

CHAPTER SIX: *Linking with Automation*

In this chapter, discover

- *What Automation is*
- *How to use the VBS command*

What is Automation ?

OVERVIEW

Automation is fundamental to the instrument. Automation enables you to control programs from your own applications as if you were using a keyboard and a mouse. For example, if you want to use Excel, Mathcad, MATLAB, or other proprietary programs, you can do so within the chain of operations of the oscilloscope, without having to go outside the WaveMaster software to instantiate the proprietary software. Here we offer a simple introduction to help those who have existing software for LeCroy oscilloscopes.

SOME DETAILS

Here is the start of a typical instrument setup file. You don't necessarily have to know about this to use the instrument, but any setup file is a convenient source of examples to illustrate the workings of Automation in the instruments.

```
' WaveMaster ConfigurationVBScript ...

On Error Resume Next
set WaveMaster = CreateObject("LeCroy.WaveMasterApplication")

' AladdinPersona ...
WaveMaster.HideClock = False
WaveMaster.TouchScreenEnable = True

Set Display = WaveMaster.Display
' Display ...
Display.GridMode = "Dual"           ' The scope variables are
Display.TraceStyle = "Line"         ' in groups, such as Display,
Display.GridIntensity = 70          ' Acquisition, Horizontal,
Display.GridOnTop = False           ' and Trigger.
Display.AxisLabels = False
Display.NumSegmentsDisplayed = 1     ' The dot notation is well
Display.StartSegment = 1            ' known in object-oriented
Display.SegmentMode = "Adjacent"    ' programming. It makes
Display.ScreenSaveEnable = True     ' hierarchies easy to create.
Display.ScreenSaveTimeout = 60      ' Properties can be handed down
Display.LockPersistence = "AllLocked" ' through a hierarchy.
Display.PersistenceStyle = "ColorGraded"
Display.Persisted = False
```

PART ONE: ABOUT REMOTE CONTROL

```
Display.PersistenceMonoChrome = True ' To save a lot of typing, you
Display.Persistence3d = False         ' can write statements like
Display.Persist3DQuality = "Solid"    ' Set Display = WaveMaster.Display
Display.AxisXRotation = 45            ' like a kind of shorthand.
Display.AxisYRotation = 20
Display.PersistenceSaturation = 50     ' Automation remote control
Display.PersistenceTime = "Infinite"   ' commands are merely copies of
Display.PersistenceLastTrace = False   ' statements like these.

Set Acquisition = WaveMaster.Acquisition
' Acquisition ...                      ' Automation queries are merely
Acquisition.TriggerMode = "Stopped"    ' the same things reversed.
                                       ' To find out the state of the
Set C1 = Acquisition.C1                ' grid you can write
' C1 ...                              ' GM = Display.GridMode
C1.View = True                        ' Where you have already defined
C1.UseGrid = "YT1"                    ' GM as a string variable.
C1.UseDotJoin = True
C1.Persisted = False
C1.PersistenceSaturation = 50
C1.PersistenceMonoChrome = True
```

The complete setup file shows that the variables are grouped as follows.

- Acquisition
- Cursors
- Display
- Math
- Measure
- Memory
- PassFail
- SaveRecall
- Utility

The X-Stream DSO differs from earlier scopes in that this same language and structure can be employed by the scope user to send commands from external computers; to send commands from programs inside the instrument, using ActiveDSO; to write scripts; and to run many other applications.

The setup file is a means of setting the scope very quickly into a given state. Individual commands can be sent as required, to make changes during the running of an application. Because the setup files are in ASCII text

format, they can be edited very simply, using a program such as Notepad. This makes your system very much less vulnerable to changes in the scope system. It also enables people to share information very easily.

HOW TO USE THE VBS COMMAND

The key to using automation commands in an existing GPIB program is the VBS command. This is described in detail in Part Two of this manual. Please note that “app” refers to the instrument application program. It can be defined by a statement like this:

```
Set WaveMaster = CreateObject("LeCroy.WaveMasterApplication")
```

Here are some examples of VBS, with the older GPIB equivalents. The command syntax is
VBS <automation command>.

```
CMD$="VBS `app.Acquisition.C1.VerScale=0.05`  
      CMD$ "C1:VDIV 50 MV" (Earlier equivalent)
```

```
CMD$="VBS `app.Horizontal.HorScale = 500e-9`  
      CMD$ "TDIV 0.5e-6" (Earlier equivalent)
```

```
CMD$="VBS `app.Display.GridMode="Dual"`  
      CMD$ "GRID DUAL" (Earlier equivalent)
```

Note that where the command includes double quotation marks, you may have to split up the string and use a code to replace the quotation marks. In QuickBasic you would use Chr\$(34). In Visual Basic you would use Chr(34).

Here is an example of setting up the Amplitude parameter:

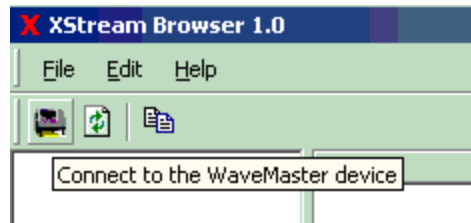
```
VBS "Measure.Pl.View = True"  
VBS "Measure.Pl.ParamEngine = "AMPL""  
VBS "Measure.Pl.Source1 = "C1""  
VBS "Measure.Pl.GateStart = 1.667"  
VBS "Measure.Pl.GateStop = 3.333"
```

HOW TO USE X-STREAM BROWSER

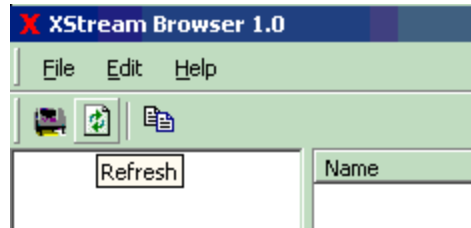
The number of different variables and methods in a complete setup is obviously large. To facilitate the job of creating control statements, LeCroy has produced the program XStreamBrowser. Using this program, you can quickly find the information that corresponds to any part of the instrument.

On entering XstreamBrowser, you will see three icons at the top left of the screen.

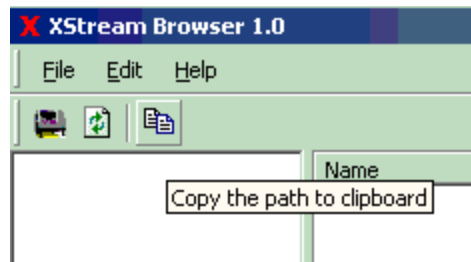
The first one makes the connection between XstreamBrowser and the instrument:



The second icon refreshes the connection. If you make changes to the instrument setup, you must click on this icon if you want XStreamBrowser to capture the changes:

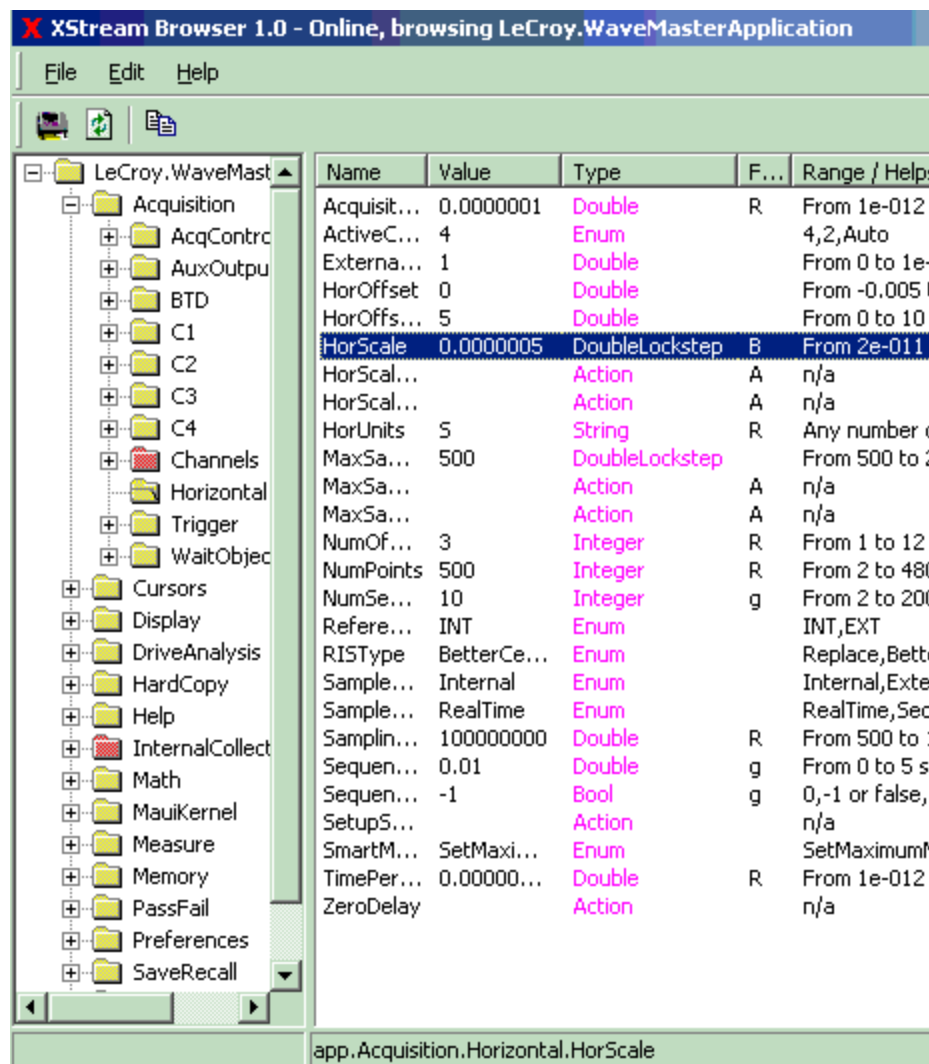


The third icon places the current statement in the clipboard for pasting into your script or program:



PART ONE: ABOUT REMOTE CONTROL

Here is an example of the selection of a line to go into the clipboard:



The statement at the bottom of the screen is the one that will be placed in the clipboard when the icon is clicked. The column labeled **F** contains Flags and Status values. For example, **R** means read and **W** means write. **HorScale** is equivalent to the older-style command **TDIV**. The Range/Help column provides short form information about the possible values that the variable can take.

You can, for example, write `app.Acquisition.Horizontal.HorScale = 0.001` to set the timebase.

You can also, for example, write `HS = app.Acquisition.Horizontal.HorScale` to read the timebase setting.

Typical variable types are as follows:

Single Single precision floating point number

Double Double precision floating point number

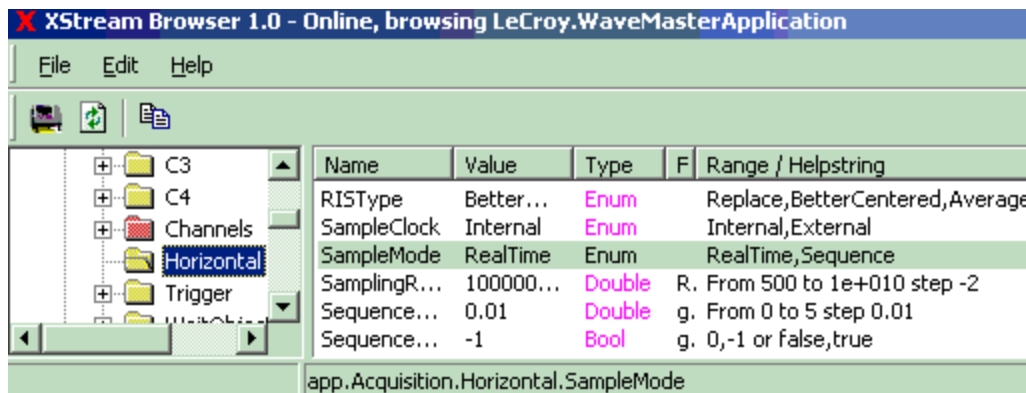
Integer Integer

Long Long integer

Enum Member of list

String String

In the case of enum variables, you may specify the value using the actual values (for example, "INT" or "EXT") for Reference source in the example already given. You may also give the position in the list for the value you want to use (for example, 0 or 1 in the given example). All lists begin with the zeroth element. In the next figure we see that the variable for SampleMode is an enum type, and that the scope is currently running in real time mode.



The following inputs are allowed.

Real Time Mode	<code>app.Acquisition.Horizontal.SampleMode = 0</code>
Real Time Mode	<code>app.Acquisition.Horizontal.SampleMode = "RealTime"</code>
Sequence Mode	<code>app.Acquisition.Horizontal.SampleMode = 1</code>
Sequence Mode	<code>app.Acquisition.Horizontal.SampleMode = "Sequence"</code>

PART ONE: ABOUT REMOTE CONTROL

Were you to click on the Clipboard icon at this point the line `app.Acquisition.Horizontal.SampleMode` would be copied into the clipboard for use in your own application program.

You will see that some lines in XStreamBrowser are classified as “Action” rather than as a variable type. These actions are performed simply by sending the Action name with no argument, for example –

```
app.InternalCollection("Display").ClearSweeps
```

This would clear all the data from a persistence trace, for example. You can often reduce the amount of typing by the following kind of statements:

```
Set Acquisition = WaveMaster.Acquisition
Set Horizontal = Acquisition.Horizontal
```

```
Horizontal.SampleClock = "Internal"
Horizontal.ExternalClockRate = 1
Horizontal.HorScale = 0.0000005
Horizontal.HorOffset = 0
Horizontal.HorOffsetOrigin = 5
Horizontal.SampleMode = "Sequence"
```

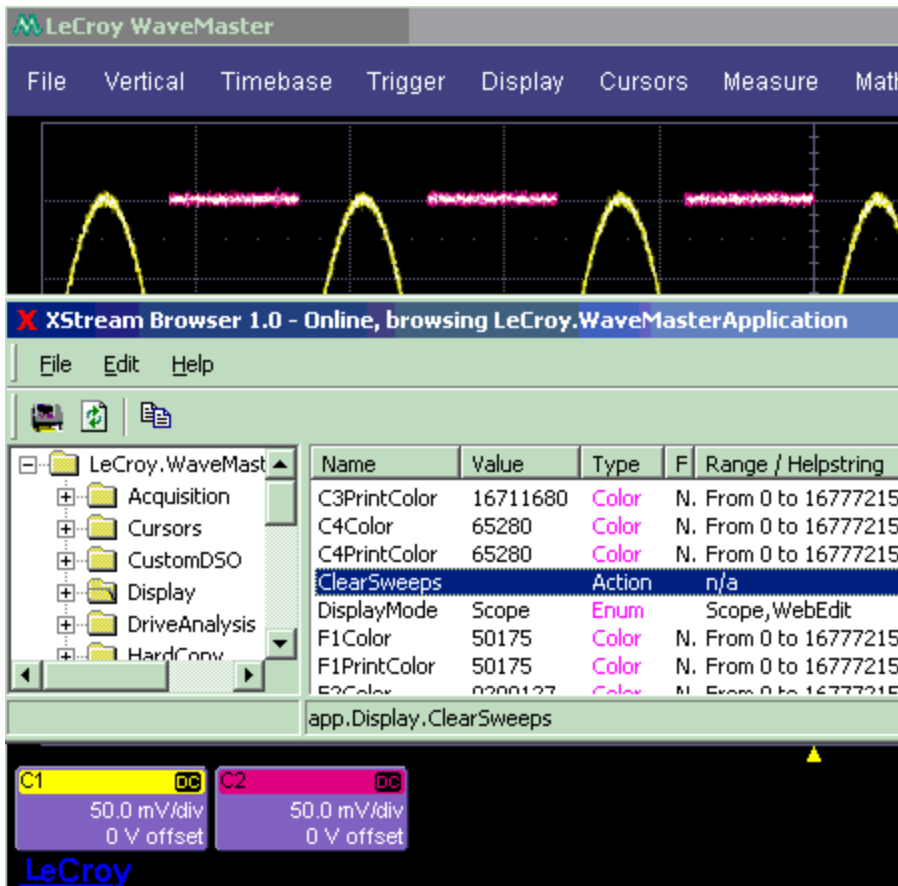
etc.

You could of course choose a much shorter word than `Horizontal`. For example, you could write:

```
Set Horiz = WaveMaster.Acquisition.Horizontal
Horiz.HorScale = 0.000001
```

ACTIONS

Besides the Control Variables, automation also provides for Actions. For example, we might want to Clear Sweeps on a particular trace. The XStreamBrowser example below shows the command for clearing a persistence trace.



The statement in your script or program would be as follows –

VBS 'app.InternalCollection("Display").ClearSweeps'

Other examples of actions are as follows:

```
app.Display.FactoryDefault
app.Acquisition.Horizontal.ZeroDelay
app.Acquisition.Trigger.ZeroLevel
app.Measure.SetGateToDefault
app.Memory.ClearAllMem
```

§ § §