gnd

**Deskew**

**Range:** From –0.1 to 0.1, step 1e-012

**Description**
Sets/Queries the deskew of input channel Cx to produce a required alignment with another trace.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the deskew of channel C2 to 3.0 ns
app.Acquisition.C2.Deskew = 3.0e-9

**InterpolateType**

**Description**
Sets/Queries the type of interpolation used for input channel Cx. Because Sinx/x interpolation increases the size of the trace by a factor of 10, beware when using this option with long records.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the interpolation for channel C3 to (sin x)/x
app.Acquisition.C3.InterpolateType = "Sinxx"
**Part Two: Reference**

### Values

<table>
<thead>
<tr>
<th>Linear</th>
<th>Linear interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sinxx</td>
<td>Sinx/x interpolation</td>
</tr>
</tbody>
</table>

**Invert**

**Description**

Sets/Queries whether input channel Cx is inverted.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set channel C2 to be inverted.
app.Acquisition.C2.Invert = True

**LabelsPosition**

**Range:** Any number of characters

**Description**

Sets/Queries the horizontal position of the label attached to acquisition trace Cx. The unit of measurement is the unit of the horizontal scale. The measurement is made from the trigger point. This control is a string, not a numeric value. This allows multiple labels to be positioned, as shown in the example below.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Add a couple of labels to trace C1, one at 0ns, and one at 55ns

app.SetToDefaultSetup
app.Acquisition.C1.ViewLabels = True
app.Acquisition.C1.LabelsPosition = "0.0,55e-9"

**LabelsText**

**Range:** Any number of characters

**Description**

Sets/Queries the text that appears in labels attached to acquisition trace Cx. Multiple labels can be specified by using comma as a delimiter.
Persist3DQuality

Description
Sets/Queries the state of the 3-D Persistence quality control, which controls the way that the persistence trace is rendered.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

app.Acquisition.C2.Persist3DQuality = "Shaded"

Values
Shaded
Solid
WireFrame

Persisted

Description
Sets/Queries the persisted state of the channel waveform. If the Display.LockPersistence control is set to “AllLocked” then the persisted state of all displayed waveforms will be the same. If the Display.LockPersistence control is set to “PerTrace” then the persisted state of each waveform can be independently controlled.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

app.Display.LockPersistence = "PerTrace"
app.Acquisition.C1.Persisted = True

Persistence3d

Description
Sets/Queries the 3-D persistence state. When True, the persistence display for this channel will be displayed as a three-dimensional surface map.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

app.Acquisition.C1.Persistence3D = True
### PersistenceMonoChrome

**Description**
Sets/Queries the monochrome persistence state. When True, the persistence display for this channel will be monochromatic, whether 2-D or 3-D.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set persistence monochrome on for trace C4.
app.Acquisition.C4.PersistenceMonoChrome = True
```

### PersistenceSaturation

**Range:** From 1 to 100, step 1

**Description**
Sets/Queries the saturation threshold for persisted waveforms. All information at this level or above will be recorded with the same color or intensity.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the persistence saturation level for trace C1.
app.Acquisition.C1.PersistenceSaturation = 60
```

### PersistenceTime

**Description**
Sets/Queries the state of the Persistence Time control. Controls the persistence decay time for this trace.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the persistence time for the persistence trace of channel C1 to 10 seconds.
app.Acquisition.C1.PersistenceTime = "10s"
```

**Values**

<table>
<thead>
<tr>
<th>Values</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5s</td>
<td>5s</td>
</tr>
<tr>
<td>10s</td>
<td>Infinite</td>
</tr>
<tr>
<td>1s</td>
<td>2s</td>
</tr>
<tr>
<td>20s</td>
<td></td>
</tr>
</tbody>
</table>
**ProbingAttenuation**

*Double*

**Range:** From 1e-006 to 10000, step 1e-006

**Description**

Sets/Queries the probe attenuation. The probe attenuation is the factor by which the signal is made smaller, for example, 10 means that the probe divides by 10, and is referred to as a ÷10 probe. Certain passive probes can be marked as "x10", even though they actually divide the input signal by a factor of 10.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the probe attenuation for channel C1 to 100
app.Acquisition.C1.ProbeAttenuation = 100

**ShowLastTrace**

*Bool*

**Description**

Sets/Queries the state of the Show Last Trace control. If True, when this trace is displayed in persistence mode, the last acquired waveform will be superimposed on the accumulating persistence map.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Makes the last acquired trace invisible for the persistence trace of channel C1.
app.Acquisition.C1.ShowLastTrace = False

**UseDotJoin**

*Bool*

**Description**

Sets/Queries the DotJoin state of the channel. If True then straight line segments will be drawn between sample points. If false then only the sample points will be shown. See **Display.TraceStyle** for a control that can change the setting for all displayed traces simultaneously.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Turn dot joining off for trace C1.
app.Acquisition.C1.UseDotJoin = False
UseGrid

**String**

Range: Any number of characters

Description

Sets/Queries the graticule on which the trace is displayed. Typical values include:

<table>
<thead>
<tr>
<th>YT1 to YT8</th>
<th>One of the YT graticules used in Single, Dual, Quad, and Octal display modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotOnGrid</td>
<td>Not displayed on any graticule</td>
</tr>
</tbody>
</table>

Example

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Switch to dual grid mode, place C1 on the lower graticule
' and C2 on the upper graticule.
app.Display.GridMode = "Dual"
app.Acquisition.C1.UseGrid = "YT2"
app.Acquisition.C2.UseGrid = "YT1"
```

VerOffset

**Double**

Range: From –0.75 to 0.75, step 0.001

Description

Sets/Queries the vertical offset of input channel Cx. The setting resolution in volts lies in the range 0.25% to 0.5%, depending on the numerical value. The available offset range depends on the current V/Div setting, and also the instrument model.

Example

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the vertical offset for C1 to 10 mV.
app.Acquisition.C1.VerOffset = 0.01
```

VerScale

**Double Lock step**

Range: From 0.002 to 1, step 0.0005, locked to 1 2 5

Description

Sets/Queries the vertical scale (in Volts/Division) of an input channel. When variable gain (VerScaleVariable control) is disabled, the control will clip values to a 1-2-5 sequence. When it is enabled, the setting resolution lies in the range 1% to 2%, depending upon the numerical value.

Example

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
```
CHAPTER TWO

Control Reference

' Set C1 to a scale of 250mV/Div in Variable Scale mode
app.Acquisition.C1.VerScaleVariable = True
app.Acquisition.C1.VerScale = 0.25

VerScaleVariable

Description
Sets/Queries the state of the variable vertical scale control for channel Cx. When the variable scale is enabled, the setting resolution lies in the range 1% to 2%, depending on the numerical value. If a knowledge of the exact value is important, the value should be read back after a setting has been made.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the variable vertical scale for C1 to On.
app.Acquisition.C1.VerScaleVariable = True

View

Description
Sets/Queries the channel's "Viewed" state. When True, the channel waveform is displayed on one of the display graticules. Even when a channel is not visible, it can be used as a source for Math, Measure, etc.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Make channel C3 visible.
app.Acquisition.C3.View = True

ViewLabels

Description
Sets/Queries whether the user-defined labels for trace Cx is visible. See Also: LabelsPosition and LabelsText controls.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Show the user-defined label for trace C2.
app.Acquisition.C2.ViewLabels = True
Properties of the type xxxx.Out.Result.yyyy are those of the last completed acquisition. They are not affected if other controls are changed after that acquisition was completed. This distinction between "Out.Result" properties and other controls is most important when the trigger mode is Single or Stopped. You should treat "Out.Result" properties as read-only.

Several of these properties mention the “frame”? the term used to describe the visible portion of the trace, which is generally smaller than the acquired waveform. For example, the frame could be used to display a 500 pt. window onto a 1 Mpt. trace; or, vertically it could be used to show the “center” 10 mV of a result.

<table>
<thead>
<tr>
<th>DataArray</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>FirstEventTime</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalFrameStart</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalFrameStop</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalOffset</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalPerStep</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalResolution</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalUnits</td>
<td>Property</td>
</tr>
<tr>
<td>IndexOfFirstSampleInFrame</td>
<td>Property</td>
</tr>
<tr>
<td>LastEventTime</td>
<td>Property</td>
</tr>
<tr>
<td>NumFrameDimensions</td>
<td>Property</td>
</tr>
<tr>
<td>NumSamplesInFrame</td>
<td>Property</td>
</tr>
<tr>
<td>Samples</td>
<td>Property</td>
</tr>
<tr>
<td>Status</td>
<td>Property</td>
</tr>
<tr>
<td>StatusDescription</td>
<td>Property</td>
</tr>
<tr>
<td>Sweeps</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalFrameStart</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalFrameStop</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalMaxPossible</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalMinPossible</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalOffset</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalPerStep</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalResolution</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalUnits</td>
<td>Property</td>
</tr>
</tbody>
</table>

DataArray

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is the array of data that can be read out to represent the input waveform. The data will have 16-bit resolution? VerticalPerStep, though the physical resolution will usually be less. See VerticalResolution. The optional boolean argument can be used to determine whether 16-bit integer data, or floating-point data is returned. True indicates that floating-point values are required; False indicates that integer values are required.</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Copy the output data array (floats) for trace C3 into an array.
CHAPTER TWO

Control Reference

Dim C3DataArray

' Emit the first two data values into a popup message box

FirstEventTime Property

Description
Queries the absolute trigger time of the acquisition, or that of the first sequence in a segmented acquisition. Times are returned encoded as a currency value (VT_CY) within a variant, which allows the use of the full 64-bit resolution of the timestamp value. Values are referenced to 1 Jan 2000, with 1 ns resolution. VT_CY values are stored as 64-bit (8 byte) two's complement integers, scaled by 10,000 to give a fixed-point number with 15 digits to the left of the decimal point, and 4 digits to the right.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the time of the first event.
MsgBox EventFirst

HorizontalFrameStart Property

Description
Reads the time, in seconds, that corresponds to the left edge of the graticule, relative to the trigger instant.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the horizontal frame start for trace C1
MsgBox HorStart

HorizontalFrameStop Property

Description
Reads the time, in seconds, that corresponds to the right-hand edge of the graticule, relative to the trigger instant.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the horizontal frame stop for trace C1
HorStop = app.Acquisition.C1.Out.Result.HorizontalFrameStop
PART TWO: REFERENCE

**HorizontalOffset**

*Property*

**Description**

Reads/Queries the time between the trigger point and the left edge of the screen. If the trigger point is off the left edge of the screen, the result is positive. If the trigger point is on the screen, the result is negative. The value is very close to: app.Acquisition.Horizontal.HorOffsetOrigin + app.Acquisition.Horizontal.HorOffset, when both are expressed in seconds. A small discrepancy occurs because the trigger point is not synchronous with the sampling clock, so variations occur with a range of one sample period.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the value of the horizontal offset for trace C2
MsgBox HorOffsetC2

**HorizontalPerStep**

*Property*

**Description**

Reads the time, in units of seconds, between successive sampling instants.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the value of the horizontal step size for trace C3
MsgBox HorStepC3

**HorizontalResolution**

*Property*

**Description**

Reads the resolution of the readout of horizontal values. It is not directly related to the sample period.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the readout resolution of the horizontal axis of trace C4
MsgBox C4HRes
CHAPTER TWO

Control Reference

**HorizontalUnits**

*Property*

**Description**

Reads the unit in which the horizontal displacements are specified for trace Cx.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the units of the horizontal axis for trace C2.
C2HorUnits = app.Acquisition.C2.Out.Result.HorizontalUnits
MsgBox C2HorUnits

**IndexOfFirstSampleInFrame**

*Property*

**Description**

Reads the index number of the first sample that appears in the frame.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the index of the first sample in the frame for trace C2.
FrameSampleC2 =
app.Acquisition.C2.Out.Result.IndexOfFirstSampleInFrame
MsgBox FrameSampleC2

**LastEventTime**

*Property*

**Description**

Queries the time of the last contributing event in a set. Useful only when the result includes data produced by a sequence acquisition, or a cumulative operation such as averaging. See description for FirstEventTime for encoding details.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the time of the last event.
MsgBox EventLast
## Part Two: Reference

### NumFrameDimensions

**Property**

**Description**

Reads the dimensionality of the trace Cx: 2 for a Y-T plot, 3 for an X-Y plot.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the dimensionality of trace C3.
MsgBox DimC3
```

### NumSamplesInFrame

**Property**

**Description**

Reads the nominal number of samples in the displayed frame. In the case of sequence mode, the frame refers to one segment, not the whole graticule.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the number of samples in the frame for trace C1
FrameSamplesC1 = app.Acquisition.C1.Out.Result.NumSamplesInFrame
MsgBox FrameSamplesC1
```

### Samples

**Property**

**Description**

Reads the number of samples in a record, as read out. It will usually be two more than NumSampleInFrame (but may be much larger) to allow for the two samples that are just to the left and right of the displayed graticule. For a trace acquired in sequence mode, “frame” refers to one segment, not the whole graticule.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the number of samples in trace C1.
C1Samples = app.Acquisition.C1.Out.Result.Samples
MsgBox C1Samples
```
**Status**

*Property*

**Description**

Queries the status of the waveform result. Status is a 64-bit bitfield, encoded in a VARIANT of VT_CY (currency) type, with the meaning associated with each bit described earlier in Chapter 1. Status should be read twice, once before reading the result, and once again after reading the result. This is due to the streaming nature of processing in the X-stream DSO software.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

MsgBox stat
```

**StatusDescription**

*Property*

**Description**

Returns a textual equivalent of the status returned in the “Status” bitfield.

**Sweeps**

*Property*

**Description**

Reads the number of trigger events (sweeps) that contributed to a cumulative result. Useful only for sequence acquisitions. Cumulative processing, such as Averaging. Accumulation, can be reset using the ClearSweeps method.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the number of accumulated sweeps for trace C1
' and display in a popup
MsgBox numSweeps
```

**VerticalFrameStart**

*Property*

**Description**

Reads the amplitude that corresponds to the bottom of the displayed frame.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the vertical frame start for trace C1.
MsgBox VerStart
```
**VerticalFrameStop**

**Property**

**Description**

Reads the amplitude that corresponds to the top of the displayed frame.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the vertical frame stop for trace C1.
MsgBox VerStop

---

**VerticalMaxPossible**

**Property**

**Description**

Reads the highest value that an actual array element can have. It will be a little less than VerticalFrameStop.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the vertical maximum possible value for trace C1.
MsgBox VerMax

---

**VerticalMinPossible**

**Property**

**Description**

Reads the lowest value that an actual array element can have. It will be a little greater than VerticalFrameStart.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the vertical minimum possible value for trace C1.
MsgBox VerMin

---

**VerticalOffset**

**Property**

**Description**

Reads the difference in potential between ground and the center of the screen. With an offset of +50 mV, the center of the screen represents –50 mV; with an offset of –21 mV, the center represents +21 mV.
CHAPTER TWO

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the vertical offset for trace C1
MsgBox VerOffset

VerticalPerStep

Property

Description

Reads the smallest step in the numerical values that can be read out, whether or not the step has physical meaning. For the basic 8-bit ADC, with values returned as 16-bit short values, the step is 1/65536 of the vertical range.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the vertical step size for trace C3.
MsgBox VerStep

VerticalResolution

Property

Description

Reads the vertical resolution, which is the actual smallest difference that can be practically resolved. For an 8-bit ADC it is 1/256 of the height of the vertical range VR. But if 16 averages are set, the resolution is improved by a factor of 4, and it becomes 1/1024 of VR, and for 100 sweeps it becomes 1/2560.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the readout resolution of the vertical axis of trace C1
MsgBox VRes

VerticalUnits

Property

Description

Reads the unit in which the vertical displacements are specified.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
'Read the units of the vertical axis of trace C1


MsgBox VertUnits
This group of variables controls the timebase, the sampling, and the trigger delay.

### Horizontal

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcquisitionDuration</td>
<td>Double</td>
</tr>
<tr>
<td>ActiveChannels</td>
<td>Enum</td>
</tr>
<tr>
<td>HorOffset</td>
<td>Double</td>
</tr>
<tr>
<td>HorOffsetControl</td>
<td>Enum</td>
</tr>
<tr>
<td>HorOffsetOrigin</td>
<td>Double</td>
</tr>
<tr>
<td>HorScale</td>
<td>DoubleLockstep</td>
</tr>
<tr>
<td>HorUnits</td>
<td>String</td>
</tr>
<tr>
<td>MaxSamples</td>
<td>DoubleLockstep</td>
</tr>
<tr>
<td>NumPoints</td>
<td>Integer</td>
</tr>
<tr>
<td>NumSegments</td>
<td>Integer</td>
</tr>
<tr>
<td>ReferenceClock</td>
<td>Enum</td>
</tr>
<tr>
<td>SampleClock</td>
<td>Enum</td>
</tr>
<tr>
<td>SampleMode</td>
<td>Enum</td>
</tr>
<tr>
<td>SampleRate</td>
<td>DoubleLockstep</td>
</tr>
<tr>
<td>SamplingRate</td>
<td>Double</td>
</tr>
<tr>
<td>SequenceTimeout</td>
<td>Double</td>
</tr>
<tr>
<td>SequenceTimeoutEnable</td>
<td>Bool</td>
</tr>
<tr>
<td>SmartMemory</td>
<td>Enum</td>
</tr>
<tr>
<td>TimePerPoint</td>
<td>Double</td>
</tr>
<tr>
<td>ZeroDelay</td>
<td>Action</td>
</tr>
</tbody>
</table>

**AcquisitionDuration**

**Double**

**Range:** From 1e-012 to 1e+012, step 1e-015

**Description**

Queries the duration of the last completed acquisition. The result may depend on the spacing of the triggers in sequence mode, and it may depend on the number of averages when a channel is in averaging mode.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Obtain the duration of the last completed acquisition.
AcqDuration = app.Acquisition.Horizontal.AcquisitionDuration
MsgBox AcqDuration
```
**ActiveChannels**  
*Enum*

**Description**
Sets/Queries the number of active DSO input channels. This is a string value, with allowed values “4”, “2”, and “Auto”; and 0, 1, and 2. Beware of using 2 as a numerical value for 2 channels: you will get Auto mode instead.

**Example**

```vbs
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the instrument to use two channels.
app.Acquisition.Horizontal.ActiveChannels = "2"
```

**Values**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Use 2 channels, with increased sample rate</td>
</tr>
<tr>
<td>4</td>
<td>Use all channels</td>
</tr>
<tr>
<td>Auto</td>
<td>Maximize sample rate based upon the # displayed channels</td>
</tr>
</tbody>
</table>

**HorOffset**  
*Double*

**Range:** From -0.0005 to 5e-007, step 1e-009

**Description**
Sets/Queries the horizontal position of the trigger time, relative to the origin set by HorOffsetOrigin, in seconds. Positive to the right, negative to the left. The setting resolution is about 1% to 2%.

**Example**

```vbs
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the horizontal trigger offset to 200 ns.
app.Acquisition.Horizontal.HorOffset = 2.0e-7
```

**HorOffsetControl**  
*Enum*

**Description**
Sets the unit for HorOffset to either Time or Div, using "Time" and "Div", or 0 and 1.

**Example**

```vbs
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the horizontal offset control to divisions.
app.Acquisition.Horizontal.HorOffsetControl = "Div"
```

**Values**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Div</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td></td>
</tr>
</tbody>
</table>
HorOffsetOrigin

**Double**

Range: From 0 to 10, step 1

Description

Sets/Queries the origin, in graticule divisions, of the time scale in which HorOffset is measured. The value 0 corresponds to the left edge of the graticule. The value 10 corresponds to the right edge of the graticule. Requesting a value outside the range will select the nearest allowed value.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the origin of the horizontal trigger offset to 4.0 divisions.

HorScale

**DoubleLock step**

Range: From 2e-011 to 10, step 5e-010, locked to 1 2 5

Description

Sets/Queries the horizontal scale in time per division.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the horizontal acquisition scale to 200 ns/div.
app.Acquisition.Horizontal.HorScale = 2.0e-7

HorUnits

**String**

Range: Any number of characters

Description

Queries the units in which the horizontal scale is measured.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Obtain the units of the horizontal scale.
HorizUnit = app.Acquisition.Horizontal.HorUnits
MaxSamples

DoubleLock step

Range: From 500 to 2.4e+007, step 1000, locked to 1 2.5 5

Description
Sets/Queries the maximum permissible number of samples to be used in the acquisition memories. At the faster sample rates, the actual number used can be less than this maximum.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the available memory length per channel to 500
app.Acquisition.Horizontal.MaxSamples = 500

NumPoints

Integer

Range: From 2 to 100000000, step 1

Description
Queries the number of samples in the current setting of the acquisition memory. For sequence mode, this refers to the number if samples per segment, not to the number in the complete set. Use MaxSamples to limit the number of samples acquired.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Obtain the number of points being used in the acquisition memory.
NumberOfPoints = app.Acquisition.Horizontal.NumPoints

NumSegments

Integer

Range: From 2 to 20000, step 1

Description
Sets/Queries the number of segments in the sequence mode of acquisition. Only valid when SampleMode = "Sequence".

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Enable sequence mode and capture 500 segments
app.Acquisition.Horizontal.SampleMode = "Sequence"
app.Acquisition.Horizontal.NumSegments = 500
ReferenceClock

Description
Sets/Queries the source of the acquisition reference clock.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the source of the reference clock to External.
app.Acquisition.Horizontal.ReferenceClock = "EXT"

Values
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>EXT</td>
<td>External reference (use rear-panel BNC)</td>
</tr>
<tr>
<td>INT</td>
<td>Internal reference clock</td>
</tr>
</tbody>
</table>

SampleClock

Description
Sets/Queries the source for the sample clock.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the sample clock to expect an external source.
app.Acquisition.Horizontal.SampleClock = "External"

Values
- External
- Internal

SampleMode

Description
Sets/Queries the mode of acquisition as real-time, sequence, or random interleaved sampling. RIS mode and sequence mode are not available over the entire range of timebases, and are not available simultaneously.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the mode of acquisition to random interleaved sampling.
app.Acquisition.Horizontal.SampleMode = "RIS"

Values
- RealTime
- RIS
- Sequence
### SampleRate

**Type:** DoubleLock step  

**Range:** From 500 to 1e+010, step 1e+008, locked to 1 2.5 5  

**Description**  
Queries the sample rate of the ADCs. If random interleaved sampling (RIS) is in use, this value will be less than the effective sampling rate of the traces.  

**Example**  
```
' Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the sample rate of the ADCs.  
SampleRate = app.Acquisition.Horizontal.SampleRate
```

### SamplingRate

**Type:** Double  

**Range:** From 500 to 1e+010, step (2 digits)  

**Description**  
Queries the sampling rate. This is the effective sampling rate of the traces, rather than the sampling rate of the ADCs. When random interleaved sampling (RIS) is not in use, both values are the same.  

**Example**  
```
' Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the effective sampling rate of the signal.  
SamplingRate = app.Acquisition.Horizontal.SamplingRate
```

### SequenceTimeout

**Type:** Double  

**Range:** From 0.01 to 100, step 0.01  

**Description**  
Sets/Queries the timeout in segment mode of acquisition if insufficient triggers are received.  

**Example**  
```
' Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the sequence mode timeout to 10 seconds  
app.Acquisition.Horizontal.SequenceTimeout = 10.0
```

### SequenceTimeoutEnable

**Type:** Bool  

**Description**  
Sets/Queries the enabling of the sequence mode timeout.  

**Example**  
```
' Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO")
```
Enable the sequence mode timeout.

`app.Acquisition.Horizontal.SequenceTimeoutEnable = True`

### SmartMemory

**Description**

Sets the mode of memory management to one of the two modes:

- **SetMaximumMemory** – Maximizes the memory length for the given timebase setting; limited by the maximum length that is compatible with the maximum sampling rate that the DSO can achieve.

- **FixedSampleRate** – Keeps the sampling rate the same when the timebase is changed; limited by the maximum sampling rate that the DSO can achieve.

**Example**

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the Smart memory mode as fixed sample rate.
app.Acquisition.Horizontal.SmartMemory = "FixedSampleRate"

**Values**

<table>
<thead>
<tr>
<th>FixedSampleRate</th>
<th>SetMaximumMemory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximizes the memory length for the given timebase setting</td>
<td>Keeps the sampling rate the same when the timebase is changed</td>
</tr>
</tbody>
</table>

### TimePerPoint

**Range:** From 1e-012 to 1e+012, step 1e-012

**Description**

Queries the time interval between successive samples in the acquisition.

**Example**

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Obtain the time per point of the acquisition.
timePerPt = app.Acquisition.Horizontal.TimePerPoint
MsgBox timePerPt`

### ZeroDelay

**Description**

Sets the trigger delay to zero, relative to the time origin set by `Horizontal.HorOffsetOrigin`.

**Example**

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the trigger delay to zero, relative to the time origin set by Horizontal.HorOffsetOrigin.
app.Acquisition.Horizontal.ZeroDelay`
This group of CVars controls all aspects of the trigger, except for trigger delay, which is in Acquisition. Names of the form `app.Acquisition.Trigger.Sources.xxxx` are aliases for simpler names, which are described in this section of the manual. Examples of alias pairs are as follows:

- `app.Acquisition.Trigger.Sources("Cx") = app.Acquisition.Trigger.Cx`

### Trigger

<table>
<thead>
<tr>
<th>CVar</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DropoutTime</td>
<td>Double</td>
</tr>
<tr>
<td>Glitch</td>
<td>Enum</td>
</tr>
<tr>
<td>GlitchHigh</td>
<td>Double</td>
</tr>
<tr>
<td>GlitchLow</td>
<td>Double</td>
</tr>
<tr>
<td>HoldoffEvents</td>
<td>Integer</td>
</tr>
<tr>
<td>HoldoffTime</td>
<td>Double</td>
</tr>
<tr>
<td>HoldoffType</td>
<td>Enum</td>
</tr>
<tr>
<td>Interval</td>
<td>Enum</td>
</tr>
<tr>
<td>IntervalDelta</td>
<td>Double</td>
</tr>
<tr>
<td>IntervalHigh</td>
<td>Double</td>
</tr>
<tr>
<td>IntervalLow</td>
<td>Double</td>
</tr>
<tr>
<td>IntervalNominal</td>
<td>Double</td>
</tr>
<tr>
<td>IntervalRange</td>
<td>Enum</td>
</tr>
<tr>
<td>PatternType</td>
<td>Enum</td>
</tr>
<tr>
<td>QualEvents</td>
<td>Integer</td>
</tr>
<tr>
<td>QualFirst</td>
<td>Bool</td>
</tr>
<tr>
<td>QualState</td>
<td>Enum</td>
</tr>
<tr>
<td>QualTime</td>
<td>Double</td>
</tr>
<tr>
<td>QualWait</td>
<td>Enum</td>
</tr>
<tr>
<td>Source</td>
<td>Enum</td>
</tr>
<tr>
<td>TrigLevel</td>
<td>Double</td>
</tr>
<tr>
<td>Type</td>
<td>Enum</td>
</tr>
<tr>
<td>ValidateSource</td>
<td>Enum</td>
</tr>
<tr>
<td>Width</td>
<td>Enum</td>
</tr>
<tr>
<td>WidthDelta</td>
<td>Double</td>
</tr>
<tr>
<td>WidthNominal</td>
<td>Double</td>
</tr>
<tr>
<td>WidthRange</td>
<td>Enum</td>
</tr>
<tr>
<td>ZeroLevel</td>
<td>Action</td>
</tr>
</tbody>
</table>

#### DropoutTime

- **Double**
- **Range:** From 2e-009 to 20, step 5e-010
- **Description:**
  - Sets/Queries the trigger dropout time. The setting resolution ranges from about 1.5% to about 2.5%, depending on the numerical value.
  - `app.Acquisition.Trigger.Line` is effective only when trigger type is set to "Dropout." See `Acquisition.Channels("Cx")` for a programming example.
Glitch

Description
Sets/Queries the type of glitch trigger.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the glitch trigger to trigger on a glitch with duration between 5 and 10us
app.Acquisition.Trigger.Type = "Glitch"
app.Acquisition.Trigger.Glitch = "InRange"
app.Acquisition.Trigger.GlitchHigh = 10e-6
app.Acquisition.Trigger.GlitchLow = 5e-6

Values

| InRange | Trigger on glitch within High/Low range specified |
| LessThan | Trigger on glitch narrower than limit specified (GlitchHigh) |

GlitchHigh

Double

Range: From 6e-010 to 20, step 2e-010

Description
Sets/Queries the upper limit for a glitch trigger. Valid in both the InRange and LessThan glitch trigger types. The setting resolution ranges from about 1.5% to about 2.5%, depending on the numerical value.

GlitchLow

Double

Range: From 6e-010 to 20, step 2e-010

Description
Sets/Queries the lower limit for a glitch trigger specified as InRange type. The setting resolution ranges from about 1.5% to about 2.5%, depending on the numerical value.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the lower limit for an in-range glitch trigger to 375 ms.
app.Acquisition.Trigger.GlitchLow = 0.375
PART TWO: REFERENCE

**HoldoffEvents**

*Integer*

**Range:** From 1 to 1000000000, step 1

**Description**

Sets/Queries the number of events by which the trigger is to be held off. The resolution is 1 at all values.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Enable 'holdoff by events' mode and configure
' to holdoff by 4095 events.
app.Acquisition.Trigger.Type = "Edge"
app.Acquisition.Trigger.HoldoffType = "Events"
app.Acquisition.Trigger.HoldOffEvents = 4095

**HoldoffTime**

*Double*

**Range:** From 2e-009 to 20, step 1e-009

**Description**

Sets/Queries the trigger holdoff time. The setting resolution ranges from about 1.5% to about 2.5%, depending on the numerical value.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Enable 'holdoff by time' mode and configure
' to holdoff by 2 seconds.
app.Acquisition.Trigger.Type = "Edge"
app.Acquisition.Trigger.HoldoffType = "Time"
app.Acquisition.Trigger.HoldoffTime = 2.0

**HoldoffType**

*Enum*

**Description**

Sets/Queries the type of hold-off trigger.

**Values**

<table>
<thead>
<tr>
<th>Events</th>
<th>Holdoff by events, specified in HoldoffEvents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>No Trigger Holdoff</td>
</tr>
<tr>
<td>Time</td>
<td>Holdoff by time, specified in HoldoffTime</td>
</tr>
</tbody>
</table>
Interval

Description
Sets/Queries the interval trigger type. This is only valid when the trigger type is set to Interval.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure the interval trigger, to trigger when the interval
' is within the limits 5us to 10us
app.Acquisition.Trigger.Type = "Interval"
app.Acquisition.Trigger.Interval = "InRange"
app.Acquisition.Trigger.IntervalRange = "Limits"
app.Acquisition.Trigger.IntervalHigh = 10e-6
app.Acquisition.Trigger.IntervalLow = 5e-6

Values

<table>
<thead>
<tr>
<th>Enum</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GreaterThan</td>
<td>Trigger when interval is &gt; specified limits</td>
</tr>
<tr>
<td>InRange</td>
<td>Trigger when interval is within range</td>
</tr>
<tr>
<td>LessThan</td>
<td>Trigger when interval is &lt; specified limits</td>
</tr>
<tr>
<td>OutOfRange</td>
<td>Trigger when interval is outside specified range</td>
</tr>
</tbody>
</table>

IntervalDelta

Double

Range: From 2e-009 to 20, step 2e-010

Description
Sets/Queries the tolerance on the pulse interval for an interval trigger specified as InRange type or OutOfRange type. The setting resolution ranges from about 1.5% to about 2.5%, depending on the numerical value.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure the interval trigger, to trigger when the interval
' is 6us +/- 34ns
app.Acquisition.Trigger.Type = "Interval"
app.Acquisition.Trigger.Interval = "InRange"
app.Acquisition.Trigger.IntervalRange = "Delta"
app.Acquisition.Trigger.IntervalDelta = 34e-9
app.Acquisition.Trigger.IntervalNominal = 6e-6
**PART TWO: REFERENCE**

**IntervalHigh**

*Double*

Range: From 2e-009 to 20, step 2e-010

Description

Sets/Queries the upper limit for an interval trigger specified as InRange type. The setting resolution ranges from about 1.5% to about 2.5%, depending on the numerical value.

**IntervalLow**

*Double*

Range: From 2e-009 to 20, step 2e-010

Description

Sets/Queries the lower limit for an interval trigger specified as InRange type. The setting resolution ranges from about 1.5% to about 2.5%, depending on the numerical value.

**IntervalNominal**

*Double*

Range: From 4e-009 to 20, step 2e-010

Description

Sets/Queries the nominal pulse interval for an interval trigger specified as InRange type or OutOfRange type. The setting resolution ranges from about 1.5% to about 2.5%, depending on the numerical value.

Example

See IntervalDelta

**IntervalRange**

*Enum*

Description

Sets/Queries the interval trigger setting as either Delta or Limits. Delta is set as a center value with a tolerance. In limits mode the criteria are a lower limit and an upper limit.

Values

<table>
<thead>
<tr>
<th>Delta</th>
<th>Specify interval as a nominal value and delta.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limits</td>
<td>Specify interval by a lower and upper limit.</td>
</tr>
</tbody>
</table>

**PatternType**

*Enum*

Description

Sets/Queries the pattern (Logic) trigger type.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the pattern trigger type to Nand.
app.Acquisition.Trigger.Type = "Logic"
app.Acquisition.Trigger.PatternType = "Nand"

Values

And          Nor
Nand         Or
QualEvents

**Range:** From 1 to 99999999, step 1

**Description**
Sets/Queries the number of events for which the qualified trigger is to wait. The setting resolution is 1 at all values.

**Example**

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the number of events for trigger holdoff to 8191.
app.Acquisition.Trigger.Type = "Qualify"
app.Acquisition.Trigger.QualWait = "Events"
app.Acquisition.Trigger.QualEvents = 8191`

QualFirst

**Description**
Sets/Queries the state of the “Qualify First” trigger. In sequence mode, this control, when set, allows subsequent triggers to be detected without qualification, after the first segment of a sequence has had a fully qualified trigger. If the control is set to False, every segment has to be qualified individually. When not in sequence mode this control is not used.

**Example**

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the qualified trigger in sequence mode to qualify in the first segment only.
app.Acquisition.Horizontal.SampleMode = "Sequence"
app.Acquisition.Trigger.Type = "Qualify"
app.Acquisition.Trigger.QualFirst = True`

QualState

**Description**
Sets/Queries the qualifying state to Above or Below the specified level for the specified input source.

**Example**

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the qualification to Below the specified level.
app.Acquisition.Trigger.QualState = "Below"`

**Values**

- Above
- Below
### QualTime

**Double**

**Range:** From 2e-009 to 20, step 2e-010

**Description**

Sets/Queries the threshold time for a qualified trigger. The setting resolution ranges from about 1.5% to about 2.5%, depending on the numerical value.

**Example**

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the tolerance for the interval trigger 144 microseconds.
app.Acquisition.Trigger.QualTime = 1.44e-4
```

### QualWait

**Enum**

**Description**

Sets/Queries the wait type for qualified trigger.

**Example**

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the qualified trigger to wait for the given time to pass.
app.Acquisition.Trigger.QualWait = "GreaterThan"
```

**Values**

- Events
- GreaterThan
- LessThan
- Off

### Source

**Enum**

**Description**

Sets/Queries the trigger source.

**Example**

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the trigger source to external.
app.Acquisition.Trigger.Source = "Ext"
```

**Values**

- C1
- ExtDivide10
- C2
- ExtTimes
- C3
- Line
- C4
- Pattern
- Ext
**TrigLevel**  
*Double*  

- **Range:** From -0.25 to 0.25, step 0.0005  
- **Description:** Sets/Queries the trigger level.  
- **Example**
  ```vbs
  ' Visual Basic Script  
  Set app = CreateObject("LeCroy.XStreamDSO")
  
  ' Set the trigger level to 20 mV.  
  app.Acquisition.Trigger.TrigLevel = 0.020
  ```

**Type**  
*Enum*  

- **Description:** Sets/Queries the trigger type (mode).  
- **Example**
  ```vbs
  ' Visual Basic Script  
  Set app = CreateObject("LeCroy.XStreamDSO")
  
  ' Set the trigger type to glitch.  
  app.Acquisition.Trigger.Type = "Glitch"
  ```

**Values**
- Dropout  
- Edge  
- Glitch  
- Interval  
- Logic  
- Qualify  
- State  
- Width

**ValidateSource**  
*Enum*  

- **Description:** Sets/Queries the second trigger source in modes with two sources.  
- **Example**
  ```vbs
  ' Visual Basic Script  
  Set app = CreateObject("LeCroy.XStreamDSO")
  
  ' Set the validation trigger source to C3.  
  app.Acquisition.Trigger.ValidateSource = "C3"
  ```

**Values**
- C1  
- C2  
- C3
PART TWO: REFERENCE

Width

Description
Sets/Queries the width trigger type.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure Width Trigger to trigger on a pulse with
' width of 10ns +/- 5ns
app.Acquisition.Trigger.Type = "Width"
app.Acquisition.Trigger.Width = "InRange"
app.Acquisition.Trigger.WidthRange = "Delta"
app.Acquisition.Trigger.WidthNominal = 10e-9
app.Acquisition.Trigger.WidthDelta = 5e-9

Values

<table>
<thead>
<tr>
<th>Enum</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GreaterThan</td>
<td>Trigger when width &gt; specified limit</td>
</tr>
<tr>
<td>InRange</td>
<td>Trigger when width is within specified limits</td>
</tr>
<tr>
<td>LessThan</td>
<td>Trigger when width &lt; specified limit</td>
</tr>
<tr>
<td>OutOfRange</td>
<td>Trigger when width is outside specified range</td>
</tr>
</tbody>
</table>

WidthDelta

Double

Range: From 2e-010 to 20, step 2e-010

Description
Sets/Queries the tolerance on pulse width for a width trigger specified as InRange type or OutOfRange type. The setting resolution ranges from about 1.5% to about 2.5%, depending on the numerical value.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the tolerance for the width trigger to 55 ns.
app.Acquisition.Trigger.WidthDelta = 5.5e-8

WidthNominal

Double

Range: From 8e-010 to 20, step 2e-010

Description
Sets/Queries the nominal pulse width for a width trigger specified as InRange type or OutOfRange type. The setting resolution ranges from about 1.5% to about 2.5%, depending on the numerical value.
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the nominal value for the width trigger 55 ns.
app.Acquisition.Trigger.WidthNominal = 5.5e-8

**WidthRange**

*Enum*

**Description**

Sets/Queries the width setting as either Delta or Limits. When in Delta mode, the controls WidthNominal and WidthDelta are used to specify the nominal value and tolerance. When in Limits mode, the GlitchLow and GlitchHigh controls are used instead.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure Width Trigger to trigger on a pulse with
' width within limits of 5ns and 10ns
app.Acquisition.Trigger.Type = "Width"
app.Acquisition.Trigger.Width = "InRange"
app.Acquisition.Trigger.WidthRange = "Limits"
app.Acquisition.Trigger.GlitchLow = 10e-9
app.Acquisition.Trigger.GlitchHigh = 5e-9

**Values**

- Delta
- Limits

**ZeroLevel**

*Action*

**Description**

Sets the trigger level to zero volts.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the trigger level to zero volts.
app.Acquisition.Trigger.ZeroLevel
This group of variables controls triggering from the input channels C1, C2, C3 and C4.

**Cx**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InputImpedance</td>
<td>Enum</td>
<td>Reads the input impedance of channel C1, in ohms.</td>
</tr>
<tr>
<td>Level</td>
<td>Double</td>
<td>Sets/Queries the trigger level for the internal trigger from channel Cx.</td>
</tr>
<tr>
<td>PatternState</td>
<td>Enum</td>
<td></td>
</tr>
<tr>
<td>Slope</td>
<td>Enum</td>
<td></td>
</tr>
</tbody>
</table>

**InputImpedance**

**Enum**

**Description**
Reads the input impedance of channel C1, in ohms.

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the input impedance of C1 trigger.
ZinC1 = app.Acquisition.Trigger.C1.InputImpedance
MsgBox ZinC1
```

**Values**

50

**Level**

**Double**

**Range:** From –0.25 to 0.25, step 0.0005

**Description**
Sets/Queries the trigger level for the internal trigger from channel Cx. The setting resolution ranges from about 1.5% to about 2.5%, depending on the numerical value.

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the trigger level to 55 mV for triggering on channel C1.
app.Acquisition.Trigger.C1.Level = 0.055
```
**PatternState**

*Enum*

**Description**

Sets/Queries the pattern state for the input channel Cx. Only valid when the trigger mode is set to “Logic.”

**Example**

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the pattern state for channel C1 to low.
app.Acquisition.Trigger.C1.PatternState = "Low"
```

**Values**

- Don'tCare
- High
- Low

---

**Slope**

*Enum*

**Description**

Sets/Queries the direction of the transition to be used for internal triggering from channel Cx.

**Example**

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the direction of the transition as negative for triggering on channel C1.
app.Acquisition.Trigger.C1.Slope = "Negative"
```

**Values**

- Negative
- Positive
This group of variables controls the external trigger.

Ext

<table>
<thead>
<tr>
<th>Coupling</th>
<th>Enum</th>
</tr>
</thead>
<tbody>
<tr>
<td>InputImpedance</td>
<td>Enum</td>
</tr>
<tr>
<td>Level</td>
<td>Double</td>
</tr>
<tr>
<td>PatternState</td>
<td>Enum</td>
</tr>
<tr>
<td>Slope</td>
<td>Enum</td>
</tr>
</tbody>
</table>

**Coupling**

Description

Sets/Reads the input coupling of the external trigger input.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the input coupling of the external trigger input.
ZinCoupling = app.Acquisition.Trigger.Ext.Coupling
MsgBox ZinCoupling

Values

DC50
Gnd
DC 50 ohms

**InputImpedance**

Description

Reads the input impedance of the external trigger.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the input impedance of external trigger input.
ZinExt = app.Acquisition.Trigger.Ext.InputImpedance
MsgBox ZinExt

Values

50
**Level**  
*Double*  
Range: From –1 to 1, step 0.001  
Description  
Sets/Queries the trigger level for the external trigger.  
Example  
```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the trigger level to 55 mV for triggering from
' the external trigger socket.
app.Acquisition.Trigger.Ext.Level = 0.055
```

**PatternState**  
*Enum*  
Description  
Sets/Queries the pattern state for the external trigger input.  
Example  
```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the pattern state to low for triggering from
' the external trigger socket.
app.Acquisition.Trigger.Ext.PatternState = "Low"
```

**Values**  
- DontCare
- High
- Low

**Slope**  
*Enum*  
Description  
Sets/Queries the direction of the transition used for the external trigger.  
Example  
```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the direction of the transition as positive for triggering
' from the external trigger socket.
```

**Values**  
- Negative
- Positive
This group of variables controls the line trigger: 50 Hz or 60 Hz.
C H A P T E R  T W O  

S O U R C E S

app.Acquisition.Trigger.Sources

Names of the form app.Acquisition.Trigger.Sources.xxxx are aliases for simpler names, which are described in this section of the manual. Examples of alias pairs are as follows:

app.Acquisition.Trigger.Sources("Cx") = app.Acquisition.Trigger.Cx
This set of variables controls the cursor system.

<table>
<thead>
<tr>
<th>Pos1</th>
<th>Pos2</th>
<th>Readout</th>
<th>Track</th>
<th>Type</th>
<th>View</th>
<th>XPos1</th>
<th>XPos2</th>
<th>YPos1</th>
<th>YPos2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double</td>
<td>Double</td>
<td>Enum</td>
<td>Bool</td>
<td>Enum</td>
<td>Bool</td>
<td>Double</td>
<td>Double</td>
<td>Double</td>
<td>Double</td>
</tr>
</tbody>
</table>

**Pos1**  

**Double**

Range: From –1.79769e+308 to 1.79769e+308, step 0

Description

Sets/Queries the position of the first cursor. For vertical cursors the range is -3.99 to 3.99 divisions. For horizontal cursors the range is from left edge to right edge of the graticule, in the units of the horizontal variable.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the position of the two cursors
app.Cursors.View = "On"
app.Cursors.Type = "HorizRel"
app.Cursors.Pos1 = 50e-9
app.Cursors.Pos2 = -50e-9

**Pos2**  

**Double**

Range: From –1.79769e+308 to 1.79769e+308, step 0

Description

Sets/Queries the position of the second cursor. For vertical cursors the range is –3.99 to 3.99 divisions. For horizontal cursors the range is from left to right of the graticule, in the units of the horizontal variable.
CHAPTER TWO

Control Reference

Readout

Description
Sets/Queries whether the readout of a pair of cursors is Absolute (two separate values), Delta (one value giving the distance between the cursors), or Slope, giving the $dv/dt$ value.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the cursors readout to absolute.
app.Cursors.Readout = "Absolute"

Values

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute</td>
<td>Readout each cursor's value</td>
</tr>
<tr>
<td>Delta</td>
<td>Readout cursor value differences</td>
</tr>
<tr>
<td>Slope</td>
<td>Readout slope, i.e. dv/dt</td>
</tr>
</tbody>
</table>

Track

Description
Sets/Queries the state of tracking of a pair of cursors. If tracking is enabled, as the first cursor is moved the second will track at a constant distance from it.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set cursors tracking on.
app.Cursors.Track = True

Type

Description
Sets/Queries the currently selected type of cursor.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the cursor type to vertical relative.
app.Cursors.View = "On"
app.Cursors.Type = "VertRel"

Values

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HorizAbs</td>
<td>Single cursor, position specified in time</td>
</tr>
<tr>
<td>HorizRel</td>
<td>Dual cursors, positions specified in time</td>
</tr>
<tr>
<td>VertAbs</td>
<td>Single cursor, position specified in divisions vertically</td>
</tr>
<tr>
<td>VertRel</td>
<td>Dual cursors, positions specified in divisions vertically</td>
</tr>
</tbody>
</table>
View

Description
Sets/Queries visibility of the cursors.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Make the cursors visible.
app.Cursors.View = "On"

XPos1

Double

Description
Sets/Queries the horizontal position of the first cursor, in the units of the horizontal variable.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the horizontal position of the first cursor to 50 ns.
app.Cursors.XPos1 = 50e-9

XPos2

Double

Description
Sets/Queries the horizontal position of the second cursor, in the units of the horizontal variable.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the horizontal position of the second cursor to 4.5 ms.
app.Cursors.XPos2 = 4.5e-3

YPos1

Double

Description
Sets/Queries the vertical position of the first cursor, in graticule divisions.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
' Set the vertical position of the first cursor.
app.Cursors.YPos1 = 3.4

**YPos2**

*Double*

Range: From –3.99 to 3.99, step 0.01

Description
Sets/Queries the vertical position of the second cursor, in graticule divisions.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the vertical position of the second cursor.
app.Cursors.YPos2 = 2.1
This set of variables controls the CustomDSO system. CustomDSO operates in one of two modes. "Basic" mode, where panel setups can be assigned to named buttons that can be displayed at the bottom of the instrument's display, and "Plug-In" mode, where user-created ActiveX controls can be embedded into the instrument's menu system.

### CustomDSO

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActionEnable1</td>
<td>Bool</td>
</tr>
<tr>
<td>ActionEnable2</td>
<td>Bool</td>
</tr>
<tr>
<td>ActionEnable3</td>
<td>Bool</td>
</tr>
<tr>
<td>ActionEnable4</td>
<td>Bool</td>
</tr>
<tr>
<td>ActionEnable5</td>
<td>Bool</td>
</tr>
<tr>
<td>ActionEnable6</td>
<td>Bool</td>
</tr>
<tr>
<td>ActionEnable7</td>
<td>Bool</td>
</tr>
<tr>
<td>ActionEnable8</td>
<td>Bool</td>
</tr>
<tr>
<td>ActionScript1</td>
<td>FileName</td>
</tr>
<tr>
<td>ActionScript2</td>
<td>FileName</td>
</tr>
<tr>
<td>ActionScript3</td>
<td>FileName</td>
</tr>
<tr>
<td>ActionScript4</td>
<td>FileName</td>
</tr>
<tr>
<td>ActionScript5</td>
<td>FileName</td>
</tr>
<tr>
<td>ActionScript6</td>
<td>FileName</td>
</tr>
<tr>
<td>ActionScript7</td>
<td>FileName</td>
</tr>
<tr>
<td>ActionScript8</td>
<td>FileName</td>
</tr>
<tr>
<td>Mode</td>
<td>Enum</td>
</tr>
<tr>
<td>PlugIn1Install</td>
<td>Action</td>
</tr>
<tr>
<td>PlugIn1ProgId</td>
<td>String</td>
</tr>
<tr>
<td>PlugIn1Remove</td>
<td>Action</td>
</tr>
<tr>
<td>PresentAtPowerUp</td>
<td>Bool</td>
</tr>
</tbody>
</table>

**ActionEnable1**

**Description**

Sets/Queries enabling of the first button in the Basic mode of CustomDSO. Disabled buttons will be grayed out.

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Enable the action of button 1 of basic CustomDSO.
app.CustomDSO.ActionEnable1 = True
```

**ActionEnable2**

**Description**

See ActionEnable1.

**Example**

See ActionEnable1.
# Chapter Two

## Control Reference

<table>
<thead>
<tr>
<th>Control</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActionEnable3</td>
<td>Bool</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>See ActionEnable1.</td>
<td></td>
</tr>
<tr>
<td>ActionEnable4</td>
<td>Bool</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>See ActionEnable1.</td>
<td></td>
</tr>
<tr>
<td>ActionEnable5</td>
<td>Bool</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>See ActionEnable1.</td>
<td></td>
</tr>
<tr>
<td>ActionEnable6</td>
<td>Bool</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>See ActionEnable1.</td>
<td></td>
</tr>
<tr>
<td>ActionEnable7</td>
<td>Bool</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>See ActionEnable1.</td>
<td></td>
</tr>
<tr>
<td>ActionEnable8</td>
<td>Bool</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>See ActionEnable1.</td>
<td></td>
</tr>
<tr>
<td>ActionScript1</td>
<td>FileName</td>
</tr>
<tr>
<td>Range: Any number of characters</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Sets/Queries the name of the script file to be recalled by button 1 of basic CustomDSO.</td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td></td>
</tr>
<tr>
<td><code>Visual Basic Script</code></td>
<td></td>
</tr>
<tr>
<td><code>Set app = CreateObject(&quot;LeCroy.XStreamDSO&quot;)</code></td>
<td></td>
</tr>
<tr>
<td><code>Set the name of the script file to be recalled by button 1 of basic CustomDSO.</code></td>
<td></td>
</tr>
<tr>
<td><code>app.CustomDSO.ActionScript1 = &quot;C:\LeCroy\XStream\CustomDSO\CustomSet3Script1.lss&quot;</code></td>
<td></td>
</tr>
<tr>
<td>ActionScript2</td>
<td>FileName</td>
</tr>
<tr>
<td>Range: Any number of characters</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>See ActionScript1.</td>
<td></td>
</tr>
</tbody>
</table>
**Part Two: Reference**

**ActionScript3**
- **FileName**
- **Range:** Any number of characters
- **Description**
  - See ActionScript1.

**ActionScript4**
- **FileName**
- **Range:** Any number of characters
- **Description**
  - See ActionScript1.

**ActionScript5**
- **FileName**
- **Range:** Any number of characters
- **Description**
  - See ActionScript1.

**ActionScript6**
- **FileName**
- **Range:** Any number of characters
- **Description**
  - See ActionScript1.

**ActionScript7**
- **FileName**
- **Range:** Any number of characters
- **Description**
  - See ActionScript1.

**ActionScript8**
- **FileName**
- **Range:** Any number of characters
- **Description**
  - See ActionScript1.

**Mode**
- **Enum**
- **Description**
  - Sets/Queries the current mode of Custom DSO.
- **Example**
  - `Visual Basic Script`
  - `Set app = CreateObject("LeCroy.XStreamDSO")`

  - `Set the CustomDSO mode to Basic`
  - `app.CustomDSO.Mode = "Basic"`

<table>
<thead>
<tr>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic</strong></td>
<td>Basic mode, assign setups to named buttons</td>
</tr>
<tr>
<td><strong>Off</strong></td>
<td>CustomDSO Disabled</td>
</tr>
<tr>
<td><strong>PlugIn</strong></td>
<td>Plug-In mode, use ActiveX controls to define menu</td>
</tr>
</tbody>
</table>
**Chapter Two**

*Control Reference*

---

**PlugIn1Install**

**Description**
Installs the currently nominated COM program for use with the Plug-In mode of Custom DSO.

**Example**

```vbs
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Install the currently specified plug-in program.
app.CustomDSO.PlugIn1ProgID = "LeCroy.CustomDSODemo.1"
app.CustomDSO.PlugIn1Install
```

**PlugIn1ProgId**

**String**

**Range:** Any number of characters

**Description**
Sets/Queries the ProgID of the ActiveX control to use during Plug-In mode of CustomDSO. The ProgID “LeCroy.CustomDSODemo.1” can be used to test the system since this simple ActiveX control is installed by default in all X-Stream instruments.

**Example**

```vbs
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Sets the name of the current plug-in program.
app.CustomDSO.PlugIn1ProgID = "LeCroy.CustomDSODemo.1"
```

**PlugIn1Remove**

**Action**

**Description**
Removes the ActiveX component that is currently in use with the Plug-In mode of Custom DSO.

**Example**

```vbs
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Remove the currently installed plug-in program.
app.CustomDSO.PlugIn1Remove
```

**PresentAtPowerUp**

** Bool**

**Description**
Sets/Queries whether the Custom DSO menu is to be shown at power up, and when no standard instrument menu is open.

**Example**

```vbs
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set CustomDSO to be in use at power-up.
app.CustomDSO.PresentAtPowerUp = True
```
This set of variables controls the properties of the screen display of the instrument.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AxisLabels</td>
<td>Bool</td>
</tr>
<tr>
<td>AxisXRotation</td>
<td>Integer</td>
</tr>
<tr>
<td>AxisYRotation</td>
<td>Integer</td>
</tr>
<tr>
<td>C1Color</td>
<td>Color</td>
</tr>
<tr>
<td>C1PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>C2Color</td>
<td>Color</td>
</tr>
<tr>
<td>C2PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>C3Color</td>
<td>Color</td>
</tr>
<tr>
<td>C3PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>C4Color</td>
<td>Color</td>
</tr>
<tr>
<td>C4PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>ClearSweeps</td>
<td>Action</td>
</tr>
<tr>
<td>DisplayMode</td>
<td>Enum</td>
</tr>
<tr>
<td>F1Color</td>
<td>Color</td>
</tr>
<tr>
<td>F1PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>F2Color</td>
<td>Color</td>
</tr>
<tr>
<td>F2PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>F3Color</td>
<td>Color</td>
</tr>
<tr>
<td>F3PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>F4Color</td>
<td>Color</td>
</tr>
<tr>
<td>F4PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>F5Color</td>
<td>Color</td>
</tr>
<tr>
<td>F5PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>F6Color</td>
<td>Color</td>
</tr>
<tr>
<td>F6PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>F7Color</td>
<td>Color</td>
</tr>
<tr>
<td>F7PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>F8Color</td>
<td>Color</td>
</tr>
<tr>
<td>F8PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>FactoryDefault</td>
<td>Action</td>
</tr>
<tr>
<td>GridIntensity</td>
<td>Integer</td>
</tr>
<tr>
<td>GridMode</td>
<td>Enum</td>
</tr>
<tr>
<td>GridOnTop</td>
<td>Bool</td>
</tr>
<tr>
<td>LockPersistence</td>
<td>Enum</td>
</tr>
<tr>
<td>M1Color</td>
<td>Color</td>
</tr>
<tr>
<td>M1PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>M2Color</td>
<td>Color</td>
</tr>
<tr>
<td>M2PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>M3Color</td>
<td>Color</td>
</tr>
<tr>
<td>M3PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>M4Color</td>
<td>Color</td>
</tr>
<tr>
<td>M4PrintColor</td>
<td>Color</td>
</tr>
<tr>
<td>NumSegmentsDisplayed</td>
<td>Integer</td>
</tr>
<tr>
<td>Persist3DQuality</td>
<td>Enum</td>
</tr>
<tr>
<td>Persisted</td>
<td>Bool</td>
</tr>
<tr>
<td>Persistence3d</td>
<td>Bool</td>
</tr>
</tbody>
</table>
### AxisLabels

**Bool**

**Description**
Sets/Queries the visibility of the labels that show the horizontal and vertical limits of each grid.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Show the axis labels.
app.Display.AxisLabels = True
```

### AxisXRotation

**Integer**

**Range:** From −90 to 90, step 1

**Description**
Sets/Queries the rotation angle of the 3-D persistence display about the X-axis. The X-axis runs horizontally in the plane of the screen. Positive or negative angles can be used, in the range −90 to +90 degrees. Zero produces a direct plan view if AxisYRotation is also zero.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the rotation about the X axis to 45 degrees.
app.Display.AxisXRotation = 45
```

### AxisYRotation

**Integer**

**Range:** From −90 to 90, step 1

**Description**
Sets/Queries the rotation angle of the 3-D persistence display about the Y-axis. The Y-axis runs vertically in the plane of the screen. Positive or negative angles in the range −90 to +90 degrees can be used. A positive angle makes the left side look closer than the right side.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
```

### PersistenceLastTrace

**Bool**

### PersistenceMonoChrome

**Bool**

### PersistenceSaturation

**Integer**

### PersistenceStyle

**Enum**

### PersistenceTime

**Enum**

### PreviewPrintColors

**Action**

### ResetAll

**Action**

### SegmentMode

**Enum**

### StartSegment

**Integer**

### TraceStyle

**Enum**

### AxisLabels

**Description**
Sets/Queries the visibility of the labels that show the horizontal and vertical limits of each grid.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Show the axis labels.
app.Display.AxisLabels = True
```

### AxisXRotation

**Range:** From −90 to 90, step 1

**Description**
Sets/Queries the rotation angle of the 3-D persistence display about the X-axis. The X-axis runs horizontally in the plane of the screen. Positive or negative angles can be used, in the range −90 to +90 degrees. Zero produces a direct plan view if AxisYRotation is also zero.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the rotation about the X axis to 45 degrees.
app.Display.AxisXRotation = 45
```

### AxisYRotation

**Range:** From −90 to 90, step 1

**Description**
Sets/Queries the rotation angle of the 3-D persistence display about the Y-axis. The Y-axis runs vertically in the plane of the screen. Positive or negative angles in the range −90 to +90 degrees can be used. A positive angle makes the left side look closer than the right side.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
```
' Set the rotation about the Y axis to 35 degrees.
app.Display.AxisYRotation = 35

C1Color

Description
Sets/Queries the color of trace C1, using a number in the range 0 to FFFFFFF in hexadecimal. The possible colors are made from any combination of the primary colors, which are set in hexadecimal as Blue = &HFF0000, Green = &HFF00, Red = &HFF. The value may be entered in decimal or in hexadecimal form, though hexadecimal is usually more convenient. If the intensity of a color is to be reduced or increased by a numerical factor, an AND operation must be used afterwards, to prevent corruption of other primary colors.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

Red = &Hff: Green = &H80: Blue = &H00

' Set the color of channel C1 trace to orange
app.Display.C1Color = (Blue * &H10000) + (Green * &H100) + Red

C1PrintColor

Description
Sets/Queries the color, in the printing palette, of trace C1, using a number in the range 0 to FFFFFFF in hexadecimal. The primary colors are Blue = &HFF0000, Green = &HFF00, Red = &HFF in hexadecimal. The value may be entered in decimal or in hexadecimal form.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

Red = &Hff: Green = &H80: Blue = &H00

' Set the color of channel C1 trace to orange for printing.
app.Display.C1PrintColor = (Blue * &H10000) + (Green * &H100) +
C2Color  
Range: From 0 to 16777215  
Description  
See C1Color.

C2PrintColor  
Range: From 0 to 16777215  
Description  
See C1Printcolor.

C3Color  
Range: From 0 to 16777215  
Description  
See C1Color.

C3PrintColor  
Range: From 0 to 16777215  
Description  
See C1Printcolor.

C4Color  
Range: From 0 to 16777215  
Description  
See C1Color.

C4PrintColor  
Range: From 0 to 16777215  
Description  
See C1Printcolor.

ClearSweeps  
Description  
Initiates a Clear Sweeps operation. Clears history only for persistence traces, see the main Clear Sweeps control app.ClearSweeps, or the ClearSweeps control in other subsystems for other options.

Example  
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Initiate a clear sweeps action for persistence traces.
app.Display.ClearSweeps
**DisplayMode**

*Enum*

**Description**

Sets/Queries the display mode as either "Scope," showing the normal instrument screen, or "WebEdit," showing the web processor editing panel. WebEdit mode is available only with certain software options, including XMATH and XMAP.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Switch to WebEdit mode
app.Display.DisplayMode = "WebEdit"
```

**Values**

- Scope
- WebEdit

**F1Color**

*Color*

**Range:** From 0 to 16777215

**Description**

See C1Color.

**F1PrintColor**

*Color*

**Range:** From 0 to 16777215

**Description**

See C1Printcolor.

**F2Color**

*Color*

**Range:** From 0 to 16777215

**Description**

See C1Color.

**F2PrintColor**

*Color*

**Range:** From 0 to 16777215

**Description**

See C1Printcolor.

**F3Color**

*Color*

**Range:** From 0 to 16777215

**Description**

See C1Color.

**F3PrintColor**

*Color*

**Range:** From 0 to 16777215

**Description**

See C1Printcolor.
CHAPTER TWO  

Control Reference

<table>
<thead>
<tr>
<th>Control</th>
<th>Color</th>
<th>Range: From 0 to 16777215</th>
</tr>
</thead>
<tbody>
<tr>
<td>F4Color</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F4PrintColor</td>
<td></td>
<td>See C1Printcolor.</td>
</tr>
<tr>
<td>F5Color</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F5PrintColor</td>
<td></td>
<td>See C1Printcolor.</td>
</tr>
<tr>
<td>F6Color</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F6PrintColor</td>
<td></td>
<td>See C1Printcolor.</td>
</tr>
<tr>
<td>F7Color</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F7PrintColor</td>
<td></td>
<td>See C1Printcolor.</td>
</tr>
<tr>
<td>F8Color</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F8PrintColor</td>
<td></td>
<td>See C1Printcolor.</td>
</tr>
</tbody>
</table>
F8PrintColor  
**Color**  
Range: From 0 to 16777215  
Description  
See C1Printcolor.

FactoryDefault  
**Action**  
Description  
Restores the display of the instrument to the factory default settings.

Example  
’ Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO")

’ Restore the display to the factory pre-set state.  
app.Display.FactoryDefault

GridIntensity  
**Integer**  
Range: From 0 to 100, step 1  
Description  
Sets/Queries the grid intensity as a percentage of the maximum value, with a resolution of 1%.

Example  
’ Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO")

’ Set the grid intensity to 60% of the maximum.  
app.Display.GridIntensity = 60

GridMode  
**Enum**  
Description  
Sets/Queries the grid mode. For example, the commands "Single" and "Dual" set the grid mode until countermanded. "Auto" allows the instrument to set the grid mode most suitable for the current number of visible traces.

Example  
’ Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO")

’ Enter Octal grid mode  
app.Display.GridMode = "Octal"

Values

<table>
<thead>
<tr>
<th>Auto</th>
<th>Automatically choose grid mode, one trace per grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dual</td>
<td>Dual grid mode</td>
</tr>
<tr>
<td>Octal</td>
<td>Octal grid mode</td>
</tr>
</tbody>
</table>
## Chapter Two

### Control Reference

<table>
<thead>
<tr>
<th>Grid</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad</td>
<td>Quad grid mode</td>
<td><code>app.Display.GridOnTop = True</code></td>
</tr>
<tr>
<td>Single</td>
<td>Single grid mode</td>
<td></td>
</tr>
<tr>
<td>XY</td>
<td>XY grid mode</td>
<td></td>
</tr>
<tr>
<td>XYDual</td>
<td>XY + Dual grid mode</td>
<td></td>
</tr>
<tr>
<td>XYSingle</td>
<td>XY + Single grid mode</td>
<td></td>
</tr>
</tbody>
</table>

#### GridOnTop

**Type:** `Bool`

**Description:** Sets/Queries whether the grid lines lie over the traces or vice versa.

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Display.GridOnTop = True
```

#### LockPersistence

**Type:** `Enum`

**Description:** Sets/Queries whether the persistence states of the visible traces are locked together or separate.

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Display.LockPersistence = "PerTrace"
```

**Values**
- `AllLocked`
- `PerTrace`

#### M1Color

**Type:** `Color`

**Range:** From 0 to 16777215

**Description:** See C1Color.

#### M1PrintColor

**Type:** `Color`

**Range:** From 0 to 16777215

**Description:** See C1Printcolor.

#### M2Color

**Type:** `Color`

**Range:** From 0 to 16777215

**Description:** See C1Color.
PART TWO: REFERENCE

M2PrintColor  Color
   Range: From 0 to 16777215
   Description
   See C1Printcolor.

M3Color  Color
   Range: From 0 to 16777215
   Description
   See C1Color.

M3PrintColor  Color
   Range: From 0 to 16777215
   Description
   See C1Printcolor.

M4Color  Color
   Range: From 0 to 16777215
   Description
   See C1Color.

M4PrintColor  Color
   Range: From 0 to 16777215
   Description
   See C1Printcolor.

NumSegmentsDisplayed  Integer
   Range: From 1 to 80, step 1
   Description
   In sequence mode, sets/queries the number of segments displayed on the screen.
   Example
   ' Visual Basic Script
   Set app = CreateObject("LeCroy.XStreamDSO")

   ' Set the number of displayed segments to 20
   app.Acquisition.Horizontal.SampleMode = "Sequence"
   app.Acquisition.Horizontal.NumSegments = 20
   app.Display.NumSegmentsDisplayed = 20
   app.Display.SegmentMode = "Mosaic"

Persist3DQuality  Enum
   Description
   Sets/Queries the type of 3-D plot that is displayed.
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the type of the 3-D persistence plot.
app.DisplayPersist3DQuality = "WireFrame"

Values

- Shaded
- Solid
- WireFrame

Persisted

Description

Sets/Queries whether persistence mode is in use. If the previously set persistence mode is per trace, the persisted cvar will be set to True by this command, even if none of the traces has been set to persistence.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the state of persistence mode.
Persist = app.DisplayPersisted

Persistence3d

Description

Sets/Queries whether the persistence 3-D mode is activated.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the 3-D display to off.
app.DisplayPersistence3d = False

PersistenceLastTrace

Description

Sets/Queries whether the last created trace is shown over the persistence trace.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the persistence display to show the last trace on top of the persistence trace.
app.DisplayPersistenceLastTrace = True
## PersistenceMonoChrome

**Description**
Sets/Queries whether the persistence mode is monochrome.

**Example**
```vbescript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the persistence mode as color.
app.Display.PersistenceMonoChrome = False
```

## PersistenceSaturation

**Description**
Sets/Queries the population level, relative to the maximum possible level, at which the persistence traces reach maximum intensity, and above which there are no further changes in color or intensity.

**Example**
```vbescript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the persistence saturation level to 60%.
app.Display.PersistenceSaturation = 60
```

## PersistenceStyle

**Description**
Sets/Queries the type of persistence trace displayed.

**Example**
```vbescript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the persistence style to color graded.
app.Display.PersistenceStyle = "ColorGraded"
```

**Values**
- 3d
- Analog
- ColorGraded

## PersistenceTime

**Description**
Sets/Queries decay time for trace persistence, expressed as a number of seconds or as infinite.

**Example**
```vbescript
' Visual Basic Script
```
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the persistence time to 10 seconds.
app.Display.PersistenceTime = "10s"

Values

0.5s
10s
1s
20s
2s
5s
Infinite

PreviewPrintColors

Action

Description

Shows the instrument display in the current color scheme selected for printing.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Show the current color scheme selected for printing.
app.Display.PreviewPrintColors

ResetAll

Action

Description

Turns off persistence on all traces where it has been turned on.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Reset all persistence traces to non-persisted mode.
app.Display.ResetAll

SegmentMode

Enum

Description

Sets/Queries the display mode for segmented input channels. All visible channels are set to the same display mode by a single command.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the display mode for segments in C2 to perspective.
app.Acquisition.C2.SegmentMode = "Perspective"
### StartSegment

**Range:** From 1 to 100000, step 1  
**Description:**
Sets/Queries the selection of the first segment to be shown on the screen.  
**Example:**
```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the first shown segment to 10.
app.Display.StartSegment = 10
```

### TraceStyle

**Description:**
Sets/Queries the style in which traces are drawn.  
**Example:**
```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the state of the persistence mode.
TraceStyle = app.Display.TraceStyle
```

<table>
<thead>
<tr>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Line</strong></td>
<td>Connect adjacent samples with straight lines</td>
</tr>
<tr>
<td><strong>Points</strong></td>
<td>Show only the sample points</td>
</tr>
</tbody>
</table>
This set of variables controls the transfer of information about the screen display to destinations such as disk files, internal memories, printers, and remote computers.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination</td>
<td>Enum</td>
<td>Sets/Queries the destination for hard copy.</td>
</tr>
<tr>
<td>Directory</td>
<td>FileName</td>
<td>Sets/Queries the directory for hard copy to a file.</td>
</tr>
<tr>
<td>EMailMessage</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>GridAreaOnly</td>
<td>Bool</td>
<td></td>
</tr>
<tr>
<td>HardcopyArea</td>
<td>Enum</td>
<td></td>
</tr>
<tr>
<td>ImageFileFormat</td>
<td>Enum</td>
<td></td>
</tr>
<tr>
<td>Orientation</td>
<td>Enum</td>
<td></td>
</tr>
<tr>
<td>PreferredFilename</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Print</td>
<td>Action</td>
<td></td>
</tr>
<tr>
<td>PromptForMessage</td>
<td>Bool</td>
<td></td>
</tr>
<tr>
<td>SelectedPrinter</td>
<td>Enum</td>
<td></td>
</tr>
<tr>
<td>StripChart</td>
<td>Bool</td>
<td></td>
</tr>
<tr>
<td>StripChartFactor</td>
<td>Enum</td>
<td></td>
</tr>
<tr>
<td>UsePrintPalette</td>
<td>Bool</td>
<td></td>
</tr>
</tbody>
</table>

### Destination

**Description**
Sets/Queries the destination for hard copy.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the destination for hard copy to e-mail.
app.Hardcopy.Destination = "EMail"
```

**Values**

- **Clipboard**: Send to clipboard for pasting into other applications
- **EMail**: Send image in an E-Mail
- **File**: Store image in a file
- **Printer**: Print to a local, or networked printer

### Directory

**Range**: Any number of characters

**Description**
Sets/Queries the directory for hard copy to a file.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the directory for hard copy to files as "D:\HC"
app.Hardcopy.Directory = "D:\HC"
```
**EMailMessage**

- **Range:** Any number of characters
- **Description:** Sets/Queries the e-mail message.
- **Example**
  ```visualbasic
  ' Visual Basic Script
  Set app = CreateObject("LeCroy.XStreamDSO")
  app.Hardcopy.EMailMessage = "Results for run 89"
  ```

**GridAreaOnly**

- **Description:** Sets/Queries whether hard copy is of grid area only.
- **Example**
  ```visualbasic
  ' Visual Basic Script
  Set app = CreateObject("LeCroy.XStreamDSO")
  GridArea = app.Hardcopy.GridAreaOnly
  ```

**HardcopyArea**

- **Description:** Sets/Queries the area of the screen to be included in a hard copy.
- **Example**
  ```visualbasic
  ' Visual Basic Script
  Set app = CreateObject("LeCroy.XStreamDSO")
  app.Hardcopy.HardCopyArea = "DSOWindow"
  ```
<table>
<thead>
<tr>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DSOWindow</strong></td>
<td>Include only the DSO window</td>
</tr>
<tr>
<td><strong>FullScreen</strong></td>
<td>Include the full display screen</td>
</tr>
<tr>
<td><strong>GridAreaOnly</strong></td>
<td>Include the grid area only (doesn't include menus)</td>
</tr>
</tbody>
</table>
CHAPTER TWO

Control Reference

ImageFileFormat

Enum

Description
Sets/Queries the file format for hard copy data.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Select the format PNG for a file of hard copy data.
app.Hardcopy.Destination = "File"
appp.Hardcopy.ImageFileFormat = "PNG"

Values

<table>
<thead>
<tr>
<th>Enum</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP</td>
<td>Windows Bitmap</td>
</tr>
<tr>
<td>BMPCOMP</td>
<td>8-bit Windows Bitmap</td>
</tr>
<tr>
<td>JPEG</td>
<td>JPEG - JFIF Compliant</td>
</tr>
<tr>
<td>PNG</td>
<td>Portable Network Graphics</td>
</tr>
<tr>
<td>PSD</td>
<td>Adobe Photoshop 3.0</td>
</tr>
<tr>
<td>TIFF</td>
<td>Tagged Image File Format</td>
</tr>
</tbody>
</table>

Orientation

Enum

Description
Sets/Queries the orientation for hard copy to landscape. Valid only when outputting to a printer as opposed to a file, the clipboard, or an e-mail.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the orientation for hardcopy to landscape.
appp.Hardcopy.Orientation = "Landscape"

Values

<table>
<thead>
<tr>
<th>Enum</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Landscape</td>
<td></td>
</tr>
<tr>
<td>Portrait</td>
<td></td>
</tr>
</tbody>
</table>

PreferredFilename

String

Range: Any number of characters

Description
Sets/Queries the preferred filename to use for hard copy.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the preferred filename to PrintFile.
appp.Hardcopy.PreferredFilename = "PrintFile"
**Print**

**Action**

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiates a hard copy.</td>
<td><code>Set app = CreateObject(&quot;LeCroy.XStreamDSO&quot;)</code></td>
</tr>
</tbody>
</table>

**PromptForMessage**

**Bool**

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets/Queries whether a prompt will be given when screen dump to e-mail is requested. The prompt offers the possibility of sending a message with the e-mail.</td>
<td><code>Set app = CreateObject(&quot;LeCroy.XStreamDSO&quot;)</code></td>
</tr>
</tbody>
</table>

**SelectedPrinter**

**Enum**

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets/Queries the selection of the printer for hard copy. White space and punctuation are removed from the string.</td>
<td><code>Set app = CreateObject(&quot;LeCroy.XStreamDSO&quot;)</code></td>
</tr>
</tbody>
</table>

**StripChart**

**Bool**

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets/Queries the status of strip chart mode of printing. Valid only when outputting to the internal printer.</td>
<td></td>
</tr>
</tbody>
</table>

**StripChartFactor**

**Enum**

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets/Queries the scale factor for strip chart printing. Valid only when outputting to the internal printer.</td>
<td></td>
</tr>
</tbody>
</table>

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Initiate a hard copy.
app.Hardcopy.Print

' Enable the prompt for message with e-mail.
app.Hardcopy.PromptForMessage = True

' Select BarbondaleTintJet as the printer for hardcopy
app.Hardcopy.SelectedPrinter = "BarbondaleTintJet"

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the strip chart scale to 5 cm/division.
app.Hardcopy.StripChartFactor = "5cmdiv"

Values
100cmdiv
10cmdiv
1cmdiv
200cmdiv
20cmdiv
2cmdiv
50cmdiv
5cmdiv

UsePrintPalette Bool

Description
Sets/Queries the status of the color palette for hard copy. The screen palette or the print palette can be used. The print palette defaults to a white background, but keeps the approximate colors of all other objects. This mode saves ink/toner when printing onto paper.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Ignore the special print color palette for hard copy.
app.Hardcopy.UsePrintPalette = False
Variables of the form app.Math.xxxx control the mathematical functions F1 through F8.
Names of the form app.Math.Functions("Fx").xxxx are aliases for simpler names, which are described in this section of the manual. Examples of alias pairs are as follows:

\[
\text{app.Math.Functions("Fx") = app.Math.Fx} \\
\]

<table>
<thead>
<tr>
<th>ClearSweeps</th>
<th>Action</th>
</tr>
</thead>
</table>

**ClearSweeps**

**Description**
Clears sweeps for history functions such as average, histogram, and trend. See also the general **app.ClearSweeps** control which clears accumulated data for all subsystems, including persistence.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Clear sweeps for all history functions.
app.Math.ClearSweeps

**ResetZoom**

**Description**
Resets zoom to its default settings.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Reset zoom.
app.Math.ResetZoom
FUNCTIONS

Names of the form app.Math.Functions("Fx").xxxx are aliases for simpler names, which are described in the section of the manual devoted to app.Math. Examples of alias pairs are as follows:

app.Math.Functions("Fx") = app.Math.Fx

See Acquisition.Channels for a programming example.

FX

This set of variables controls the math functions F1 through F8.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AxisXRotation</td>
<td>Integer</td>
</tr>
<tr>
<td>AxisYRotation</td>
<td>Integer</td>
</tr>
<tr>
<td>ClearSweeps</td>
<td>Action</td>
</tr>
<tr>
<td>DoResetZoom</td>
<td>Action</td>
</tr>
<tr>
<td>DoStoreToMemoryTrace</td>
<td>Action</td>
</tr>
<tr>
<td>Equation</td>
<td>String</td>
</tr>
<tr>
<td>LabelsPosition</td>
<td>String</td>
</tr>
<tr>
<td>LabelsText</td>
<td>String</td>
</tr>
<tr>
<td>MathMode</td>
<td>Enum</td>
</tr>
<tr>
<td>MeasureOp</td>
<td>Enum</td>
</tr>
<tr>
<td>Operator1</td>
<td>Enum</td>
</tr>
<tr>
<td>Operator2</td>
<td>Enum</td>
</tr>
<tr>
<td>Persist3DQuality</td>
<td>Enum</td>
</tr>
<tr>
<td>Persisted</td>
<td>Bool</td>
</tr>
<tr>
<td>Persistence3d</td>
<td>Bool</td>
</tr>
<tr>
<td>PersistenceMonoChrome</td>
<td>Bool</td>
</tr>
<tr>
<td>PersistenceSaturation</td>
<td>Integer</td>
</tr>
<tr>
<td>PersistenceTime</td>
<td>Enum</td>
</tr>
<tr>
<td>ShowLastTrace</td>
<td>Bool</td>
</tr>
<tr>
<td>Source1</td>
<td>Enum</td>
</tr>
<tr>
<td>Source2</td>
<td>Enum</td>
</tr>
<tr>
<td>Source3</td>
<td>Enum</td>
</tr>
<tr>
<td>UseDotJoin</td>
<td>Bool</td>
</tr>
<tr>
<td>UseGrid</td>
<td>String</td>
</tr>
<tr>
<td>View</td>
<td>Bool</td>
</tr>
<tr>
<td>ViewLabels</td>
<td>Bool</td>
</tr>
</tbody>
</table>

AxisXRotation

**Description**

Sets/Queries the state of the X-axis rotation control, used only in 3-D persistence mode to control the apparent viewing position.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

Set the rotation about the X-axis to 35 degrees for trace F3.
app.Acquisition.F3.AxisXRotation = 35

**AxisYRotation**

*Integer*

**Range:** From –90 to 90, step 1

**Description**

Sets/queries the state of the Y-axis rotation control, used only in 3-D persistence mode to control the apparent viewing position.

**Example**

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

Set the rotation about the Y-axis to 25 degrees for trace F3.
app.Acquisition.F3.AxisYRotation = 25`

**ClearSweeps**

*Action*

**Description**

Clears accumulated data for a single function trace.

**Example**

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Reset accumulation for trace F1

**DoResetZoom**

*Action*

**Description**

Resets the zoom state of math trace Fx.

**Example**

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Reset zoom of math function F3.
app.Math.F3.DoResetZoom`

**DoStoreToMemoryTrace**

*Action*

**Description**

Stores data from math function Fx to a memory trace.

Destination for F1 will be M1, F2 will be M2, etc.
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Store math function F2 to a memory trace.
app.Math.F2.DoStoreToMemoryTrace

Equation

Range: Any number of characters.

Description
Queries the equation that defines the math function Fx.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the definition of math function F3.
MsgBox EquationF3

GraphOp

Description
Sets/Queries the graph operator of math function Fx. This can be used to produce a Histogram, Track, or Trend of a selected measurement directly within the Math subsystem, without the need to use the measurement subsystem. This control is only valid when MathMode is set to "Graphing."

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1 to show the histogram of the amplitude of C1
app.Math.F1.View = True
app.Math.F1.MathMode = "Graphing"
app.Math.F1.GraphOp = "Histogram"
app.Math.F1.Source1 = "C1"

Values
Histogram
Track
Trend

LabelsPosition

Range: Any number of characters.
**Part Two: Reference**

**Description**

Sets/Queries the horizontal position of the label attached to the acquisition trace Fx. The unit of measure is the unit of the horizontal scale. Measurement is made from the trigger point.

This control is a string, not a numeric value, which allows multiple labels to be positioned, as shown in the following example.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Add a couple of labels to trace F1, one at 0ns, and one at 55ns

app.SetToDefaultSetup
app.Math.F1.View = True
app.Math.F1.ViewLabels = True
app.Math.F1.LabelsPosition = "0.0, 55e-9"
```

**LabelsText**

*String*

**Range:** Any number of characters.

**Description**

Sets/Queries the text that appears in labels attached to acquisition trace Cx. Multiple labels can be specified by using a comma as a delimiter.

**MathMode**

*Enum*

**Description**

Sets/Queries the math mode.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the mode of the math function F1
```

**Values**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphing</td>
<td>Graphing mode, chain a measurement and a graphing operator</td>
</tr>
<tr>
<td>OneOperator</td>
<td>Single math operator</td>
</tr>
<tr>
<td>TwoOperators</td>
<td>Chain two math operators</td>
</tr>
<tr>
<td>WebEdit</td>
<td>Measurement is defined using the Processing Web Editor</td>
</tr>
</tbody>
</table>

**MeasureOp**

*Enum*

**Description**

Sets/Queries the measure operator of math function Fx. Valid only when MathMode = "Graph".
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the measure operator of math function F1
app.Math.F1.View = True
app.Math.F1.MathMode = "Graph"
app.Math.F1.GraphOp = "Histogram"

Values
Amplitude
Area
Base
Cycles
Delay
DeltaDelay
DeltaPeriodAtLevel
DeltaTimeAtLevel
DeltaWidthAtLevel
Duration
DutyAtLevel
DutyCycle
EdgeAtLevel
ExcelParam
ExtinctionRatio
EyeAmplitude
EyeAvgPower
EyeBER
eyecrossing
EyeHeight
EyeOneLevel
EyeQ
EyeWidth
EyeZeroLevel
Fall
Fall8020
FallAtLevel
FirstPoint
Frequency
FrequencyAtLevel
FullWidthAtHalfMaximum
FullWidthAtXX
HalfPeriod
HistogramAmplitude
HistogramBase
HistogramMaximum
HistogramMean
HistogramMedian
HistogramMinimum
PART TWO: REFERENCE

HistogramRms
HistogramSdev
HistogramTop
HoldTime
LastPoint
LevelAtX
MathcadParam
MATLABParameter
Maximum
MaximumPopulation
Mean
Median
Minimum
Mode
NarrowBandPhase
NarrowBandPower
npoints
Null
NumberOfModes
OvershootNegative
OvershootPositive
ParamScript
Peaks
PeakToPeak
Percentile
Period
PeriodAtLevel
Phase
PopulationAtX
Range:
Rise
Rise2080
RiseAtLevel
RootMeanSquare
Setup
Skew
StandardDeviation
TIE
TimeAtLevel
Top
TotalPopulation
Width
WidthAtLevel
XAtMaximum
XAtMinimum
XAtPeak

Operator1

Enum

Description

Sets/Queries the first operator of math function Fx. When MathMode = "OneOperator", this is the only math operator, when MathMode = "TwoOperators", this is the first of two operators. When MathMode = "Graph", this control has no effect. Note also that the list of available math operators varies, depending upon the instrument
model number and the list of installed software.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Define the first operator of math function F1 as an Average
app.Math.F1.View = True
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Average"

Values
AbsoluteValue
Average
Boxcar
Copy
Correlation
Derivative
Deskew
Difference
EnhancedResolution
Envelope
ExcelMath
Exp
Exp10
FFT
Filter
Floor
Histogram
Integral
Interpolate
Invert
Ln
Log10
MathcadMath
MATLABWaveform
Null
PersistenceHistogram
PersistenceTraceMean
PersistenceTraceRange
PersistenceTraceSigma
Product
Ratio
Reciprocal
Rescale
options.Roof
SegmentSelect
SinXOverX
Sparse
Square
SquareRoot
PART TWO: REFERENCE

<table>
<thead>
<tr>
<th>Sum</th>
<th>Track</th>
<th>Trend</th>
<th>WaveScript</th>
<th>Zoom</th>
</tr>
</thead>
</table>

Operator2

**Description**
Sets/Queries the second operator of math function Fx. This control is valid only when MathMode = "TwoOperators", indicating that two math operators are chained to produce a single result.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Define the first operator of math function F1 as an Average
app.Math.F1.View = True
app.Math.F1.Operator1 = "Invert"
app.Math.F1.Operator2 = "Average"
```

**Values**

- AbsoluteValue
- Average
- Boxcar
- Copy
- Correlation
- Derivative
- Deskew
- Difference
- EnhancedResolution
- Envelope
- ExcelMath
- Exp
- Exp10
- FFT
- Filter
- Floor
- Histogram
- Integral
- Interpolate
- Invert
- Ln
- Log10
- MathcadMath
- MATLABWaveform
- Null
- PersistenceHistogram
- PersistenceTraceMean
- PersistenceTraceRange
PersistenceTraceSigma
Product
Ratio
Reciprocal
Rescale
Roof
SegmentSelect
SinXOverX
Sparse
Square
SquareRoot
Sum
Track
Trend
WaveScript
Zoom

Persist3DQuality

Enum
Description
Sets/Queries the state of the 3-D Persistence quality control. Controls the way that the persistence trace is rendered.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set persistence 3-D to shaded for trace F1
app.Math.F1.Persist3DQuality = "Shaded"

Values
Shaded
Solid
WireFrame

Persisted

Bool
Description
Sets/Queries the persisted state of the function waveform. If the Display.LockPersistence control is set to “AllLocked,” the persisted state of all displayed waveforms will be the same. If the Display.LockPersistence control is set to “PerTrace,” the persisted state of each waveform can be independently controlled.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set persistence on for trace F3.
**Part Two: Reference**

**Persistence3D**

**Bool**

**Description**

Sets/Queries the 3-D persistence state. When True, the persistence display for this channel will be displayed as a three dimensional surface map.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

Set persistence plot as 3-D for trace F4.
app.Acquisition.F4.Persistence3D = True
```

**PersistenceMonoChrome**

**Bool**

**Description**

Sets/Queries the monochrome persistence state. When True, the persistence display for this channel will be monochromatic, whether 2-D or 3-D.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set persistence monochrome on for trace F1
app.Math.F1.PersistenceMonoChrome = True
```

**PersistenceSaturation**

**Integer**

**Range:** From 1 to 100, step 1

**Description**

Sets/Queries the saturation threshold for persisted waveforms. All information at this level or higher will be recorded with the same color or intensity.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the persistence saturation level for trace F1.
app.Math.F1.PersistenceSaturation = 40
```

**PersistenceTime**

**Enum**

**Description**

Sets/Queries the state of the Persistence Time control. Controls the persistence decay time for this trace.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
```
' Set the persistence time for the trace F1 to 10 seconds.

app.Math.F1.PersistenceTime = "10s"

Values

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5s</td>
</tr>
<tr>
<td>10s</td>
</tr>
<tr>
<td>1s</td>
</tr>
<tr>
<td>20s</td>
</tr>
<tr>
<td>2s</td>
</tr>
<tr>
<td>5s</td>
</tr>
<tr>
<td>Infinite</td>
</tr>
</tbody>
</table>

**ShowLastTrace**

**Bool**

**Description**

Sets/Queries the state of the Show Last Trace control. If True, then when this trace is displayed in persistence mode, the last acquired waveform will be superimposed on the accumulating persistence map. See the general description above for a discussion of the locked and unlocked persistence modes.

**Example**

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Do not show the last trace for the persistence trace of trace F1.

**Source1**

**Enum**

**Description**

Sets/Queries the first source of the first operator in Fx. The two possible sources of Operator1 are Source1 and Source2. Source3 is the second source to Operator2, with the first source of Operator2 being the output of Operator1. The list of available sources depends on the instrument model and its installed software options.

**Example**

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Define the first source of math function F1 as C3.

app.Math.F1.Source1 = "C3"

Values

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BadBits</td>
</tr>
<tr>
<td>Bits</td>
</tr>
<tr>
<td>C1</td>
</tr>
<tr>
<td>C2</td>
</tr>
<tr>
<td>C3</td>
</tr>
<tr>
<td>C4</td>
</tr>
<tr>
<td>Eye</td>
</tr>
</tbody>
</table>
### Source2

**Description**

Sets/Queries the second source of the first operator in Fx. The list of available sources depends on the instrument model and its installed software options.

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Define the second source of math function F2 as C4.
app.Math.F1.Source2 = "C4"
```

**Values**

<table>
<thead>
<tr>
<th>BadBits</th>
<th>Bits</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>Eye</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
</tr>
</thead>
</table>
Source3

Description
Sets/Queries the second source of the second operator in Fx. Only valid when MathMode = "Dual". The list of available sources depends on the instrument model and its installed software options.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Define the third source of math function F1 as C1.
app.Math.F1.Source3 = "C1"

Values
BadBits
Bits
C1
C2
C3
C4
Eye
F2
F3
F4
F5
F6
F7
F8
M1
M2
M3
M4
None
PRBS

UseDotJoin

Description
Sets/Queries the DotJoin state of the trace. When True, straight line segments will be drawn between sample points. When False, only the sample points will be shown. See Display.TraceStyle for a control that can change the setting for all displayed traces simultaneously.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Turn dot join off for trace F1.
app.Math.F1.UseDotJoin = False
### UseGrid

**Range:** Any number of characters.

**Description**
Sets/Queries the grid in use for the math trace Fx. See also `app.Acquisition.Cx.UseGrid`.

**Example**

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Place math trace F3 on grid YT4.
app.Math.F3.UseGrid = "YT4"
```

### View

**Description**
Sets/Queries whether the trace of math function Fx is visible. Even when math traces are not visible, but are being used as inputs to other math functions and/or measurements, they are computed.

**Example**

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Show math trace F3.
```

### ViewLabels

**Description**
Sets/Queries whether trace labels, defined with LabelsText and LabelsPosition controls, are shown.

**Example**

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Show the user-defined trace label for trace F1
app.Math.F1.ViewLabels = True
```
This node is dynamically created, and will contain the controls for the operator currently selected into Operator1. See the Math/Measure Control reference at the end of this manual for a list of these controls.
Properties of the type xxxx.Out.Result.yyyy are those of the last completed acquisition. They are not affected if other CVars are changed after that acquisition was completed. This distinction between "Out.Result" properties and other CVars is most important when the trigger mode is Single or Stopped. You should treat "Out.Result" properties as read-only.

This set of variables controls the zoom functions for math trace Fx.

**Zoom**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>HorPos</td>
<td>Double</td>
</tr>
<tr>
<td>HorZoom</td>
<td>Double</td>
</tr>
<tr>
<td>ResetZoom</td>
<td>Action</td>
</tr>
<tr>
<td>VariableHorZoom</td>
<td>Bool</td>
</tr>
<tr>
<td>VariableVerZoom</td>
<td>Bool</td>
</tr>
<tr>
<td>VerPos</td>
<td>Double</td>
</tr>
<tr>
<td>VerZoom</td>
<td>Double</td>
</tr>
</tbody>
</table>

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Turn on trace F1, will default to Zoom-Only
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"

' Zoom trace F1 by a factor of 2 horizontally and vertically
app.Math.F1.Zoom.HorZoom = 2.0
```

**HorPos**

**Double**

*Range:* From –0.5 to 0.5, step (8 digits)

*Description*

Sets/Queries the horizontal position of the center of the grid on the zoomed trace Fx. The unit of measure is the screen width, that is, 0.3 means a shift of three of the ten divisions. A positive value moves the trace to the left.

**HorZoom**

**Double**

*Range:* From 0.1 to 1e+006, step (8 digits)

*Description*

Sets/Queries the horizontal magnification of the trace Fx. The magnification will be in a 1 2 5 10 sequence unless variable horizontal magnification has been set.

**ResetZoom**

**Action**

*Description*

Resets the trace Fx to x1 zoom and zero offset in both axes, so that it is identical to its input trace.
**VariableHorZoom**

**Description**
Sets/Queries the ability to zoom horizontally by a continuously variable factor. If a horizontal zoom of 0.9 is set, while variable zoom is off, the horizontal zoom will be set to 1.0. If the variable zoom is then enabled, the factor of 0.9 will have been remembered, and it will be used.

**VariableVerZoom**

**Description**
Sets/Queries the ability to zoom vertically by a continuously variable factor. If a vertical zoom of 0.9 is set, while variable zoom is off, the vertical zoom will be set to 1.0. If the variable zoom is then enabled, the factor of 0.9 will have been remembered, and it will be used.

**VerPos**

**Range:** From –1.5 to 1.5, step (8 digits)

**Description**
Sets/Queries the vertical position of the center of the grid on the zoomed trace Fx. The unit of measure is the screen height, that is, 0.375 means a shift of three of the eight divisions. A positive value moves the trace downwards.

**VerZoom**

**Range:** From 0.1 to 100, step (8 digits)

**Description**
Sets/Queries the vertical magnification of the trace Fx. The magnification will be in a 1 2 5 10 sequence unless VariableVerZoom has been set to True, in which case it will be continuously variable.
This set of variables controls the display of data in X-vs.-Y mode. Only valid when the instrument is in XY, XYSingle, or XYDual display modes.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AxisXRotation</td>
<td>Integer</td>
<td>Sets/Queries the state of the X-axis rotation control, used only in 3-D persistence modes to control the viewing position. See the general description above for a discussion of the locked and unlocked persistence modes.</td>
</tr>
<tr>
<td>AxisYRotation</td>
<td>Integer</td>
<td>Sets/Queries the state of the Y-axis rotation control, used only in 3-D persistence modes to control the viewing position.</td>
</tr>
<tr>
<td>ClearSweeps</td>
<td>Action</td>
<td></td>
</tr>
<tr>
<td>InputX</td>
<td>Enum</td>
<td></td>
</tr>
<tr>
<td>InputY</td>
<td>Enum</td>
<td></td>
</tr>
<tr>
<td>Persist3DQuality</td>
<td>Enum</td>
<td></td>
</tr>
<tr>
<td>Persisted</td>
<td>Bool</td>
<td></td>
</tr>
<tr>
<td>Persistence3d</td>
<td>Bool</td>
<td></td>
</tr>
<tr>
<td>PersistenceMonoChrome</td>
<td>Bool</td>
<td></td>
</tr>
<tr>
<td>PersistenceSaturation</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>PersistenceTime</td>
<td>Enum</td>
<td></td>
</tr>
<tr>
<td>ShowLastTrace</td>
<td>Bool</td>
<td></td>
</tr>
<tr>
<td>UseDotJoin</td>
<td>Bool</td>
<td></td>
</tr>
</tbody>
</table>

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Switch to XY+Dual Grid Mode
app.Display.GridMode = "XYDual"

' Configure XY to show C1 vs. C2 in 3D Persistence mode
app.Math.XY.InputX = "C1"
app.Math.XY.InputY = "C2"
app.Math.XY.Persisted = True
app.Math.XY.Persistence3d = True
app.Math.XY.Persistencemonochrome = False
app.Math.XY.PersistenceTime = "Infinite"
```
## PART TWO: REFERENCE

### ClearSweeps

**Action**

Description

Clears the persistence X-Y plot.

### InputX

**Enum**

**Description**

Sets/Queries the name of the input channel for the X-axis of the X-Y plot.

**Values**

- BadBits
- Bits
- C1
- C2
- C3
- C4
- Eye
- F1
- F2
- F3
- F4
- F5
- F6
- F7
- F8
- M1
- M2
- M3
- M4
- PRBS

### InputY

**Enum**

**Description**

Sets/Queries the name of the input channel for the Y-axis of the X-Y plot.

**Values**

- BadBits
- Bits
- C1
- C2
- C3
- C4
- Eye
- F1
- F2
- F3
- F4
- F5
- F6
- F7
Persist3DQuality  
**Enum**

Description
Sets/Queries the state of the 3-D Persistence quality control. Controls the way that the persistence trace is rendered.

Values
- Shaded
- Solid
- WireFrame

Persisted  
**Bool**

Description
Sets/Queries the persisted state of the X-Y plot. If the Display.LockPersistence control is set to “AllLocked”, the persisted state of all displayed waveforms will be the same. If the Display.LockPersistence control is set to “PerTrace”, the persisted state of each waveform can be independently controlled.

Persistence3d  
**Bool**

Description
Sets/Queries the 3-D persistence state. When True, the persistence display for the X-Y plot will be displayed as a three dimensional surface map.

PersistenceMonoChrome  
**Bool**

Description
Sets/Queries the monochrome persistence state. When True, the persistence display for the X-Y plot will be monochromatic, whether 2-D or 3-D.

PersistenceSaturation  
**Integer**

Range: From 1 to 100, step 1

Description
Sets/Queries the saturation threshold for persisted X-Y plot. All information at this level or above will be recorded with the same color or intensity.

PersistenceTime  
**Enum**

Description
Sets/Queries the state of the Persistence Time control. Controls the persistence decay time for the X-Y persistence.

Values
- 0.5s
- 10s
PART TWO: REFERENCE

| 1s | 20s | 2s | 5s | Infinite |

**ShowLastTrace**

**Description**
Sets/Queries the state of the Show Last Trace control. If True, when this trace is displayed in persistence mode, the last acquired waveform will be superimposed on the accumulating persistence map.

**UseDotJoin**

**Description**
Sets/Queries whether dot joining is used in the X-Y plot.
Properties of the type xxxx.Out.Result,yyyy are those of the last completed acquisition. They are not affected if other CVars are changed after that acquisition was completed. This distinction between "Out.Result" properties and other CVars is most important when the trigger mode is Single or Stopped. You should treat "Out.Result" properties as read-only.

This XY result object is very similar to, but not identical to, the result object exposed by the channel and math traces. The differences are due to the fact that the XY trace returns pairs of data values, one for X, one for Y.

<table>
<thead>
<tr>
<th>DataArray</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>FirstEventTime</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalFrameStart</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalFrameStop</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalOffset</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalPerStep</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalResolution</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalUnits</td>
<td>Property</td>
</tr>
<tr>
<td>LastEventTime</td>
<td>Property</td>
</tr>
<tr>
<td>NumFrameDimensions</td>
<td>Property</td>
</tr>
<tr>
<td>Samples</td>
<td>Property</td>
</tr>
<tr>
<td>Sweeps</td>
<td>Property</td>
</tr>
<tr>
<td>XFrameStart</td>
<td>Property</td>
</tr>
<tr>
<td>XFrameStop</td>
<td>Property</td>
</tr>
<tr>
<td>XMaxPossible</td>
<td>Property</td>
</tr>
<tr>
<td>XMinPossible</td>
<td>Property</td>
</tr>
<tr>
<td>XOffset</td>
<td>Property</td>
</tr>
<tr>
<td>XPerStep</td>
<td>Property</td>
</tr>
<tr>
<td>XResolution</td>
<td>Property</td>
</tr>
<tr>
<td>XUnits</td>
<td>Property</td>
</tr>
<tr>
<td>YFrameStart</td>
<td>Property</td>
</tr>
<tr>
<td>YFrameStop</td>
<td>Property</td>
</tr>
<tr>
<td>YMaxPossible</td>
<td>Property</td>
</tr>
<tr>
<td>YMinPossible</td>
<td>Property</td>
</tr>
<tr>
<td>YOffset</td>
<td>Property</td>
</tr>
<tr>
<td>YPerStep</td>
<td>Property</td>
</tr>
<tr>
<td>YResolution</td>
<td>Property</td>
</tr>
<tr>
<td>YUnits</td>
<td>Property</td>
</tr>
</tbody>
</table>

**DataArray Description**

This is the array of data that can be read out to represent the calculated waveform. The returned array will have numSamples rows, and two columns. The first column will contain X values, and the second column Y values. For example: (0, 0) indicates the X value of the first sample, (10, 1) indicates the Y value of the 10th sample. If DataArray(True) is specified, floating-point values are retrieved. If DataArray(False) is specified, 16-bit integer values are returned.

**FirstEventTime Description**

This is the array of data that can be read out to represent the calculated waveform. The returned array will have numSamples rows, and two columns. The first column will contain X values, and the second column Y values. For example: (0, 0) indicates the X value of the first sample, (10, 1) indicates the Y value of the 10th sample. If DataArray(True) is specified, floating-point values are retrieved. If DataArray(False) is specified, 16-bit integer values are returned.
PART TWO: REFERENCE

Reads the time of the first event.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the time of the first event.

---

**HorizontalFrameStart**

*Property*

**Description**

Reads the start of the horizontal scale of the Y-T traces that contribute to the X-Y plot. If the input traces have been shifted to different offsets, this CVar refers to the X-trace. The values for both traces can be obtained using app.Math.Cx.Out.Result.HorizontalFrameStart.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the horizontal frame start for the input traces of an X-Y plot.
HorStart = app.Math.XY.Out.Result.HorizontalFrameStart

---

**HorizontalFrameStop**

*Property*

**Description**

Reads the end of the horizontal scale of the Y-T traces that contribute to the X-Y plot. If the input traces have been shifted to different offsets, this control variable refers to the X-trace.

---

**HorizontalOffset**

*Property*

**Description**

Reads the offset of the horizontal scale of the Y-T traces that contribute to the X-Y plot. If the input traces have been shifted to different offsets, this cvar refers to the X-trace. The values for both traces can be obtained using app.Math.Cx.Out.Result.HorizontalFrameStart.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the horizontal offset of the input traces of an XY plot.
HorOffset = app.Math.XY.Out.Result.HorizontalOffset

---

**HorizontalPerStep**

*Property*

**Description**

This is the time between successive sampling instants of the input traces of an X-Y plot.

**Example**
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the value of the horizontal step size for the input traces
' of an X-Y plot.
HorStepXY = app.Math.XY.Out.Result.HorizontalPerStep

### HorizontalResolution Property

**Description**

This is the resolution of the readout of horizontal values of the input traces of an X-Y plot. It is not directly related to the sampling period.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the readout resolution of the horizontal axis of input
' traces of an X-Y plot.

### HorizontalUnits Property

**Description**

Reads the name of the horizontal units of the data of the X-Y plot. The horizontal dimension is not visible on the X-Y plot, but is implicit in the data. If both inputs are normal channel traces, units are in seconds; while if both are FFTs, units are in Hz.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the horizontal units of the input traces of an XY plot.
HUnits = app.Math.XY.Out.Result.HorizontalUnits

### LastEventTime Property

**Description**

Reads the time of the last event.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the time of the last event.
EventLast = app.Math.XY.Out.Result.LastEventTime
PART TWO: REFERENCE

NumFrameDimensions

Property
Description
Reads the dimensionality of trace XY: 2 for a Y-T plot, 3 for an X-Y plot.
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the dimensionality of trace XY.
DimXY = Math.XY.Out.Result.NumFrameDimensions

Samples

Property
Description
Reads the number of points in the input traces of an X-Y plot.
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the number of samples in the input traces of an X-Y plot.
XYSamples = app.Math.XY.Out.Result.Samples

Sweeps

Property
Description
This is the number of sweeps since the last clearance, for example a clearance caused by the last change in time per division.
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the number of sweeps for trace XY.

XFrameStart

Property
Description
Reads the value at the left edge of the X-Y plot.
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the value at the left hand edge for the X-Y plot.
XLeft = app.Math.XY.Out.Result.XFrameStart
CHAPTER TWO

Control Reference

XFrameStop

Property

Description
See the corresponding control variable in app.Acquisition.Cx.Out.Result.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the value at the right hand edge for the X-Y plot.
XRight = app.Math.XY.Out.Result.XFrameStop

XMaxPossible

Property

Description
This is the highest value that an actual array element can have. It will be a little less than XFrameStop.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the maximum possible value of X for trace XY.
XMaxXY = app.Math.XY.Out.Result.XMaxPossible

XMinPossible

Property

Description
This is the lowest value that an actual array element can have. It will be a little greater than XFrameStart.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the minimum possible value of X for trace XY.
XMinXY = app.Math.XY.Out.Result.XMinPossible

XOffset

Property

Description
See the corresponding control variable in app.Acquisition.Cx.Out.Result.

XPerStep

Property

Description
This is the smallest step in the numerical X values that can be read out, whether or not the step has physical meaning. For the basic 8-bit ADC the step is 1/65536 of the Xrange.
PART TWO: REFERENCE

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the step size in X for trace XY.
XStepXY = app.Math.XY.Out.Result.XPerStep

XResolution

Property

Description

X resolution is the actual smallest difference that can be practically resolved. For an 8-bit ADC it is 1/256 of the height of the vertical range. But if 16 averages are set, the resolution is improved by a factor of 4, and it becomes 1/1024 of the vertical range, and for 100 sweeps it becomes 1/2560.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the resolution of the X-axis of trace XY.
XRes = app.Math.XY.Out.Result.Xresolution

XUnits

Property

Description

Reads the name of the units of the horizontal scale of an XY plot. This is not the same as the horizontal unit, app.Math.XY.Out.Result.HorizontalUnits, which refers to the horizontal units of the original traces that contribute to the XY plot.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the units of the horizontal axis of an XY plot.
XUnits = app.Math.XY.Out.Result.Xunits

YFrameStart

Property

Description

Reads the value at the bottom edge for the X-Y plot.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the value at the bottom edge for the X-Y plot.
YBottom = app.Math.XY.Out.Result.YFrameStart
YFrameStop  

**Property**

Description
Reads the value at the top edge for the X-Y plot.

Example

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the value at the top edge for the X-Y plot.
YTop = app.Math.XY.Out.Result.YFrameStop
```

YMaxPossible  

**Property**

Description
This is the highest value that an actual array element can have. It will be a little less than YFrameStop.

Example

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the maximum possible value of Y for trace XY.
YMaxXY = app.Math.XY.Out.Result.YMaxPossible
```

YMinPossible  

**Property**

Description
This is the lowest value that an actual array element can have. It will be a little greater than YFrameStart.

Example

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the minimum possible value of Y for trace XY.
YMinXY = app.Math.XY.Out.Result.YMinPossible
```

YOffset  

**Property**

Description
Please see the corresponding control variable in app.Acquisition.Cx.Out.Result.

YPerStep  

**Property**

Description
This is the smallest step in the numerical Y values that can be read out, whether or not the step has physical meaning. For the basic 8-bit ADC, the step is 1/65536 of the Y-range.
**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the step size in Y for trace XY.
YStepXY = app.Math.XY.Out.Result.YPerStep
```

### YResolution

**Property**

**Description**

Y-resolution is the actual smallest difference that can be practically resolved. For an 8-bit ADC, it is 1/256 of the height of the Y-range. But if 16 averages are set in the channel, the resolution is improved by a factor of 4, and it becomes 1/1024. For 100 sweeps it becomes 1/2560.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the resolution of the Y-axis of trace XY.
```

### YUnits

**Property**

**Description**

Reads the name of the units of the vertical scale of an XY plot.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the units of the vertical axis of an XY plot.
YUnits = app.Math.XY.Out.Result.Yunits
```
Variables of the form `app.Measure` control the parameters P1 through P8, and their associated statistical results and histicons.

Names of the form `app.Measure.Measure("Premote").xxxx` and `app.Measure.Measure("Px").xxxx` are aliases for simpler names, which are described in this section of the manual. Examples of alias pairs are as follows:

```plaintext
```

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ClearAll</code></td>
<td>Action - Resets all parameter setups, turning each of the parameters view to &quot;off&quot;, the MeasurementType to &quot;measure&quot; and the selected paramEngine to &quot;Null&quot;. See <code>Acquisition.Channels</code> for a programming example.</td>
</tr>
<tr>
<td><code>ClearSweeps</code></td>
<td>Action - Clears the accumulated statistics for parameters P1 to P8, as well as the accumulated statistics for their associated histicons.</td>
</tr>
<tr>
<td><code>HelpMarkers</code></td>
<td>Enum - Enables or disables visual markers for the measurement.</td>
</tr>
<tr>
<td><code>HistoOn</code></td>
<td>Bool - Enables or disables the display of histograms.</td>
</tr>
<tr>
<td><code>MeasureMode</code></td>
<td>Enum - Sets the measurement mode to &quot;measure&quot;, &quot;average&quot;, or &quot;peak&quot;.</td>
</tr>
<tr>
<td><code>SetGateToDefault</code></td>
<td>Action - Sets the gate settings to the default values.</td>
</tr>
<tr>
<td><code>StatsOn</code></td>
<td>Bool - Enables or disables the display of statistical results.</td>
</tr>
<tr>
<td><code>StdGateStart</code></td>
<td>Double - Sets the start value of the standard gate.</td>
</tr>
<tr>
<td><code>StdGateStop</code></td>
<td>Double - Sets the stop value of the standard gate.</td>
</tr>
<tr>
<td><code>StdSource</code></td>
<td>Enum - Sets the standard source to &quot;derived&quot;, &quot;internal&quot;, or &quot;null&quot;.</td>
</tr>
</tbody>
</table>

Example

```plaintext
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' get into the custom parameter mode
app.Measure.MeasureMode = "MyMeasure"
app.Measure.ClearAll
app.Measure.StatsOn = True
app.Measure.HistoOn = False

' Configure P1 to measure amplitude
app.Measure.P1.View = True
app.Measure.P1.ParamEngine = "ampl"
```
**PART TWO: REFERENCE**

**HelpMarkers**  
*Enum*

**Description**

Sets/Queries the level of detail for help markers (if any of the selected parameter definitions have help markers). These markers are displayed on the source traces, but only if those traces are viewed simultaneously with the parameter measurements. **Note:** This setting is global for all Px.

**Values**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Detailed</strong></td>
<td>Detailed help markers</td>
</tr>
<tr>
<td><strong>Off</strong></td>
<td>No help markers</td>
</tr>
<tr>
<td><strong>Simple</strong></td>
<td>Simple help markers</td>
</tr>
</tbody>
</table>

**HistOn**  
*Bool*

**Description**

Sets/Queries the visibility of the histicons of the parameters that are viewed.

**MeasureMode**  
*Enum*

**Description**

Sets/Queries the set of parameters to be displayed.

**Values**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MyMeasure</strong></td>
<td>Completely customizable user defined parameter list, .</td>
</tr>
<tr>
<td><strong>Off</strong></td>
<td></td>
</tr>
<tr>
<td><strong>StdHorizontal</strong></td>
<td>Standard horizontal parameters</td>
</tr>
<tr>
<td><strong>StdVertical</strong></td>
<td>Standard vertical parameters</td>
</tr>
</tbody>
</table>

**SetGateToDefault**  
*Action*

**Description**

Sets the measure gate to its default state. Valid only when in either Std. Vertical or Std. Horizontal measurement modes. For MyMeasure, see the equivalent controls under Px.

**StatsOn**  
*Bool*

**Description**

Sets/Queries the display of parameter statistics. **Note:** Statistics are accumulated whether or not they are displayed, i.e., you do not need to set StatsOn = "On" to collect statistics.

**StdGateStart**  
*Double*

**Range:** From 0 to 10, step 0.01

**Description**

Sets/Queries the position of the left limit of the measure gate (in divisions). Valid only in either Std. Vertical or Std. Horizontal measurement modes. For MyMeasure, see the equivalent controls under Px.
**StdGateStop**

*Double*

**Range:** From 0 to 10, step 0.01

**Description**

Sets/Queries the position of the right-hand limit of the measure gate (in divisions). Valid only in either Std. Vertical or Std. Horizontal measurement modes. For MyMeasure see the equivalent controls under Px.

**StdSource**

*Enum*

**Description**

Sets/Queries the channel that is the source of ALL standard voltage or time parameters. In “My Measure” mode, each parameter has its own Source selection, and this setting is ignored.

**Values**

- BadBits
- Bits
- C1
- C2
- C3
- C4
- Ctie
- Eye
- F1
- F2
- F3
- F4
- F5
- F6
- F7
- F8
- M1
- M2
- M3
- M4
- PRBS
- TIE
Names of the form app.Measure.Measure("Premote").xxxx and app.Measure.Measure("Px").xxxx are aliases for simpler names, which are described in the section of the manual devoted to app.Measure. Examples of alias pairs are as follows:


PX

This set of variables controls parameters P1 through P8, and the statistical results and histicons that depend on them. Valid only when the MeasureMode is "MyMeasure", otherwise they are predefined.

<table>
<thead>
<tr>
<th>ArithEngine</th>
<th>Enum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equation</td>
<td>String</td>
</tr>
<tr>
<td>GateStart</td>
<td>Double</td>
</tr>
<tr>
<td>GateStop</td>
<td>Double</td>
</tr>
<tr>
<td>MeasurementType</td>
<td>Enum</td>
</tr>
<tr>
<td>ParamEngine</td>
<td>Enum</td>
</tr>
<tr>
<td>PSource1</td>
<td>Enum</td>
</tr>
<tr>
<td>PSource2</td>
<td>Enum</td>
</tr>
<tr>
<td>Source1</td>
<td>Enum</td>
</tr>
<tr>
<td>Source2</td>
<td>Enum</td>
</tr>
<tr>
<td>View</td>
<td>Bool</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

app.Measure.MeasureMode = "MyMeasure"

' Set parameter P1 to math on parameters.
App.Measure.P1.MeasurementType = "math"

ArithEngine

Description
Sets/Queries the parameter arithmetic function for parameter Px. The setting is only used when Px.MeasurementType is set to "math".

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the parameter arithmetic for P1 to product of two parameters.
app.Measure.MeasureMode = "MyMeasure"
app.Measure.P1.MeasurementType = "math"
app.Measure.P1.ArithEngine = "ParamProduct"
app.Measure.P1.PSource1 = "P2"
app.Measure.P1.PSource2 = "P3"

'of course to get an answer, you should also define P2 and P3
app.Measure.P2.MeasurementType = "measure"
app.Measure.P2.Source1 = "C1"

app.Measure.P3.MeasurementType = "measure"
app.Measure.P3.ParamEngine = "Mean"

Values
MathcadParamArith
ParamConst
ParamDifference
ParamProduct
ParamRatio
ParamScript
ParamSum

Equation
String
Range: Any number of characters.
Description
Reads the text equation that summarizes the setup for parameter Px.

GateStart
Double
Range: From 0 to 10, step 0.01
Description
Sets/Reads the position of the left edge of the measure gate for parameter Px.

GateStop
Double
Range: From 0 to 10, step 0.01
Description
Sets/Reads the position of the right-hand edge of the measure gate for parameter Px.

MeasurementType
Enum
Description
Sets/Queries the measurement type of the parameter Px.

Values
<table>
<thead>
<tr>
<th>math</th>
<th>Measurement is a mathematical combination of one or two other Py,Pz</th>
</tr>
</thead>
<tbody>
<tr>
<td>measure</td>
<td>Standard measurement mode (parametric of a trace waveform)</td>
</tr>
<tr>
<td>WebEdit</td>
<td>Measurement is defined using the Processing Web Editor</td>
</tr>
</tbody>
</table>
PART TWO: REFERENCE

ParamEngine  

Enum

Description

Sets/Queries the parameter (measurement on a trace) for Px. This setting applies only if the MeasurementType control is set to "measure".

Values

Amplitude
Area
Base
Cycles
Delay
DeltaDelay
DeltaPeriodAtLevel
DeltaTimeAtLevel
Duration
DutyAtLevel
DutyCycle
EdgeAtLevel
ExcelParam
ExtinctionRatio
EyeAmplitude
EyeAvgPower
EyeBER
eyecrossing
EyeHeight
EyeOneLevel
EyeQ
EyeWidth
EyeZeroLevel
Fall
Fall8020
FallAtLevel
FirstPoint
Frequency
FrequencyAtLevel
FullWidthAtHalfMaximum
FullWidthAtXX
HalfPeriod
HistogramAmplitude
HistogramBase
HistogramMaximum
HistogramMean
HistogramMedian
HistogramMinimum
HistogramRms
HistogramSdev
HistogramTop
HoldTime
LastPoint
LevelAtX
MathcadParam
MATLABParameter
Maximum
MaximumPopulation
Mean
Median
Minimum
Mode
NarrowBandPhase
NarrowBandPower
npoints
Null
NumberOfModes
OvershootNegative
OvershootPositive
ParamScript
Peaks
PeakToPeak
Percentile
Period
PeriodAtLevel
Phase
PopulationAtX
Range:
Rise
Rise2080
RiseAtLevel
RootMeanSquare
Setup
Skew
StandardDeviation
TIE
TimeAtLevel
Top
TotalPopulation
Width
WidthAtLevel
XAtMaximum
XAtMinimum
XAtPeak

PSource1  Enum

Description
Sets/Queries the first parameter source of parameter Px. This choice is only applicable if the MeasurementType is set to "math". The choices are limited to other parameters.

Values
P1
P2
P3
P4
P5
P6
**P R A R T T W O : R E F E R E N C E**

**PSource2**  
*Enum*

Description

Sets/Queries the second parameter source of the parameter Pₓ, when the MeasurementType is "math".

Values

P₁  
P₂  
P₃  
P₄  
P₅  
P₆  
P₇  
P₈

**Source1**  
*Enum*

Description

Sets/Queries the first trace source of the parameter Pₓ. Used only when MeasurementType = "measure". For MeasurementType = "math", refer to PSource1.

Values

C₁  
C₂  
C₃  
C₄  
F₁  
F₂  
F₃  
F₄  
F₅  
F₆  
F₇  
F₈  
M₁  
M₂  
M₃  
M₄

**Source2**  
*Enum*

Description

Sets/Queries the second trace source of the parameter Pₓ. Only applicable when the MearementType is "measure". The choices are limited to waveforms.

Values

C₁  
C₂  
C₃  
C₄
<table>
<thead>
<tr>
<th>View</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Sets/Queries the display of parameter Px.</td>
</tr>
</tbody>
</table>

**Note:** It is not necessary for a parameter to be displayed in order for it to be used in combinatorial math (i.e., as a source for another parameter in MeasurementType "math").
This path specifies that the selected ParamEngine or ArithEngine control variables are “here.”
RESULT

app.Measure.Px.Out.Result

Properties of the type xxxx.Out.Result.yyyy are those of the last completed acquisition. They are not affected if other CVars are changed after that acquisition was completed. This distinction between "Out.Result" properties and other CVars is most important when the trigger mode is Single or Stopped. You should treat "Out.Result" properties as read-only.

<table>
<thead>
<tr>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>HorizontalResolution</td>
</tr>
<tr>
<td>HorizontalUnits</td>
</tr>
<tr>
<td>Value</td>
</tr>
<tr>
<td>VerticalResolution</td>
</tr>
<tr>
<td>VerticalUnits</td>
</tr>
</tbody>
</table>

**HorizontalResolution Property**

Description
- Reads the readout resolution of horizontal values. This is not directly related to the sample rate.

Example
```vbs
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the readout resolution of the horizontal axis of the trace
' on which the parameter P4 is based.
```

**HorizontalUnits Property**

Description
- Reads the horizontal unit of the trace on which the parameter Px is based.

Example
```vbs
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the horizontal unit for parameter P1.
HorUnit = app.Measure.P1.Out.Result.HorizontalUnits
```

**Value Property**

Description
- Reads the value of the parameter.

Example
```vbs
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the value of the parameter P3.
```
VerticalResolution Property

Description

Vertical resolution is the actual smallest difference that can be practically resolved. For an 8-bit ADC, it is 1/256 of the height of the vertical range (VR). But if 16 averages are set in the channel, the resolution is improved by a factor of 4, and it becomes 1/1024 of VR. For 100 sweeps it becomes 1/2560.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the readout resolution of the vertical axis of the trace
' on which the parameter P4 is based.

VerticalUnits Property

Description

Reads the vertical unit of the trace on which parameter Px is based.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical unit for parameter P1.
STATISTICS

This set of variables controls the statistical summaries that are provided for all the parameters.

RESULT

This set of variables provides data from the histicons that are provided for all the parameters.

<table>
<thead>
<tr>
<th>Base</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>BinPopulations</td>
<td>Property</td>
</tr>
<tr>
<td>Bins</td>
<td>Property</td>
</tr>
<tr>
<td>BinWidth</td>
<td>Property</td>
</tr>
<tr>
<td>FirstPopulatedBin</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalFrameStart</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalFrameStop</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalUnits</td>
<td>Property</td>
</tr>
<tr>
<td>LastPopulatedBin</td>
<td>Property</td>
</tr>
<tr>
<td>Max</td>
<td>Property</td>
</tr>
<tr>
<td>MaxPopulation</td>
<td>Property</td>
</tr>
<tr>
<td>MaxPopulationBin</td>
<td>Property</td>
</tr>
<tr>
<td>Mean</td>
<td>Property</td>
</tr>
<tr>
<td>Min</td>
<td>Property</td>
</tr>
<tr>
<td>OffsetAtLeftEdge</td>
<td>Property</td>
</tr>
<tr>
<td>PeakInfo</td>
<td>Property</td>
</tr>
<tr>
<td>Peaks</td>
<td>Property</td>
</tr>
<tr>
<td>PopulationInside</td>
<td>Property</td>
</tr>
<tr>
<td>PopulationOver</td>
<td>Property</td>
</tr>
<tr>
<td>PopulationUnder</td>
<td>Property</td>
</tr>
<tr>
<td>Rms</td>
<td>Property</td>
</tr>
<tr>
<td>Sdev</td>
<td>Property</td>
</tr>
<tr>
<td>Sweeps</td>
<td>Property</td>
</tr>
<tr>
<td>Top</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalFrameStart</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalFrameStop</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalUnits</td>
<td>Property</td>
</tr>
</tbody>
</table>

Base

Description
Reads the base value of the distribution.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
' Read the base value in the histicon for parameter P1.
**PART TWO: REFERENCE**

---

**BinPopulations**

**Property**

Description

Reads the array of bin populations for the distribution.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
' Collect the number of bins in the histogram of parameter P1.
' Collect the array of bin populations for the histogram of parameter P1.
dim BinPop(100)
for i = 0 to 100
   BinPop(i) =
next

---

**Bins**

**Property**

Description

Reads the number of bins in the histogram.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"

' Read the number of bins in the histogram for parameter P1.

---

**BinWidth**

**Property**

Description

Reads the width of the bins of the histogram.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure

' Inspect the histogram bin width for parameter P8.
CHAPTER TWO

Control Reference

FirstPopulatedBin Property

Description
Reads the position of the leftmost populated bin of the histogram.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
' Read the position of the leftmost populated bin
' of the histogram for parameter P1.

HorizontalFrameStart Property

Description
Queries the left edge of the frame of the histicon for parameter Px.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the horizontal frame start of the histicon for parameter P3.

HorizontalFrameStop Property

Description
Reads the right-hand edge of the frame of the histicon of parameter Px.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the horizontal frame stop of the histicon for parameter P3.

HorizontalUnits Property

Description
Reads the horizontal dimensional or physical units for the histogram. Note: These are nominally identical to the physical units of the parametric measurement that has been histogrammed.
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
' Inspect the horizontal unit for the histicon for parameter P1.
HorUnit =

**LastPopulatedBin Property**

Description

Reads the index of the rightmost populated bin of the histogram.

**Max Property**

Description

Reads the maximum value for the histicon of parameter Px.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
' Read the maximum value for the histicon for parameter P1.
coordinateOfRightmostPopulated =

**MaxPopulation Property**

Description

Reads the population of the most populated bin of the histogram.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the population of the most populated bin
' of the histicon for parameter P6.
BinMaxHP6 =

**MaxPopulationBin Property**

Description

Reads the index of the most populated bin of the histogram.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
' Read the index of the bin with the maximum population
indexOfMaxPopulation =

Mean Property

Description
Reads the mean of the distribution of the histogram.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
' Inspect the mean of the histicon for the parameter P1.
mean = app.Measure.P1.Statistics("histo").Mean

Min Property

Description
Reads the minimum value of the histogram.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
' Read the maximum value for the histicon for parameter P1.
coordinateOfLeftmostPopulated =

OffsetAtLeftEdge Property

Description
Reads the horizontal coordinate of the left edge of the leftmost bin in the histogram.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Look at the left edge offset of the histicon for parameter P1.
leftEdge =

PeakInfo Property

Description
Reads three values about a peak of the histicon of parameter Px. The first two results are the mean and the standard deviation. The third value is unused.
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Collects the information array for second peak of
' the histicon of parameter P3.
ReDim InfoArray2

Peaks

Description
Reads the number of peaks in the distribution (according to a peak identification process).

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the number of peaks found

PopulationInside

Description
Reads the total number of data points inside the frame of the histicon of parameter Px.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the population inside the histicon for parameter P3.

PopulationOver

Description
Reads the number of data points outside the histicon to the right of the last bin in the histogram.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the population to the right of the frame
' of the histicon for parameter P1.
populationInOverflow =
PopulationUnder Property

Description
Reads the number of data points outside the histogram to the left of the leftmost bin.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
' Read the population that are NOT binned because the values were to the left of the leftmost bin
numUnderflows =

Rms Property

Description
Reads the root mean square of the distribution.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
' Read the rms value of the histicon for parameter P1.

Sdev Property

Description
Reads the standard deviation of the distribution.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the standard deviation of the histogrammed distribution

Sweeps Property

Description
Reads the number of sweeps contributing to the histogram.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the number of sweeps for the histicon for parameter P1.
Top

Property

Description
Reads the top value of the histicon for the parameter Px.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the top value in the histicon for parameter P6.

VerticalFrameStart

Property

Description
Reads the value at the bottom of the frame (display graticule) of the histogram.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical frame start of the histicon for parameter P1.

VerticalFrameStop

Property

Description
Reads the value at the top of the frame of the histogram.

VerticalUnits

Property

Description
Reads the vertical units for the histogram. These are almost always "Events."

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical units of the histicon of parameter P4.
RESULT

This set of variables controls the readout of data from the last event in the statistics that are provided for all the parameters.

<table>
<thead>
<tr>
<th>Value</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueArray</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalResolution</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalUnits</td>
<td>Property</td>
</tr>
</tbody>
</table>

**Value**

**Description**
Reads the last recorded value of parameter Px.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the last recorded value for the parameter P3.

**ValueArray**

**Description**
Some measurements return multiple values for a single waveform input. This array contains all of the values.

**VerticalResolution**

**Description**
Reads the vertical resolution of the result.

**VerticalUnits**

**Description**
Reads the vertical unit of measurement for parameter Px.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical unit of measurement for the parameter P1.
VerUnit = app.Measure.P1.Statistics("last").Result.VerticalUnits
This set of variables controls the readout of data about the maximum value in the statistics that are provided for all the parameters.

### HorizontalResolution

**Property**

Description

*Reads the horizontal resolution for the parameter Px.*

Example

```visualBasic
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the horizontal resolution for the parameter P1.
```

### HorizontalUnits

**Property**

Description

*Reads the horizontal units of measurement for parameter Px.*

Example

```visualBasic
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the horizontal unit for the parameter P1.
HorUnit = app.Measure.P1.Statistics("max").Result.HorizontalUnits
```

### Value

**Property**

Description

*Reads the maximum recorded value of parameter Px.*

Example

```visualBasic
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the highest recorded value for the parameter P5.
```
**VerticalResolution**

*Property*

**Description**
Reads the vertical resolution for the parameter Px.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical resolution for the parameter P1.
```

**VerticalUnits**

*Property*

**Description**
Reads the vertical unit of measurement for parameter Px.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical unit for the parameter P5.
VerUnit = app.Measure.P5.Statistics("max").Result.VerticalUnits
```
PART TWO: REFERENCE

RESULT

app.Measure.Px.Statistics("mean").Result

This set of variables controls the readout of data about the mean value in the statistics that are provided for all the parameters.

Result

<table>
<thead>
<tr>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>HorizontalResolution</td>
</tr>
<tr>
<td>HorizontalUnits</td>
</tr>
<tr>
<td>Value</td>
</tr>
<tr>
<td>VerticalResolution</td>
</tr>
<tr>
<td>VerticalUnits</td>
</tr>
</tbody>
</table>

**HorizontalResolution**

Description
Reads the horizontal resolution for parameter Px.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the horizontal resolution for the parameter P1.
HorRes =

**HorizontalUnits**

Description
Reads the horizontal units of measurement for parameter Px.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the horizontal unit for the parameter P1.
HorUnit = app.Measure.P1.Statistics("mean").Result.HorizontalUnits

**Value**

Description
Reads the mean value of parameter Px.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the mean recorded value for the parameter P3.
## VerticalResolution Property

**Description**

Reads the vertical resolution for parameter Px.

**Example**

```vbscript
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical resolution for the parameter P1.
```

## VerticalUnits Property

**Description**

Reads the vertical unit of measurement for the parameter Px.

**Example**

```vbscript
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical unit for the parameter P1.
```
This set of variables controls the readout of data about the minimum value in the statistics that are provided for all the parameters.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>HorizontalResolution</td>
<td>Property</td>
<td>' Visual Basic Script</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Set app = CreateObject(&quot;LeCroy.XStreamDSO&quot;)</td>
</tr>
<tr>
<td></td>
<td>Reads the horizontal resolution for parameter Px.</td>
<td>' Inspect the horizontal resolution for the parameter P1.</td>
</tr>
<tr>
<td>HorizontalUnits</td>
<td>Property</td>
<td>' Visual Basic Script</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Set app = CreateObject(&quot;LeCroy.XStreamDSO&quot;)</td>
</tr>
<tr>
<td></td>
<td>Reads the horizontal units of measurement for parameter Px.</td>
<td>' Inspect the horizontal unit for the parameter P1.</td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td>HorUnit = app.Measure.P1.Statistics(&quot;min&quot;).Result.HorizontalUnits</td>
</tr>
<tr>
<td>Value</td>
<td>Property</td>
<td>' Visual Basic Script</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Set app = CreateObject(&quot;LeCroy.XStreamDSO&quot;)</td>
</tr>
<tr>
<td></td>
<td>Reads the lowest recorded value of parameter Px.</td>
<td>' Inspect the minimum recorded value for the parameter P3.</td>
</tr>
</tbody>
</table>
VerticalResolution Property

Description
Reads the vertical resolution for parameter Px.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical resolution for the parameter P1.

VerticalUnits Property

Description
Reads the vertical unit of measurement for parameter Px.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical unit for the parameter P1.
This set of variables controls the readout of data about the number of data values in the statistics that are provided for all the parameters.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
| **HorizontalResolution**  | Reads the horizontal resolution for parameter Px. | ' Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO") |
|                           |                                                  | ' Inspect the horizontal resolution for the parameter P1.               |
| **HorizontalUnits**       | Reads the horizontal units of measurement for parameter Px. | ' Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO") |
|                           |                                                  | ' Inspect the horizontal unit for the parameter P1.                    |
|                           |                                                  | HorUnit = app.Measure.P1.Statistics("num").Result.HorizontalUnits       |
| **Value**                 | Reads the number of recorded values of parameter Px. | ' Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO") |
|                           |                                                  | ' Inspect the number of values of the parameter P3.                    |
CHAPTER TWO

VerticalResolution

Property

Description
Reads the vertical resolution for parameter Px.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical resolution for the parameter P1.

VerticalUnits

Property

Description
Reads the vertical unit of measurement parameter Px.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical unit for the parameter P1.
This set of variables controls the readout of data about the standard deviation of the values in the statistics that are provided for all the parameters.

**Result**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HorizontalResolution</td>
<td>Reads the horizontal resolution for the parameter Px.</td>
</tr>
<tr>
<td>HorizontalUnits</td>
<td>Reads the horizontal units of measure for parameter Px.</td>
</tr>
<tr>
<td>Value</td>
<td>Reads the standard deviation of the recorded values of parameter Px.</td>
</tr>
</tbody>
</table>

**Example**

### HorizontalResolution

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the horizontal resolution for the parameter P1.
```

### HorizontalUnits

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the horizontal unit for the parameter P1.
HorUnit = app.Measure.P1.Statistics("sdev").Result.HorizontalUnits
```

### Value

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the standard deviation of the recorded values of parameter P5.
' of the parameter P5.
```
VerticalResolution

Description

Reads the vertical resolution for parameter Px.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical resolution for the parameter P1.

VerticalUnits

Description

Reads the vertical units for parameter Px.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the vertical unit for the parameter P1.
MEMORY

Variables of the form app.Memory.xxxx control the memories (M1 to M4). Names of the form app.Memory.Memories("Mx").xxxx are aliases for simpler names, which are described in this section of the manual. Examples of alias pairs are as follows:

app.Memory.Memories("Mx").Zoom = app.Memory.Mx.Zoom

See app.Acquisition.Channels("Cx") for a programming example.

**ClearAllMem**

**Action**

**Description**

Clears the contents of all trace memories.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Clear the contents of all trace memories.
app.Memory.ClearAllMem
MEMORIES

Names of the form app.Memory.Memories("Mx").xxxx are aliases for simpler names, which are described in the section of the manual devoted to app.Memory. Examples of alias pairs are as follows:

app.Memory.Memories("Mx").Zoom = app.Memory.Mx.Zoom

See under Acquisition.Channels for a programming example.
This set of variables controls the memories (M1 to M4).

<table>
<thead>
<tr>
<th>ClearMem</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy</td>
<td>Action</td>
</tr>
<tr>
<td>IncludedInMZgroup</td>
<td>Bool</td>
</tr>
<tr>
<td>LabelsPosition</td>
<td>String</td>
</tr>
<tr>
<td>LabelsText</td>
<td>String</td>
</tr>
<tr>
<td>Source1</td>
<td>Enum</td>
</tr>
<tr>
<td>SourceTimeText</td>
<td>String</td>
</tr>
<tr>
<td>UseDotJoin</td>
<td>Bool</td>
</tr>
<tr>
<td>UseGrid</td>
<td>String</td>
</tr>
<tr>
<td>UserText</td>
<td>String</td>
</tr>
<tr>
<td>View</td>
<td>Bool</td>
</tr>
<tr>
<td>ViewLabels</td>
<td>Bool</td>
</tr>
</tbody>
</table>

**ClearMem**  
Description  
Initiates a clear memory operation for memory Mx.

**Copy**  
Description  

**IncludedInMZgroup**  
Description  
Sets/Queries the inclusion of trace MZ in the group to be multi-zoomed.

Example

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

app.Memory.M2.IncludedInMZgroup = True
```

**LabelsPosition**  
Range:  Any number of characters

**LabelsText**  
Range:  Any number of characters.

**Source1**  
Values
- BadBits
- Bits
- C1
- C2
- C3
- C4
CHAPTER TWO

Control Reference

Eye
F1
F2
F3
F4
F5
F6
F7
F8
M2
M3
M4
PRBS

SourceTimeText

String

Range: Any number of characters.

UseDotJoin

Bool

Description
Sets/Queries the style of memory trace Mx.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set memory trace M2 to dot joining off.
app.Memory.M2.UseDotJoin = False

UseGrid

String

Range: Any number of characters.

Description
Sets/Queries the grid used for memory trace Mx.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set memory trace M2 to use grid YT3.
app.Memory.M2.UseGrid = "YT3"

UserText

String

Range: Any number of characters

View

Bool

Description
Sets/Queries whether memory trace Mx is visible.
**ViewLabels**

**Bool**

**Description**
Sets/Queries whether labels are visible for trace Mx.

**RESULT**

*app.Memory.Mx.Out.Result*

See *app.Acquisition.Cx.Out.Result* for a definition of methods and properties used to access the Mx waveform result.
This set of variables controls zooming of the memory traces (M1 to M4).

### Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>HorPos</td>
<td>Double</td>
</tr>
<tr>
<td>HorZoom</td>
<td>Double</td>
</tr>
<tr>
<td>ResetZoom</td>
<td>Action</td>
</tr>
<tr>
<td>VariableHorZoom</td>
<td>Bool</td>
</tr>
<tr>
<td>VariableVerZoom</td>
<td>Bool</td>
</tr>
<tr>
<td>VerPos</td>
<td>Double</td>
</tr>
<tr>
<td>VerZoom</td>
<td>Double</td>
</tr>
</tbody>
</table>

### Example

**Visual Basic Script**

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

' Save C1 into M1
app.SaveRecall.Waveform.SaveTo = "Memory"
app.SaveRecall.Waveform.SaveSource = "C1"
app.SaveRecall.Waveform.SaveDestination = "M1"
app.SaveRecall.Waveform.DoSave

' Zoom M1 by x10 Horizontally, and x2 Vertically
app.Memory.M1.Zoom.ResetZoom
app.Memory.M1.Zoom.HorZoom = 10.0
app.Memory.M1.Zoom.HorPos = 0.0
app.Memory.M1.Zoom.VerPos = 0.0
```

**HorPos**

- **Type**: Double
- **Range**: From –0.5 to 0.5, step (8 digits)
- **Description**
  
  Sets/Queries the horizontal position of the center of the grid on the zoomed trace Mx. The unit of measure is the screen width, that is, 0.3 means a shift of three of the ten divisions. A positive value moves the trace to the left.

**HorZoom**

- **Type**: Double
- **Range**: From 0.1 to 100000, step (8 digits)
- **Description**
  
  Sets/Queries the horizontal magnification of trace Mx. The magnification will be in a 1-2-5-10 sequence unless variable horizontal magnification has been set.
## PART TWO: REFERENCE

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ResetZoom</strong></td>
<td>Resets the zoom for trace Mx.</td>
</tr>
<tr>
<td><strong>VariableHorZoom</strong></td>
<td>Sets/Queries the ability to zoom horizontally by a continuously variable factor as opposed to a factor that follows a 1-2-5 sequence.</td>
</tr>
<tr>
<td><strong>VariableVerZoom</strong></td>
<td>Sets/Queries the ability to zoom vertically by a continuously variable factor as opposed to a factor that follows a 1-2-5 sequence.</td>
</tr>
<tr>
<td><strong>VerPos</strong></td>
<td>Sets/Queries the vertical position of the center of the grid on zoomed trace Mx. The unit of measure is the screen height, that is, 0.375 means a shift of three of the eight divisions. A positive value moves the trace downwards.</td>
</tr>
<tr>
<td><strong>VerZoom</strong></td>
<td>Sets/Queries the vertical magnification of trace Mx. The magnification will be in a 1-2-5-10 sequence unless variable vertical magnification has been set.</td>
</tr>
</tbody>
</table>
Names of the form app.PassFail("Qremote").xxxx and app.PassFail("Qx").xxxx are aliases for simpler names, which are described in this section of the manual. Examples of alias pairs are as follows:


<table>
<thead>
<tr>
<th>ActionOn</th>
<th>Enum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm</td>
<td>Bool</td>
</tr>
<tr>
<td>EnableActions</td>
<td>Bool</td>
</tr>
<tr>
<td>ExecuteScript</td>
<td>Bool</td>
</tr>
<tr>
<td>PredefinedConditions</td>
<td>Enum</td>
</tr>
<tr>
<td>PrintScreen</td>
<td>Bool</td>
</tr>
<tr>
<td>Pulse</td>
<td>Bool</td>
</tr>
<tr>
<td>Save</td>
<td>Bool</td>
</tr>
<tr>
<td>Stop</td>
<td>Bool</td>
</tr>
<tr>
<td>StopAfter</td>
<td>Integer</td>
</tr>
<tr>
<td>StopTesting</td>
<td>Bool</td>
</tr>
<tr>
<td>Testing</td>
<td>Bool</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Setup Parameter P1 to be the amplitude of C1
app.Measure.MeasureMode = "MyMeasure"
app.Measure.P1.ParamEngine = "Ampl"
app.Measure.P1.Source1 = "C1"
app.Measure.P1.View = True

' Setup Pass/Fail condition Q1 to be a parameter comparison of P1

' condition is false when Ampl(C1) < 100mV
app.PassFail.Q1.Present = True
app.PassFail.Q1.Source1 = "P1"
app.PassFail.Q1.Operator.Limit = 0.1
app.PassFail.Q1.View = True

' Enable the Alarm action, enable actions, and start testing
app.PassFail.EnableActions = True
app.PassFail.ActionOn = "Fail"
app.PassFail.Alarm = True

app.PassFail.PredefinedConditions = "AnyTrue"

### ActionOn

**Enum**

**Description**
Sets/Queries whether a Pass condition or a Fail condition will initiate the pre-selected actions.

See `app.Acquisition.Channels("Cx")` for a programming example.

**Values**
- Fail
- Pass

### Alarm

**Bool**

**Description**
Sets/Queries whether Alarm is included in the PassFail actions.

### EnableActions

**Bool**

**Description**
Sets/Queries whether the selected actions will be executed if the selected PassFail condition is met.

### ExecuteScript

**Bool**

**Description**
Sets/Queries whether ExecuteScript is included in the PassFail actions.

### PredefinedConditions

**Enum**

**Description**
Sets/Queries the logical criteria that must be met in a pass-fail test. For example, the condition "AnyTrue" means that the pass-fail criterion is met if at least one of the test conditions results in a True result.

**Values**
- AllFalse
- AllQ1ToQ4OrAllQ5ToQ8
- AllTrue
- AnyFalse
- AnyQ1ToQ4AndAnyQ5ToQ8
- AnyTrue

### PrintScreen

**Bool**

**Description**
Sets/Queries whether Print Screen is included in the PassFail actions.

### Pulse

**Bool**

**Description**
Sets/Queries whether Pulse is included in the PassFail actions. This action emits a pulse from the Aux output socket.
<table>
<thead>
<tr>
<th>Control Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Save</strong></td>
<td><strong>Bool</strong></td>
</tr>
<tr>
<td>Description</td>
<td>Sets/Queries whether Save is included in the PassFail actions.</td>
</tr>
<tr>
<td><strong>Stop</strong></td>
<td><strong>Bool</strong></td>
</tr>
<tr>
<td>Description</td>
<td>Sets/Queries whether Stop is included in the PassFail actions.</td>
</tr>
<tr>
<td><strong>StopAfter</strong></td>
<td><strong>Integer</strong></td>
</tr>
<tr>
<td>Range:</td>
<td>From 1 to 1000000000, step 1</td>
</tr>
<tr>
<td>Description</td>
<td>Sets/Queries the maximum number of sweeps that will be acquired before testing is halted.</td>
</tr>
<tr>
<td><strong>StopTesting</strong></td>
<td><strong>Bool</strong></td>
</tr>
<tr>
<td>Description</td>
<td>If Enabled, testing will stop after a number of sweeps defined by the StopAfter control.</td>
</tr>
<tr>
<td><strong>Testing</strong></td>
<td><strong>Bool</strong></td>
</tr>
<tr>
<td>Description</td>
<td>Sets/Queries whether PassFail testing is on.</td>
</tr>
</tbody>
</table>
This set of variables controls the tests Q1 through Q8 in the pass fail system.

<table>
<thead>
<tr>
<th>ConditionEngine</th>
<th>Enum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equation</td>
<td>String</td>
</tr>
<tr>
<td>Source1</td>
<td>Enum</td>
</tr>
<tr>
<td>Source2</td>
<td>Enum</td>
</tr>
<tr>
<td>View</td>
<td>Bool</td>
</tr>
</tbody>
</table>

**ConditionEngine**

*Description*

Sets/Queries whether pass-fail test Qx uses mask testing or parameter comparison.

*Values*

- MaskTestCondition
- ParameterCompare

**Equation**

*Range: Any number of characters*

*Description*

Inspects the equation for pass-fail test Qx. A typical equation would be “All P3 < 0.7071”.

**Source1**

*Description*

Sets/Queries the first (parameter) source of pass-fail test Qx.

*Values*

- P1
- P2
- P3
- P4
- P5
- P6
- P7
- P8

**Source2**

*Description*

Sets/Queries the second (waveform) source of pass-fail test Qx.

*Values*

- C1
- C2
- C3
- C4
- F1
- F2
## View

<table>
<thead>
<tr>
<th>View</th>
<th>Bool</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Description**

Sets/Queries whether pass-fail test Qx is visible.
PART TWO: REFERENCE

RESULT

Properties of the type xxxx.Out.Result.yyyy are those of the last completed acquisition. They are not affected if other CVars are changed after that acquisition was completed. This distinction between "Out.Result" properties and other CVars is most important when the trigger mode is Single or Stopped. You should treat "Out.Result" properties as read-only.

Result

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HorizontalResolution</td>
<td>Reads the readout resolution of horizontal values.</td>
</tr>
<tr>
<td>HorizontalUnits</td>
<td>Reads the horizontal unit of the horizontal axis.</td>
</tr>
<tr>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>StatusDescription</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td></td>
</tr>
<tr>
<td>ValueArray</td>
<td></td>
</tr>
</tbody>
</table>

HorizontalResolution Property

Description

Reads the readout resolution of horizontal values.

HorizontalUnits Property

Description

Reads the horizontal unit of the horizontal axis.

Status Property

StatusDescription Property

Value Property

Description

Reads the result of test Qx in the pass-fail system.

ValueArray Property
PREFERENCES

This set of variables controls user preferences for the instrument setup and operation.

<table>
<thead>
<tr>
<th>Preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudibleFeedback</td>
</tr>
<tr>
<td>OffsetControl</td>
</tr>
<tr>
<td>Performance</td>
</tr>
</tbody>
</table>

AudibleFeedback

Description
Sets/Queries whether audible feedback is enabled, to sound when a control is touched.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Turn on the audible feedback function.
app.Preferences.AudibleFeedback = True

OffsetControl

Description
Sets/Queries whether the Vertical Offset constant is in Volts or Divisions when the vertical scale control is changed.

Values
Div
Volts

Performance

Description
Sets/Queries the variable value that controls the Optimization of the instrument in terms of analysis or display. When set to “Analysis” the display is given low priority and will update less frequently. Use this mode where analysis performance is much more important than display rate.

Values
Analysis
AnalysisMid
Default
Display
DisplayMid
This set of variables controls user preferences for the instrument e-mail system. E-mail can be sent when the hardcopy button is pressed, with the hardcopy system appropriately configured. Two standards are supported, SMTP (Simple Mail Transport Protocol), and MAPI (Messaging Application Programming).

<table>
<thead>
<tr>
<th>DefaultRecipient</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>Enum</td>
</tr>
<tr>
<td>OriginatorAddress</td>
<td>String</td>
</tr>
<tr>
<td>SendTestMail</td>
<td>Action</td>
</tr>
<tr>
<td>SMTPServer</td>
<td>String</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure the originator and recipient addresses, replace these with appropriate values for your corporate network.
app.Preferences.Email.DefaultRecipient = "recipientAddress@domain.com"
app.Preferences.Email.Mode = "SMTP"
app.Preferences.Email.OriginatorAddress = "myScope@myDomain.com"
app.Preferences.Email.SMTPServer = "companySMTPServer.companydomain.com"

' Send a simple test mail
app.Preferences.Email.SendTestMail

DefaultRecipient

Range: Any number of characters

Description
Sets/Queries the default recipient of e-mail transmissions.

Mode

Description
Sets/Queries the transmission mode for e-mail.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set e-Mail mode to MAPI.
app.Preferences.EMail.Mode = "MAPI"
Values

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPI</td>
<td>Messaging Application Programming Interface (Uses Outlook Express by default)</td>
</tr>
<tr>
<td>SMTP</td>
<td>Simple Mail Transfer Protocol, requires an SMTP server</td>
</tr>
</tbody>
</table>

**OriginatorAddress**

*String*

**Description**

Sets/Queries the originator address for e-mail. This can be any address, and will be used when the recipient replies to an e-mail. The instrument doesn't necessarily have to have its own E-Mail account in order to use this.

**SendTestMail**

*Action*

**Description**

Sends a message by e-mail to test the system.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Send an e-Mail message to test the system.
app.Preferences.EMail.SendTestMail

**SMTPServer**

*String*

**Description**

Sets/Queries the name of the SMTP Server for e-mail. Ask your system administrator if you are unsure of what value to set this to.
Controls for the Save/Recall subsystem. Includes nodes for saving and recalling Waveforms and Panels (Setups).
These are controls for Saving and Recalling instrument setups.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoRecallDefaultNvlPanel</td>
<td>Action</td>
</tr>
<tr>
<td>DoRecallDefaultPanel</td>
<td>Action</td>
</tr>
<tr>
<td>DoRecallPanel</td>
<td>Action</td>
</tr>
<tr>
<td>DoSavePanel</td>
<td>Action</td>
</tr>
<tr>
<td>InternalName1</td>
<td>String</td>
</tr>
<tr>
<td>InternalName2</td>
<td>String</td>
</tr>
<tr>
<td>InternalName3</td>
<td>String</td>
</tr>
<tr>
<td>InternalName4</td>
<td>String</td>
</tr>
<tr>
<td>InternalName5</td>
<td>String</td>
</tr>
<tr>
<td>InternalName6</td>
<td>String</td>
</tr>
<tr>
<td>PanelFilename</td>
<td>FileName</td>
</tr>
<tr>
<td>RecallInternal1</td>
<td>Action</td>
</tr>
<tr>
<td>RecallInternal2</td>
<td>Action</td>
</tr>
<tr>
<td>RecallInternal3</td>
<td>Action</td>
</tr>
<tr>
<td>RecallInternal4</td>
<td>Action</td>
</tr>
<tr>
<td>RecallInternal5</td>
<td>Action</td>
</tr>
<tr>
<td>RecallInternal6</td>
<td>Action</td>
</tr>
<tr>
<td>SaveInternal1</td>
<td>Action</td>
</tr>
<tr>
<td>SaveInternal2</td>
<td>Action</td>
</tr>
<tr>
<td>SaveInternal3</td>
<td>Action</td>
</tr>
<tr>
<td>SaveInternal4</td>
<td>Action</td>
</tr>
<tr>
<td>SaveInternal5</td>
<td>Action</td>
</tr>
<tr>
<td>SaveInternal6</td>
<td>Action</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Reset to default setup
app.SaveRecall.Setup.DoRecallDefaultPanel

' Store the current setup into the first of the 6 setup stores.
app.SaveRecall.Setup.InternalName1 = "My Setup1"
app.SaveRecall.Setup.SaveInternal1

DoRecallDefaultNvlPanel      Action

Description
Recalls the factory set NVL (preference) panel settings. These are controls that are not affected when
the default panel is recalled, and includes items such as the color preferences, remote control
preferences, etc. Caution: Use with care, especially when invoking by means of the VBS? remote
command via GPIB or TCP/IP, which could result in the controller being disconnected when the default
port is selected.
**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

'Recall the factory default nvl panel settings.
app.SaveRecall.Setup.DoRecallDefaultNvlPanel

**DoRecallDefaultPanel**

**Action**

**Description**
Recalls the factory set panel settings.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

'Recall the factory default panel settings.
app.SaveRecall.Setup.DoRecallDefaultPanel

**DoRecallPanel**

**Action**

**Description**
Recall the panel file named in the PanelFilename control.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Create the filename for the next panel setup to be recalled.
app.SaveRecall.Setup.PanelFilename = "Setup89"

'Recall the panel setup from the named file.
app.SaveRecall.Setup.DoRecallPanel

**DoSavePanel**

**Action**

**Description**
Saves the current panel settings to the previously specified file. If the filename already exists, the file will be overwritten without a prompt.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Create the filename for the next panel setup save.
app.SaveRecall.Setup.PanelFilename = "TestSave"
'Save the panel setup to the named file.
app.SaveRecall.Setup.DoSavePanel

InternalName1
String
Range: Any number of characters
Description
Sets/Queries the name of internal panel setup memory 1.

InternalName2
String
Range: Any number of characters
Description
Sets/Queries the name of internal panel setup memory 2.

InternalName3
String
Range: Any number of characters
Description
Sets/Queries the name of internal panel setup memory 3.

InternalName4
String
Range: Any number of characters
Description
Sets/Queries the name of internal panel setup memory 4.

InternalName5
String
Range: Any number of characters
Description
Sets/Queries the name of internal panel setup memory 5.

InternalName6
String
Range: Any number of characters
Description
Sets/Queries the name of internal panel setup memory 6.

PanelFilename
FileName
Range: Any number of characters
Description
Sets/Queries the current filename for saving a panel setup. An “.lss” extension is automatically appended if not supplied.

RecallInternal1
Action
Description
Recalls the settings that are stored in internal panel memory 1.
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Recall the settings from internal panel memory 1.
app.SaveRecall.Setup.RecallInternal1

RecallInternal2
Action
Description
Recalls the settings that are stored in internal panel memory 2.

RecallInternal3
Action
Description
Recalls the settings that are stored in internal panel memory 3.

RecallInternal4
Action
Description
Recalls the settings that are stored in internal panel memory 4.

RecallInternal5
Action
Description
Recalls the settings that are stored in internal panel memory 5.

RecallInternal6
Action
Description
Recalls the settings that are stored in internal panel memory 6.

SaveInternal1
Action
Description
Saves the current instrument settings into internal panel memory 1.
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Save the current settings into internal panel memory 1.
app.SaveRecall.Setup.SaveInternal1

SaveInternal2
Action
Description
Saves the current instrument settings into internal panel memory 2.
CHAPTER TWO

Control Reference

SaveInternal3
Description
Saves the current instrument settings into internal panel memory 3.

SaveInternal4
Description
Saves the current instrument settings into internal panel memory 4.

SaveInternal5
Description
Saves the current instrument settings into internal panel memory 5.

SaveInternal6
Description
Saves the current instrument settings into internal panel memory 6.
These controls are used to manage files and folders, including the ability to create and delete folders, and to delete files.

### Utilities

<table>
<thead>
<tr>
<th><strong>Utilities</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CreateDir</strong></td>
</tr>
<tr>
<td><strong>DeleteAll</strong></td>
</tr>
<tr>
<td><strong>DeleteAllWithPrompt</strong></td>
</tr>
<tr>
<td><strong>DeleteFile</strong></td>
</tr>
</tbody>
</table>

#### CreateDir

**Description**

Creates the directory specified in the Directory control.

**Example**

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Create a named directory
app.SaveRecall.Utilities.Directory = "C:\MyDir"
app.SaveRecall.Utilities.CreateDir
```

#### DeleteAll

**Description**

Deletes all files in the directory specified by the Directory control without a cautionary prompt. Use with care! Files cannot be recovered if deleted accidentally.

**Example**

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Delete all files without showing a yes/no prompt.
app.SaveRecall.Utilities.Directory = "C:\MyDir"
app.SaveRecall.Utilities.DeleteAll
```

#### DeleteAllWithPrompt

**Description**

Deletes all files, but only after showing a prompt that allows the action to be abandoned. Until this prompt is acknowledged by the user, automation control is blocked.

**Example**

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
```
Delete all files after showing a yes/no prompt.

app.SaveRecall.Utilities.Directory = "C:\MyDir"
app.SaveRecall.Utilities.DeleteAllWithPrompt

DeleteFile

Description

Deletes the file named by the Filename control.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Delete the named file
app.SaveRecall.Utilities.Filename = "C:\MyDir\MyFile.txt"
app.SaveRecall.Utilities.DeleteFile
Contains controls used for saving and recalling waveforms.

<table>
<thead>
<tr>
<th>Delimiter</th>
<th>Enum</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoRecall</td>
<td>Action</td>
</tr>
<tr>
<td>DoSave</td>
<td>Action</td>
</tr>
<tr>
<td>RecallDestination</td>
<td>Enum</td>
</tr>
<tr>
<td>RecallFrom</td>
<td>Enum</td>
</tr>
<tr>
<td>RecallSource</td>
<td>Enum</td>
</tr>
<tr>
<td>SaveDestination</td>
<td>Enum</td>
</tr>
<tr>
<td>SaveSource</td>
<td>Enum</td>
</tr>
<tr>
<td>SaveTo</td>
<td>Enum</td>
</tr>
<tr>
<td>SubFormat</td>
<td>Enum</td>
</tr>
<tr>
<td>TraceTitle</td>
<td>String</td>
</tr>
<tr>
<td>WaveFormat</td>
<td>Enum</td>
</tr>
<tr>
<td>WaveformDir</td>
<td>FileName</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Save C1 into M1
app.SaveRecall.Waveform.SaveTo = "Memory"
app.SaveRecall.Waveform.SaveSource = "C1"
app.SaveRecall.Waveform.SaveDestination = "M1"
app.SaveRecall.Waveform.DoSave

' Save C2 into C:\MyDir in binary format
app.SaveRecall.Waveform.SaveTo = "File"
app.SaveRecall.Waveform.SaveSource = "C2"
app.SaveRecall.Waveform.WaveformDir = "C:\MyDir"
app.SaveRecall.Waveform.WaveFormat = "Binary"
app.SaveRecall.Waveform.DoSave

' Save C3 into C:\MyDir in text format
app.SaveRecall.Waveform.SaveTo = "File"
app.SaveRecall.Waveform.SaveSource = "C3"
app.SaveRecall.Waveform.WaveformDir = "C:\MyDir"
app.SaveRecall.Waveform.WaveFormat = "ASCII"
app.SaveRecall.Waveform.DoSave
Delimiter

Description
Sets/Queries the delimiter to use when saving data in ASCII text mode.

Values
- Comma
- Semicolon
- Space
- Tab

DoRecall

Description
Recall waveform data into a trace memory. Source can be either an internal memory (M1 to M4), or a file on a mass-storage device, depending on the state of the “RecallFrom” control.

DoSave

Description
Save waveform data into an internal memory, or file on a mass-storage device, using the pre-specified source and destination.

RecallDestination

Description
Sets/Queries the destination for waveform recall. When the DoRecall action is executed the waveform will be transferred into this destination trace.

Values
- M1
- M2
- M3
- M4

RecallFrom

Description
Sets/Queries the type of source for waveform recall.

Values
<table>
<thead>
<tr>
<th>File</th>
<th>Recall from file on a mass-storage device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>Recall from one of the internal memories (M1..M4)</td>
</tr>
</tbody>
</table>

RecallSource

Description
Sets/Queries the source for recalling waveform data. Used only when recalling from an internal memory, with RecallSource = “Memory”.

Values
- M1
- M2


**PART TWO: REFERENCE**

**SaveDestination**

*Enum*

Description

Sets/Queries the destination to which waveform data will be saved. Used only when `SaveTo = "Memory"`.

Example

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Setup to store trace C2 into M4 and perform the save operation

app.SaveRecall.Waveform.SaveTo = "Memory"
app.SaveRecall.Waveform.SaveSource = "C2"
app.SaveRecall.Waveform.SaveDestination = "M4"
```

Values

- M1
- M2
- M3
- M4

**SaveSource**

*Enum*

Description

Sets/Queries the source from which waveform data will be saved.

Example

```vbscript
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the destination to memory for waveform save.
app.SaveRecall.Waveform.SaveTo = "Memory"

' Set the source to C2, for saving a waveform.
app.SaveRecall.Waveform.SaveSource = "C2"

' Set the destination to memory M4, for saving a waveform.
app.SaveRecall.Waveform.SaveDestination = "M4"

' Save waveform data as previously specified.
app.SaveRecall.Waveform.DoSave
```

Values

- AllDisplayed
- BadBits
- Bits
- C1
**SaveTo**

**Enum**

Description

Sets/Queries the type of destination for waveform save.

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the destination to Memory for waveform save.
app.SaveRecall.Waveform.SaveTo = "Memory"
```

Values

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>Save into file on a mass-storage device</td>
</tr>
<tr>
<td>Memory</td>
<td>Save into an internal memory (M1..M4)</td>
</tr>
</tbody>
</table>

**SubFormat**

**Enum**

Description

Sets/Queries the sub-format of the waveform data when saving in ASCII mode. Defines whether data is saved as Amplitude values only, Time and Amplitude pairs, or a standard format that includes the header.

Values

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmplitudeOnly</td>
<td></td>
</tr>
<tr>
<td>Header</td>
<td></td>
</tr>
<tr>
<td>TimeAmplitude</td>
<td></td>
</tr>
</tbody>
</table>

**TraceTitle**

**String**

Range: Any number of characters.

Description

Sets/Queries the title (prefix) to use when naming saved traces. This prefix will have the family (sequence) number appended to it when forming the filename.
PART TWO: REFERENCE

WaveFormat

Description
Sets/Queries the format to use when saving waveform data into a file. “Binary” is the most efficient, storing one or two bytes per data sample, depending upon the number of significant bits. In ASCII mode, the Subformat and Delimiter controls define the data format.

Values

<table>
<thead>
<tr>
<th>Enum</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>Plain ASCII files with choice of various delimiters</td>
</tr>
<tr>
<td>Binary</td>
<td>LeCroy's standard binary waveform format</td>
</tr>
<tr>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>MathCad</td>
<td></td>
</tr>
<tr>
<td>MATLAB</td>
<td></td>
</tr>
</tbody>
</table>

WaveformDir

FileName

Description
Sets/Queries the directory for storing waveform files.
These controls are used for SDA (Serial Data Analyzer) models only.

<table>
<thead>
<tr>
<th>Control</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERParamsOn</td>
<td>Bool</td>
</tr>
<tr>
<td>CalcType</td>
<td>Enum</td>
</tr>
<tr>
<td>ClockSource</td>
<td>Enum</td>
</tr>
<tr>
<td>DarkCalLevel</td>
<td>Double</td>
</tr>
<tr>
<td>DataSource</td>
<td>Enum</td>
</tr>
<tr>
<td>ErrorMapOn</td>
<td>Bool</td>
</tr>
<tr>
<td>EyeMode</td>
<td>Enum</td>
</tr>
<tr>
<td>FindFrequency</td>
<td>Action</td>
</tr>
<tr>
<td>FindScale</td>
<td>Action</td>
</tr>
<tr>
<td>HiPassFreq</td>
<td>Double</td>
</tr>
<tr>
<td>LowPassFreq</td>
<td>Double</td>
</tr>
<tr>
<td>MaskFailTraceOn</td>
<td>Bool</td>
</tr>
<tr>
<td>MaskFileName</td>
<td>String</td>
</tr>
<tr>
<td>NumPatternBits</td>
<td>Integer</td>
</tr>
<tr>
<td>PLLOn</td>
<td>Bool</td>
</tr>
<tr>
<td>RefReceiver</td>
<td>Bool</td>
</tr>
<tr>
<td>SDAMode</td>
<td>Enum</td>
</tr>
<tr>
<td>ShowFailLocation</td>
<td>Bool</td>
</tr>
<tr>
<td>SignalFrequency</td>
<td>Double</td>
</tr>
<tr>
<td>SignalMode</td>
<td>Enum</td>
</tr>
<tr>
<td>SignalType</td>
<td>Enum</td>
</tr>
<tr>
<td>StartN</td>
<td>Integer</td>
</tr>
<tr>
<td>StepN</td>
<td>Integer</td>
</tr>
<tr>
<td>StopN</td>
<td>Integer</td>
</tr>
<tr>
<td>Units</td>
<td>Enum</td>
</tr>
<tr>
<td>UserSignal</td>
<td>Enum</td>
</tr>
<tr>
<td>VerAutoFit</td>
<td>Bool</td>
</tr>
</tbody>
</table>

**BERParamsOn**

Description

Sets/Queries whether the bit error rate parameters are shown.

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the BER parameters on.
app.SDA.BERParamsOn = True
```

**CalcType**

Description

Sets/Queries the type of SDA calculation.

Example
```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Sets the calculation type as peak to peak.
app.SDA.CalcType = "PeaktoPeak"

Values
Average
PeaktoPeak

ClockSource

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets/Queries the clock source trace for SDA.</td>
</tr>
</tbody>
</table>

Example
```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the instrument into SDA mode.
app.SDA.SDAMode = "MaskTest"
' Set the clock source as trace C2.
app.SDA.ClockSource = "C2"

Values
C1
C2
C3
C4
F1
F2
F3
F4
F5
F6
F7
F8
M1
M2
M3
M4
PRBS
```

<table>
<thead>
<tr>
<th>DarkCalLevel</th>
<th>Double</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range: From –0.05 to 0.05, step 1e-006</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Sets/Queries the dark calibration level.</td>
<td></td>
</tr>
</tbody>
</table>
**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the dark calibration level to 0.01.
app.SDA.DarkCalLevel = 0.01
' Visual Basic Script
```

**Description**

Sets/Queries the data source trace for SDA.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the instrument into SDA mode.
app.SDA.SDAMode = "MaskTest"
' Set the data source as trace C3.
app.SDA.DataSource = "C3"
```

**Values**

- C1
- C2
- C3
- C4
- F1
- F2
- F3
- F4
- F5
- F6
- F7
- F8
- M1
- M2
- M3
- M4
- PRBS

**ErrorMapOn**

**Bool**

**Description**

Sets/Queries whether the error map is on.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
```
' Set the error map on.
app.SDA.ErrorMapOn = True

EyeMode

Description
Sets/Queries the type of eye diagram.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the eye mode as traditional.
app.SDA.EyeMode = "Traditional"

Values
Sequential
Traditional

FindFrequency

Description
Instruct the instrument to find the frequency of the signal.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Find the frequency.
app.SDA.FindFrequency

FindScale

Description
Find the scale for SDA Jitter NCycle vs. N Plot.

HiPassFreq

Double
Range: From 10 to 1e+010, step 1
Description
Set the SDA jitter filter high-pass frequency.

LowPassFreq

Double
Range: From 10 to 1e+010, step 1
Description
Set the SDA jitter filter low-pass frequency.

MaskFailTraceOn

Bool
Description
Sets/Queries whether the failed section of the trace should be shown.
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set mask fail trace on.
app.SDA.MaskFailTraceOn = True

**MaskFileName**

*String*

Range: Any number of characters.

Description

Sets/Queries the current mask filename.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the instrument into SDA mode.
app.SDA.SDAMode = "MaskTest"
' Set the mask filename.
app.SDA.MaskFileName = "D:\XStreamMasks\FIB4181"

**NumPatternBits**

*Integer*

Range: From 1 to 2147483647, step 1

Description

Number of bits in the data pattern. Used for the periodic jitter parameter.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the number of bits as 32.
app.SDA.NumPatternBits = 32

**PLLOn**

*Bool*

Description

Sets/Queries the use of a PLL to track the clock frequency.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the PLL off.
app.SDA.PLLOn = False
RefReceiver

Description
Sets/Queries the status of a reference receiver. With an optical-to-electrical converter connected to the
instrument, this variable enables or disables the use of information from the receiver.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set reference receiver as on.
app.SDA.RefReceiver = True

SDAMode

Description
Sets/Queries the mode of operation of the instrument. Values: Scope, MaskTest, Jitter, BER, Clock,
Summary for SDA+ASDA. Values: Scope, MaskTest for SDM

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the instrument into Mask Test mode.
app.SDA.SDAMode = "MaskTest"

Values (SDA, ASDA)
Scope
MaskTest
Jitter
BER
Clock
Summary

Values (SDM)
Scope
MaskTest

ShowFailLocation

Description
Sets/Queries whether mask failures are to be shown by markers.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Show locations of failures.
app.SDA.ShowFailLocation = True
SignalFrequency

*Double*

Range: From 5e+007 to 4e+010, step 1

**Description**

Sets/Queries the signal frequencies.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

'Set the signal frequency to 15 MHz.
app.SDA.SignalFrequency = 15e6

SignalMode

*Enum*

**Description**

Values for this control depend upon the SDA standard set by the “SignalType” control.

**Values**

- Receiver
- TransAbs
- TransNrm

SignalType

*Enum*

**Description**

Sets/Queries the signal type for SDA.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

'Set the instrument into SDA mode.
app.SDA.SDAMode = "MaskTest"
'Set signal type as STM16.
app.SDA.SignalType = "STM16"

**Values**

- 1000BaseCX
- 1000BaseLX
- 1000BaseSX
- 1000baseX
- 10GBASELX4
- Custom
- DVI
- FC1063
- FC133
- FC2125
- FC266
- FC531
- IEEE1394b
PART TWO: REFERENCE

Infini2.5Gbs
OC1
OC12
OC3
OC48
PCIExpress
SATA1.5
STM1
STM16
STM4
STS1Eye
STS3Interface
STS3Transmit
USB2.0
XAUI

StartN

Range: From 1 to 100000, step 1

Description
Sets/Queries the unit interval where measurement is to start.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the UI where measurement is to start.
app.SDA.StartN = 5

StepN

Range: From 1 to 100000, step 1

Description
Sets/Queries the gap between measurement groups, in unit intervals.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the step between measurements.
app.SDA.StepN = 2

StopN

Range: From 1 to 100000, step 1

Description
Sets/Queries the unit interval where measurement is to stop.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
" Set the UI where measurement is to stop.
app.SDA.StopN = 5000

Units

<table>
<thead>
<tr>
<th>Description</th>
<th>Enum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets/Queries the unit of measure.</td>
<td></td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the units as UI
app.SDA.Units = "UI"

Values
S
UI

UserSignal

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A trace selected by the user to be displayed with horizontal scaling set by the Mask Test failed bit locator. For example, data source is C1 and user signal is C2. There is a mask failure on the data signal C2 between 40 ns and 80 ns. C2 would zoom to 40 ns–80 ns.</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the instrument into SDA mode.
app.SDA.SDAMode = "MaskTest"

' Set the user signal source as trace C4.
app.SDA.UserSignal = "C4"

Values
C1
C2
C3
C4
F1
F2
F3
F4
F5
F6
F7
F8
VerAutoFit

Description

Controls whether data is fit vertically to the mask automatically.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set vertical auto fitting off.
app.SDA.VerAutoFit = False
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsLevel</td>
<td>Double</td>
</tr>
<tr>
<td>AxisXRotation</td>
<td>Integer</td>
</tr>
<tr>
<td>AxisYRotation</td>
<td>Integer</td>
</tr>
<tr>
<td>BitsInLocator</td>
<td>Integer</td>
</tr>
<tr>
<td>C1ReceiverStandard</td>
<td>Enum</td>
</tr>
<tr>
<td>C2ReceiverStandard</td>
<td>Enum</td>
</tr>
<tr>
<td>C3ReceiverStandard</td>
<td>Enum</td>
</tr>
<tr>
<td>C4ReceiverStandard</td>
<td>Enum</td>
</tr>
<tr>
<td>ClearSweeps</td>
<td>Action</td>
</tr>
<tr>
<td>ClockTIESlope</td>
<td>Enum</td>
</tr>
<tr>
<td>DarkCal</td>
<td>Action</td>
</tr>
<tr>
<td>DarkCalLevel</td>
<td>Double</td>
</tr>
<tr>
<td>DataSource</td>
<td>Enum</td>
</tr>
<tr>
<td>EyeMode</td>
<td>Enum</td>
</tr>
<tr>
<td>EyeThresholdType</td>
<td>Enum</td>
</tr>
<tr>
<td>FailCursorsOn</td>
<td>Bool</td>
</tr>
<tr>
<td>FailedList</td>
<td>Enum</td>
</tr>
<tr>
<td>FailedSymbolsFilter</td>
<td>Enum</td>
</tr>
<tr>
<td>LabelsPosition</td>
<td>String</td>
</tr>
<tr>
<td>LabelsText</td>
<td>String</td>
</tr>
<tr>
<td>MaskFailTraceOn</td>
<td>Bool</td>
</tr>
<tr>
<td>MaskFailX</td>
<td>Double</td>
</tr>
<tr>
<td>MaskFailY</td>
<td>Double</td>
</tr>
<tr>
<td>MaskType</td>
<td>Enum</td>
</tr>
<tr>
<td>MaxFailures</td>
<td>Integer</td>
</tr>
<tr>
<td>MeasurementMode</td>
<td>Enum</td>
</tr>
<tr>
<td>MonochromeEye</td>
<td>Enum</td>
</tr>
<tr>
<td>PercentLevel</td>
<td>Integer</td>
</tr>
<tr>
<td>Persist3DQuality</td>
<td>Enum</td>
</tr>
<tr>
<td>Persisted</td>
<td>Bool</td>
</tr>
<tr>
<td>Persistence3d</td>
<td>Bool</td>
</tr>
<tr>
<td>PersistenceMonoChrome</td>
<td>Bool</td>
</tr>
<tr>
<td>PersistenceSaturation</td>
<td>Integer</td>
</tr>
<tr>
<td>PersistenceTime</td>
<td>Enum</td>
</tr>
<tr>
<td>RefReceiver</td>
<td>Bool</td>
</tr>
<tr>
<td>SDAMode</td>
<td>Enum</td>
</tr>
<tr>
<td>ShowFailLocation</td>
<td>Bool</td>
</tr>
<tr>
<td>ShowLastTrace</td>
<td>Bool</td>
</tr>
<tr>
<td>SignalFrequency</td>
<td>Double</td>
</tr>
<tr>
<td>SignalType</td>
<td>Enum</td>
</tr>
<tr>
<td>SliceWidth</td>
<td>Integer</td>
</tr>
<tr>
<td>Stop</td>
<td>Bool</td>
</tr>
<tr>
<td>TrackMaskFail</td>
<td>Action</td>
</tr>
<tr>
<td>UseDotJoin</td>
<td>Bool</td>
</tr>
<tr>
<td>UseGrid</td>
<td>String</td>
</tr>
<tr>
<td>UserSignal</td>
<td>Enum</td>
</tr>
<tr>
<td>VerAutoFit</td>
<td>Bool</td>
</tr>
<tr>
<td>View</td>
<td>Bool</td>
</tr>
<tr>
<td>ViewLabels</td>
<td>Bool</td>
</tr>
<tr>
<td>XMargin</td>
<td>Integer</td>
</tr>
<tr>
<td>YMargin</td>
<td>Integer</td>
</tr>
</tbody>
</table>
PART TWO: REFERENCE

AbsLevel

Double

Range: From –100 to 100, step 1e-007

Description
Voltage level to set in the TIE at level.

AxisXRotation

Integer

Range: From –90 to 90, step 1

Description
Refer to the corresponding variable in Display, using SDA.BadBits.Persisted.

AxisYRotation

Integer

Range: From –90 to 90, step 1

Description
Refer to the corresponding variable in Display, using SDA.BadBits.Persisted.

BitsInLocator

Integer

Range: From 1 to 101, step 1

Description
Number of bits to display in the Mask Test bottom grid, where the bit that failed the mask is displayed in the center of the grid. For example, BitsInLocator = 5 means that 2 bits before and 2 bits after the failing bit are going to be displayed in the bottom grid.

C1ReceiverStandard

Enum

Description
Read only. Indicates what reference receiver filter the optical-to-electrical converter uses on the SDA signal.

Values
DISABLED
FC1063
FC2125
L1000BASE
OC12
OC3
OC48
OTHER

C2ReceiverStandard

Enum

Description
Read only. Indicates what reference receiver filter the optical-to-electrical converter uses on the SDA signal.

Values
DISABLED
FC1063
FC2125
L1000BASE
OC12
CHAPTER TWO

OC3
OC48
OTHER

C3ReceiverStandard

Description
Read only. Indicates what reference receiver filter the optical-to-electrical converter uses on the SDA signal.

Values
DISABLED
FC1063
FC2125
L1000BASE
OC12
OC3
OC48
OTHER

C4ReceiverStandard

Description
Read only. Indicates what reference receiver filter the optical-to-electrical converter uses on the SDA signal.

Values
DISABLED
FC1063
FC2125
L1000BASE
OC12
OC3
OC48
OTHER

ClearSweeps

Description
Refer to the corresponding variable in Acquisition.Cx, using SDA.BadBits.Persisted.

ClockTIESlope

Description
Sets the preferred edge for eye diagram alignment.

Values
Both
Negative
Positive
**PART TWO: REFERENCE**

**DarkCal**

**Description**
Press to start a no-signal calibration of an optical-to-electrical probe.

**DarkCalLevel**

**Double**

**Range:** From -0.05 to 0.05, step 1e-006

**Description**
A read only variable that shows the dark level after calibration.

**DataSource**

**Enum**

**Description**
Using SDA.BadBits, refer to the corresponding variable in SDA.

**Values**
- C1
- C2
- C3
- C4
- F1
- F2
- F3
- F4
- F5
- F6
- F7
- F8
- M1
- M2
- M3
- M4
- PRBS

**EyeMode**

**Enum**

**Description**
Refer to the corresponding variable in SDA, using SDA.BadBits.

**Values**
- Sequential
- Traditional

**EyeThresholdType**

**Enum**

**Description**
Threshold type for Eye Diagram TIE level. Change the Eye Diagram TIE settings to match the expected eye crossings, for example, SDA default for eye crossing at 50%. You can set it to EyeThresholdType = Percent and PercentLevel=32.

**Values**
- Absolute
- Percent
### FailCursorsOn

**Type:** *Bool*

**Description:**
Turns on round cursors surrounding points in the eye diagram that penetrate the mask.

### FailedList

**Type:** *Enum*

**Description:**
Reads the indices of the bits that failed the mask test.

### FailedSymbolsFilter

**Type:** *Enum*

**Description:**
Sets what failed indices to get in the FailedList. Values: All, NearXY. For NearXY, see `MaskFailX` and `MaskFailY`.

### LabelsPosition

**Type:** *String*

**Description:**
Refer to the corresponding variable in `Acquisition.Cx`, using SDA.BadBitsPersisted.

### LabelsText

**Type:** *String*

**Description:**
Refer to the corresponding variable in `Acquisition.Cx`, using SDA.BadBitsPersisted.

### MaskFailTraceOn

**Type:** *Bool*

**Description:**
Refer to the corresponding variable in `SDA`, using SDA.BadBits.

### MaskFailX

**Type:** *Double*

**Description:**
Shows the relative horizontal position of the selected mask failure cursors. For example, the user clicks on a mask failure cursor in the middle of the display. `MaskFailX` shows the selected horizontal position that would be 0.5.

### MaskFailY

**Type:** *Double*

**Description:**
Shows the relative vertical position of the selected mask failure cursors. For example, the user clicks on a mask failure cursor in the middle of the display.

### MaskType

**Type:** *Enum*

**Description:**
Refer to the corresponding variable in `SDA`, using SDA.BadBits.

**Values:**
- Absolute
Part Two: Reference

MaxFailures

*Integer*

Range: From 1 to 10000, step 1

Description

Sets the number of failed bits to display in FailedList.

MeasurementMode

*Enum*

Description

Display a set of parameters measuring various properties of the eye diagram.

Values

Amplitude
Eye
Off
Timing

MonochromeEye

*Enum*

Values

Colored
Monochrome

PercentLevel

*Integer*

Range: From 0 to 100, step 1

Persist3DQuality

*Enum*

Description

Refer to the corresponding variable in *Acquisition.Cx*, using SDA.BadBits.Persisted.

Values

Shaded
Solid
WireFrame

Persisted

*Bool*

Description

Refer to the corresponding variable in *Acquisition.Cx*, using SDA.BadBits.Persisted.

Persistence3d

*Bool*

Description

Refer to the corresponding variable in *Acquisition.Cx*, using SDA.BadBits.Persisted.

PersistenceMonoChrome

*Bool*

Description

Refer to the corresponding variable in *Acquisition.Cx*, using SDA.BadBits.Persisted.

PersistenceSaturation

*Integer*

Range: From 1 to 100, step 1
### PersistenceTime

**Enum**

Description
Refer to the corresponding variable in `Acquisition.Cx`, using SDA.BadBits.Persisted.

**Values**
- 0.5s
- 10s
- 1s
- 20s
- 2s
- 5s
- Infinite

### RefReceiver

**Bool**

Description
Refer to the corresponding variable in `SDA`, using SDA.BadBits.

### SDAMode

**Enum**

Description
Refer to the corresponding variable in `SDA`, using SDA.BadBits.

**Values**
- MaskTest
- Scope

### ShowFailLocation

**Bool**

Description
Refer to the corresponding variable in `SDA`, using SDA.BadBits.

### ShowLastTrace

**Bool**

Description
Refer to the corresponding variable in `Acquisition.Cx`, using SDA.BadBits.Persisted.

### SignalFrequency

**Double**

Range: From 5e+007 to 4e+010, step 1

Description
Refer to the corresponding variable in `SDA`, using SDA.BadBits.

### SignalType

**Enum**

Description
Refer to the corresponding variable in `SDA`, using SDA.BadBits.
## Values

- 1000BaseCX
- 1000BaseLX
- 1000BaseSX
- 1000baseX
- 10GBASELX4
- Custom
- DVI
- FC1063
- FC133
- FC2125
- FC266
- FC531
- IEEE1394b
- Infini2.5Gbs
- OC1
- OC12
- OC3
- OC48
- PCIExpress
- SATA1.5
- STM1
- STM16
- STM4
- STS1Eye
- STS3Interface
- STS3Transmit
- USB2.0
- XAU1

### SliceWidth

**Integer**

**Range:** From 0 to 100, step 1

**Description**

Eye Diagram measurement aid. Applies a vertical slice around the middle of the eye diagram for narrowing measurement areas.

### Stop

**Bool**

**Description**

Stop acquisition when the signal penetrates the mask.

### TrackMaskFail

**Action**

**Description**

Refer to the corresponding variable in SDA, using SDA.BadBits.

### UseDotJoin

**Bool**

**Description**

Refer to the corresponding variable in Acquisition.Cx, using SDA.BadBits.
UseGrid

Range: Any number of characters

Description
Using SDA.BadBits.Persisted, please refer to the corresponding variable in Acquisition.Cx

UserSignal

Description
Using SDA.BadBits, refer to the corresponding variable in SDA.

Values
C1
C2
C3
C4
F1
F2
F3
F4
F5
F6
F7
F8
M1
M2
M3
M4
NONE

VerAutoFit

Description
Using SDA.BadBits, refer to the corresponding variable in SDA.

View

Description
Using SDA.BadBits.Persisted, please refer to the corresponding variable in Acquisition.Cx

ViewLabels

Description
Using SDA.BadBits.Persisted, please refer to the corresponding variable in Acquisition.Cx

XMargin

Range: From 0 to 100, step 1

Description
Inflate the mask horizontally. Values: 0-100 in percent. 0 Means original mask; 100% means the mask stretches all the way over the horizontal span.
Y Margin

Range: From 0 to 100, step 1

Description

Inflate the mask vertically. Values: 0-100 in percent. 0 Means original mask; 100% means the mask stretches all the way over the vertical
RESULT

<table>
<thead>
<tr>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataArray</td>
</tr>
<tr>
<td>ExtendedStatus</td>
</tr>
<tr>
<td>FirstEventTime</td>
</tr>
<tr>
<td>HorizontalFrameStart</td>
</tr>
<tr>
<td>HorizontalFrameStop</td>
</tr>
<tr>
<td>HorizontalOffset</td>
</tr>
<tr>
<td>HorizontalPerStep</td>
</tr>
<tr>
<td>HorizontalResolution</td>
</tr>
<tr>
<td>HorizontalUnits</td>
</tr>
<tr>
<td>HorizontalVarianceArray</td>
</tr>
<tr>
<td>HorizontalVariances</td>
</tr>
<tr>
<td>IndexOfFirstSampleInFrame</td>
</tr>
<tr>
<td>LastEventTime</td>
</tr>
<tr>
<td>NumFrameDimensions</td>
</tr>
<tr>
<td>NumSamplesInFrame</td>
</tr>
<tr>
<td>Samples</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>StatusDescription</td>
</tr>
<tr>
<td>Sweeps</td>
</tr>
<tr>
<td>UpdateTime</td>
</tr>
<tr>
<td>VerticalFrameStart</td>
</tr>
<tr>
<td>VerticalFrameStop</td>
</tr>
<tr>
<td>VerticalMaxPossible</td>
</tr>
<tr>
<td>VerticalMinPossible</td>
</tr>
<tr>
<td>VerticalOffset</td>
</tr>
<tr>
<td>VerticalPerStep</td>
</tr>
<tr>
<td>VerticalResolution</td>
</tr>
<tr>
<td>VerticalUnits</td>
</tr>
</tbody>
</table>

DataArray

Description

ExtendedStatus

Description

FirstEventTime

Description

HorizontalFrameStart

Description
### Part Two: Reference

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HorizontalFrameStop</td>
<td>Using SDA.BadBits.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>HorizontalOffset</td>
<td>Using SDA.BadBits.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>HorizontalPerStep</td>
<td>Using SDA.BadBits.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>HorizontalUnits</td>
<td>Using SDA.BadBits.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>HorizontalVariances</td>
<td>Using SDA.BadBits.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>IndexOfFirstSampleInFrame</td>
<td>Using SDA.BadBits.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>LastEventTime</td>
<td>Using SDA.BadBits.Out.Result, please refer to the corresponding variable in Acquisition.Cx.</td>
</tr>
<tr>
<td>NumSamplesInFrame</td>
<td>Using SDA.BadBits.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
</tbody>
</table>
Samples

Description

Status

Description

StatusDescription

Description

Sweeps

Description

UpdateTime

Description

VerticalFrameStart

Description

VerticalFrameStop

Description

VerticalMaxPossible

Description

VerticalMinPossible

Description

VerticalOffset

Description

VerticalPerStep

Description
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VerticalUnits</td>
<td>Using SDA.BadBits.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>AxisXRotation</td>
<td>Integer</td>
</tr>
<tr>
<td>AxisYRotation</td>
<td>Integer</td>
</tr>
<tr>
<td>ClearSweeps</td>
<td>Action</td>
</tr>
<tr>
<td>EyeMode</td>
<td>Enum</td>
</tr>
<tr>
<td>LabelsPosition</td>
<td>String</td>
</tr>
<tr>
<td>LabelsText</td>
<td>String</td>
</tr>
<tr>
<td>MaskType</td>
<td>Enum</td>
</tr>
<tr>
<td>Persist3DQuality</td>
<td>Enum</td>
</tr>
<tr>
<td>Persisted</td>
<td>Bool</td>
</tr>
<tr>
<td>Persistence3d</td>
<td>Bool</td>
</tr>
<tr>
<td>PersistenceMonoChrome</td>
<td>Bool</td>
</tr>
<tr>
<td>PersistenceSaturation</td>
<td>Integer</td>
</tr>
<tr>
<td>PersistenceTime</td>
<td>Enum</td>
</tr>
<tr>
<td>ShowLastTrace</td>
<td>Bool</td>
</tr>
<tr>
<td>SignalFrequency</td>
<td>Double</td>
</tr>
<tr>
<td>SignalType</td>
<td>Enum</td>
</tr>
<tr>
<td>UseDotJoin</td>
<td>Bool</td>
</tr>
<tr>
<td>UseGrid</td>
<td>String</td>
</tr>
<tr>
<td>VerAutoFit</td>
<td>Bool</td>
</tr>
<tr>
<td>View</td>
<td>Bool</td>
</tr>
<tr>
<td>ViewLabels</td>
<td>Bool</td>
</tr>
</tbody>
</table>

- **AxisXRotation**: Integer
  - Range: From –90 to 90, step 1
- **AxisYRotation**: Integer
  - Range: From –90 to 90, step 1
- **ClearSweeps**: Action
- **EyeMode**: Enum
- **LabelsPosition**: String
- **LabelsText**: String
- **MaskType**: Enum
- **Persist3DQuality**: Enum
- **Persisted**: Bool
- **Persistence3d**: Bool
- **PersistenceMonoChrome**: Bool
- **PersistenceSaturation**: Integer
- **PersistenceTime**: Enum
- **ShowLastTrace**: Bool
- **SignalFrequency**: Double
- **SignalType**: Enum
- **UseDotJoin**: Bool
- **UseGrid**: String
- **VerAutoFit**: Bool
- **View**: Bool
- **ViewLabels**: Bool
**PART TWO: REFERENCE**

**LabelsText**
- **Type**: String
- **Range**: Any number of characters
- **Description**: Using SDA.Bits.View, please refer to the corresponding variable in Acquisition.Cx

**MaskType**
- **Type**: Enum
- **Description**: Using SDA.Bits, please refer to the corresponding variable in SDA.
- **Values**:
  - Absolute
  - Normalized

**Persist3DQuality**
- **Type**: Enum
- **Description**: Using SDA.Bits.View, please refer to the corresponding variable in Acquisition.Cx
- **Values**:
  - Shaded
  - Solid
  - WireFrame

**Persisted**
- **Type**: Bool
- **Description**: Using SDA.Bits.View, please refer to the corresponding variable in Acquisition.Cx

**Persistence3d**
- **Type**: Bool
- **Description**: Using SDA.Bits.View, please refer to the corresponding variable in Acquisition.Cx

**PersistenceMonoChrome**
- **Type**: Bool
- **Description**: Using SDA.Bits.View, please refer to the corresponding variable in Acquisition.Cx

**PersistenceSaturation**
- **Type**: Integer
- **Range**: From 1 to 100, step 1
- **Description**: Using SDA.Bits.View, please refer to the corresponding variable in Acquisition.Cx

**PersistenceTime**
- **Type**: Enum
- **Description**: Using SDA.Bits.View, please refer to the corresponding variable in Acquisition.Cx
CHAPTER TWO

Control Reference

Values

0.5s
10s
1s
20s
2s
5s
Infinite

ShowLastTrace

Description

Using SDA.Bits.View, please refer to the corresponding variable in Acquisition.Cx

SignalFrequency

Description

Sets/Queries the signal frequency.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the signal frequency to 2.13 MHz
app.SDA.Bits.SignalFrequency = 2.13e6

SignalType

Description

Using SDA.Bits, please refer to the corresponding variable in SDA.

Values

1000BaseCX
1000BaseLX
1000BaseSX
1000BaseX
10GBASELX4
Custom
DVI
FC1063
FC133
FC2125
FC266
FC531
IEEE1394b
Infini2.5Gbs
OC1
OC12
OC3
OC48
PCIExpress
**PART TWO: REFERENCE**

SATAT1.5
STM1
STM16
STM4
STS1Eye
STS3Interface
STS3Transmit
USB2.0
XAUI

**UseDotJoin**

*Bool*

Description
Using SDA.Bits.View, please refer to the corresponding variable in Acquisition.Cx

**UseGrid**

*String*

Range: Any number of characters

Description
Using SDA.Bits.View, please refer to the corresponding variable in Acquisition.Cx

**VerAutoFit**

*Bool*

Description
Using SDA.Bits, please refer to the corresponding variable in SDA.

**View**

*Bool*

Description
Using SDA.Bits.View, please refer to the corresponding variable in Acquisition.Cx

**ViewLabels**

*Bool*

Description
Using SDA.Bits.View, please refer to the corresponding variable in Acquisition.Cx
RESULT

app.SDA.Bits.Out.Result

Properties of the type xxxx.Out.Result.yyyy are those of the last completed acquisition. They are not affected if other CVars are changed after that acquisition was completed. This distinction between "Out.Result" properties and other CVars is most important when the trigger mode is Single or Stopped. You should treat "Out.Result" properties as read-only.

<table>
<thead>
<tr>
<th>DataArray</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtendedStatus</td>
<td>Property</td>
</tr>
<tr>
<td>FirstEventTime</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalFrameStart</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalFrameStop</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalOffset</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalPerStep</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalResolution</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalUnits</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalVarianceArray</td>
<td>Property</td>
</tr>
<tr>
<td>HorizontalVariances</td>
<td>Property</td>
</tr>
<tr>
<td>IndexOfFirstSampleInFrame</td>
<td>Property</td>
</tr>
<tr>
<td>LastEventTime</td>
<td>Property</td>
</tr>
<tr>
<td>NumFrameDimensions</td>
<td>Property</td>
</tr>
<tr>
<td>NumSamplesInFrame</td>
<td>Property</td>
</tr>
<tr>
<td>Samples</td>
<td>Property</td>
</tr>
<tr>
<td>Status</td>
<td>Property</td>
</tr>
<tr>
<td>StatusDescription</td>
<td>Property</td>
</tr>
<tr>
<td>Sweeps</td>
<td>Property</td>
</tr>
<tr>
<td>UpdateTime</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalFrameStart</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalFrameStop</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalMaxPossible</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalMinPossible</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalOffset</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalPerStep</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalResolution</td>
<td>Property</td>
</tr>
<tr>
<td>VerticalUnits</td>
<td>Property</td>
</tr>
</tbody>
</table>

DataArray

   Description

ExtendedStatus

   Description

FirstEventTime

   Description
### HorizontalFrameStart
**Property**

**Description**

### HorizontalFrameStop
**Property**

**Description**

### HorizontalOffset
**Property**

**Description**

### HorizontalPerStep
**Property**

**Description**

### HorizontalResolution
**Property**

**Description**

### HorizontalUnits
**Property**

**Description**

### HorizontalVarianceArray
**Property**

**Description**

### HorizontalVariances
**Property**

**Description**

### IndexOfFirstSampleInFrame
**Property**

**Description**

### LastEventTime
**Property**

**Description**

### NumFrameDimensions
**Property**

**Description**
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumSamplesInFrame</td>
<td>Using SDA.Bits.Out.Result, please refer to the corresponding variable in Acquistion.Cx.Out.Result</td>
</tr>
</tbody>
</table>
## PART TWO: REFERENCE

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VerticalPerStep</strong></td>
<td>Using SDA.Bits.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td><strong>VerticalUnits</strong></td>
<td>Using SDA.Bits.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
</tbody>
</table>
## Control Reference

### EYE

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AxisXRotation</td>
<td>Integer</td>
</tr>
<tr>
<td>AxisYRotation</td>
<td>Integer</td>
</tr>
<tr>
<td>ClearSweeps</td>
<td>Action</td>
</tr>
<tr>
<td>DataSource</td>
<td>Enum</td>
</tr>
<tr>
<td>EyeMode</td>
<td>Enum</td>
</tr>
<tr>
<td>LabelsPosition</td>
<td>String</td>
</tr>
<tr>
<td>LabelsText</td>
<td>String</td>
</tr>
<tr>
<td>MaskType</td>
<td>Enum</td>
</tr>
<tr>
<td>Persist3DQuality</td>
<td>Enum</td>
</tr>
<tr>
<td>Persisted</td>
<td>Bool</td>
</tr>
<tr>
<td>Persistence3d</td>
<td>Bool</td>
</tr>
<tr>
<td>PersistenceMonoChrome</td>
<td>Bool</td>
</tr>
<tr>
<td>PersistenceSaturation</td>
<td>Integer</td>
</tr>
<tr>
<td>PersistenceTime</td>
<td>Enum</td>
</tr>
<tr>
<td>ShowLastTrace</td>
<td>Bool</td>
</tr>
<tr>
<td>SignalFrequency</td>
<td>Double</td>
</tr>
<tr>
<td>SignalType</td>
<td>Enum</td>
</tr>
<tr>
<td>UseDotJoin</td>
<td>Bool</td>
</tr>
<tr>
<td>UseGrid</td>
<td>String</td>
</tr>
<tr>
<td>UseMonochrome</td>
<td>Bool</td>
</tr>
<tr>
<td>VerAutoFit</td>
<td>Bool</td>
</tr>
<tr>
<td>View</td>
<td>Bool</td>
</tr>
<tr>
<td>ViewLabels</td>
<td>Bool</td>
</tr>
</tbody>
</table>

**AxisXRotation**

*Integer*

**Range:** From −90 to 90, step 1

**Description:**

Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.

**AxisYRotation**

*Integer*

**Range:** From −90 to 90, step 1

**Description:**

Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.

**ClearSweeps**

*Action*

**Description:**

Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.

**DataSource**

*Enum*

**Description:**

Sets/Queries the source of the data for the eye diagram.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
```
' Set the source for the eye diagram to input channel C3.
app.SDA.eye.DataSource = "C3"

Values
- C1
- C2
- C3
- C4
- F1
- F2
- F3
- F4
- F5
- F6
- F7
- F8
- M1
- M2
- M3
- M4
- PRBS

**EyeMode**

*Enum*

Description
Using SDA.Eye, please refer to the corresponding variable in SDA.

Values
- Sequential
- Traditional

**LabelsPosition**

*String*

Range: Any number of characters.

Description
Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.

**LabelsText**

*String*

Range: Any number of characters

Description
Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.

**MaskType**

*Enum*

Description
Using SDA.Eye, please refer to the corresponding variable in SDA.

Values
- Absolute
- Normalized
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persist3DQuality</td>
<td>Enum</td>
<td>Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Values</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shaded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WireFrame</td>
</tr>
<tr>
<td>Persisted</td>
<td>Bool</td>
<td>Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.</td>
</tr>
<tr>
<td>Persistence3d</td>
<td>Bool</td>
<td>Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.</td>
</tr>
<tr>
<td>PersistenceMonoChrome</td>
<td>Bool</td>
<td>Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.</td>
</tr>
<tr>
<td>PersistenceSaturation</td>
<td>Integer</td>
<td>From 1 to 100, step 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.</td>
</tr>
<tr>
<td>PersistenceTime</td>
<td>Enum</td>
<td>Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Infinite</td>
</tr>
<tr>
<td>ShowLastTrace</td>
<td>Bool</td>
<td>Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.</td>
</tr>
<tr>
<td>SignalFrequency</td>
<td>Double</td>
<td>From 5e+007 to 4e+010, step 1</td>
</tr>
</tbody>
</table>
### SignalType

<table>
<thead>
<tr>
<th>Description</th>
<th>Enum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using SDA.Eye, please refer to the corresponding variable in SDA.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000BaseCX</td>
</tr>
<tr>
<td>1000BaseLX</td>
</tr>
<tr>
<td>1000BaseSX</td>
</tr>
<tr>
<td>1000baseX</td>
</tr>
<tr>
<td>10GBASELX4</td>
</tr>
<tr>
<td>Custom</td>
</tr>
<tr>
<td>DVI</td>
</tr>
<tr>
<td>FC1063</td>
</tr>
<tr>
<td>FC133</td>
</tr>
<tr>
<td>FC2125</td>
</tr>
<tr>
<td>FC266</td>
</tr>
<tr>
<td>FC531</td>
</tr>
<tr>
<td>IEEE1394b</td>
</tr>
<tr>
<td>Infini2.5Gbs</td>
</tr>
<tr>
<td>OC1</td>
</tr>
<tr>
<td>OC12</td>
</tr>
<tr>
<td>OC3</td>
</tr>
<tr>
<td>OC48</td>
</tr>
<tr>
<td>PCIExpress</td>
</tr>
<tr>
<td>SATA1.5</td>
</tr>
<tr>
<td>STM1</td>
</tr>
<tr>
<td>STM16</td>
</tr>
<tr>
<td>STM4</td>
</tr>
<tr>
<td>STS1Eye</td>
</tr>
<tr>
<td>STS3Interface</td>
</tr>
<tr>
<td>STS3Transmit</td>
</tr>
<tr>
<td>USB2.0</td>
</tr>
<tr>
<td>XAUI</td>
</tr>
</tbody>
</table>

### UseDotJoin

<table>
<thead>
<tr>
<th>Description</th>
<th>Bool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.</td>
<td></td>
</tr>
</tbody>
</table>

### UseGrid

<table>
<thead>
<tr>
<th>Range: Any number of characters</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.</td>
<td></td>
</tr>
</tbody>
</table>

### UseMonochrome

<table>
<thead>
<tr>
<th>Description</th>
<th>Bool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets/Queries whether the display is monochrome.</td>
<td></td>
</tr>
</tbody>
</table>

---

Please see the corresponding variable in app.Measure.Px.Operator (ParamEngine = "Dperiod@level").
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set monochrome off.
app.SDA.Eye.UseMonochrome = False

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VerAutoFit</td>
<td>Bool</td>
<td>Using SDA.Eye, please refer to the corresponding variable in SDA.</td>
</tr>
<tr>
<td>View</td>
<td>Bool</td>
<td>Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.</td>
</tr>
<tr>
<td>ViewLabels</td>
<td>Bool</td>
<td>Using SDA.Eye, please refer to the corresponding variable in Acquisition.Cx.</td>
</tr>
</tbody>
</table>
Properties of the type xxxx.Out.Result.yyyy are those of the last completed acquisition. They are not affected if other CVars are changed after that acquisition was completed. This distinction between "Out.Result" properties and other CVars is most important when the trigger mode is Single or Stopped. You should treat "Out.Result" properties as read-only.

### Columns

<table>
<thead>
<tr>
<th>Description</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries the number of columns in the display.</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the number of columns.
```

---

### DataArray


---

### FirstEventTime

### HorizontalFrameStart

**Property**

**Description**


### HorizontalFrameStop

**Property**

**Description**


### HorizontalOffset

**Property**

**Description**


### HorizontalPerColumn

**Property**

**Description**

Queries the column spacing.

**Example**

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the horizontal spacing.
HSpace = app.SDA.Eye.Out.Result.HorizontalPerColumn`

### HorizontalUnits

**Property**

**Description**


### LastEventTime

**Property**

**Description**


### MaxPopulationInRectangle

**Property**

**Description**

Queries the highest population in the array.

**Example**

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the total population.
MaxPop = app.SDA.Eye.Out.Result.MaxPopulationInRectangle`
**NumFrameDimensions**

**Property**

**Description**


**PopulationOfRectangle**

**Property**

**Description**

Queries the total population of the array.

**Example**

```visualbasic
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the total population.
```

**Rows**

**Property**

**Description**

Queries the number of rows in the display.

**Example**

```visualbasic
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Inspect the number of rows.
Rows = app.SDA.Eye.Out.Result.Rows
```

**Sweeps**

**Property**

**Description**


**UpdateTime**

**Property**

**Description**


**VerticalFrameStart**

**Property**

**Description**


**VerticalFrameStop**

**Property**

**Description**

<table>
<thead>
<tr>
<th>VerticalOffset</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VerticalPerRow</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Queries the vertical spacing.</td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td></td>
</tr>
<tr>
<td>' Visual Basic Script</td>
<td></td>
</tr>
<tr>
<td>Set app = CreateObject(&quot;LeCroy.XStreamDSO&quot;)</td>
<td></td>
</tr>
<tr>
<td>' Inspect the vertical spacing.</td>
<td></td>
</tr>
<tr>
<td>VSpace = app.SDA.Eye.Out.Result.VerticalPerRow</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VerticalUnits</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
</tr>
</tbody>
</table>
**AbsLevel**

*Double*

Range: From −100 to 100, step 1e-007

Description

Please see the corresponding variable in app.Measure.Px.Operator (ParamEngine = "TIE@level")

**EyeThresholdType**

*Enum*

Description

Sets/Queries whether the eye threshold is measured in absolute units or percentage.

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the eye threshold to percent.
app.SDA.TIE.EyeThresholdType = "Absolute"
```

Values

Absolute

Percent

**FindFrequency**

*Action*

Description

Determine the signal frequency from the waveform data.

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Calculate the signal frequency from the trace data.
app.SDA.TIE.FindFrequency
```

**PercentLevel**

*Integer*

Range: From 0 to 100, step 1

Description

Please see the corresponding variable in app.Measure.Px.Operator (ParamEngine = "TIE@level")
PLLOn

Description
Please see the corresponding variable in app.SDA.

SignalFrequency

Description
Sets/Queries the signal frequency for TIE.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the signal frequency to 2.00 MHz
app.SDA.TIE.SignalFrequency = 2.0e6

Slope

Description
Please see the corresponding variable in app.Measure.Px.Operator (ParamEngine = "TIE@level")

Values
Both
Neg
Pos

View

Description
Sets/Queries the visibility of the function.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Show the TIE function.
app.SDA.TIE.View = True
Properties of the type `xxx.Out.Result.yyy` are those of the last completed acquisition. They are not affected if other cvars are changed after that acquisition was completed. This distinction between "Out.Result" properties and other cvars is most important when the trigger mode is Single or Stopped. You should treat "Out.Result" properties as read-only.

### Result

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtendedStatus</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>FirstEventTime</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>HorizontalResolution</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>HorizontalUnits</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>LastEventTime</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>NumFrameDimensions</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>Status</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>StatusDescription</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>UpdateTime</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>Value</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>ValueArray</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>VerticalResolution</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>VerticalUnits</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
<tr>
<td>Status</td>
<td>Property</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Description</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>StatusDescription</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UpdateTime</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ValueArray</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VerticalResolution</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VerticalUnits</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Using SDA.TIE.Out.Result, please refer to the corresponding variable in Acquisition.Cx.Out.Result</td>
</tr>
</tbody>
</table>
This set of variables controls the date and time setup. In addition to manual controls for hh/mm/ss and dd/mm/yy, there is the ability to set the time and date from an Internet clock, using the SNTP protocol.

<table>
<thead>
<tr>
<th>CurrentDateAndTime</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day</td>
<td>Integer</td>
</tr>
<tr>
<td>Hour</td>
<td>Integer</td>
</tr>
<tr>
<td>Minute</td>
<td>Integer</td>
</tr>
<tr>
<td>Month</td>
<td>Integer</td>
</tr>
<tr>
<td>Second</td>
<td>Integer</td>
</tr>
<tr>
<td>SetFromSNTP</td>
<td>Action</td>
</tr>
<tr>
<td>Validate</td>
<td>Action</td>
</tr>
<tr>
<td>Year</td>
<td>Integer</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set time/date from the NIST Internet clock
app.Utility.DateTimeSetup.SetFromSNTP

**CurrentDateAndTime**  
*String*

Range: Any number of characters.  
Description
Reads the current date and time from the real-time calendar and clock.  
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the current date and time from the real-time calendar and clock.
app.Utility.DateTimeSetup.CurrentDateAndTime

**Day**  
*Integer*

Range: From 1 to 31, step 1  
Description
Sets/Queries the day of the month setting of the real-time clock as a number. The value will not be accepted by the clock until app.Utility.DateTimeSetup.Validate is sent. All time/date controls are validated at the same time.  
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the day of the month as 21.
app.Utility.DateTimeSetup.Day = 21
app.Utility.DateTimeSetup.Validate

Hour  
**Integer**

Range: From 0 to 23, step 1

Description
Sets/Queries the hours setting of the real-time clock as a number. The value will not be accepted by the clock until app.Utility.DateTimeSetup.Validate is sent. All time/date controls are validated at the same time.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the hour as 13.
app.Utility.DateTimeSetup.Hour = 13
app.Utility.DateTimeSetup.Validate

Minute  
**Integer**

Range: From 0 to 59, step 1

Description
Sets/Queries the minutes setting of the real-time clock as a number. The value will not be accepted by the clock until app.Utility.DateTimeSetup.Validate is sent. All time/date controls are validated at the same time.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the minute as 34.
app.Utility.DateTimeSetup.Minute = 34
app.Utility.DateTimeSetup.Validate

Month  
**Integer**

Range: From 1 to 12, step 1

Description
Sets/Queries the month setting of the real-time clock as a number. The value will not be accepted by the clock until app.Utility.DateTimeSetup.Validate is sent. All time/date controls are validated at the same time.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the month as August.
app.Utility.DateTimeSetup.Month = 8
**Second**

*Integer*

**Range:** From 0 to 59, step 1

**Description**
Sets/Queries the seconds setting of the real-time clock as a number. The value will not be accepted by the clock until `app.Utility.DateTimeSetup.Validate` is sent. All time/date controls are validated at the same time.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the seconds as 55.
app.Utility.DateTimeSetup.Second = 55
app.Utility.DateTimeSetup.Validate

**SetFromSNTP**

*Action*

**Description**
Sets the real time clock from the simple network time protocol.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the real time clock from the simple network time protocol.
app.Utility.DateTimeSetup.SetFromSNTP

**Validate**

*Action*

**Description**
Validates any new settings. This action = clicking “Validate Changes” on the Date/Time page.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the day, hour, and minute, and validate.
app.Utility.DateTimeSetup.Day = 3
app.Utility.DateTimeSetup.Hour = 5
app.Utility.DateTimeSetup.Minute = 8

app.Utility.DateTimeSetup.Validate
YEAR

Description

Sets/Queries the year setting of the real-time clock as a number. The value will not be accepted by the clock until app.Utility.DateTimeSetup.Validate is sent. All time/date controls are validated at the same time.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the year as 2003.
app.Utility.DateTimeSetup.Year = 2003
app.Utility.DateTimeSetup.Validate
**Part Two: Reference**

**Options**

The options subsystem contains controls to query the list of installed software and hardware options.

<table>
<thead>
<tr>
<th>InstalledHWOptions</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstalledSWOptions</td>
<td>String</td>
</tr>
<tr>
<td>ScopeID</td>
<td>String</td>
</tr>
</tbody>
</table>

**InstalledHWOptions**

Range: Any number of characters.

Description

Shows a list of the installed hardware options.

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the list of installed hardware options and present
' in a popup dialog
MsgBox app.Utility.Options.InstalledHWOptions
```

**InstalledSWOptions**

Range: Any number of characters.

Description

Shows list of installed software options.

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the list of installed software options and display
' in a popup dialog
MsgBox app.Utility.Options.InstalledSWOptions
```

**ScopeID**

Range: Any number of characters.

Description

Queries the ID of the instrument. This ID should be specified when purchasing software options for your instrument.

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Read the ID of the instrument.
MsgBox app.Utility.Options.ScopeID
```
These controls are related to the remote control section of the instrument. In this context Automation is not considered part of “Remote”. Remote control currently includes control using ASCII remote commands from GPIB or TCP/IP.

### Control Reference

<table>
<thead>
<tr>
<th>Control</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllowControlFrom</td>
<td>String</td>
</tr>
<tr>
<td>Assistant</td>
<td>Enum</td>
</tr>
<tr>
<td>GpibAddress</td>
<td>Integer</td>
</tr>
<tr>
<td>Interface</td>
<td>Enum</td>
</tr>
<tr>
<td>RestrictControl</td>
<td>Enum</td>
</tr>
<tr>
<td>SetToErrorsOnlyAndClearAtStartup</td>
<td>Bool</td>
</tr>
</tbody>
</table>

**AllowControlFrom**

**String**

**Range:** Any number of characters.

**Description**

Sets/Queries an IP address from which remote control is allowed, if remote control has been restricted to specified clients using the RestrictControl control. This string contains a comma-delimited list of addresses, which can be supplied either in IP or DNS form. For example: "126.2.2.34,dansWorkstation"

**Assistant**

**Enum**

**Description**

Sets/Queries the setting of the remote assistant.

**Values**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EO</td>
<td>Log errors only</td>
</tr>
<tr>
<td>FD</td>
<td>Log all remote commands/queries</td>
</tr>
<tr>
<td>OFF</td>
<td>Turn the assistant off</td>
</tr>
</tbody>
</table>

**GpibAddress**

**Integer**

**Range:** From 1 to 30, step 1

**Description**

Queries the current GPIB address for remote control.

**Interface**

**Enum**

**Description**

Sets/Queries the currently selected type of remote control interface.

**Values**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPIB</td>
<td></td>
</tr>
<tr>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>TCPIP</td>
<td></td>
</tr>
</tbody>
</table>

**RestrictControl**

**Enum**

**Description**

Sets/Queries whether remote control is restricted to certain hosts, where the host name is defined either by IP address or DNS name.
Values

No
Yes

SetToErrorsOnlyAndClearAtStartup  Bool

Description

Enables the resetting of the remote assistant to “Errors Only” mode whenever the instrument is reset. Also ensures that the remote assistant log is cleared upon startup. This control is set by default to lower the risk that the remote assistant will be set to “Full Dialog” mode and be forgotten, causing a decrease in remote control performance.
This set of variables controls the web editor, which shows the paths for data flow in the instrument. This feature is not supported on all instruments. Currently it is supported on DDA and SDA models, and models with XMAP and/or XMATH software options.

### WebEditor

<table>
<thead>
<tr>
<th>Method/Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddConnection</td>
<td>Add a connection to the web editor</td>
</tr>
<tr>
<td>AddPreview</td>
<td>Add a preview window</td>
</tr>
<tr>
<td>AddProcessor</td>
<td>Add a processor to the web editor</td>
</tr>
<tr>
<td>ClearSweeps</td>
<td>Clear all sweeps</td>
</tr>
<tr>
<td>GetProcessor</td>
<td>Get a processor from the web editor</td>
</tr>
<tr>
<td>RemoveAll()</td>
<td>Remove all processors</td>
</tr>
<tr>
<td>RemoveConnection</td>
<td>Remove a connection</td>
</tr>
<tr>
<td>RemovePreview</td>
<td>Remove a preview window</td>
</tr>
<tr>
<td>RemoveProcessor</td>
<td>Remove a processor</td>
</tr>
</tbody>
</table>

### Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Enter auto-trigger mode
app.Acquisition.TriggerMode = "Auto"

' Show the web editor and remove all processors from it
app.ActiveView = "WebEdit"
app.WebEditor.RemoveAll

' Create a Waveform Adder, name it "MyAvg", and place it at x=200, y=30

' Connect the output of channel 1 "C1Exec", to the first input of the adder, and the output of channel 2 "C2Exec" to the second
app.WebEditor.AddConnection "MyAdder", 0, "C1Exec", 0
app.WebEditor.AddConnection "MyAdder", 1, "C2Exec", 0

' Add a preview window to show the averaged output
app.WebEditor.AddPreview "MyAdder", 0, "MyAvgPreview", 370, 40
```
PART TWO: REFERENCE

Place function F1 into WebEdit mode, and hook the adder output to it

app.Math.F1.View = True


Description

Adds a connection between two ‘pins’ of nodes placed within the Web Editor. Pins are described by the name of the node, and the zero-based index of the pin on that node.


Description

Adds a Preview to the specified pin of the specified node. The coordinates specify where the preview will appear on the Web, with 0,0 being the top-left corner.


Description

Adds a named “processor” to the web. To determine the name of a processor, just place it on the web using the GUI and hover the mouse over the node. The “ProgID” of the node, in the format ‘LeCroy.<procName>’ will appear. When processors are added from automation, there is no distinction between Measure, Math, and Pass/Fail processors.

ClearSweeps

Description

Clears any accumulated data for nodes such as Average, Persistence, etc. that reside in the processing web.

GetProcessor([in] VARIANT processor)

Description

Retrieves a reference to a processor that has been added to the Web. This reference may then be used to access the processor’s controls. See the Math/Measure control reference section of this manual for a list of the available controls for each processor.

Example

Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

Show the web editor and remove all processors from it

app.ActiveView = "WebEdit"

app.WebEditor.RemoveAll

Create a Waveform Averager, name it "MyAvg", and place it at
app.WebEditor.AddProcessor "LeCroy.Average", "MyAvg", 200, 30

' Retrieve a pointer to the averager and set it's number of sweeps
' to the value 1234
set myAverager = app.WebEditor.GetProcessor("MyAvg")

RemoveAll() Method
Description
Removes all processors from the web.

RemoveConnection([in] VARIANT destProcessor, [in] VARIANT destInputPin) Method
Description
Removes a connection between two pins on the web.

RemovePreview([in] VARIANT processor) Method
Description
Removes the named preview display.

RemoveProcessor([in] VARIANT processor) Method
Description
Removes the named processor from the Web.
This section provides a reference guide for all controls used by the various ‘processors’ available in the instrument’s Math and Measure subsystems.
Math/Measure Control Reference

This section of the manual is concerned with documenting the controls used by the numerous Math/Measure “processors” in the X-Stream DSO. The basic structure is very similar to that used in the preceding section, with one basic exception.

In the Math and Measure sections of the DSO, the automation hierarchy is more dynamic than it is elsewhere. When a math or measure operator is selected, its controls “appear” in the hierarchy, switching out the controls used by the previously selected operator.

For example, when the FFT operator is selected as Operator 1 of Math Function F1:

```plaintext
app.Math.F1.Operator1 = "FFT"
```

The following controls will be available:

```plaintext
app.Math.F1.Operator1Setup.FillType
app.Math.F1.Operator1Setup.Type
```

The nodes in the automation hierarchy that behave in this way include:

```plaintext
app.Math.Fx.Operator1Setup
app.Math.Fx.Operator2Setup

app.Measure.Fx.Operator

app.PassFail.Q1.Operator
```

In order to clarify this dependency, the titles of each of the sections in this chapter follow this format:

```
AVERAGE
app.Math.Fx.OperatorYSetup (Operator = "Average")
```

Path to controls, where:  
Fx := \{ F1, F2, F3, F4, F5, F6, F7, F8 \} and  
OperatorYSetup := \{ Operator1Setup, Operator2Setup \}
AVERAGE

app.Math.Fx.OperatorYSetup (Operator = "Average")

Description
Waveform Averaging.

<table>
<thead>
<tr>
<th>AverageType</th>
<th>Enum</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClearSweeps</td>
<td>Action</td>
</tr>
<tr>
<td>Sweeps</td>
<td>Integer</td>
</tr>
</tbody>
</table>

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Turn trace F1 on and setup to average the data from C1
' Average mode is set to Continuous
app.Math.F1.View = True
app.Math.F1.Operator1 = "Average"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Source1 = "C1"
app.Math.F1.Operator1Setup.AverageType = "Continuous"

AverageType

Description
Sets/Queries the averaging mode. Continuous and Summation modes are supported.

Values
Continuous
Summed

ClearSweeps

Description
Clears all averaged sweeps.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Clear sweeps for average in trace F1.
Sweeps

Range: From 1 to 1000000, step 1

Description

Sets/Queries the number of sweeps to be averaged when trace Fx is set to averaging: continuous or summed.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set number of sweeps to be averaged in trace F1 as 20.
BOXCAR

app.Math.Fx.OperatorYSetup (Operator = "Boxcar")

Rectangular BoxCar filter (local running average) of specified length.

Length

<table>
<thead>
<tr>
<th>Range: From 2 to 50, step 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
</tbody>
</table>

Sets / Queries the length, in samples, of the boxcar FIR filter (i.e., the running average of a local set of "length" points)

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the filter length for the boxcar function in trace F1
app.Math.F1.View = True
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Boxcar"
CORRELATION  \texttt{app.Math.Fx.OperatorYSetup (Operator = \char"\textquoteleft\textquoteleft Correlation\textquoteright\textquoteright\}}

Correlates a portion of one waveform to another.

<table>
<thead>
<tr>
<th>CorrLength</th>
<th>Double</th>
</tr>
</thead>
<tbody>
<tr>
<td>CorrStart</td>
<td>Double</td>
</tr>
</tbody>
</table>

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure correlation in F3 using a length of 3.5 divisions,
' starting at the first division.
```

CorrLength  \textit{Double}

\textbf{Range:} From 0.001 to 10, step 0.001

\textbf{Description}

Sets/Queries the length (in graticule divisions) of the section of the first input trace that is used in the calculation of correlation.

CorrStart  \textit{Double}

\textbf{Range:} From 0 to 9.999, step 0.001

\textbf{Description}

Sets/Queries the position (in graticule divisions) of the start of the section of trace 1 that is used in the correlation function in trace Fx.
DERIVATIVE

app.Math.Fx.OperatorYSetup (Operator = "Derivative")

Computes the derivative of the waveform: \( \frac{\text{next \_sample \_value} - \text{this \_sample \_value}}{\text{horizontal \_sample \_interval}} \).

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnableAutoScale</td>
<td>Bool</td>
</tr>
<tr>
<td>FindScale</td>
<td>Action</td>
</tr>
<tr>
<td>VerOffset</td>
<td>Double</td>
</tr>
<tr>
<td>VerScale</td>
<td>DoubleLockstep</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Start a find scale operation for derivative function trace F1
app.Math.F1.View = True
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Derivative"
app.Math.F1.Operator1Setup.FindScale

EnableAutoScale

Description
Sets/Queries whether the Autoscale function is enabled for derivative function trace Fx. If enabled, an Autoscale operation is performed whenever the setup changes.

FindScale

Description
Initiates a Find Scale action, to set a suitable vertical scale for the derivative function trace Fx.

VerOffset

Range: From -1e+006 to 1e+006, step 1e-009
Description
Sets/Queries the vertical offset of derivative function trace Fx.

VerScale

Range: From 1e-012 to 1e+013, step 10000, locked to 1.25
Description
Sets/Queries the vertical scale of derivative function Fx.
**DESKEW**

Deskew waveform by shifting it in time.

Example

```vbs
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set the displacement of the trace F3 to 3.7e-9
app.Math.F3.Operator1 = "Deskew"
```

### WaveDeskew

**Type:** Double

**Range:** From -0.1 to 0.1, step 1e-012

**Description**

Sets/Queries the displacement in time of trace Fx. A positive value delays the signal: a negative one makes it appear earlier.
ENVELOPE

Envelope of minimum and maximum values for an ensemble of sweeps.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Math.F3.Source1 = "C1"
app.Math.F3.Operator1 = "Envelope"

ClearSweeps

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sweeps</td>
</tr>
</tbody>
</table>

| Integer |

Description

Initiates a Clear Sweeps operation for envelope function trace Fx.

Sweeps

Range: From 1 to 1000000, step 1

Description

Sets/Queries the maximum number of sweeps to be used by envelope function trace Fx.
Enhanced resolution achieved through FIR filtering, using well behaved filters, with precalculated noise gain.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F3 to be an ERes of C1
app.Math.F3.Source1 = "C1"

Bits

<table>
<thead>
<tr>
<th>Enum</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sets/Queries the increase in resolution in bits for the enhanced resolution function. Be sure to specify the value as a string, and not as a floating point number, for example, &quot;0.5&quot;, &quot;1&quot;, &quot;1.5&quot;, &quot;2&quot;, &quot;2.5&quot;, or &quot;3&quot;.</td>
</tr>
</tbody>
</table>

Values

<table>
<thead>
<tr>
<th>0.5</th>
<th>1</th>
<th>1.5</th>
<th>2</th>
<th>2.5</th>
<th>3</th>
</tr>
</thead>
</table>
EXCELMATH  \( app.Math.Fx.OperatorYSetup (Operator = "ExcelMath") \)

Performs Math in Excel. Transfers 1 or 2 waveforms into Excel and reads the resulting waveform.

<table>
<thead>
<tr>
<th>addChart</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddLabels</td>
<td>Action</td>
</tr>
<tr>
<td>Advanced</td>
<td>Bool</td>
</tr>
<tr>
<td>ClearSheet</td>
<td>Action</td>
</tr>
<tr>
<td>CreateDemoSheet</td>
<td>Action</td>
</tr>
<tr>
<td>FindScale</td>
<td>Action</td>
</tr>
<tr>
<td>NewSheet</td>
<td>Bool</td>
</tr>
<tr>
<td>OutputCell</td>
<td>String</td>
</tr>
<tr>
<td>OutputEnable</td>
<td>Bool</td>
</tr>
<tr>
<td>OutputHeaderCell</td>
<td>String</td>
</tr>
<tr>
<td>Scaling</td>
<td>Enum</td>
</tr>
<tr>
<td>Source1Cell</td>
<td>String</td>
</tr>
<tr>
<td>Source1Enable</td>
<td>Bool</td>
</tr>
<tr>
<td>Source1HeaderCell</td>
<td>String</td>
</tr>
<tr>
<td>Source2Cell</td>
<td>String</td>
</tr>
<tr>
<td>Source2Enable</td>
<td>Bool</td>
</tr>
<tr>
<td>Source2HeaderCell</td>
<td>String</td>
</tr>
<tr>
<td>SpreadsheetFilename</td>
<td>FileName</td>
</tr>
<tr>
<td>Status</td>
<td>String</td>
</tr>
<tr>
<td>WithHeader</td>
<td>Bool</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F3 to process C1 in Excel using a demo-sheet
app.Math.F3.Source1 = "C1"

AddChart **Action**

**Description**

Adds a chart to the current Excel spreadsheet.

AddLabels **Action**

**Description**

Adds labels to the cells of the array headers in the Excel spreadsheet.
**Advanced**

Description

Enables/Disables/Queries the advanced Excel settings. By default, the cell ranges used to store the input waveform, and to retrieve the calculated waveform, are preset.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set function trace F3 to be an Excel function.

' Enable the use of the advanced settings.

---

**ClearSheet**

Description

Clears the contents of the current Excel spreadsheet.

**CreateDemoSheet**

Description

Creates a “demo sheet,” an excel spreadsheet pre-labeled, and with the output column equation preset to invert the input data.

**FindScale**

Description

Sets a suitable scale for the output data from Excel on the instrument graticule when scaling has been set to manual.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set function trace F3 to be an Excel function.

' Set the scaling from the Excel spreadsheet to automatic.

' Find a suitable scale for the output data
' on the instrument graticule.
NewSheet

Description
Enables/Disables/Queries the creation of a new Excel spreadsheet. If a new sheet is not to be created, an existing file name must be specified in the SpreadsheetFilename control.

Example
'S Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

'S Set function trace F3 to be an Excel function.

'S Enable the creation of a new Excel spreadsheet.

OutputCell

Description
Sets/Queries the cell label for output in the Excel function Fx. This cell marks the start (top) of the array of data to be taken from Excel into the instrument.

OutputEnable

Description
Enables/Disables/Queries the transfer of output data from Excel to the instrument. If a one-way computation is required, where results of the Excel processing are not required, this should be set to False to increase performance.

OutputHeaderCell

Description
Sets/Queries the header cell label for output in Excel function Fx. This is the starting cell for the header that carries setup information about the output waveform, from Excel to the instrument. Only used if the WithHeader control is set to True.
Scaling

Description
Sets/Queries the method of scaling the output trace from the Excel spreadsheet.

Values

<table>
<thead>
<tr>
<th>Automatic</th>
<th>Automatically scale the output waveform to full-scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>FromSheet</td>
<td>Retrieve scaling information from the output header in the spreadsheet</td>
</tr>
<tr>
<td>Manual</td>
<td>Manually auto-scale when FindScale is pressed</td>
</tr>
</tbody>
</table>

Source1Cell

Description
Sets/Queries the cell label for source 1 in Excel function Fx. This cell marks the start (top) of the array into which data from the first source waveform is transferred.

Source1Enable

Description
Enables/Disables/Queries the transfer of source 1 data from the instrument to Excel.

Source1HeaderCell

Description
Sets/Queries the header cell label for source 1 in Excel function Fx. This is the starting cell for the header that carries setup information about waveform 1, from the instrument to Excel. This information includes waveform length, vertical and horizontal units, vertical and horizontal framing information, and horizontal scaling and offset information. Only used when the WithHeader control is set to True.

Source2Cell

Description
Sets/Queries the cell label for source 2 in Excel function Fx. This cell marks the start (top) of the array into which data from the second source waveform is transferred.

Source2Enable

Description
Enables/Disables/Queries the transfer of source 2 data from the instrument to Excel.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
' Set function trace F1 to be an Excel function.

' Enable the transfer of source 2 data from the instrument to Excel.

**Source2HeaderCell**

*String*

**Range:** Any number of characters

**Description**

Sets/Queries the header cell label for source 2 in Excel function Fx. This is the starting cell for the header that carries setup information about waveform 2 from the instrument to Excel. This information includes waveform length, vertical and horizontal units, vertical and horizontal framing information, and horizontal scaling and offset information. Only used when the WithHeader control is set to True.

**SpreadsheetFilename**

*FileName*

**Range:** Any number of characters

**Description**

Sets/Queries the file name of the current Excel spreadsheet.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set function trace F3 to be an Excel function.

' Disable the creation of a new Excel spreadsheet.

' Select the filename of the existing Excel spreadsheet to be used.

**Status**

*String*

**Range:** Any number of characters

**Description**

Inspects the status of the Excel-and-instrument combination. Examples are "OK", or "Excel not installed".
Part Two: Reference

**WithHeader**

**Description**

Enables/Disables/Queries the presence of headers with the waveform
CHAPTER THREE

Math/Measure Control Reference

FFT

app.Math.Fx.OperatorYSetup (Operator = “FFT”)

Fast Fourier Transform of waveform data.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Enum</th>
</tr>
</thead>
<tbody>
<tr>
<td>FillType</td>
<td>Enum</td>
</tr>
<tr>
<td>SuppressDC</td>
<td>Bool</td>
</tr>
<tr>
<td>Type</td>
<td>Enum</td>
</tr>
<tr>
<td>Window</td>
<td>Enum</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F3 to perform an FFT of C1
app.Math.F3.Source1 = "C1"
app.Math.F3.Operator1 = "FFT"

Algorithm

Description
Sets/Queries the algorithm for the FFT in function trace Fx.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set function trace F4 to FFT.
app.Math.F4.Operator1 = "FFT"
' Set the FFT algorithm to power of two.

Values

LeastPrime
Power2

FillType

Description
Sets/Queries the type of trace fill to use in the FFT function trace Fx.
Values
  Truncate
  ZeroFill

SuppressDC
  Description
  Enables/Disables suppression of the value at zero frequency in the FFT spectrum.

Type
  Description
  Sets/Queries the type of FFT spectrum for function trace Fx.
  Values

<table>
<thead>
<tr>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imaginary</td>
<td>Imaginary part of the complex spectrum</td>
</tr>
<tr>
<td>Magnitude</td>
<td>Magnitude with linear vertical scale</td>
</tr>
<tr>
<td>Phase</td>
<td>Phase</td>
</tr>
<tr>
<td>PowerDensity</td>
<td>Power Density</td>
</tr>
<tr>
<td>PowerSpectrum</td>
<td>Power Spectrum</td>
</tr>
<tr>
<td>Real</td>
<td>Real part of the complex spectrum</td>
</tr>
</tbody>
</table>

Window
  Description
  Sets/Queries the type of window for FFT function trace Fx.
  Values

<table>
<thead>
<tr>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlackmanHarris</td>
<td>FlatTop</td>
</tr>
<tr>
<td>Hamming</td>
<td>Rectangular</td>
</tr>
<tr>
<td>VonHann</td>
<td></td>
</tr>
</tbody>
</table>
FILTER

\texttt{app.Math.Fx.OperatorYSetup (Operator = "Filter")}

Processes waveform using specified digital filter.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoLength</td>
<td>Bool</td>
</tr>
<tr>
<td>CosineBeta</td>
<td>Double</td>
</tr>
<tr>
<td>CustomFilename</td>
<td>FileName</td>
</tr>
<tr>
<td>FilterKind</td>
<td>Enum</td>
</tr>
<tr>
<td>FilterType</td>
<td>Enum</td>
</tr>
<tr>
<td>FirOrIir</td>
<td>Enum</td>
</tr>
<tr>
<td>GaussianBT</td>
<td>Double</td>
</tr>
<tr>
<td>HighFreqPass</td>
<td>Double</td>
</tr>
<tr>
<td>LowFreqPass</td>
<td>Double</td>
</tr>
<tr>
<td>NumberOfTaps</td>
<td>Integer</td>
</tr>
<tr>
<td>PassBandAttenuation</td>
<td>Double</td>
</tr>
<tr>
<td>PassBandRipple</td>
<td>Double</td>
</tr>
<tr>
<td>Rolloff</td>
<td>Double</td>
</tr>
<tr>
<td>StopBandAttenuation</td>
<td>Double</td>
</tr>
<tr>
<td>TransitionWidth</td>
<td>Double</td>
</tr>
<tr>
<td>Window</td>
<td>Enum</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1 to filter C1
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Filter"
app.Math.F1.Operator1Setup.FirOrIir = "FIR"

\textbf{AutoLength} \hspace{2cm} \textit{Bool}

Description

Enables/Disables/Queries status of the auto-length feature for the filter.

\textbf{CosineBeta} \hspace{2cm} \textit{Double}

Range: From 0 to 100, step 1

Description

Sets/Queries the constant beta for the raised root cosine filter Fx, as a percentage.
**CustomFilename**

**FileName**

Range: Any number of characters

Description

Sets/Queries the name of the current custom file for filter Fx. Valid only when FilterKind is set to "Custom".

**FilterKind**

**Enum**

Description

Sets/Queries kind of filter to use in function Fx.

Values

- BandPass
- BandStop
- Custom
- Gaussian
- HighPass
- LowPass
- RaisedCosine
- RaisedRootCosine

**FilterType**

**Enum**

Description

Sets/Queries the type of filter to use in function Fx.

Values

- Bessel
- Butterworth
- Chebyshev
- InverseChebyshev

**FirOrIir**

**Enum**

Description

Sets/Queries whether filter Fx is an FIR filter or an IIR filter.

Values

<table>
<thead>
<tr>
<th>FIR</th>
<th>Finite Impulse Response Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIR</td>
<td>Infinite Impulse Response Filter</td>
</tr>
</tbody>
</table>
**GaussianBT**

*Double*

- **Range:** From 0 to 100, step 1
- **Description:**
  Sets/Queries the value of the Gaussian BT constant for filter Fx. Valid only when FilterKind is set to "Gaussian".

**HighFreqPass**

*Double*

- **Range:** From 1000 to 1e+10, step 1
- **Description:**
  Sets/Queries the higher cut-off frequency for high-pass filter Fx.

**LowFreqPass**

*Double*

- **Range:** From 1000 to 1e+10, step 1
- **Description:**
  Sets/Queries the lower cut-off frequency for band-pass filter Fx.

**NumberOfTaps**

*Integer*

- **Range:** From 0 to 2001, step 1
- **Description:**
  Sets/Queries the number of taps in filter Fx. Valid only when the AutoLength control is set to False.

**PassBandAttenuation**

*Double*

- **Range:** From 0.5 to 20, step 0.1
- **Description:**
  Sets/Queries the pass-band attenuation of filter Fx.

**PassBandRipple**

*Double*

- **Range:** From 0.5 to 20 step 0.1
- **Description:**
  Sets/Queries the pass-band ripple.
**PART TWO: REFERENCE**

**Rolloff**

*Description*
Sets/Queries the roll-off of filter Fx.

**StopBandAttenuation**

*Description*
Sets/Queries the stop-band attenuation of filter Fx.

**TransitionWidth**

*Description*
Sets/Queries the width of the transition in the frequency spectrum of filter Fx.

**Window**

*Description*
Sets/Queries the type of window for filter Fx.

*Values*
Bartlett
Blackman
Hamming
Hanning
Kaiser
Rectangular
FLOOR

\texttt{app.Math.Fx.OperatorYSetup (Operator = "Floor")}

Most negative or minimum values for an ensemble of sweeps.

<table>
<thead>
<tr>
<th>ClearSweeps</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sweeps</td>
<td>Integer</td>
</tr>
</tbody>
</table>

**Example**

```vbs
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1 to measure the Floor of the first 1000 sweeps of C1
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Floor"
```

**ClearSweeps**

**Action**

- Description
  - Initiates a Clear Sweeps action for Floor function trace Fx.

**Sweeps**

**Integer**

- **Range:** From 1 to 1000000, step 1
- **Description**
  - Sets/Queries the maximum number of sweeps for Floor function trace Fx.
HISTOGRAM

`app.Math.Fx.OperatorYSetup (Operator = "Histogram")`

Histogram of the values of a parameter. Or, if a waveform is used as the input, histogram the waveform sample amplitudes.

<table>
<thead>
<tr>
<th>AutoFindScale</th>
<th>Bool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bins</td>
<td>DoubleLockstep</td>
</tr>
<tr>
<td>Center</td>
<td>Double</td>
</tr>
<tr>
<td>ClearSweeps</td>
<td>Action</td>
</tr>
<tr>
<td>FindScale</td>
<td>Action</td>
</tr>
<tr>
<td>HorScale</td>
<td>DoubleLockstep</td>
</tr>
<tr>
<td>Values</td>
<td>Integer</td>
</tr>
<tr>
<td>VerScaleType</td>
<td>Enum</td>
</tr>
</tbody>
</table>

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1 to histogram the first 200000 sample
' values from source waveform C1 into 50 bins.
' Auto find-scale is enabled.
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Histogram"
```

**AutoFindScale**

*Bool*

Description

Enables/Disables automatic scale setting for histogram function Fx.

**Bins**

*DoubleLockstep*

Range: From 20 to 2000, step 1, locked to 1-2-5

Description

Sets/Queries the number of bins in histogram function Fx.
CHAPTER THREE  
Math/Measure Control Reference

Center  
Range: -1e+010 to 1e+010, step 1e-012 
Description  
Sets/Queries the horizontal value at the center of the graticule of histogram function Fx.

ClearSweeps  
Description  
Clears the contents of all the bins of histogram function Fx.

FindScale  
Description  
Creates a suitable horizontal position and scale to include all the non-empty bins of histogram Fx.

HorScale  
Range: From 1e-012 to 1e+012, step 0.01, locked to 1-2-5 
Description  
Sets/Queries the horizontal scale in units per division for histogram function Fx. Use the FindScale control to automatically determine the scale by looking at the non-zero populated bins.

Values  
Range: From 20 to 2000000000, step 1 
Description  
Sets/Queries the maximum number of values from the source result to include in the histogram function Fx.

VerScaleType  
Description  
Sets/Queries the way that the vertical scale is calculated as histogram Fx grows.

Example  
Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO")

  ' Set function F1 as histogram.
  app.Math.F1.Operator1 = "Histogram"
  ' Set the vertical scale type to linear with constant maximum.
  app.Math.F1.Operator1Setup.VerScaleType = "LinConstMax"

Values  

<table>
<thead>
<tr>
<th>LinConstMax</th>
<th>Linear scale with constant maximum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>Linear scale</td>
</tr>
</tbody>
</table>
INTEGRAL

`app.Math.Fx.OperatorYSetup (Operator = "Integral")`

Integral of the linearly rescaled (multiplier and adder) input.

<table>
<thead>
<tr>
<th>Adder</th>
<th>Double</th>
</tr>
</thead>
<tbody>
<tr>
<td>FindScale</td>
<td>Action</td>
</tr>
<tr>
<td>Multiplier</td>
<td>Double</td>
</tr>
<tr>
<td>VerOffset</td>
<td>Double</td>
</tr>
<tr>
<td>VerScale</td>
<td>DoubleLockstep</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1 to integrate C1
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Integral"
app.Math.F1.Operator1Setup.Adder = 0.1
app.Math.F1.Operator1Setup.FindScale

Adder

Double

Range: From -1e-009 to 1e-009, step 1e-002

Description

Sets/Queries the additive A for integral function Fx.

FindScale

Action

Description

Initiates an action to find suitable vertical offset and scale for integral function trace Fx.

Multiplier

Double

Range: From -1e+006 to 1e+006, step 1e-006

Description

Sets/Queries the multiplying constant M for integral function Fx.
VerOffset

Range: From -1e+006 to 1e+006, step 1e-012

Description
Sets/Queries the vertical offset for integral function trace Fx.

VerScale

Range: From 1e-009 to 1e+007 step 0.01, locked to 1-2-5

Description
Sets/Queries the vertical scale for integral function trace Fx.
INTERPOLATE  \texttt{app.Math.Fx.OperatorYSetup(Operator = "Interpolate")}

Interpolate, producing more points in the resulting waveform using linear, cubic, or weighted \(\sin(x)/x\) algorithms.

<table>
<thead>
<tr>
<th>Expand</th>
<th>DoubleLockstep</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enum</td>
</tr>
</tbody>
</table>

**Example**

' Visual Basic Script

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1 to interpolate C1
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Interpolate"
app.Math.F1.Operator1Setup.InterpolateType = "Cubic"
```

**Expand**

**DoubleLock step**

*Range:* From 2 to 50, step 0.1, locked to 1-2-5

**Description**

Sets/Queries the sampling expansion ratio for the interpolation function \(F_x\).

**InterpolateType**

**Enum**

**Description**

Sets/Queries the type of interpolation for function trace \(F_x\).

**Values**

- Cubic
- Linear
- \(\sin xx\)
**MATH SCRIPT**  

`app.Math.Fx.OperatorYSetup (Operator = "WaveScript")`

Visual basic script that produces a waveform from one or two input waveforms.

**Math script**

<table>
<thead>
<tr>
<th>Code</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Enum</td>
</tr>
<tr>
<td>Status</td>
<td>String</td>
</tr>
</tbody>
</table>

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Place sample program code in a string
' This will invert the first 200 points of the waveform
code = "function Update()" + vbCrLf
code = code + "    unscaledData = InResult.DataArray(False)" + vbCrLf
code = code + "    For i = 0 to 200" + vbCrLf
code = code + "        unscaledData(i) = -unscaledData(i)" + vbCrLf
code = code + "    next" + vbCrLf
code = code + "    OutResult.DataArray(False) = unscaledData" + vbCrLf
code = code + "end Function"

' Configure F1 to integrate C1
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
```

**Code**  

**String**

*Range: Any number of characters*

**Description**

Sets/Queries the scripting code used in math script function Fx. For complete details of programming the instrument in VBScript, please view the online Help and the relevant section of this manual.
Language

<table>
<thead>
<tr>
<th>Enum</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JScript</td>
<td>Sets/Queries the language used for math script function F.x.</td>
</tr>
<tr>
<td>Java Script</td>
<td></td>
</tr>
<tr>
<td>VBScrip t</td>
<td>Visual Basic Script</td>
</tr>
</tbody>
</table>

Values

Example

```javascript
```

Status

<table>
<thead>
<tr>
<th>String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any number of characters</td>
<td>Inspects the status of the script operation. A typical message would be &quot;Error at line 23, Type mismatch&quot;; or &quot;OK&quot; if the supplied code executed successfully.</td>
</tr>
</tbody>
</table>
MATHCADMATH  

*app.Math.Fx.OperatorY*Setup (Operator = "MathcadMath")

Produces a waveform using a user specified Mathcad function.

<table>
<thead>
<tr>
<th>Advanced</th>
<th>Bool</th>
</tr>
</thead>
<tbody>
<tr>
<td>FindScale</td>
<td>Action</td>
</tr>
<tr>
<td>NewSheet</td>
<td>Bool</td>
</tr>
<tr>
<td>OutputEnable</td>
<td>Bool</td>
</tr>
<tr>
<td>OutputHeaderVar</td>
<td>String</td>
</tr>
<tr>
<td>OutputVar</td>
<td>String</td>
</tr>
<tr>
<td>Reload</td>
<td>Action</td>
</tr>
<tr>
<td>Scaling</td>
<td>Enum</td>
</tr>
<tr>
<td>Source1Enable</td>
<td>Bool</td>
</tr>
<tr>
<td>Source1HeaderVar</td>
<td>String</td>
</tr>
<tr>
<td>Source1Var</td>
<td>String</td>
</tr>
<tr>
<td>Source2Enable</td>
<td>Bool</td>
</tr>
<tr>
<td>Source2HeaderVar</td>
<td>String</td>
</tr>
<tr>
<td>Source2Var</td>
<td>String</td>
</tr>
<tr>
<td>Status</td>
<td>String</td>
</tr>
<tr>
<td>WithHeader</td>
<td>Bool</td>
</tr>
<tr>
<td>WorksheetFilename</td>
<td>FileName</td>
</tr>
</tbody>
</table>

**Example**

' Visual Basic Script  
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1 to process C1 using Mathcad  
app.Math.F1.View = True  
app.Math.F1.Source1 = "C1"  
app.Math.F1.MathMode = "OneOperator"  

**Advanced**  

Description

Enables/Disables/Queries the use of the advanced features. When in advanced mode the names used for source and output vectors, in addition to names used for source and output headers, may be modified from their default values.

**FindScale**

Description

Sets a suitable vertical scale of the Mathcad output trace on the instrument graticule. Valid only when Manual scaling is specified.
NewSheet
  Description
  Enables/Disables/Queries the creation of a new Mathcad worksheet.
  Example
  ' Visual Basic Script
  Set app = CreateObject("LeCroy.XStreamDSO")
  ' Enable creation of a new Mathcad worksheet.

OutputEnable
  Description
  Enables/Disables/Queries the transmission of output data from Mathcad to the instrument.
  Example
  ' Visual Basic Script
  Set app = CreateObject("LeCroy.XStreamDSO")
  ' Enable transmission of output data.

OutputHeaderVar
  Range: Any number of characters
  Description
  Sets/ Queries the name in Mathcad of the output header variable.
  Example
  ' Visual Basic Script
  Set app = CreateObject("LeCroy.XStreamDSO")
  ' Enables use of headers.
  app.Math.F1.Operator1Setup.WithHeader = True
  ' Sets the name of the output header variable
  app.Math.F1.Operator1Setup.OutputHeaderVar = "header1"
OutputVar

String
Range: Any number of characters
Description
Sets/Queries the name in Mathcad of the output variable.
Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set function F1 as Mathcad calculation.
' Sets the name of the output variable in Mathcad.

Reload

Action
Description
Reloads a specified Mathcad worksheet. If the worksheet does exist, the system creates an empty one
with a name of the form "UntitledN", where N is an integer.
Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set function F1 as Mathcad calculation.
' Specifies a Mathcad worksheet name.
app.Math.F1.Operator1Setup.WorksheetFilename = "XStream34.mcd"
' Reload a Mathcad worksheet.

Scaling

Enum
Description
Sets/Queries the method of vertical scaling of the Mathcad output trace on the instrument graticule.
Values
Automatic
Manual

Source1Enable

Bool
Description
Enables/Disables/Queries the transmission of source 1 data from the instrument to Mathcad.
Source1HeaderVar
Range: Any number of characters
Description
Sets/Queries the name in Mathcad of input 1 header variable.

Source1Var
Range: Any number of characters
Description
Sets/Queries the name in Mathcad of input variable 1.

Source2Enable
Description
Enables/Disables/Queries the transmission of source 2 data from the instrument to Mathcad.

Source2HeaderVar
Range: Any number of characters
Description
Sets/Queries the name in Mathcad of input 2 header variable.

Source2Var
Range: Any number of characters
Description
Sets/Queries the name in Mathcad of input variable 2.

Status
Range: Any number of characters
Description
Inspects the status of the Mathcad calculation.

WithHeader
Description
Enables/Disables/Queries inclusion of headers in the Mathcad calculation.

WorksheetFilename
Range: Any number of characters
Description
Sets/Queries a Mathcad worksheet file name.
MATLAB MATH

Math/Measure Control Reference

CHAPTER THREE

MATLAB MATH

app.Math.Fx.OperatorYSetup (Operator = “MATLABWaveform”)

Produces a waveform using a user specified MATLAB function.

<table>
<thead>
<tr>
<th>MATLABCode</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLABPlot</td>
<td>Bool</td>
</tr>
<tr>
<td>MATLABScalePerDiv</td>
<td>Double</td>
</tr>
<tr>
<td>MATLABZeroOffset</td>
<td>Double</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1 to process C1 using MATLAB
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "MATLAB math"
app.Math.F1.Operator1Setup.MATLABCode = "WformOut = -0.5 * WformIn"

MATLABCode

String

Range: Any number of characters
Description

Sends/Inspects the MATLAB code.

MATLABPlot

Bool

Description

Enables/Disables a MATLAB plot.

MATLABScalePerDiv

Double

Range: From 1e-009 to 1e+009, step 1e-009
Description

Sets/Queries the vertical scale in units per division, of the MATLAB output trace on the instrument graticule.
MATLABZeroOffset  

**Description**

Sets/Queries the vertical zero offset of the MATLAB output trace on the scope graticule. This is the position on the graticule where zero is found: a positive offset moves the trace downwards; a negative offset moves it upwards.
PHISTOGRAM  

\[ \text{app.Math.Fx.OperatorYSetup (Operator = "PersistenceHistogram")} \]

Histogram of a slice of a persistence map.

<table>
<thead>
<tr>
<th>CenterCursor</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClearSweeps</td>
<td>Action</td>
</tr>
<tr>
<td>CutDirection</td>
<td>Enum</td>
</tr>
<tr>
<td>HorCutCenter</td>
<td>Double</td>
</tr>
<tr>
<td>HorCutWidth</td>
<td>Double</td>
</tr>
<tr>
<td>PctCutWidth</td>
<td>Double</td>
</tr>
<tr>
<td>VerCutCenter</td>
<td>Double</td>
</tr>
<tr>
<td>VerCutWidth</td>
<td>Double</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1 to generate a slice of the persistence map of C1
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "PHistogram"

CenterCursor  
Action

Description
Centralizes the cut on the graticule for persistence histogram Fx.

ClearSweeps  
Action

Description
Initiates a clearance of the contents of persistence histogram Fx.

CutDirection  
Enum

Description
Sets/Queries the direction of the cut the persistence histogram Fx.

<table>
<thead>
<tr>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Horizontal</strong></td>
</tr>
<tr>
<td><strong>Vertical</strong></td>
</tr>
</tbody>
</table>
**Part Two: Reference**

**HorCutCenter**  
*Double*  
Range: From -10 to 10, step 0.0001  
Description  
Sets/Queries the position of the center of the horizontal cut (measured in the units of vertical scale) of the slice to be used in the persistence histogram Fx.

**HorCutWidth**  
*Double*  
Range: From -10 to 10, step 0.0001  
Description  
Sets/Queries the width of the horizontal cut (measured in the units of vertical scale) of the slice to be used in persistence histogram Fx.

**PctCutWidth**  
*Double*  
Range: From 0 to 100, step 0.1  
Description  
Sets/Queries the width of the cut in percent.

**VerCutCenter**  
*Double*  
Range: From -1 to 1, step 1e-009  
Description  
Sets/Queries the position of the center of the vertical cut (measured in the units of horizontal scale) of the slice to be used in persistence histogram Fx.

**VerCutWidth**  
*Double*  
Range: From -1 to 1, step 1e-009  
Description  
Sets/Queries the width of the vertical cut (measured in the units of horizontal scale) of the slice to be used in the persistence histogram Fx.
PTRACE MEAN

Create a waveform from the mean of a persistence map.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1 to generate a waveform from the
' mean value of the persistence map of C1
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Ptrace mean"

ClearSweeps

Description

Clears the contents of persistence mean trace Fx.
PTRACE RANGE

Generates a waveform with a width derived from a population range of a persistence map.

<table>
<thead>
<tr>
<th>ClearSweeps</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>PctPopulation</td>
<td>Double</td>
</tr>
</tbody>
</table>

**Example**

' Visual Basic Script

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Ptrace range"
```

**ClearSweeps**

*Action*

Description

Clears the contents of persistence range trace Fx.

**PctPopulation**

*Double*

| Range: From 0.5 to 100, step 0.5 |

Description

Sets/Queries the percentage of the persistence population that is spanned by persistence range trace Fx.
**PTRACE SIGMA**

Generates a waveform with a width derived from the sigma of a persistence map.

<table>
<thead>
<tr>
<th><strong>ClearSweeps</strong></th>
<th><strong>Action</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sigma</strong></td>
<td><strong>Double</strong></td>
</tr>
</tbody>
</table>

**Example**

```Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Ptrace sigma"
app.Math.F1.Operator1Setup.Sigma = 5.0
```

**ClearSweeps**

- **Description**
  - Clears the contents of persistence sigma trace Fx.

**Sigma**

- **Range:** From 0.5 to 10, step 0.1
- **Description**
  - Sets/Queries the number of standard deviations of the persistence population that is spanned by sigma trace Fx.
RESCALE

Linearly transform the vertical values of a waveform.

Example

Visual Basic Script

```vbscript
Set app = CreateObject("LeCroy.XStreamDSO")

app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Rescale"
app.Math.F1.Operator1Setup.CustomUnit = True
app.Math.F1.Operator1Setup.Unit = "DEG"
```

**Adder**

<table>
<thead>
<tr>
<th>Adder</th>
<th>Double</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range:</td>
<td>From -1.79769e+308 to 1.79769e+308, step 0</td>
</tr>
<tr>
<td>Description</td>
<td>Sets/Queries the additive constant A in the rescale function Fx = M.Input + A</td>
</tr>
</tbody>
</table>

**CustomUnit**

<table>
<thead>
<tr>
<th>CustomUnit</th>
<th>Bool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Enables/Disables the application of a custom unit of measurement to rescale function trace Fx.</td>
</tr>
</tbody>
</table>

**Multiplier**

<table>
<thead>
<tr>
<th>Multiplier</th>
<th>Double</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range:</td>
<td>From -1.79769e+308 to 1.79769e+308 step 0</td>
</tr>
<tr>
<td>Description</td>
<td>Sets/Queries the multiplicative constant M in the rescale function Fx = M.Input + A.</td>
</tr>
</tbody>
</table>
### Unit

**Range:** Any number of characters

**Description:**
Sets/Queries the custom unit for rescale function trace Fx. Only used when the CustomUnit control is set to True.
ROOF

\textit{app.Math.Fx.OperatorYSetup (Operator = "Roof")}

The most positive or maximum values for an ensemble of sweeps, or "Roof."

<table>
<thead>
<tr>
<th>ClearSweeps</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sweeps</td>
<td>Integer</td>
</tr>
</tbody>
</table>

**Example**

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1 to measure the Roof of the first 1000 sweeps of C1
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Roof"

**ClearSweeps**

**Description**

Initiates a clear sweeps action for roof function trace Fx.

**Sweeps**

**Range:** From 1 to 1000000, step 1

**Description**

Sets/Queries the maximum number of sweeps for Roof function trace Fx.

**Example**

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Set function trace F2 to roof.
app.Math.F2.Operator1 = "Roof"

' Set the maximum number of sweeps to 150.
SEGMENT

`app.Math.Fx.OperatorYSetup (Operator = "SegmentSelect")`

Selects one waveform from a group of waveforms.

**Example**

```plaintext
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Enable sequence acquisition mode, collect 10 segments
app.Acquisition.Horizontal.NumSegments = 10
app.Acquisition.Horizontal.SampleMode = "Sequence"

' Configure F1 to show the 5th of the 10 collected segments
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1 = "Segment"
```

**SelectedSegment**

*Integer*

- **Range:** From 1 to 1000000000, step 1
- **Description:**
  In sequence mode, sets/queries the number of the segment selected from a set.
**PART TWO: REFERENCE**

**SPARSE**

`app.Math.Fx.OperatorYSetup (Operator = "Sparse")`

Waveform sparser, will reduce the number of points in the output waveform by skipping points in the input, and starting at a given offset.

<table>
<thead>
<tr>
<th>SparsingFactor</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>SparsingPhase</td>
<td>Integer</td>
</tr>
</tbody>
</table>

**Example**

`Visual Basic Script`

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

' Configure F1 to sparse C1 by a factor of 100
app.Math.F1.View = True
app.Math.F1.Source1 = "C1"
app.Math.F1.MathMode = "OneOperator"
app.Math.F1.Operator1Setup.Spar singFactor = 100
```

**SparsingFactor**

*Integer*

**Range:** From 1 to 1000000, step 1

**Description**

Sets/Queries the factor by which the number of samples is reduced in the sparsing function Fx.

**SparsingPhase**

*Integer*

**Range:** From 0 to 0, step 1

**Description**

Sets/Queries the number of the first sample that will be accepted by the sparsing function Fx.
**TRACK**

*app.Math.Fx.OperatorYSetup (Operator = "Track")*

The "Track" function gives a waveform of equivalent horizontal scale to the source waveform, but of a measurement on that waveform.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoFindScale</strong></td>
<td><strong>Bool</strong></td>
</tr>
<tr>
<td><strong>Center</strong></td>
<td><strong>Double</strong></td>
</tr>
<tr>
<td><strong>FindScale</strong></td>
<td><strong>Action</strong></td>
</tr>
<tr>
<td><strong>VerScale</strong></td>
<td><strong>DoubleLockstep</strong></td>
</tr>
</tbody>
</table>

**AutoFindScale**

Description

Enables/Disables the automatic setting of the vertical scale and vertical offset for Track trace Fx.

**Center**

Range: From \(-1e+010\) to \(1e+010\), step \(1e-012\)

Description

Sets/Queries the vertical position of the center of Track trace Fx.

**FindScale**

Description

Sets the vertical scale and offset to optimum values to display Track trace Fx.

**VerScale**

Range: From \(1e-012\) to \(1e+012\) step 0.01, locked to 1-2-5

Description

Sets/Queries the vertical scale of Track trace Fx.
TRENDF

app.Math.Fx.OperatorYSetup (Operator = "Trend")

Trend of the values of a parameter, if connected to a parameter result source, or a trend of the sample values of a waveform, if connected to a waveform result.

<table>
<thead>
<tr>
<th>AutoFindScale</th>
<th>Bool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>Double</td>
</tr>
<tr>
<td>ClearSweeps</td>
<td>Action</td>
</tr>
<tr>
<td>FindScale</td>
<td>Action</td>
</tr>
<tr>
<td>Values</td>
<td>Integer</td>
</tr>
<tr>
<td>VerScale</td>
<td>DoubleLockstep</td>
</tr>
</tbody>
</table>

AutoFindScale descriptor

Description

Enables/Disables the automatic setting of the vertical scale and vertical offset for Trend trace Fx.

Center descriptor

Range: From -1e+010 to 1e+010, step 1e-012

Description

Sets/Queries the vertical position of the center of Trend trace Fx.

ClearSweeps descriptor

Description

Clears the contents of Trend trace Fx.

FindScale descriptor

Description

Sets the vertical scale and offset to optimum values to display Trend trace Fx.

Values descriptor

Range: From 20 to 1000000, step 1

Description

Sets/Queries the number of visible values in Trend trace Fx.

VerScale descriptor

Range: From 1e-012 to 1e+012 step 0.01, locked to 1-2-5

Description

Sets/Queries the vertical scale of Trend trace Fx.
MATHCADPARAMARITH  
\texttt{app.Measure.Px.Operator (ArithEngine = "MathcadParamArith")}

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced</td>
<td>Bool</td>
</tr>
<tr>
<td>NewSheet</td>
<td>Bool</td>
</tr>
<tr>
<td>OutputEnable</td>
<td>Bool</td>
</tr>
<tr>
<td>OutputHeaderVar</td>
<td>String</td>
</tr>
<tr>
<td>OutputVar</td>
<td>String</td>
</tr>
<tr>
<td>Reload</td>
<td>Action</td>
</tr>
<tr>
<td>Source1Enable</td>
<td>Bool</td>
</tr>
<tr>
<td>Source1HeaderVar</td>
<td>String</td>
</tr>
<tr>
<td>Source1Var</td>
<td>String</td>
</tr>
<tr>
<td>Source2Enable</td>
<td>Bool</td>
</tr>
<tr>
<td>Source2HeaderVar</td>
<td>String</td>
</tr>
<tr>
<td>Source2Var</td>
<td>String</td>
</tr>
<tr>
<td>Status</td>
<td>String</td>
</tr>
<tr>
<td>WithHeader</td>
<td>Bool</td>
</tr>
<tr>
<td>WorksheetFilename</td>
<td>FileName</td>
</tr>
</tbody>
</table>

**Advanced**

Description

Enables/Disables/Queries use of advanced features for Mathcad parameter Px.

Example

' Visual Basic Script

\begin{verbatim}
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 as Mathcad calculation.
' Enables use of advanced features.
\end{verbatim}

**NewSheet**

Description

Enables/Disables/Queries the use of a new Mathcad worksheet for parameters.

Example

' Visual Basic Script

\begin{verbatim}
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 as Mathcad calculation.
' Enable new Mathcad worksheet.
\end{verbatim}
**PART TWO: REFERENCE**

---

**OutputEnable**

**Description**

Enables/Disables/Queries transmission of output data from Mathcad to the instrument.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 as Mathcad calculation.
' Enables transmission of Mathcad output to the instrument.

---

**OutputHeaderVar**

**String**

<table>
<thead>
<tr>
<th>Range: Any number of characters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Sets/Queries the name of the output variable header in Mathcad parameter Px.</td>
</tr>
</tbody>
</table>

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 as Mathcad calculation.
' Sets the name of the Mathcad output header variable

---

**OutputVar**

**String**

<table>
<thead>
<tr>
<th>Range: Any number of characters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Sets/Queries the name of the output variable in Mathcad parameter Px.</td>
</tr>
</tbody>
</table>

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 as Mathcad calculation.
' Sets the name of the Mathcad output variable
**Reload**

**Description**
Reloads the specified Mathcad file.

**Example**
```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 as Mathcad calculation.
' Specify a Mathcad worksheet filename.
' Reload the specified Mathcad file.
```

**Source1Enable**

**Description**
Enables/Disables/Queries transmission of source data 1 from instrument to Mathcad.

**Example**
```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 as Mathcad calculation.
' Enables transmission of source 1 data to instrument.
```

**Source1HeaderVar**

**Range:** Any number of characters

**Description**
Sets/Queries the name of the input variable 1 header in Mathcad parameter Px.

**Example**
```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 as Mathcad calculation.
' Sets the name of the Mathcad source 1 header variable
app.Measure.P3.Operator.Source1HeaderVar = "input1header"
```
Source1Var

String

Range: Any number of characters

Description
Sets/Queries the name of input variable 1 in Mathcad parameter Px.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 as Mathcad calculation.
' Sets the name of the source 1 variable
app.Measure.P3.Operator.Source1Var = "input1"

Source2Enable

Bool

Description
Enables/Disables/Queries transmission of source data 2 from instrument to Mathcad.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 as Mathcad calculation.
' Enables transmission of source 2 data to instrument.

Source2HeaderVar

String

Range: Any number of characters

Description
Sets/Queries the name of input variable 2 header in Mathcad parameter Px.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 as Mathcad calculation.
' Sets the name of the Mathcad source 2 header variable
app.Measure.P3.Operator.Source2HeaderVar = "input2header"
Source2Var  
- **String**
- **Range:** Any number of characters
- **Description:** Sets/Queries the name of input variable 2 in Mathcad parameter Px.
- **Example**
  ```vbnet
  ' Visual Basic Script
  Set app = CreateObject("LeCroy.XStreamDSO")
  
  ' Set parameter P3 as Mathcad calculation.
  ' Sets the name of the source 2 variable
  ```

Status  
- **String**
- **Range:** Any number of characters
- **Description:** Inspects the status of Mathcad parameter calculation Px.
- **Example**
  ```vbnet
  ' Visual Basic Script
  Set app = CreateObject("LeCroy.XStreamDSO")
  
  ' Set parameter P3 as Mathcad calculation.
  ' Inspect status of Mathcad parameter calculation.
  ```

WithHeader  
- **Bool**
- **Description:** Enables/Disables/Queries the presence of headers with variables with Mathcad parameter Px.
- **Example**
  ```vbnet
  ' Visual Basic Script
  Set app = CreateObject("LeCroy.XStreamDSO")
  
  ' Set parameter P3 as Mathcad calculation.
  ' Enables inclusion of headers with data.
  ```
WorksheetFilename

Range: Any number of characters

Description
Sets/Queries the name of the current Mathcad file for parameter Px.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 as Mathcad calculation.
' Specify a Mathcad worksheet filename.
**P INVERT**  
`app.Measure.Px.Operator (ArithEngine = "ParamInvert")`

This engine produces the reciprocal of the input parameter.

**CycleForTimeUnits**  
*Bool*

**Description**

Sets/Queries the use of period as the measurement for time units, rather than 1/Hz.

When true, if there are seconds in the vertical dimensions of the source parameter, it is assumed that there is an implicit "cycles" per second. Thus the resulting units upon inversion is Hertz (cycles per second).
**PART TWO: REFERENCE**

**P SCRIPT**

```
app.Measure.Px.Operator (ArithEngine = "ParamScript")
```

Calculates a parameter result from one or two parameter inputs

<table>
<thead>
<tr>
<th>Code</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Enum</td>
</tr>
<tr>
<td>Status</td>
<td>String</td>
</tr>
<tr>
<td>Timeout</td>
<td>Double</td>
</tr>
</tbody>
</table>

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Place sample program code in a string
' This will the square of the input value times 3.1
code =        "function Update()" + vbCrLf
code = code + "    value = InResult.Value(False)" + vbCrLf
code = code + "    OutResult.Value(False) = value * value * 3.1"
+ vbCrLf
code = code + "end Function"

' Configure P1 as a scripting component
app.Measure.P1.View = True
app.Measure.P1.Source1 = "C1"
app.Measure.P1.MeasurementType = "math"
app.Measure.P1.ArithEngine = "P Script"
```

**Code**

*String*

**Range:** Any number of characters

**Description**

Sets/Queries the code used to calculate parameter Px. This code is, of course, text.

**Language**

*Enum*

**Description**

Sets/Queries the language for parameter script Fx (i.e., choice of scripting language).

**Values**

- JScript
- VBScript
CHAPTER THREE

Math/Measure Control Reference

Status  String
Range: Any number of characters
Description
Inspects the status of parameter script Px. A typical message is "Error in line 14, Expected end of statement."

Timeout  Double
Range: From 1 to 12000, step 0.001
Description
Sets/Queries the time-out for parameter script Px.
Q

app.Measure.Px.Operator (ParamEngine = "EyeQ")

Calculates the Q-factor based on a vertical slice from the (assumed input) eye diagram. The width of the vertical slice is specified through PctCutWidth.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.P1.ParamEngine = "EyeQ"

PctCutWidth

Double

Range: From 0 to 100, step 0.1

Description
Sets/Queries the vertical slice or cut width in percent.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Sets the cut width to 8%.
## AREA

`app.Measure.Px.Operator (ParamEngine = "Area")`

Calculates the area of the input waveform relative to zero.

**Example**

```vbs
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to area.
app.Measure.P1.View = True
app.Measure.P1.MeasurementType = "measure"
app.Measure.P1.ParamEngine = "Area"
app.Measure.P1.Source1 = "C1"

' Enable cyclic calculation of area.

' Set markers to simple.
```

<table>
<thead>
<tr>
<th>Cyclic</th>
<th>Bool</th>
</tr>
</thead>
</table>

**Description**

Enables/Disables cyclic calculation of area parameter Px, which is calculated using a whole number of cycles of the signal.

**Note:** Help Markers aid in selecting the region of the waveform where measurement is made.
AVG POWER \texttt{app.Measure.Px.} \texttt{Operator (ParamEngine = "EyeAvgPower")}

The measurement is assumed to be on an eye-diagram, and the percentage of a unit-interval at the center of the eye is used to estimate the average power.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")


' Sets the cut width to 8\% of the unit interval.

\textbf{PctCutWidth} \hspace{1cm} \textit{Double}

\begin{itemize}
  \item Range: From 0 to 100, step 0.1
  \item Description: Sets/Queries the cut width in percent.
\end{itemize}
Delta-period at level, the difference between successive (adjacent periods), also known as "cycle-to-cycle jitter."

### Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsLevel</td>
<td>Double</td>
</tr>
<tr>
<td>BaseFrequency</td>
<td>Double</td>
</tr>
<tr>
<td>FindBaseFrequency</td>
<td>Action</td>
</tr>
<tr>
<td>FindLevel</td>
<td>Action</td>
</tr>
<tr>
<td>GroupSize</td>
<td>Integer</td>
</tr>
<tr>
<td>Hysteresis</td>
<td>Double</td>
</tr>
<tr>
<td>LevelType</td>
<td>Enum</td>
</tr>
<tr>
<td>PercentLevel</td>
<td>Double</td>
</tr>
<tr>
<td>SignalType</td>
<td>Enum</td>
</tr>
<tr>
<td>Slope</td>
<td>Enum</td>
</tr>
<tr>
<td>StartCycle</td>
<td>Integer</td>
</tr>
<tr>
<td>StdBaseFrequency</td>
<td>Enum</td>
</tr>
<tr>
<td>UseBaseFrequency</td>
<td>Enum</td>
</tr>
</tbody>
</table>

#### Example

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta period at level.
app.Measure.MeasureMode = "MyMeasure"
app.Measure.P1.MeasurementType = "measure"
app.Measure.P1.ParamEngine = "DeltaPeriodAtLevel"
' Set level type to absolute.
app.Measure.P1.Operator.LevelType = "Absolute"
' Set the measurement level to 5 mV.
app.Measure.P1.Operator.AbsLevel = 0.005
```

**AbsLevel**

**Double**

Range: From -100 to 100, step 1e-007

Description

Sets/Queries the absolute measuring level for parameter period at level Px.
**BaseFrequency**

*Double*

**Range:** From 1 to 4e+010, step 1000

**Description**

Sets/Queries the base frequency of the signal for the parameter delta period at level. This only has effect when the UseBaseFrequency = "Custom", otherwise the base frequency is set directly from the standard selection.

When the period measurements are made on a data-stream instead of a clock, the engine needs to be told, or to learn, the base frequency (effective clock frequency). Since precision is exceptionally important, it's most often the case that the engine is asked to find the frequency, and then this control is queried to verify that the correct frequency has been found.

**FindBaseFrequency**

*Action*

**Description**

Finds the base frequency of the signal by inspecting the signal, if you are not supplying a standard value or a custom value.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta period at level.
app.Measure.P1.ParamEngine = "DeltaPeriodAtLevel"

' Find the base frequency of the signal.

**FindLevel**

*Action*

**Description**

Finds the 50 percent level for delta period measurements. Level type may be either absolute or percentage.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta period at level.
app.Measure.P1.ParamEngine = "DeltaPeriodAtLevel"

' Find the 50 percent level.
GroupSize

**Integer**

**Range:** From 1 to 128, step 1

**Description**
Sets/Queries the size of a group of consecutive cycles for the delta period at level parameter Px. This is an advanced feature.

**Example**
Visual Basic Script

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta period at level.
app.Measure.P1.ParamEngine = "DeltaPeriodAtLevel"
' Set up for clock signal.
app.Measure.P1.Operator.SignalType = "Clock"
' Set the group size to 8.
```

Hysteresis

**Double**

**Range:** From 0 to 10, step 0.1

**Description**
Sets the hysteresis range for parameter delta-period at level Px, specified in divisions. Hysteresis can be used to obtain good noise rejection.

**Example**
Visual Basic Script

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta period at level.
app.Measure.P1.ParamEngine = "DeltaPeriodAtLevel"
' Set the hysteresis range to 0.9 division. (+- 0.45 divisions)
app.Measure.P1.Operator.Hysteresis = 0.9
```

LevelType

**Enum**

**Description**
Sets/Queries whether the level is absolute or a percentage of the amplitude of the signal for Px.

**Example**
Visual Basic Script

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta period at level.
app.Measure.P1.ParamEngine = "DeltaPeriodAtLevel"
```
```
' Set level type to absolute.
app.Measure.P1.Operator.LevelType = "Absolute"
```

**Values**

- Absolute
- Percent

---

### PercentLevel

**Description**

Sets/Queries the percentage measuring level for parameter delta-period at level P_x.

**Example**

```vbs```
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta period at level.
app.Measure.P1.ParamEngine = "DeltaPeriodAtLevel"

' Set level type to percentage.
app.Measure.P1.Operator.LevelType = Percent

' Set the measurement level to 55%
```

---

### SignalType

**Description**

Sets/Queries the type of data specified for the measurement of delta-period at level.

**Example**

```vbs```
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta period at level.
app.Measure.P1.ParamEngine = "DeltaPeriodAtLevel"

' Set the signal type to data.
app.Measure.P1.Operator.SignalType = "Data"
```

**Values**

- Clock
- Data
## Slope

<table>
<thead>
<tr>
<th>Enum</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
| Both   | Sets/Queries the polarity of the transitions for measuring parameter delta-period at level. | Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta period at level.
app.Measure.P1.ParamEngine = "DeltaPeriodAtLevel"

' Set transtion polarity to negative.
app.Measure.P1.Operator.LevelType = "Neg"

| Values | Both | Neg | Pos |
**StartCycle**

**Integer**

**Range:** From 0 to 0, step 1

**Description**

Sets/Queries the start cycle when measuring over groups of clock cycles

**Example**

Visual Basic Script

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta period at level.
app.Measure.P1.ParamEngine = "DeltaPeriodAtLevel"
' Set up for clock signal.
app.Measure.P1.Operator.SignalType = "Clock"
' Set the group size to 8.
' Set the start cycle to 2.
```

**StdBaseFrequency**

**Enum**

**Description**

Sets/Queries the standard base frequency, sending or receiving a string from the list of standard frequencies.

**Example**

Visual Basic Script

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta period at level.
app.Measure.P1.ParamEngine = "DeltaPeriodAtLevel"
' Set up to use the standard frequency
' Set the standard base frequency to 2.048 MHz.
app.Measure.P1.Operator.StdBaseFrequency = "2.048MHz"
```

**Values**

- 1.544MHz
- 139.264MHz
- 155.52MHz
- 2.048MHz
- 2488.32MHz
- 34.368MHz
- 44.736MHz
51.84MHz
622.08MHz
8.448MHz

**UseBaseFrequency**

**Enum**

**Description**
Sets/Queries the choice of a specified base frequency as custom or standard. When "custom" is selected, you must specify or find the base frequency.

**Example**
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta period at level.
app.Measure.P1.ParamEngine = "DeltaPeriodAtLevel"
' Set up to use a standard base frequency.

**Values**
Custom
Standard
DTIME@LEVEL

A calculation of the time between two highly specific transitions.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsLevel1</td>
<td>Double</td>
</tr>
<tr>
<td>AbsLevel2</td>
<td>Double</td>
</tr>
<tr>
<td>FindLevel1</td>
<td>Action</td>
</tr>
<tr>
<td>FindLevel2</td>
<td>Action</td>
</tr>
<tr>
<td>Hysteresis1</td>
<td>Double</td>
</tr>
<tr>
<td>Hysteresis2</td>
<td>Double</td>
</tr>
<tr>
<td>LevelType1</td>
<td>Enum</td>
</tr>
<tr>
<td>LevelType2</td>
<td>Enum</td>
</tr>
<tr>
<td>PercentLevel1</td>
<td>Double</td>
</tr>
<tr>
<td>PercentLevel2</td>
<td>Double</td>
</tr>
<tr>
<td>Slope1</td>
<td>Enum</td>
</tr>
<tr>
<td>Slope2</td>
<td>Enum</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.MeasureMode = "MyMeasure"
app.Measure.P1.MeasurementType = "measure"
app.Measure.P1.ParamEngine = "DeltaTimeAtLevel"
' Set level type for trace 1 to absolute.
app.Measure.P1.Operator.LevelType1 = "Absolute"
' Set the measurement level for trace 1 to 25 mV
app.Measure.P1.Operator.AbsLevel1 = 0.025

AbsLevel1

Double

Range: From -100 to 100, step 1e-005

Description

Sets/Queries the absolute measuring level for the first transition time, if LevelType1 = "Absolute".

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.P1.ParamEngine = "DeltaTimeAtLevel"
' Set level type for trace 1 to absolute.
app.Measure.P1.Operator.LevelType1 = "Absolute"
' Set the measurement level for trace 1 to 25 mV
app.Measure.P1.Operator.AbsLevel1 = 0.025
AbsLevel2  
*Double*

Range: From -100 to 100, step 1e-005  
Description

Sets/Queries the absolute measuring level for the second transition time, if LevelType2 = “Absolute”.

FindLevel1  
*Action*

Description

Sets the measurement level automatically for the first transition time. (Works for both percent and absolute level modes.)

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.P1.ParamEngine = "DeltaTimeAtLevel"
' Find the level for trace 1.
app.Measure.P1.Operator.FindLevel1

FindLevel2  
*Action*

Description

Sets the measurement level automatically for second transition (for LevelType2 either “Percent” or “Absolute”).

Hysteresis1  
*Double*

Range: From 0 to 10, step 0.1  
Description

Sets/Queries the hysteresis level in graticule divisions for first transition time. (This is an advanced feature.)

Hysteresis2  
*Double*

Range: From 0 to 10, step 0.1  
Description

Sets/Queries the hysteresis level in graticule divisions for the second transition. (This is an advanced feature.)
**LevelType1**

*Enum*

Description

Sets/Queries whether to use absolute level or percent level for the first transition time.

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.P1.ParamEngine = "DeltaTimeAtLevel"
' Set level type for trace 1 to percentage.
app.Measure.P1.Operator.LevelType1 = "Percent"
' Set the measurement level for trace 1 to 55%
```

Values

Absolute
Percent

**LevelType2**

*Enum*

Description

Sets/Queries whether to use absolute level or percent level for specifying the second transition.

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.P1.ParamEngine = "DeltaTimeAtLevel"
' Set level type for trace 2 to absolute.
app.Measure.P1.Operator.LevelType2 = "Absolute"
' Set the measurement level for trace 2 to 15 mV
app.Measure.P1.Operator.AbsLevel2 = 0.015
```

Values

Absolute
Percent

**PercentLevel1**

*Double*

Range: From 0 to 100, step 1

Description

Sets/Queries the measuring level, in percent, for the first transition if LevelType1 = “Percent”.

---

3-70

ISSUED: July 2003

901783
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.P1.ParamEngine = "DeltaTimeAtLevel"
' Set level type for trace 1 to percentage.
app.Measure.P1.Operator.LevelType1 = "Percent"
' Set the measurement level for trace 1 to 55%

PercentLevel2

Double
Range: From 0 to 100, step 1
Description
Sets/Queries the measuring level, in percent, for the second transition if LevelType2 = "Percent".
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.P1.ParamEngine = "DeltaTimeAtLevel"
' Set level type for trace 2 to percentage.
app.Measure.P1.Operator.LevelType2 = "Percent"
' Set the measurement level for trace 2 to 45%

Slope1

Enum
Description
Sets/Queries the transition polarity of trace 1 for delta time at level Fx.
Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.P1.ParamEngine = "DeltaTimeAtLevel"
' Set edge polarity for trace 1 to positive.

Values
Both
Neg
Pos
Slope2

Description
Sets/Queries the transition polarity of the second transition time.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.P1.ParamEngine = "DeltaTimeAtLevel"
' Set edge polarity for the second transition to both.
app.Measure.P1.Operator.Slope2 = "Both"

Values
Both
Neg
Pos
**DUTY@LEVEL**

```markdown
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AbsLevel</strong></td>
<td>Double</td>
</tr>
<tr>
<td><strong>FindLevel</strong></td>
<td>Action</td>
</tr>
<tr>
<td><strong>Hysteresis</strong></td>
<td>Double</td>
</tr>
<tr>
<td><strong>HysteresisPct</strong></td>
<td>Double</td>
</tr>
<tr>
<td><strong>HysteresisType</strong></td>
<td>Enum</td>
</tr>
<tr>
<td><strong>LevelType</strong></td>
<td>Enum</td>
</tr>
<tr>
<td><strong>PercentLevel</strong></td>
<td>Double</td>
</tr>
<tr>
<td><strong>Slope</strong></td>
<td>Enum</td>
</tr>
</tbody>
</table>
```

**AbsLevel**

*Double*

- **Range:** From -100 to 100, step 1e-007
- **Description:** Sets/Queries the absolute level used if LevelType = "Absolute"

**FindLevel**

*Action*

- **Description:** Causes the engine to find a suitable level for either LevelType ("Absolute" or "Percent").

**Hysteresis**

*Double*

- **Range:** From 0 to 10, step 0.1
- **Description:** Sets/Queries the hysteresis range in divisions for duty at level Px.

**Example**

' Visual Basic Script

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

app.Measure.P1.ParamEngine = "DutyAtLevel"
app.Measure.P1.Operator.HysteresisType = "Divisions"
app.Measure.P1.Operator.Hysteresis = 0.8
```
HysteresisPct

**Range:** From 0 to 100, step 0.1

**Description**
Sets/Queries the percent hysteresis range for duty at level Px.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.P1.ParamEngine = "DutyAtLevel"
' Set hysteresis measurement type to percentage.
app.Measure.P1.Operator.HysteresisType = "Percent"
' Set the percent hysteresis to 55%
```

HysteresisType

**Enum**

**Description**
Sets/Queries whether to set the hysteresis range in divisions or percent of the signal amplitude.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.P1.ParamEngine = "DutyAtLevel"
' Set hysteresis measurement type to percentage.
app.Measure.P1.Operator.HysteresisType = "Percent"
' Set the percent hysteresis to 55%
```

LevelType

**Enum**

**Values**

```
Absolute
Percent
```

---

**PART TWO: REFERENCE**

**HysteresisPct**

*Double*

**Range:** From 0 to 100, step 0.1

**Description**
Sets/Queries the percent hysteresis range for duty at level Px.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.P1.ParamEngine = "DutyAtLevel"
' Set hysteresis measurement type to percentage.
app.Measure.P1.Operator.HysteresisType = "Percent"
' Set the percent hysteresis to 55%
```

**HysteresisType**

*Enum*

**Description**
Sets/Queries whether to set the hysteresis range in divisions or percent of the signal amplitude.

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.P1.ParamEngine = "DutyAtLevel"
' Set hysteresis measurement type to percentage.
app.Measure.P1.Operator.HysteresisType = "Percent"
' Set the percent hysteresis to 55%
```

**LevelType**

*Enum*

**Values**

```
Absolute
Percent
```
PercentLevel

<table>
<thead>
<tr>
<th>Double</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range:</td>
</tr>
<tr>
<td>Description</td>
</tr>
</tbody>
</table>

Using ParamEngine = "DutyAtLevel", refer to the corresponding variable for the Delta Time At Level parameter.

Slope

<table>
<thead>
<tr>
<th>Enum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
</tbody>
</table>

Sets/Queries the polarity of the first transition used (i.e., classic duty factor is for setting "Pos")

<table>
<thead>
<tr>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neg</td>
</tr>
<tr>
<td>Pos</td>
</tr>
</tbody>
</table>
**PART TWO: REFERENCE**

**DWIDTH@LEVEL**

*app.Measure.Px.Operator (ParamEngine = "DeltaWidthAtLevel")*

Calculates the Difference (delta) between adjacent width measurements.

<table>
<thead>
<tr>
<th>AbsLevel</th>
<th>Double</th>
</tr>
</thead>
<tbody>
<tr>
<td>FindLevel</td>
<td>Action</td>
</tr>
<tr>
<td>Hysteresis</td>
<td>Double</td>
</tr>
<tr>
<td>LevelType</td>
<td>Enum</td>
</tr>
<tr>
<td>PercentLevel</td>
<td>Double</td>
</tr>
<tr>
<td>Slope</td>
<td>Enum</td>
</tr>
</tbody>
</table>

**AbsLevel**

*Double*

**Range:** From -100 to 100, step 1e-007

**Description**

Sets/Queries the absolute level used if LevelType = “Absolute”

**FindLevel**

*Action*

**Description**

Causes the engine to find a suitable level for either LevelType (“Absolute” or “Percent”)

**Hysteresis**

*Double*

**Range:** From 0 to 10, step 0.1

**Description**

Sets/Queries the hysteresis setting for this measurement. (This is an advanced feature.)

**LevelType**

*Enum*

**Description**

Sets/Queries whether to use absolute level or percent level for the last (ending) transition time.

**Values**

- Absolute
- Percent

**PercentLevel**

*Double*

**Range:** From 0 to 100, step 1

**Description**

Sets/Queries the percentage level to be used if the LevelType = “Percent”.

**Slope**

*Enum*

**Description**

Sets/Queries which polarity transition is used as the start (first) of the width measurement.
Values
  Both
  Neg
  Pos
**EDGE@LEVEL**

*app.Measure.Px.Operator (ParamEngine = "EdgeAtLevel")*

Counts the number of transitions or edges in the input waveform.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsLevel</td>
<td>Double</td>
<td>Sets/Queries the absolute level used to detect edges, when LevelType = &quot;Absolute&quot;.</td>
</tr>
<tr>
<td>FindLevel</td>
<td>Action</td>
<td>Causes the engine to find a suitable level for either LevelType (&quot;Absolute&quot; or &quot;Percent&quot;)</td>
</tr>
<tr>
<td>Hysteresis</td>
<td>Double</td>
<td>Sets/Queries the hysteresis used for edge detection in divisions. (This is an advanced feature.)</td>
</tr>
<tr>
<td>LevelType</td>
<td>Enum</td>
<td>Sets/Queries whether to use absolute or percent levels.</td>
</tr>
<tr>
<td>PercentLevel</td>
<td>Double</td>
<td>Sets/Queries the percent level threshold used for detection of edges when LevelType = “Percent”.</td>
</tr>
</tbody>
</table>

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

app.Measure.P1.ParamEngine = "EdgeAtlevel"
```

AbsLevel

*Double*

Range: From -100 to 100, step 1e-007

Description

Sets/Queries the absolute level used to detect edges, when LevelType = "Absolute".

FindLevel

*Action*

Description

Causes the engine to find a suitable level for either LevelType ("Absolute" or "Percent")

Hysteresis

*Double*

Range: From 0 to 10, step 0.1

Description

Sets/Queries the hysteresis used for edge detection in divisions. (This is an advanced feature.)

LevelType

*Enum*

Description

Sets/Queries whether to use absolute or percent levels.

Values

- Absolute
- Percent

PercentLevel

*Double*

Range: From 0 to 100, step 1

Description

Sets/Queries the percent level threshold used for detection of edges when LevelType = “Percent”. 

---

**PART TWO: REFERENCE**

**3-78**

ISSUED: July 2003

901783
## Slope

**Description**
Sets/Queries the polarity of the edges detected.

**Values**
- Both
- Neg
- Pos

**Enum**
## EXCELPARAM

*app.Measure.Px.Operator (ParamEngine = "ExcelParam")*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddChart</td>
<td>Action</td>
</tr>
<tr>
<td>AddLabels</td>
<td>Action</td>
</tr>
<tr>
<td>Advanced</td>
<td>Bool</td>
</tr>
<tr>
<td>ClearSheet</td>
<td>Action</td>
</tr>
<tr>
<td>CreateDemoSheet</td>
<td>Action</td>
</tr>
<tr>
<td>NewSheet</td>
<td>Bool</td>
</tr>
<tr>
<td>OutputCell</td>
<td>String</td>
</tr>
<tr>
<td>OutputEnable</td>
<td>Bool</td>
</tr>
<tr>
<td>OutputHeaderCell</td>
<td>String</td>
</tr>
<tr>
<td>Source1Cell</td>
<td>String</td>
</tr>
<tr>
<td>Source1Enable</td>
<td>Bool</td>
</tr>
<tr>
<td>Source1HeaderCell</td>
<td>String</td>
</tr>
<tr>
<td>Source2Cell</td>
<td>String</td>
</tr>
<tr>
<td>Source2Enable</td>
<td>Bool</td>
</tr>
<tr>
<td>Source2HeaderCell</td>
<td>String</td>
</tr>
<tr>
<td>SpreadsheetFilename</td>
<td>FileName</td>
</tr>
<tr>
<td>Status</td>
<td>String</td>
</tr>
<tr>
<td>WithHeader</td>
<td>Bool</td>
</tr>
</tbody>
</table>

**AddChart**

*Description*

Adds a chart to the output worksheet.

**AddLabels**

*Description*


**Advanced**

*Description*

Sets/Queries whether advanced features of this component are enabled.

**ClearSheet**

*Description*


**CreateDemoSheet**

*Description*

**NewSheet**

*Description*


**OutputCell**

*String*

*Range:* Any number of characters

*Description*


**OutputEnable**

*Bool*

*Description*


**OutputHeaderCell**

*String*

*Range:* Any number of characters

*Description*


**Source1Cell**

*String*

*Range:* Any number of characters

*Description*


**Source1Enable**

*Bool*

*Description*


**Source1HeaderCell**

*String*

*Range:* Any number of characters

*Description*


**Source2Cell**

*String*

*Range:* Any number of characters

*Description*


**Source2Enable**

*Bool*

*Description*

### Source2HeaderCell

- **Range:** Any number of characters
- **Description:**

### SpreadsheetFilename

- **Range:** Any number of characters
- **Description:**

### Status

- **Range:** Any number of characters
- **Description:**

### WithHeader

- **Description:**
EXT. RATIO

\textit{app.Measure.Px-operator (ParamEngine = "ExtinctionRatio")}

Extinction Ratio assumes it is operating on an eye diagram (persistence result).

<table>
<thead>
<tr>
<th>CalcType</th>
<th>Enum</th>
</tr>
</thead>
<tbody>
<tr>
<td>PctCutWidth</td>
<td>Double</td>
</tr>
</tbody>
</table>

Example

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to ExtinctionRatio
app.Measure.P1.ParamEngine = "ExtinctionRatio"

CalcType

Description

Sets/Queries the calculation type.

Example

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

app.Measure.P1.ParamEngine = "ExtinctionRatio"

' Set the calculation type to linear.
app.Measure.P1.Operator.CalcType = "Linear"

Values

\[ \text{dB}, \text{linear} \]

PctCutWidth

Description

Sets/Queries the percent cut width. This specifies which region of the eye diagram (about the center of the eye) will be used to estimate extinction ratio.

Example

`' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to delta time at level.
app.Measure.P1.ParamEngine = "ExtinctionRatio"

' Set the percent cut width to 15%.
EYE AMPLITUDE

Example

Visual Basic Script

```
Set app = CreateObject("LeCroy.XStreamDSO")

app.Measure.P1.ParamEngine = "EyeAmplitude"
```

PctCutWidth

- **Type:** Double

- **Range:** From 0 to 100, step 0.1

- **Description:**
  Sets/Queries the percent cut width. This specifies which region of the eye diagram (about the center of the eye) will be used to estimate eye amplitude.
EYE BER  

Estimates the BER based on a cut through the (assumed input) Eye Diagram.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.P1.ParamEngine = "EyeBER"

PctCutWidth  

Description
Sets/Queries the percent cut width. This specifies which region of the eye diagram (about the center of the eye) will be used to estimate Bit Error Rate
EYE HEIGHT

Estimates the difference between the high (one level) and low (zero level) of the (assumed input) eye diagram, based on a vertical slice centered on the eye diagram of a specified width, PctCutWidth.

Example

Visual Basic Script

```vbscript
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.P1.ParamEngine = "EyeHeight"
```

**CalcUnits**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using ParamEngine = &quot;EyeHeight&quot;, refer to the corresponding variable for the Extinction Ratio parameter.</td>
</tr>
</tbody>
</table>

**PctCutWidth**

<table>
<thead>
<tr>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>dB</td>
</tr>
<tr>
<td>linear</td>
</tr>
</tbody>
</table>

Description

Sets/Queries the percent cut width. This specifies which region of the eye diagram (about the center of the eye) will be used to estimate eye height.
**FALL@LEVEL**  \( \text{app.Measure.Px.Operator (ParamEngine = "FallAtLevel")} \)

Calculates the fall time at specified arbitrary levels.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HighAbs</strong></td>
<td>Double</td>
</tr>
<tr>
<td><strong>HighPct</strong></td>
<td>Double</td>
</tr>
<tr>
<td><strong>LevelsAre</strong></td>
<td>Enum</td>
</tr>
<tr>
<td><strong>LowAbs</strong></td>
<td>Double</td>
</tr>
<tr>
<td><strong>LowPct</strong></td>
<td>Double</td>
</tr>
</tbody>
</table>

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.P1.ParamEngine = "FallAtLevel"

' Set measurement level type to percent.
' Set the higher percent level to 85 percent.
' Set the higher percent level to 15 percent.

**HighAbs**  \( \text{Double} \)

Range: From -100 to 100, step 1e-007

Description
Sets/Queries the upper (first) transition level, if LevelsAre = “Absolute”.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to fall time at set levels.
app.Measure.P1.ParamEngine = "FallAtLevel"
' Set measurement level type to absolute.
' Set the higher absolute level to 120 mV.

**HighPct**  \( \text{Double} \)

Range: From 11 to 95, step 1

Description
Sets/Queries the upper (first) transition level, if LevelsAre = “Percent”.

Example
LevelsAre  
<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets/Queries whether the measurement levels are absolute or relative to the trace.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute</td>
</tr>
<tr>
<td>Percent</td>
</tr>
</tbody>
</table>

LowAbs  
<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets/Queries the lower (second) transition level, if LevelsAre = “Absolute”.</td>
</tr>
</tbody>
</table>

LowPct  
<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets/Queries the lower (second) transition level, if LevelsAre = “Percent”.</td>
</tr>
</tbody>
</table>
FREQ@LEVEL

**app.Measure.Px.Operator (ParamEngine = "FrequencyAtLevel")**

Calculates the frequency based on a specified transition level.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsLevel</td>
<td>Double</td>
</tr>
<tr>
<td>BaseFrequency</td>
<td>Double</td>
</tr>
<tr>
<td>FindBaseFrequency</td>
<td>Action</td>
</tr>
<tr>
<td>Hysteresis</td>
<td>Double</td>
</tr>
<tr>
<td>LevelType</td>
<td>Enum</td>
</tr>
<tr>
<td>PercentLevel</td>
<td>Double</td>
</tr>
<tr>
<td>SignalType</td>
<td>Enum</td>
</tr>
<tr>
<td>Slope</td>
<td>Enum</td>
</tr>
<tr>
<td>StdBaseFrequency</td>
<td>Enum</td>
</tr>
<tr>
<td>UseBaseFrequency</td>
<td>Enum</td>
</tr>
</tbody>
</table>

**Example**

' Visual Basic Script

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.P1.ParamEngine = "FrequencyAtLevel"
```

**AbsLevel**

*Double*

Range: From -100 to 100, step 1e-007

Description

Sets/Queries the transition level, if LevelType = "Percent".

**BaseFrequency**

*Double*

Range: From 1 to 4e+010, step 1000

Description

Sets/Queries a custom (non-standard) base frequency to be used when SignalType = "Data" and when UseBaseFrequency = "Custom"

**Example**

' Visual Basic Script

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to frequency at level.
app.Measure.P1.ParamEngine = "FrequencyAtLevel"

' Set the instrument to treat the input waveform as a data stream.
app.Measure.P1.Operator.SignalType = "Data"

' Set the instrument to use a non-standard base frequency.
app.Measure.P1.Operator.UseBaseFrequency = "Custom"

' Set the base frequency to 23.79 MHz. (note this is a number not
a string,
'however a string would work just as well)

**FindBaseFrequency**

**Action**

**Description**
Set the base frequency by inspection of the data signal, is SignalType = "Data"

**Example**
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to frequency at level.
app.Measure.P1.ParamEngine = "FrequencyAtLevel"
' Set the instrument to use a non-standard base frequency.
app.Measure.P1.Operator.UseBaseFrequency = "Custom"
' Find the base frequency from the signal.

**FindLevel**

**Action**

**Description**
Causes the engine to find a suitable level for either LevelType ("Absolute" or "Percent").

**Hysteresis**

**Double**

**Range:** From 0 to 10, step 0.1

**Description**
Sets/Queries the hysteresis level. (This is an advanced feature.)

**LevelType**

**Enum**

**Description**
Sets/Queries whether the levels used are specified in "Percent" or "Absolute".

**Values**

Absolute
Percent

**PercentLevel**

**Double**

**Range:** From 0 to 100, step 1

**Description**
Sets/Queries the transition level, if LevelType = "Percent".
SignalType  

**Enum**

Description
Sets/Queries the input signal type. When "Data" is selected, the periods are the same as for an NRZ datastream, with changing states on whole periods of the assumed "clock."

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to frequency at level.
app.Measure.P1.ParamEngine = "FrequencyAtLevel"
' Set input signal type to data
app.Measure.P1.Operator.SignalType = "Data"

Values
Clock
Data

Slope  

**Enum**

Description
Sets/Queries the polarity of the transitions to be used to estimate whole cycles.

Values
Neg
Pos

StdBaseFrequency  

**Enum**

Description
Sets/Queries the selection of standard frequencies (an enumerated list). This setting only has effect if UseBaseFrequency = "Standard".

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to frequency at level.
app.Measure.P1.ParamEngine = "FrequencyAtLevel"
' Set the instrument to use a standard base frequency.
' Set the standard base frequency to 155.52MHz
app.Measure.P1.Operator.StdBaseFrequency = "155.52MHz"
' or alternately set to third choice in (zero based index) list
Values

1.544MHz
139.264MHz
155.52MHz
2.048MHz
2488.32MHz
34.368MHz
44.736MHz
51.84MHz
622.08MHz
8.448MHz

**UseBaseFrequency**

**Enum**

**Description**
Sets/Queries whether to use a custom base frequency or a standard base frequency for frequency at level.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to frequency at level.
app.Measure.P1.ParamEngine = "FrequencyAtLevel"

' Set the instrument to use a standard base frequency.

**Values**

Custom
Standard
FWXX

app.Measure.Px.Operator (ParamEngine = "FullWidthAtXX")

Calculates the full width (of a histogram or distribution) at a specified fraction of the maximum height. (This is a generalized case of FWHM, or full width at half maximum.)

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.P1.ParamEngine = "FullWidthAtXX"
' set the fraction to 25 percent
app.Measure.P1.Operator.HFractionHt = .25

HFractionHt

Double

Range: From 0 to 100, step 1

Description

Sets/Queries the fraction (as a percentage) of the maximum height at which the full width of a histogram will be measured.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Sets the fraction of the height for parameter P2 to 25%.
app.Measure.P2.Operator.HFractionHt = 25
## PART TWO: REFERENCE

### HALF PERIOD

`app.Measure.Px.Operator (ParamEngine = "HalfPeriod")`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsLevel</td>
<td>Double</td>
</tr>
<tr>
<td>FindLevel</td>
<td>Action</td>
</tr>
<tr>
<td>Hysteresis</td>
<td>Double</td>
</tr>
<tr>
<td>LevelType</td>
<td>Enum</td>
</tr>
<tr>
<td>PercentLevel</td>
<td>Double</td>
</tr>
<tr>
<td>Slope</td>
<td>Enum</td>
</tr>
</tbody>
</table>

#### Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.P1.ParamEngine = "HalfPeriod"
```

---

### AbsLevel

- **Parameter**: AbsLevel
- **Type**: Double
- **Range**: From -100 to 100, step 1e-007
- **Description**: Sets/queries the transition level, if LevelType = "Absolute".

### FindLevel

- **Parameter**: FindLevel
- **Type**: Action
- **Description**: Causes the engine to find a suitable level for either LevelType ("Absolute" or "Percent").

### Hysteresis

- **Parameter**: Hysteresis
- **Type**: Double
- **Range**: From 0 to 10, step 0.1
- **Description**: Sets/queries the hysteresis level in divisions (this is an advanced feature).

### LevelType

- **Parameter**: LevelType
- **Type**: Enum
- **Description**: Sets/queries whether the levels used are specified in "Percent" or "Absolute".
- **Values**:
  - Absolute
  - Percent

### PercentLevel

- **Parameter**: PercentLevel
- **Type**: Double
- **Range**: From 0 to 100, step 1
- **Description**: Sets/queries the transition level, if LevelType = "Percent".
### Slope

**Description**

Sets/Queries the polarity of the transitions to be used to estimate whole cycles.

**Values**

- Both
- Neg
- Pos

**Enum**
**HOLD TIME**

*app.MeasurePx.Operator (ParamEngine = "HoldTime")*

Calculates the "Hold" time for a data or clock signal.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClockAbsLevel</td>
<td>Double</td>
</tr>
<tr>
<td>ClockFindLevel</td>
<td>Action</td>
</tr>
<tr>
<td>ClockHysteresis</td>
<td>Double</td>
</tr>
<tr>
<td>ClockLevels</td>
<td>Enum</td>
</tr>
<tr>
<td>ClockPctLevel</td>
<td>Double</td>
</tr>
<tr>
<td>ClockSlope</td>
<td>Enum</td>
</tr>
<tr>
<td>CursorDisplay</td>
<td>Enum</td>
</tr>
<tr>
<td>DataAbsLevel</td>
<td>Double</td>
</tr>
<tr>
<td>DataFindLevel</td>
<td>Action</td>
</tr>
<tr>
<td>DataHysteresis</td>
<td>Double</td>
</tr>
<tr>
<td>DataLevels</td>
<td>Enum</td>
</tr>
<tr>
<td>DataPctLevel</td>
<td>Double</td>
</tr>
<tr>
<td>DataSlope</td>
<td>Enum</td>
</tr>
</tbody>
</table>

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to hold time
app.Measure.P1.ParamEngine = "HoldTime"

' Set clock level type to absolute.

' Set the absolute clock level to 30 mV.
app.Measure.P1.Operator.ClockAbsLevel = 0.03

' or alternatively to 32 millivolts
app.Measure.P1.Operator.ClockAbsLevel = "32mV"
```

**ClockAbsLevel**

*Double*

**Range:** From -100 to 100, step 1e-007

**Description**

Sets/Queries the absolute measuring level for the clock input, if ClockLevels = "Absolute".

**ClockFindLevel**

*Action*

**Description**

Causes the engine to find a suitable level for either ClockLevels setting ("Absolute" or "Percent").

**Example**

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
```
' Set parameter P1 to hold time at set levels.
app.Measure.P1.ParamEngine = "HoldTime"
' Find the measuring level for the clock signal.
app.Measure.P1.Operator.ClockFindLevel

ClockHysteresis Double
Range: from 0 to 10, step 0.1
Description
Sets/Queries the hysteresis range in divisions for the clock input signal.
Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to hold time at set levels.
app.Measure.P1.ParamEngine = "HoldTime"
' Set the clock hysteresis to 0.8 divisions.
app.Measure.P1.Operator.ClockHysteresis = 0.8

ClockLevelIs Enum
Description
Sets/Queries whether the clock level is set as absolute or percent for hold time P1.
Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to hold time at set levels.
app.Measure.P1.ParamEngine = "HoldTime"
' Set clock level type to absolute.

Values
Absolute
Percent
ClockPctLevel  
**Double**

**Range:** From 0 to 100, step 1  
**Description:** Sets/Queries the percent measuring level for the clock input.

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to hold time at set levels.
app.Measure.P1.ParamEngine = "HoldTime"
' Set clock level type to percent.
' Set the percent clock level to 45
```

ClockSlope  
**Enum**

**Description:** Sets/Queries the slope for the measured transitions for the clock input signal.

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to hold time at set levels.
app.Measure.P1.ParamEngine = "HoldTime"
' Set the clock slope to positive.
```

**Values**
- Both
- Neg
- Pos

CursorDisplay  
**Enum**

**Description:** Sets/Queries the type of cursor display for hold time.

**Example**

```vbnet
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to hold time at set levels.
```
app.Measure.P1.ParamEngine = "HoldTime"

' Set the type of cursor display as simple.

Values

- Detailed
- Off
- Simple

**DataAbsLevel**

**Double**

Range: From -100 to 100, step 1e-007

Description

Sets/Queries the absolute measuring level for the data input.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to hold time at set levels.
app.Measure.P1.ParamEngine = "HoldTime"

' Set data level type to absolute.

' Set the absolute data level to 30 mV.
app.Measure.P1.Operator.DataAbsLevel = 0.03

**DataFindLevel**

**Action**

Description

Automatically finds a suitable level for the Data signal, for either DataLevelIs ("Absolute" or "Percent").

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to hold time at set levels.
app.Measure.P1.ParamEngine = "HoldTime"

' Find the measuring level for the data signal.
**DataHysteresis**  
*Double*

**Range:** From 0 to 10, step 0.1  
**Description:**
Sets/Queries the hysteresis range in divisions for the data input signal.  
**Example**
```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to hold time at set levels.
app.Measure.P1.ParamEngine = "HoldTime"
' Set the data hysteresis to 0.8 divisions.
app.Measure.P1.Operator.DataHysteresis = 0.8
```

**DataLevels**  
*Enum*

**Description:**
Sets/Queries whether the data level is set as absolute or percent for hold time Px.  
**Example**
```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to hold time at set levels.
app.Measure.P1.ParamEngine = "HoldTime"
' Set data level type to percent.
```

**Values**
- Absolute
- Percent

**DataPctLevel**  
*Double*

**Range:** From 0 to 100, step 1  
**Description:**
Sets/Queries the percent measuring level for the data input.  
**Example**
```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to hold time at set levels.
app.Measure.P1.ParamEngine = "HoldTime"
' Set data level type to percent.
```
' Set the percent data level to 55

**DataSlope**

*Enum*

**Description**

Sets/Queries the polarity of the transitions measured for the data input signal.

**Example**

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to hold time at set levels.
app.Measure.P1.ParamEngine = "HoldTime"
' Set the data slope to negative.

**Values**

- Both
- Neg
- Pos
**CursorShape**

**Description**
Sets/Queries the cursor shape.

**Example**
```
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to level at X
app.Measure.P1.ParamEngine = "LevelAtX"

' Sets the cursor to difference.
```

**Values**
- Absolute
- Difference
- Reference

---

**HorValue**

**Description**
Sets/Queries the horizontal position at which the level is to be measured.

**Example**
```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to level at X
app.Measure.P1.ParamEngine = "LevelAtX"

' Set the horizontal value to 120 ns.
app.Measure.P1.Operator.HorValue = 0.12e-6
```
LevelCursor

Description
Sets/Queries whether the level cursor is visible.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to level at X
app.Measure.P1.ParamEngine = "LevelAtX"
' Remove the level cursor.

Marker

Description
Sets/Queries whether the marker is visible.

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to level at X
app.Measure.P1.ParamEngine = "LevelAtX"
' Remove the marker.

PinToData

Description
Sets/Queries whether the measurement is taken from the nearest data point (pin to data) or not (based on the interpolated value at X).

Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to level at X
app.Measure.P1.ParamEngine = "LevelAtX"
' Disables pin to data to allow interpolated measurements.
## TimeCursor

<table>
<thead>
<tr>
<th>Description</th>
<th>Bool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets/Queries whether the time cursor is visible.</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

Visual Basic Script

```vbs
Set app = CreateObject("LeCroy.XStreamDSO")

Set parameter P1 to level at X
```

```vbs
app.Measure.P1.ParamEngine = "LevelAtX"

' Remove the time cursor.
```
Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.

Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.

Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.

Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.

Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.

Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.
PART TWO: REFERENCE

Reload
Description
Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.

Source1Enable
Type: Bool
Description
Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.

Source1HeaderVar
Type: String
Range: Any number of characters
Description
Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.

Source1Var
Type: String
Range: Any number of characters
Description
Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.

Source2Enable
Type: Bool
Description
Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.

Source2HeaderVar
Type: String
Range: Any number of characters
Description
Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.

Source2Var
Type: String
Range: Any number of characters
Description
Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.
CHAPTER THREE  Math/Measure Control Reference

Status

String

Range: Any number of characters

Description

Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.

WithHeader

Bool

Description

Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.

WorksheetFilename

FileName

Range: Any number of characters

Description

Using ParamEngine = "MathcadParam", refer to the corresponding variable for the MathcadMath function.
MATLAB PARAM

```
app.Measure.Pr.Operator (ParamEngine = "MATLABParameter")
```

<table>
<thead>
<tr>
<th>MATLABCode</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLABPlot</td>
<td>Bool</td>
</tr>
<tr>
<td>MATLABScalePerDiv</td>
<td>Double</td>
</tr>
<tr>
<td>MATLABZeroOffset</td>
<td>Double</td>
</tr>
</tbody>
</table>

**MATLABCode**

- **Range:** Any number of characters
- **Description:**
  
  Using ParamEngine = "MATLAB param", refer to the corresponding variable for the MATLABMath function.

**MATLABPlot**

- **Description:**
  
  Using ParamEngine = "MATLAB param", refer to the corresponding variable for the MATLABMath function.

**MATLABScalePerDiv**

- **Range:** From 1e-009 to 1e+009, step 1e-009
- **Description:**
  
  Using ParamEngine = "MATLAB param", refer to the corresponding variable for the MATLABMath function.

**MATLABZeroOffset**

- **Range:** From -1e+009 to 1e+009, step 1e-009
- **Description:**
  
  Using ParamEngine = "MATLAB param", refer to the corresponding variable for the MATLABMath function.
MAXIMUM


Calculates the maximum vertical value of the waveform.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.P1.ParamEngine = "Maximum"
MEAN

\texttt{app.Measure.Px.Operator (ParamEngine = "Mean")}

Calculates the mean value of the input waveform's vertical values. When \texttt{Cyclic = true}, the range of values used is limited to a whole number of cycles.

Example

\begin{verbatim}
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to mean.
app.Measure.P1.ParamEngine = "Mean"

' Set the mean parameter for cyclic measurements.
\end{verbatim}

Cyclic

\textbf{Bool}

Description

Sets/Queries whether the mean parameter \texttt{Px} is to be measured over a number of complete cycles.

Example

\begin{verbatim}
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P2 to mean.
app.Measure.P2.ParamEngine = "Mean"

' Set the mean parameter for cyclic measurements.
\end{verbatim}
MEDIAN

app.Measure.Px.Operator (ParamEngine = "Median")

Calculates the median (division between two halves) of the probability distribution of an input waveform. For periodic signals it is advisable to use Cyclic = True.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.P1.ParamEngine = "Median"
' Set the measurement for a periodic signal

Cyclic

Description
Sets/Queries whether the median parameter Px is to be measured over an integral number of complete cycles.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P2 to median.
app.Measure.P2.ParamEngine = "Median"
' Set the median parameter for cyclic measurements.
MINIMUM

`app.Measure.Px.Operator (ParamEngine = "Minimum")`

Calculates the minimum value of a waveform.

Example

`Visual Basic Script`

```vbscript
Set app = CreateObject("LeCroy.XStreamDSO")

```
NB PHASE

Estimates the phase of the frequency component at the specified Frequency.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to narrow band phase.
app.Measure.P1.ParamEngine = "NarrowBandPhase"
' Set the frequency at which the phase is to be measured to 691 MHz.
app.Measure.P1.Operator.Frequency = "691MHz"
' or alternatively to 299MHz

Frequency

Double

Range: From 10 to 1e+010, step 1
Description
Sets/Queries the frequency at which the narrow band phase is to be measured.
NB POWER

\texttt{app.Measure.Px-operator\ (ParamEngine = "NarrowBandPower")}

Measures the power found at a given frequency.

Example

\begin{verbatim}
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to narrow band phase.
app.Measure.P1.ParamEngine = "NarrowBandPower"

' Set the frequency at which the power is to be measured to 235 MHz.

Frequency
\textit{Double}

Range: From 10 to 1e+010, step 1

Description
Sets/Queries the frequency at which the narrow band power is to be measured.
NPTS

Counts the number of (sample or data) points in the input waveform.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.P3.ParamEngine = "npoints"

' Also count the points extending off the grid
app.Measure.P3.Operator.UsePointsInFrame = false

UsePointsInFrame

Description

Sets/Queries whether to use the number of points in the graticule, or the total number of points in the trace.

Note: In many cases there are points off screen in the source waveform; but in particular, in the X-Stream standard architecture, waveforms are often "clipped to frame" already. This setting is most useful when using the WebEdit mode.
ONE LEVEL

Estimates the high level for the (assumed input) eye diagram, based on a slice through the middle of the eye diagram of specified width, PctCutWidth.

Example

```visualBasic
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.P1.ParamEngine = "EyeOneLevel"
```

PctCutWidth

<table>
<thead>
<tr>
<th>Description</th>
<th>Double</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>From 0 to 100, step 0.1</td>
</tr>
</tbody>
</table>

Sets/Queries the width of the vertical slice through the eye, from which the one level is estimated.
PARAM SCRIPT  

app.Measure.Px.Operator (ParamEngine = "ParamScript")

Calculates a parameter from a waveform based on VBScript or JavaScript.

<table>
<thead>
<tr>
<th>Code</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Enum</td>
</tr>
<tr>
<td>Status</td>
<td>String</td>
</tr>
<tr>
<td>Timeout</td>
<td>Double</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Place sample program code in a string
' This will calculate the mean of all of the data points
code =        "Function Update()" + vbCrLf
code = code + "    numSamples = InResult.Samples" + vbCrLf
code = code + "    unscaledData = InResult.DataArray(False)" + vbCrLf
code = code + "    mean = 0" + vbCrLf
code = code + "    For i = 0 to numSamples-1" + vbCrLf
code = code + "        mean = mean + unscaledData(i)" + vbCrLf
code = code + "    next" + vbCrLf
code = code + "    if numSamples > 0 then mean = mean/numSamples" + vbCrLf
code = code + "    OutResult.Value = mean" + vbCrLf
code = code + "end Function"

' Configure P1 as a scripting component
app.Measure.P1.View = True
app.Measure.P1.Source1 = "C1"
app.Measure.P1.MeasurementType = "measure"
app.Measure.P1.ParamEngine = "ParamScript"

Code

Range: Any number of characters
### Language

**Description**
Sets/Queries which scripting language is to be used

**Values**
- JScript
- VBScript

### Status

**Range:** Any number of characters

**Description**
Queries status, as reported by the scripting engine (errors).

### Timeout

**Range:** From 1 to 12000, step 0.001

**Description**
Sets/Queries Timeout for scripting calculation.
PERCENTILE \hspace{1cm} \textit{app.Measure.Px.Operator (ParamEngine = "Percentile")}

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.P1.ParamEngine = "Percentile"

' Set the percentile level to 67%. (i.e. will calculate percentile @ 67%)

\textbf{HPctPop} \hspace{1cm} \textit{Double}

\textbf{Range:} \hspace{1cm} From 0 to 100, step 1

\textbf{Description}

Sets/Queries the percentage of the population that falls to the left of (or below) the desired percentile. For example, the median is the 50\textsuperscript{th} percentile, or the horizontal coordinate of the histogram at which 50% of the population falls to the left.
PERIOD@LEVEL

Calculates the period of the input, using a specified level and slope.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsLevel</td>
<td>Double</td>
</tr>
<tr>
<td>BaseFrequency</td>
<td>Double</td>
</tr>
<tr>
<td>FindBaseFrequency</td>
<td>Action</td>
</tr>
<tr>
<td>FindLevel</td>
<td>Action</td>
</tr>
<tr>
<td>Hysteresis</td>
<td>Double</td>
</tr>
<tr>
<td>LevelType</td>
<td>Enum</td>
</tr>
<tr>
<td>PercentLevel</td>
<td>Double</td>
</tr>
<tr>
<td>SignalType</td>
<td>Enum</td>
</tr>
<tr>
<td>Slope</td>
<td>Enum</td>
</tr>
<tr>
<td>StdBaseFrequency</td>
<td>Enum</td>
</tr>
<tr>
<td>UseBaseFrequency</td>
<td>Enum</td>
</tr>
</tbody>
</table>

Example

Visual Basic Script

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.P1.ParamEngine = "PeriodAtLevel"
```

AbsLevel

**Double**

Range: From -100 to 100, step 1e-007

Description

Using ParamEngine = "PeriodAtLevel", refer to the corresponding variable for the Delta Time At Level parameter.

BaseFrequency

**Double**

Range: From 1 to 4e+010, step 1000

Description

Sets/Queries the frequency to be used as a reference when SignalType = "Data"; otherwise, not used.

FindBaseFrequency

**Action**

Description

Automatically finds (by analyzing the input) the base frequency for a SignalType = "Data".
## Math/Measure Control Reference

### FindLevel

**Action**

Description

Causes the engine to find a suitable level for either LevelType ("Absolute" or "Percent")

### Hysteresis

**Double**

**Range:** From 0 to 10, step 0.1

**Description**

Sets/Queries the Hysteresis level used for edge or transition detection. (This is an advanced feature.)

### LevelType

**Enum**

**Description**

Sets/Queries whether the levels of the signal amplitude are "Percent" or "Absolute".

**Values**

- Absolute
- Percent

### PercentLevel

**Double**

**Range:** From 0 to 100, step 1

**Description**

Sets/Queries the percent level to be used when LevelType = "Percent".

### SignalType

**Enum**

**Description**

Using ParamEngine = "PeriodAtLevel", refer to the corresponding variable for the Frequency At Level parameter.

**Values**

- Clock
- Data
Slope  

Description  

Using ParamEngine = "PeriodAtLevel", refer to the corresponding variable for the Delta Time At Level parameter.

Values  

Neg  

Pos  

StdBaseFrequency  

Description  

Using ParamEngine = "PeriodAtLevel", refer to the corresponding variable for the Frequency At Level parameter.

Values  

1.544MHz  

139.264MHz  

155.52MHz  

2.048MHz  

2488.32MHz  

34.368MHz  

44.736MHz  

51.84MHz  

622.08MHz  

8.448MHz  

UseBaseFrequency  

Description  

Sets/Queries which frequency to use, a "Standard" frequency or "Custom" frequency, as specified by BaseFrequency. This control only has effect when the SignalType = “Data”.

Values  

Custom  

Standard
### PHASE

**app.Measure.Px.Operator (ParamEngine = "Phase")**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>OutputType</td>
<td>Enum</td>
</tr>
<tr>
<td>RefAbsLevel</td>
<td>Double</td>
</tr>
<tr>
<td>RefFindLevel</td>
<td>Action</td>
</tr>
<tr>
<td>RefHysteresis</td>
<td>Double</td>
</tr>
<tr>
<td>RefLevelType</td>
<td>Enum</td>
</tr>
<tr>
<td>RefPercentLevel</td>
<td>Double</td>
</tr>
<tr>
<td>RefSlope</td>
<td>Enum</td>
</tr>
<tr>
<td>SigAbsLevel</td>
<td>Double</td>
</tr>
<tr>
<td>SigFindLevel</td>
<td>Action</td>
</tr>
<tr>
<td>SigHysteresis</td>
<td>Double</td>
</tr>
<tr>
<td>SigLevelType</td>
<td>Enum</td>
</tr>
<tr>
<td>SigPercentLevel</td>
<td>Double</td>
</tr>
<tr>
<td>SigSlope</td>
<td>Enum</td>
</tr>
</tbody>
</table>

#### OutputType

**Enum**

Description

Sets/Queries the output type for Phase Px.

Example

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to phase difference.
app.Measure.P1.ParamEngine = "Phase"

' Set the output unit as radians.
app.Measure.P1.Operator.OutputType = "Radians"

Values

- Degrees
- Percent
- Radians

#### RefAbsLevel

**Double**

Range: From -100 to 100, step 1e-007

Description

Sets/Queries the test level for the reference trace in absolute units.

Example

' Visual Basic Script

Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to phase difference.
app.Measure.P1.ParamEngine = "Phase"
PART TWO: REFERENCE

' Set the reference trace test level in absolute units as 10 mV.
app.Measure.P1.Operator.RefAbsLevel = 0.01

RefFindLevel
Action
Description
Finds the test level for the reference trace.
Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 to phase difference.
' Find the test level for the reference trace.

RefHysteresis
Double
Range: From 0 to 10, step 0.1
Description
Sets/Queries the hysteresis range for the reference trace.
Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to phase difference.
app.Measure.P1.ParamEngine = "Phase"
' Set the reference hysteresis in graticule divisions.
app.Measure.P1.Operator.RefHysteresis = 0.7

RefLevelType
Enum
Description
Sets/Queries the unit of measurement for the test level of the reference trace.
Example
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to phase difference.
app.Measure.P1.ParamEngine = "Phase"
' Set the reference level to be measured in absolute units.
app.Measure.P1.Operator.RefLevelType = "Absolute"
Values

Absolute
Percent

RefPercentLevel

Description
Sets/Queries the test level for the reference trace in percent.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 to phase difference.
' Set the reference test level in percent.

RefSlope

Description
Sets/Queries the polarity of the measured reference transitions.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P1 to phase difference.
app.Measure.P1.ParamEngine = "Phase"
' Set the reference slope to negative.

Values
Both
Neg
Pos

SigAbsLevel

Description
Sets/Queries the test level for the signal in absolute units.

SigFindLevel

Description
Causes the engine to find a suitable level for either SigLevelType ("Absolute" or "Percent").
SigHysteresis

Description
Sets/Queries the hysteresis range for the signal.

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P3 to phase difference.
' Set the signal hysteresis in graticule divisions.
app.Measure.P3.Operator.SigHysteresis = 0.7

SigLevelType

Description
Sets/Queries which level to use "Percent" or "Absolute" for transitions on the signal.

Values
Absolute
Percent

SigPercentLevel

Description
Sets/Queries the test level for the signal in percent.

SigSlope

Description
Sets/Queries the polarity of the measured signal transitions.

Values
Both
Neg
Pos
RISE@LEVEL

Calculates the rise time (transition time on a rising edge) using two specified levels.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HighAbs</td>
<td>Double</td>
</tr>
<tr>
<td>HighPct</td>
<td>Double</td>
</tr>
<tr>
<td>LevelsAre</td>
<td>Enum</td>
</tr>
<tr>
<td>LowAbs</td>
<td>Double</td>
</tr>
<tr>
<td>LowPct</td>
<td>Double</td>
</tr>
</tbody>
</table>

Example

Visual Basic Script

```
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
app.Measure.P1.ParamEngine = "RiseAtLevel"
```

**HighAbs**

- **Double**
- **Range:** From -100 to 100, step 1e-007
- **Description**
  - Sets/Queries the higher (second) level crossing used, if LevelsAre = "Absolute".

**HighPct**

- **Double**
- **Range:** From 11 to 95, step 1
- **Description**
  - Sets/Queries the higher (second) level crossing used, if LevelsAre = "Percent".

**LevelsAre**

- **Enum**
- **Description**
  - Sets/Queries whether the levels used are "Absolute" or "Percent".
- **Values**
  - Absolute
  - Percent

**LowAbs**

- **Double**
- **Range:** From -100 to 100, step 1e-007
- **Description**
  - Sets/Queries the lower (first) level crossing used, if LevelsAre = "Absolute".
### Part Two: Reference

<table>
<thead>
<tr>
<th>LowPct</th>
<th>Double</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range:</td>
<td>From 5 to 89, step 1</td>
</tr>
<tr>
<td>Description</td>
<td>Sets/Queries the lower (first) level crossing used, if LevelsAre = “Percent”.</td>
</tr>
</tbody>
</table>
CHAPTER THREE

Math/Measure Control Reference

RMS

\texttt{app.Measure.Px.Operator (ParamEngine = "RootMeanSquare")}

Calculates the root mean square of the values of the input waveform.

Example

\begin{verbatim}
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
app.Measure.P1.ParamEngine = "RMS"
\end{verbatim}

Cyclic

\textbf{Bool}

\textbf{Description}

Sets/Queries whether the RMS is measured over an integral number of cycles (suggested for periodic signals).

Example

\begin{verbatim}
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")

' Set parameter P2 to RMS.
app.Measure.P2.ParamEngine = "RMS"

' Set the RMS parameter for cyclic measurements.
\end{verbatim}
SETUP

*app.Measure.Px.Operator (ParamEngine = "Setup")*

Calculates the setup time associated with a pair of input waveforms for Clock and Data.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClockAbsLevel</td>
<td>Double</td>
</tr>
<tr>
<td>ClockFindLevel</td>
<td>Action</td>
</tr>
<tr>
<td>ClockHysteresis</td>
<td>Double</td>
</tr>
<tr>
<td>ClockLevels</td>
<td>Enum</td>
</tr>
<tr>
<td>ClockPctLevel</td>
<td>Double</td>
</tr>
<tr>
<td>ClockSlope</td>
<td>Enum</td>
</tr>
<tr>
<td>CursorDisplay</td>
<td>Enum</td>
</tr>
<tr>
<td>DataAbsLevel</td>
<td>Double</td>
</tr>
<tr>
<td>DataFindLevel</td>
<td>Action</td>
</tr>
<tr>
<td>DataHysteresis</td>
<td>Double</td>
</tr>
<tr>
<td>DataLevels</td>
<td>Enum</td>
</tr>
<tr>
<td>DataPctLevel</td>
<td>Double</td>
</tr>
<tr>
<td>DataSlope</td>
<td>Enum</td>
</tr>
<tr>
<td>Summary</td>
<td>String</td>
</tr>
</tbody>
</table>

Example

```Visual Basic Script```

```vbnet
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
app.Measure.P1.ParamEngine = "Setup"
```

**ClockAbsLevel**

*Double*

**Range:** From -100 to 100, step 1e-007

**Description**

Sets/Queries the level used if ClockLevels = “Absolute”.

**ClockFindLevel**

*Action*

**Description**

Causes the engine to find a suitable level for either ClockLevels ("Absolute" or "Percent").

**ClockHysteresis**

*Double*

**Range:** From 0 to 10, step 0.1

**Description**

Using ParamEngine = "Setup", refer to the corresponding variable for the Hold Time parameter.
ClockLevels  
Enum  
Description  
Sets/Queries whether the Clock signal levels are specified in "Percent" or "Absolute".

Values  
Absolute  
Percent  

ClockPctLevel  
Double  
Range: From 0 to 100 step 1  
Description  
Using ParamEngine = "Setup", please refer to the corresponding variable for the Hold Time parameter.

ClockSlope  
Enum  
Description  
Sets/Queries the polarity of transitions of the Clock signal.

Values  
Both  
Neg  
Pos  

CursorDisplay  
Enum  
Description  
Using ParamEngine = "Setup", refer to the corresponding variable for the Hold Time parameter.

Values  
Detailed  
Off  
Simple  

DataAbsLevel  
Double  
Range: From -100 to 100, step 1e-007  
Description  
Sets/Queries the level used if the ClockLevels = "Percent"
**DataFindLevel**

*Action*

Description

Causes the engine to find a suitable level for either DataLevellns ("Absolute" or "Percent")

**DataHysteresis**

*Double*

Range: From 0 to 10, step 0.1

Description

Using ParamEngine = "Setup", refer to the corresponding variable for the Hold Time parameter.

**DataLevellns**

*Enum*

Description

Sets/Queries whether the Data signal level is DataAbsLevel or DataPctLevel.

Values

Absolute
Percent

**DataPctLevel**

*Double*

Range: From 0 to 100, step 1

Description

Using ParamEngine = "Setup", refer to the corresponding variable for the Hold Time parameter.

**DataSlope**

*Enum*

Description

Sets/Queries the polarity of transitions to be used for the Data signal.

Values

Both
Neg
Pos

**Summary**

*String*

Range: Any number of characters

Description

Using ParamEngine = "Setup", refer to the corresponding variable for the Hold Time parameter.
SKEW

Calculates the skew between two clock signal waveforms.

```
app.Measure.Px.Operator (ParamEngine = "Skew")
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock1AbsLevel</td>
<td>Double</td>
<td>Range: From -100 to 100, step 1e-007</td>
</tr>
<tr>
<td>Clock1FindLevel</td>
<td>Action</td>
<td>Description: Automatically find a suitable level for Clock1, for either &quot;Percent&quot; or &quot;Absolute&quot; levels.</td>
</tr>
<tr>
<td>Clock1Hysteresis</td>
<td>Double</td>
<td>Range: From 0 to 10, step 0.1 Description: Sets/Queries hysteresis for transition detection used for Clock1.</td>
</tr>
<tr>
<td>Clock1Levels</td>
<td>Enum</td>
<td>Description: Sets/Queries whether to use Percent or Absolute levels for Clock1. Values: Absolute, Percent</td>
</tr>
<tr>
<td>Clock2AbsLevel</td>
<td>Double</td>
<td></td>
</tr>
<tr>
<td>Clock2FindLevel</td>
<td>Action</td>
<td></td>
</tr>
<tr>
<td>Clock2Hysteresis</td>
<td>Double</td>
<td></td>
</tr>
<tr>
<td>Clock2Levels</td>
<td>Enum</td>
<td></td>
</tr>
<tr>
<td>Clock2PctLevel</td>
<td>Double</td>
<td></td>
</tr>
<tr>
<td>Clock2Slope</td>
<td>Enum</td>
<td></td>
</tr>
</tbody>
</table>

Example

```
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
app.Measure.P1.ParamEngine = "Skew"
```
PART TWO: REFERENCE

Clock1PctLevel

*Double*

Range: From 0 to 100, step 1
Description
Sets/Queries the "Percent" of the amplitude of Clock1 to use for a transition level, if Clock1Levels = "Percent".

Clock1Slope

*Enum*

Description
Sets/Queries the polarity of transitions detected on Clock1.
Values
Both
Neg
Pos

Clock2AbsLevel

*Double*

Range: From -100 to 100, step 1e-007
Description
Using ParamEngine = "Skew", refer to the corresponding variable for the Hold Time parameter.

Clock2FindLevel

*Action*

Description
Using ParamEngine = "Skew", refer to the corresponding variable for the Hold Time parameter.

Clock2Hysteresis

*Double*

Range: From 0 to 10, step 0.1
Description
Sets/Queries hysteresis for transition detection used for Clock1.

Clock2Levels

*Enum*

Description
Sets/Queries whether to use Percent or Absolute levels for Clock2.
Values
Absolute
Percent

Clock2PctLevel

*Double*

Range: From 0 to 100, step 1
Description
Sets/Queries the "Percent" of the amplitude of Clock2 to use for a transition level, if Clock2Levels = "Percent".
Clock2Slope

Description
Sets/Queries the polarity of transitions detected on Clock2.

Values
- Both
- Neg
- Pos
STD DEV \hspace{1cm} \text{app.Measure.Px.Operator (ParamEngine = "StandardDeviation")}

Calculates the standard deviation of the input waveform’s vertical values.

Example

\begin{verbatim}
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
' Set parameter P1 to standard deviation.
' Set the RMS parameter for cyclic measurements.
\end{verbatim}

Cyclic \hspace{1cm} \text{Bool}

Description

Sets/Queries whether the standard deviation is to be measured over an integral number of complete cycles.
CHAPTER THREE

Math/Measure Control Reference

TIE@LEVEL

app.Measure.Px.Operator (ParamEngine = "TIE")

Calculates the Time Interval Error: the errors in observed transition times relative to a series of expected times (virtual clock).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsLevel</td>
<td>Sets/Queries the absolute transition level to be used when LevelType = &quot;Absolute&quot;.</td>
</tr>
<tr>
<td>BaseFrequency</td>
<td>Using ParamEngine = &quot;TIE&quot;, refer to the corresponding variable for the Delta Period At Level parameter.</td>
</tr>
<tr>
<td>CutOffDivisor</td>
<td>Sets/Queries the cut-off divisor for the &quot;Golden PLL&quot;.</td>
</tr>
<tr>
<td>DatasNRZ</td>
<td></td>
</tr>
<tr>
<td>FindBaseFrequency</td>
<td></td>
</tr>
<tr>
<td>FindLevel</td>
<td></td>
</tr>
<tr>
<td>Hysteresis</td>
<td></td>
</tr>
<tr>
<td>IncludeVirtualEdges</td>
<td></td>
</tr>
<tr>
<td>LevelType</td>
<td></td>
</tr>
<tr>
<td>PercentLevel</td>
<td></td>
</tr>
<tr>
<td>ResultScaling</td>
<td></td>
</tr>
<tr>
<td>SignalType</td>
<td></td>
</tr>
<tr>
<td>Slope</td>
<td></td>
</tr>
<tr>
<td>StdBaseFrequency</td>
<td></td>
</tr>
<tr>
<td>UseBaseFrequency</td>
<td></td>
</tr>
<tr>
<td>UseGoldenPLL</td>
<td></td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
app.Measure.P1.ParamEngine = "TIE"

AbsLevel

Range: From -100 to 100, step 1e-007

BaseFrequency

Range: From 1 to 4e+010, step 1000

CutOffDivisor

Range: From 20 to 10000, step 1
**PART TWO: REFERENCE**

---

**DataIsNRZ**

*Type:* Bool  

*Description:* Sets/Queries whether the signal is of NRZ type.

---

**FindBaseFrequency**

*Type:* Action  

*Description:* Automatically determines the Base Frequency for the expected virtual clock.

---

**FindLevel**

*Type:* Action  

*Description:* Using ParamEngine = "TIE", refer to the corresponding variable for the Delta Period At Level parameter.

---

**Hysteresis**

*Type:* Double  

*Range:* From 0 to 10, step 0.1  

*Description:* Sets/Queries hysteresis for transition detection used for the input waveform.

---

**IncludeVirtualEdges**

*Type:* Bool  

*Description:* Sets/Queries whether to include virtual edges in the series of interval errors reported by this measurement.

---

**LevelType**

*Type:* Enum  

*Description:* Sets/Queries whether to use Percent or Absolute levels.  

*Values:*  

- Absolute  
- Percent

---

**PercentLevel**

*Type:* Double  

*Range:* From 0 to 100, step 1  

*Description:* Sets/Queries the "Percent" of the amplitude of the input waveform for a transition detection, if LevelType = "Percent".
CHAPTER THREE  

Math/Measure Control Reference

ResultScaling  
Enum  
Description  
Sets/Queries whether the measurement is to be displayed in seconds (Time) or in unit intervals (UI).  
Values  
Time  
UI

SignalType  
Enum  
Description  
Sets/Queries whether the input signal is a Clock or Data.  
Values  
Clock  
Data

Slope  
Enum  
Description  
Sets/Queries the polarity of transitions detected.  
Values  
Both  
Neg  
Pos

StdBaseFrequency  
Enum  
Description  
Sets/Queries a standard base frequency from a list of frequencies. This value is only used if the UseBaseFrequency = "Standard".  
Values  
1.544MHz  
139.264MHz  
155.52MHz  
2.048MHz  
2488.32MHz  
34.368MHz  
44.736MHz  
51.84MHz  
622.08MHz  
8.448MHz
## PART TWO: REFERENCE

<table>
<thead>
<tr>
<th><strong>UseBaseFrequency</strong></th>
<th><strong>Enum</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Sets/Queries whether to use the custom base frequency or one chosen from a list of standard frequencies.</td>
</tr>
<tr>
<td><strong>Values</strong></td>
<td>Custom</td>
</tr>
<tr>
<td></td>
<td>Standard</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>UseGoldenPLL</strong></th>
<th><strong>Bool</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Sets/Queries whether to use a golden PLL for the expected times.</td>
</tr>
</tbody>
</table>
**TIME@LEVEL**

```
app.Measure.Px.Operator (ParamEngine = "TimeAtLevel")
```

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsLevel</td>
<td>Double</td>
</tr>
<tr>
<td>FindLevel</td>
<td>Action</td>
</tr>
<tr>
<td>Hysteresis</td>
<td>Double</td>
</tr>
<tr>
<td>HysteresisPct</td>
<td>Double</td>
</tr>
<tr>
<td>HysteresisType</td>
<td>Enum</td>
</tr>
<tr>
<td>LevelType</td>
<td>Enum</td>
</tr>
<tr>
<td>PercentLevel</td>
<td>Double</td>
</tr>
<tr>
<td>Slope</td>
<td>Enum</td>
</tr>
</tbody>
</table>

**AbsLevel**

- **Double**
- **Range:** From -100 to 100, step 1e-007
- **Description:** Sets/Queries the level for transitions detected on the input waveform if LevelType = "Absolute".

**FindLevel**

- **Action**
- **Description:** Using ParamEngine = "TimeAtLevel", please refer to the corresponding variable for the Delta Time At Level parameter.

**Hysteresis**

- **Double**
- **Range:** From 0 to 10, step 0.1
- **Description:** Sets/Queries hysteresis for transition detection used.

**HysteresisPct**

- **Double**
- **Range:** From 0 to 100, step 0.1
- **Description:** Sets/Queries the hysteresis as a percentage of signal amplitude, if HysteresisType = "Percent".

**HysteresisType**

- **Enum**
- **Description:** Sets/Queries whether hysteresis is specified in divisions, or if it is a percentage of the input waveform amplitude.
- **Values:**
  - Divisions
  - Percent
**LevelType**

*Enum*

**Description**

Using ParamEngine = "TimeAtLevel", refer to the corresponding variable for the Delta Time At Level parameter.

**Values**

- Absolute
- Percent

**PercentLevel**

*Double*

**Range:** From 0 to 100, step 1

**Description**

Sets/Queries the "Percent" of the amplitude of the input waveform is used for detecting transitions if LevelType = "Percent".

**Slope**

*Enum*

**Description**

Sets/Queries the polarity of transitions detected.

**Values**

- Both
- Neg
- Pos
WIDTH@LEVEL  \texttt{app.Measure.Px.Operator (ParamEngine = "WidthAtLevel")}

Calculates the width of pulses in the input waveform at specified levels.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsLevel</td>
<td>Double</td>
</tr>
<tr>
<td>FindLevel</td>
<td>Action</td>
</tr>
<tr>
<td>FirstWhenBoth</td>
<td>Enum</td>
</tr>
<tr>
<td>Hysteresis</td>
<td>Double</td>
</tr>
<tr>
<td>HysteresisPct</td>
<td>Double</td>
</tr>
<tr>
<td>HysteresisType</td>
<td>Enum</td>
</tr>
<tr>
<td>LevelType</td>
<td>Enum</td>
</tr>
<tr>
<td>PercentLevel</td>
<td>Double</td>
</tr>
<tr>
<td>Slope</td>
<td>Enum</td>
</tr>
</tbody>
</table>

Example

' Visual Basic Script

\begin{verbatim}
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
' Set parameter P1 to width at level.
app.Measure.P1.ParamEngine = "WidthAtLevel"
' Set first used polarity to negative.
app.Measure.P1-operator.Slope = "Both"
app.Measure.P1-operator.FirstWhenBoth = "Neg"
\end{verbatim}

AbsLevel  \texttt{Double}

Range: From -100 to 100 step 1e-007

Description

Sets/Queries the level for transitions detected on the input waveform, if LevelType = "Absolute".

FindLevel  \texttt{Action}

Description

Automatically finds a suitable level for the width measurement, for either case of LevelType.

FirstWhenBoth  \texttt{Enum}

Description

Sets/Queries the polarity of the first accepted transition when both polarities are used.

Values

- Neg
- Pos
**Part Two: Reference**

**Hysteresis**

*Description*

Sets/Queries hysteresis for transition detection if HysteresisType = "Absolute".

*Range: From 0 to 10, step 0.1*

**HysteresisPct**

*Description*

Sets/Queries hysteresis in percent for transition detection if HysteresisType = "Percent".

*Range: From 0 to 100, step 0.1*

**HysteresisType**

*Enum*

Sets/Queries whether to use Percent or Divisions for hysteresis.

*Values*

- Divisions
- Percent

**LevelType**

*Enum*

Sets/Queries whether to use Percent or Absolute levels.

*Values*

- Absolute
- Percent

**PercentLevel**

*Double*

Sets/Queries the "Percent" of the amplitude of the input waveform, if LevelType = "Percent".

*Range: From 0 to 100, step 1*

**Slope**

*Enum*

Sets/Queries the polarity of transitions detected.

*Values*

- Both
- Neg
- Pos
X AT PEAK  

\textit{app.Measure.Px.Operator (ParamEngine = “XAtPeak”)}

Estimates the horizontal position of the \( N \)th most significant peak in a distribution

Example

\begin{verbatim}
' Visual Basic Script
Set app = CreateObject("LeCroy.XStreamDSO")
app.Measure.MeasureMode = "MyMeasure"
app.Measure.P1.ParamEngine = "XAtPeak"
' Set the parameter to measure location of the 4th peak.
app.Measure.P1.Operator.PeakNumber = 4
\end{verbatim}

\textbf{PeakNumber} \hspace{2cm} \textit{Integer}

Range: From 1 to 10000, step 1

Description

Sets/Queries the horizontal position of the \( N \)th highest peak of a histogram.
ZERO LEVEL  \texttt{app.Measure.Px.Operator (ParamEngine = "EyeZeroLevel")}

Estimates the low level for the (assumed input) eye diagram, based on a slice through the middle of the eye diagram of specified width, PctCutWidth.

\begin{verbatim}
Example
    ' Visual Basic Script
    Set app = CreateObject("LeCroy.XStreamDSO")
    app.Measure.P1.ParamEngine = "EyeZeroLevel"
\end{verbatim}

\textbf{PctCutWidth} \hspace{1cm} \textit{Double}

\begin{itemize}
    \item \textbf{Range:} From 0 to 100, step 0.1
    \item \textbf{Description:} Sets/Queries the width of the vertical slice through the eye, from which the zero level is estimated.
\end{itemize}
Glossary

ACTIVEX
Microsoft's brand name for the technologies that enable interoperability using the Component Object Model (COM). ActiveX technology includes, but is not limited to, OLE.

AUTOMATION
COM-based technology that enables binding at run time, or late binding, to an object's methods and properties and also makes possible cross-application macro programming. Formerly referred to as OLE Automation.

AUTOMATION CLIENT
An application, programming tool, or scripting language that accesses services provided by Automation objects. Formerly referred to as Automation controller.

AUTOMATION OBJECT
An instance of a class defined within an application that is exposed for access by other applications or programming tools by Automation interfaces.

AUTOMATION SERVER
An application, type library, or other source that makes Automation objects available for programming by other applications, programming tools, or scripting languages.

COM (COMPONENT OBJECT MODEL)
The programming model and binary standard on which OLE is based. COM defines how objects and their clients interact within processes or across process boundaries.

DCOM (DISTRIBUTED COMPONENT OBJECT MODEL)
Distributed form of COM, enables communication between computers using the COM standard.

DISPATCH INTERFACES (DISPINTERFACE)
An IDispatch interface that responds only to a certain fixed set of names. The properties and methods of the dispinterface are not in the virtual function table (VTBL) for the object.

EARLY BINDING
Also known as static binding (converting symbolic addresses in the program to storage-related addresses) this is binding that occurs during program compilation or linkage.

LATE BINDING
The ability to bind member names to dispatch identifiers (IDs) at run time, rather than at compile time.