WSCRIPT 2.0

J.A. Woollam Co., Inc.

Contents

WSCRIPT

Introduction

WSCRIPT is a WVASE32 utility program that allows you to script frequently repeated WVASE32 operations, and it gives you access to hardware functions not available from the WVASE32 interface.

Installation

WSCRIPT comes on 2 floppy disks or on a CD and requires 2.7MB of hard disk space. WVASE32 must be installed in order to run WSCRIPT.

To install WSCRIPT, insert Disk 1 of 2 in the floppy drive and execute the Setup.exe program. When prompted, insert Disk 2 of 2 to complete the installation. If WSCRIPT is on a CD, execute the Setup.exe program located in the folder <CDROM Drive>:\Wscript\Disk1. You will be prompted to select the destination folder for WSCRIPT. The default destination folder is <WINDISK>:\WVASE32\Tools. The files WSCRIPT.exe and Tools.cnf will be placed in this folder. Additional files required by WSCRIPT will be placed in the Windows\System folder. The setup program will also add a program icon to the Program Folder WVASE 3.0. These instructions assume that <WINDISK> is the C hard disk and that WSCRIPT has been installed in the default location.

Starting WSCRIPT

There are four ways to start WSCRIPT. The first three ways are standard Windows methods for executing programs. In the last method, WSCRIPT is launched directly from WVASE32. The standard Windows execution methods are:

- Selecting the Wscript icon from the Program Folder WVASE 3.0,
- Double-clicking the Wscript.exe icon in the C:\WVASE32\Tools folder, and
- Selecting Run from the Start Menu and executing the line C:\WVASE32\Tools\Wscript.exe.

To run WSCRIPT from WVASE32 select the menu item Global|Run WVASE Tools. A Run 'WVASE' Tools message box appears with a list of available tools. Select WSCRIPT and press the Ok button or just double-click WSCRIPT.

Quick Start

WSCRIPT operation is very simple. You create a script by typing commands into a text window, and you then run that script by pressing the Run Script button. WVASE32 executes each valid command returns a response to the WSCRIPT program. Execution of a script can be aborted by pressing the Cancel Script Button. Script execution will end after completion of the current command.

Example

WSCRIPT provides a place to enter the commands that you want to send to WVASE32 and a place to display WVASE32's responses to commands. In the example in Figure 1, on page 2, the user has entered two commands into the script area.

```
Message(Current Version of WVASE32)

GetVersion
```

When theRun Script button is pressed, WSCRIPT sends the first command, the Message command, with the parameter (Current Version of WVASE32). WVASE32 responds by echoing the parameter of the Message command back to WSCRIPT. This response is added to the response area, after which the next command in the script area is sent to WVASE32. WVASE32 responds to this command, GetVersion, with its version number, 3.255.

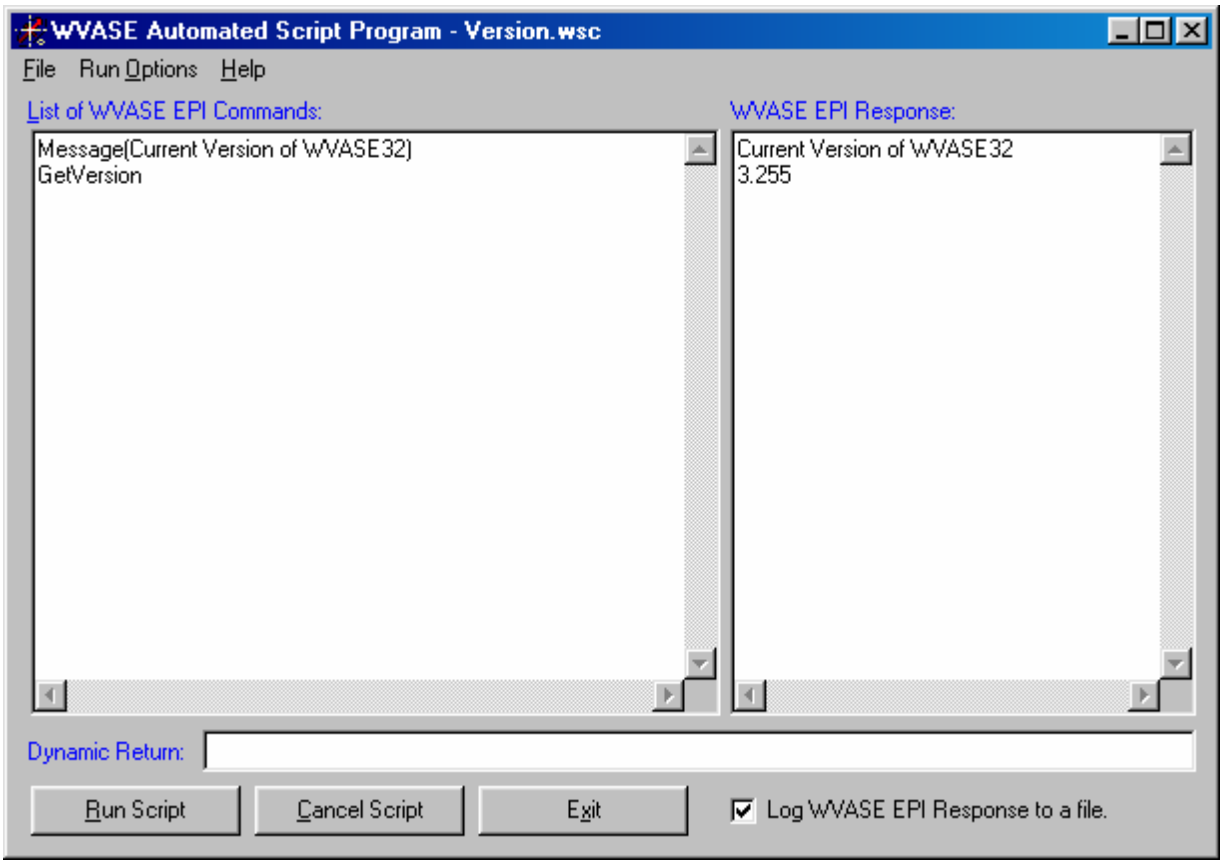

Figure 1: WSCRIPT Program

Creating a Script

When WSCRIPT is started no script is loaded, and the script area is blank. In the event that a script is loaded and you wish to create a new script select File|New. If the current script has not been saved or if you have made changes to the loaded script, you will be prompted to save the script.

You create a script by typing valid WVASE EPI commands into the script area. One command is allowed per line, and no extra characters are allowed between the beginning of the line and the beginning of the command. Valid commands and their syntax are listed in EPI Commands.

Saving a Script

You can save scripts that you have created by selecting File|Save. A file dialog box will appear that allows you to name the file and select the folder in which to save it. If the script already has a name, selecting File|Save will save the script to the current location without asking you for filename or location. This action replaces the previously saved version of the script. If you want to save the script with a new name or to a different location you must select File|Save As in order to bring up the file dialog box that allows you to specify the filename and location.

Opening a Script

To run a previously saved script select File|Open. A file dialog box appears that allows you to select a script. If you change the script you will be asked if you wish to save your changes, when you create a new file, open a different file, or exit WSCRIPT. Be sure of your wishes before responding to the dialog box.

Editing a Script

The script area responds to the usual Windows editing keys, Cut (Ctrl-X), Copy (Ctrl-C), Paste (Ctrl-V), as well as the Delete key and the Backspace key. When editing a script the same rules apply as when creating a script, one command per line, and no extra characters between the start of the line and the beginning of the command.

Running a Script

A script is executed by pressing the Run Script button. WSCRIPT sends the commands in the script area one command at a time to WVASE32. WVASE32 executes the command and returns a response to WSCRIPT when execution is completed. Then the next command in the script is sent as long as the Cancel Script button has not been pressed. WSCRIPT stops sending commands to WVASE32 when it reaches the end of the script or upon completion of the command that was executing when the Cancel Script button was pressed.

The option to save all of WVASE32's responses to commands in the script exists, and can be enabled by checking the Log WVASE EPI Responses to file check box. If this box is checked, a file dialog box will appear to allow you to specify the name and location of the response file.

Exiting Wscript

WSCRIPT shuts down when the Exit Button is pressed or File|Exit is selected. If the current script has changed, you will be prompted to save the changes. If WSCRIPT was not launched from WVASE32, WVASE32 will also shutdown.

Getting Help

This manual is included with WSCRIPT as a help file, which is accessible by pressing F1 or by selecting HelpContents. All of the EPI Commands are also listed in the help file.

EPI Commands

Modeling Commands

Command Name: **AddLayer**
Parameters: full pathname (including .mat extension) of the material name
Return String: OK
Description: Add a layer to the current model, above the current active layer.

Command Name: **CurrentDirectory**
Parameters: a valid windows path.
Return String: OK
Description: Changes the current default directory for file operations.

Command Name: **CurrentLayer**
Parameters: desired layer number.
Return String: OK
Description: Changes the current active layer in the model.

Command Name: **CurrentModel**
Parameters: desired layer number {0..9}.
Return String: OK
Description: Sets the currently active model.

Command Name: **Defaults**
Parameters:

Angstroms	Sets default wavelength units to Angstroms
nm	Sets default wavelength units to nanometers
ev	Sets default wavelength units to photon energy
um	Sets default wavelength units to microns
1/cm	Sets default wavelength units to wavenumbers
Thick-Angstroms	Sets default units for layer thickness to Angstroms
Thick-nm	Sets default units for layer thickness to nanometers
Thick-um	Sets default units for layer thickness to microns
nk	Sets default units for optical constants to n&k
e1e2	Sets default units for optical constants to e1&e2
maxdata=	Changes the maximum number of data points and erases all current data/models

Return String: OK
Description: Sets the default units for the WVASE program.
Comment: Only one of the above parameters should be specified in each call to this command; to set units for both wavelength and thickness, call this command twice.

Command Name: **DeleteLayer**
Parameters: none
Return String: OK
Description: Deletes the current active layer in the model.

Command Name: **EditLayer**
Parameters: a string of the format:
parmname1=parmvalue1,parmname2=parmvalue2,etc.
Valid Parameter Names (depends on the current layer type)

All Layers	thick={thickness in current units} rate={rate in Å/s}
EMA	nmat={1,2,3} type={0,1,2} mat1={e.g. sio2}, mat2={}, mat3={} EMA2={}, EMA3={}, EMAq={}
VIRTUAL	fixed={}, seperation={}, average={}, time={}
GRADED	slices={}, nodes={}, type={} {mat1={e.g. sio2},mat={} OR alloy={e.g. algaas}}
CAUCHY	An={}, Bn={}, Cn={}, Ak={}, Bk={}, Ck={}
ALLOY	Alloy={}
TEMP	Temp={}
ALLOY-TEMP	Temp={}, Alloy={}
LORENTZ	e1(inf)={}, Am*n*={},Br*n*={},En*n*={} *{where n=1 to 7}*
PSEMI	Sel*n*, Am*n*, Br*n*, En*n*, Lcon*n*, Rcon*n*, Dis , PolePos, PoleMag, PolePos2, PoleMag2, Lpos*n*, Lamp*n*, L2nd*n*, Rpos*n*, Ramp*n*, R2nd*n*

Return String: OK
Description: Edits parameter values of the current active layer in the model.
Example: EditLayer(thick=500,An=1.5,Bn=.1)

Command Name: **EditOptions**
Parameters: a string of the format:
parmname1=parmvalue1, parmname2=parmvalue2, etc.
Valid Parameter Names

mode=	0=ideal mode', 1=thick non-uniformity, 2=wvl bandwidth, 3=ang bandwidth, 4=all 3 types
convtype=	convolution type {0=sqaure,1=triangular,2=Gaussian}
convpts	number convolution points
thickunilayer	??
patterned=	??
ellipsometertype=	specifies the ellipsometer configuration. {0=RAE/RCE, 1=RAE w/ AR, 2=Phase modulated, 3=RCE(PSCA), 4=RCE(PCSA)}
minidelta=	minimum delta for experimental data
modambient=	if 1 specifies top layer is the ambient, {0,1}
backrefl=	
pol=	
thkuni=	

bndwidth=

angsprd=

1strefl=

wvlshift=

wazimuth=

wretard=

deloff1=

deloff2=

deloff3=

Return String: OK
Description: Edits model options for current model.

Command Name: **LayerName**
Parameters: layer number
Return String: Layername=*name* returns an error if layer number not valid
Description: Returns the material name of the specified layer for the current model.

Command Name: **LayerOptConst**
Parameters: wavelength in current units
Return String: real and imaginary parts of complex index or complex dielectric based on current units. e.g. n=1.533 k=0.012, or e1=2.35 e2=0.0368
Description: Returns optical constants at specified wavelength for current layer.

Command Name: **LayerSave**
Parameters: full path name of file including .mat extension
Return String: OK
Description: Saves the current layer as a dispersion model instead of a tabulated list.

Command Name: **LayerThick**
Parameters: layer number (invalid number returns an error)
Return String: Thick=*thick*Å
Description: Returns the thickness of specified layer in Å for the current model.

Command Name: **ModelDelete**
Parameters: none
Return String: OK
Description: Deletes the current model and removes all fitting parameters.

Command Name: **ModelOpen**
Parameters: full path name of file including .mod extension
Return String: OK if successful
Description: Replaces the current model with one from the specified file.

Command Name: **ModelSave**
Parameters: full path name of file including .mod extension
Return String: OK
Description: Saves the current model to the specified file.

Command Name: **OptConstSave**
Parameters: full path name of file including .mod extension, *comment*
Return String: OK
Description: Saves the current layer's optical constants in a tabulated material file.
Example: OptConstSave(c:\wvase\mat\new_si.mat,This is a material file)

Hardware Commands

Command Name: **DynoScan**
Parameters:
Notes: A 'flag' parameter is true or enabled if it is present in the string, and false or disabled if it is not present in the string. A 'Value' parameter requires a corresponding value to be specified, using an '=' character, i.e., ParmName=ParmValue.

revs=	Number of analyzer revolutions to average signal.
acqtime=	Specifies time in seconds for data acquisition, overloads "revs=" specification.
pol=	Input polarizer angle for data acquisition, should be set to near the psi values.
highaccuracy	Specifies that polarizer zone averaging is to be used, default is not used.
aoi=	Angle of incidence for insitu measurement.
type=	Specifies the DataType for the measurement, normally 0 for ellipsometric data.
dcoffset=	Causes a dc offset measurements to be made at the specified number of revs.
single	Causes only a single data scan to be taken.
update=	# of points between graphics updates, if not specified default is 0.
realtime=	Requests WVASE to send a 'Realtime=' message to the Client via the DDE 'hot link' at the exact 'ellipsometer' time specified.
report	Tells WVASE to update the Client's DDE control after each data point is acquired.
fit	Tells WVASE to perform a model fit to the data in real-time. *The model and fit parameters must be defined in WVASE32 prior to initiating dynamic data aquisition and fitting!*
end	Stops the data acquisition.
add	Starts data acquisition, without resetting the data acquisition timestamp to zero; the subsequent data will be appended to the existion data.
channels[*ChannelSelectionStr*]	Selects all channels for acquistion.
channelsfit[*ChannelSelectionStr*]	Selects all channels for fitting.
channelsgraph[*ChannelSelectionStr*]	Selects all channels for graphing.
realtime=	Requests WVASE to send a message to the Client at the specified time.
nosave	Tells WVASE not to save the dynamic data.
simulate	'Simulates' data acquisition without the hardware running. To use, load in a experimental file with dynamic data before executing the DynoScan command; the program will then 'pretend' that the experimental data values are being acquired in realtime by the hardware. If you set simulate=300, every 300 milliseconds the program will simulate the acquisition of experimental data.
polled	This flag tells WVASE to retrieve acquired data only in reponse

to the EPI DynoGetData command.

The *ChannelSelectionStr* used in the ‘channel’ commands can take one of the following formats:
- the keyword ‘all’; selects all channels
- a range of channels ‘10-20’; specify a beginning and ending channel, separated by ‘-‘
- specific channels ‘0 10 30 40’; channel numbers separated by spaces
- the keyword ‘skip=n’; if skip=1, every other channel is selected

Return String: if the 'report' parameters is selected, a the Client's DDE control is updated with new data after each data point is acquired. This will be either psi and delta data, or fit data, depending on the setting of the 'fit' parameter.
Description: MXX only, initiates dynamic data acquisition/fitting.
Examples:
DynoScan(revs=30,pol=20,highaccuracy,single)
=> performs a single (or static) measurement
DynoScan(revs=20,pol=15,aoi=75.2,report,fit,update=3,channelsgraph[0 20 40])
=> starts a dynamic scan, fitting a model to the data and reporting the results over the DDE ‘hot link’, and graphing channels 0, 20, and 40 (the graph is updated every third data acquisition)
DynoScan(end)
=> ends the dynamic scan
Important Note: To use the 'report' option of this command, it is necessary to set up a second DDE conversation with the WVASE32 program. This DDE link must be a 'hot' link, as the WVASE32 program will continually update the Client's control as data is being acquired. To establish this link (with Textbox named 'Text2') in Visual Basic:

```
Text2.LinkTopic = "WVASE|DDE_DynamicData"
Text2.LinkItem = "Data"
Text2.LinkTimeout = -1
Text2.LinkMode = 1
```

To establish a 'hot' link using other lanugages, a WM_DDE_ADVISE message is required, with the Server name of "WVASE", the DDE topic "DDE_DynamicData", and the DDE item "Data". The data string returned over this hot link depends on the parameters which are passed via the preceding ‘DynoScan’ command. The ‘report’ flag must be specified for any information to be passed over the hot link. A ‘Time=’ parameter will always be part of the ‘report’ string, along with ‘AlignX=’ and ‘AlignY=’ values from the alignment detector. If the ‘fit’ flag is set, the MSE and model fit parameters will also be appended to the ‘report’ string. If the ‘RawData[ChannelSelectionStr]’ parameter is included, the specified ellipsometric data will be added to the ‘report’ string, in the following format:
RawData[Channel0:Psi0 Delta0 Channel1:Psi1 Delta1 Channeln:Psin Deltan]

Command Name: **HardAlign**
Parameters:
MXX Systems

revs=	Number of analyzer revolutions to average signal.
acqtime=	Specifies time in seconds for averaging, overloads “revs=” specification. Maximum revs = 300.
nodisplay	Specifies that only data is returned, cross hairs will not be displayed on screen.
report	Causes alignment data to be returned continuously until stopped by operator.
gated	specialized alignment procedure primarily for wobbling substrates. Gating controls set via MxxGateAcq command.

VASE Systems

revs=

nodisplay

slit=

pol=

report

Return String: OK
Description: Display the alignment crosshair screen for system allignment.

Command Name: **HardCal**
Parameters:
MXX Systems

revs=	Number of analyzer revolutions to average signal.
acqtime=	Specifies time in seconds for averaging, overloads "revs=" specification. Maximum revs = 300.
pts=	Number of data points to acquire for calibration.
span=	Degree span of polarizer for calibration.
type=	0 for normal, 1 for coarse, 2 for electronics, 3 for straight through system, 4 for off-sample system.
wineffects	Specifies that window effects be included in a normal (type=0) calibration.

VASE Systems

revs=	Number of analyzer revolutions to average signal.
pts=	
span=	Degree span of polarizer for calibration.
type=	0 for normal, 1 for coarse, 2 for electronics, 3 for straight through system, 4 for off-sample system.
wvl1=	Wavelength in Å's for calibration of UV detector.
wvl2=	Wavelength in Å's for calibration of IR detector.
AOI=	Angle of incidence for calibration.

Return String: Fit results.
Description: Calibrate the hardware.

Command Name: **HardInit**
Parameters:
MXX Systems:

simulate	Specifies that preexisting data be used in place of newly acquired data.
file=	Specifies data file used for simulation, if not present, existing data in exp window is used.

VASE Systems:

warm	Directive to not reramp the analyzer moter if already initialized.
file=	Specifies data file used for simulation, if not present, existing data in exp window is used.

Return String: OK
Description: Initializes the hardware.

Command Name: **HardSignal**
Parameters:
For MXX Systems:

revs=	Number of analyzer revolutions to average signal.
acqtime=	Specifies time in seconds for averaging, overloads "revs=" specification. Maximum revs = 300.
shutter=closed	Closes shutter, if shutter is not specified it remains in current state.
shutter=open	Opens shutter, if shutter is not specified it remains in current state.
calcdcoffset	Specifies a dc offset measurement should be made before returning data.

For VASE Systems:

revs=	Number of analyzer revolutions to average signal.
noautoslit	Directive to keep slits fixed even if detector is overload, default is auto sizing.

Return String (MXX): if calabrated Ψ,Δ,δΨ,δΔ, … for each channel
if not calibrated A2,B2,δA2,δB2, … for each channel
Return String (VASE): Ψ,Δ,δΨ,δΔ,DC,Chop2A,A2,B2,δA2,δB2
Description: Returns info normally displayed on the Hardware Signal screen..

Command Name: **HardStatus**
Parameters: none
Return String: Init={0,1},Cal={0,1}
Description: Retrieves initialization and calibration status.

Command Name: **MonoInfo** (VASE only)
Parameters: none
Return String: ??
Description: Returns complete description of the monochromator's current state.

Command Name: **Mononame** (VASE only)
Parameters: none
Return String: ??
Description: Returns type of monochromator being used.

Command Name: **MoveAOI**
Parameters: home, *desired goniometer position (in degrees)*
Return String: OK
Description: Moves the angle of incidence.
Important Note: If the desired position is zero, then only the sample stage moves. Otherwise both goniometers move. Caution, it is possible to move the goniometers to positions where the protection limit switches are needed to prevent mechanical contact of detector and input arms.

Command Name: **MoveFilter** (VASE only)
Parameters: {0-4}, home, homeconst=
Return String: OK
Description: Allow direct control of the filter used on a monochromator.
Example: MoveFilter(1), MoveFilter(homeconst=34, MoveFilter(home)

Command Name: **MoveMono** (VASE only)
Parameters: desired monochromator position (in Å)
Return String: OK
Description: Move monochromator to specified wavelength.
Example: MoveMono(5000), MoveMono(0) moves to white light

Command Name: **MovePol**
Parameters: desired polarizer angle (with respect to plane of incidence)
Return String: OK
Description: Move polarizer to specified angle.
Example: MovePol(30)

Command Name: **MoveRetarder** (VASE only)
Parameters:

X=	axis#1-position in deg, if not specified current X position is maintained unless homed.
Y=	axis#2-position in cm, if not specified current Y position is maintained unless homed.
Home	Homes the stages; should not be used with X and Y parameters.
Null	

Return String: e.g. X=6.5, Y=4.5
Description: Moves retarder as specified.

Command Name: **MoveTranslator**
Parameters:

X=	x- position in cm, if not specified current X position is maintained unless homed.
Y=	y- position in cm, if not specified current Y position is maintained unless homed.
Home	Homes the translation stage; should not be used with X and Y parameters.

Return String: e.g. X=6.5, Y=4.5
Description: Move translation stage to specified position.
Example: MoveTranslator(x=14,y=-5), MoveTranslator(home),
MoveTranslator() returns current X-Y position

Command Name: **MxxGatedAcq** (MXX only)
Parameters:

gated	Indicates gating will be used, if not present gating is turned off.
xtol=	Specifies valid alignment range for accepting data.
ytol=	Specifies valid alignment range for accepting data.
threshold=	Specifies minimum intensity for accepting data.
trackfactor=	??

Return String: OK
Description: Accepts or rejects data based on current alignment. This command is used primarily with rotating substrates that have significant wobble.

Command Name: **SlitWidth** (Monochromators with variable slit width only)
Parameters: a number specifying the default width of the entrance slit in microns
Return String: OK
Description: Set the defalut width of the monochromator entrance slit.

Command Name: **SpectroScan** (VASE only)
Parameters: for most of these values, if they are not set then the existing value form the last scan is used

expections()	The parameters avalilable are: usedynamicave, pol=, zoneave, useautoretarder, fixedslit
sampletype={0..4}	0=isotropic, 1=slightly anisotropic, 2=non-regressed anisotropic, 3=highly anisotropic mode #1, 4=highly anisotropic mode #2

wvlstart=	Beginning wavelength, in current wavelength units
wvlend=	End wavelength, in current wavelength units
wvlby=	Wavelength increment, in current wavelength units
angstart	Beginning angle
angend	Ending angle
angby	Angle increment
revs=	Revolutions to average per measurement
usedynamicave	Causes dynamic averaging to be used when intensity drops below threshold
dynamicmax	Maximum number of revs to use with dynamic averaging
thresholdintensity	Intensity below which dynamic averaging used, typically =0.4
pol=	Polarizer angle to use without tracking, else tracking
trackpoltol	Polarizer tracking tolerance if not fixed
zoneave	Directive to zone average, no zone averaging is the default unless specified
autoretarder={0,1,2}	If 1 or 2 AR is used in high accuracy or fast mode respectively, if 0 or not present AR is not used
splitscan={0,1}	If 1, grating changes occur in the middle of the overlap region
slit=	Default slit width to use unless overloaded and then it will be reduced
fixedslit=	Slit width used regardless of overloads, if not present slits automatically move
transmission	Directive to move only the sample stage, if not present reflection assumed
report	Data is sent back over the dynamic dde link after every point??
append	New data is appended to existing experimental data, else existing data deleted

Return String: OK
Description: Acquires a variable angle spectroscopic ellipsometric (VASE) data scan.

Command Name: **R&TScan** (VASE only)
Parameters:

baseline	Specifies that baseline data will be aquired
baselineonsample	Specifies that the baseline is taken using the sample, is inclusive of baseline
Type={1..18}	An integer that specifies the type of data to acquire, valid only when the 'Baseline' parameter is specified.1=pT, 2=sT, 3=uT, 4=pR, 5=sR, 6=uR, 7=ATpp, 8=ATps, 9=ATss, 10=ATsp, 11=ATuu, 12=ATux, 13=Arpp, 14=ARps, 15=ARss, 16=ARsp, 17=ARuu, 18=Arux
WvlStart=	Beginning wavelength, valid only when the 'Baseline' parameter is specified
WvlEnd=	End wavelength, valid only when the 'Baseline' parameter is specified
WvlBy=	Wavelength increment, valid only when the 'Baseline' parameter is specified
AngStart=	Beginning angle, valid only when the 'Baseline' parameter is *not*specified

AngEnd=	Ending angle, valid only when the 'Baseline' parameter is *not* specified
AngBy=	Angle increment, valid only when the 'Baseline' parameter is *not*specified
Revs=	Revolutions to average per measurement
AuxDet	Directive to use an auxiliary R/T detector
Pol=	Polarizer angle for user defined polarizer R/T measurements, valid only when the 'Baseline' parameter is specified

Return String: OK
Description: Acquires Reflection and/or Transmission data.
Comment: To acquire R&T data via the WVASE EPI, the external program must first call the R&TScan command with the 'Baseline' parameter set. This will acquire a baseline scan at the specified wavelength range and data type. It is the user's responsiblity to ensure that the sample is not blocking the beam during the baseline scan. The external program can either prompt the user to remove sample, or automatically translate the sample out of the beam (using the MoveTranslator EPI command). A second call to the R&TScan command without the 'Baseline' parameter set will acquire the R&T data, using the previously specified wavelength range and data type. The angle range should be specified in this second call. Again it is the user's responsibility to have the sample mounted and aligned properly, as the WVASE program does not provide any prompts when operated via the EPI interface.

Experimental Data Commands

Command Name: **ExpAdd**
Parameters:

psi=	psi value of added data point
del=	del value of added data point
dpsi=	standard deviation on psi of added data point, default is 0.01°
ddel=	standard deviation on del of added data point, default is 0.1°
ang=	sets the angle of incidence for the data point
wvl=	sets the wavelength in ° for the data point
time=	sets the time in minutes for the data point

Return String: OK
Description: Adds dynamic data to the Experimental window.
Example: ExpAdd(psi=20,del=135,ang=75,wvl=5000,time=5.63)

Command Name: **ExpComment**
Parameters: comment text
Return String: OK
Description: Specifies a comment for the Experimental data file.

Command Name: **ExpDelete**
Parameters: none
Return String: OK
Description: Deletes all data in the Experimental window for the current model.

Command Name: **ExpMerge**
Parameters: full path of file including .dat extension
Return String: OK
Description: Merges/sorts a data file with the existing data in the Experimental window.

Command Name: **ExpOpen**
Parameters: full path of file including .dat extension
Return String: OK
Description: Opens a data file in the Experimental window for the current model.

Command Name: **ExpRange**
Parameters:

For dynamic data: tstart, tend, skip, wvls[‘all’ or list of selected channel numbers],channelsfit[{*channels*}], channelsgraph[{*channels*}]
For VASE: wvlstart, wvlend, angstart, angend
spectroscopic
adddata
removedata
alldata
datatypes[]
datatypes[all]

Return String: OK
Description: Selects a range of the Experimental data.
Examples for dynamic data:

ExpRange(tstart=1.5, tend=3.5, skip=3, wvls[all])
ExpRange(tstart=0.5, tend=1.5, skip=0, wvls[0 10 20 40])

Command Name: **ExpSave**
Parameters: full path of file including .dat extension
Return String: OK
Description: Saves the data in the Experimental window for the current model.
Important Note: If a filename is not specified, the file Save As box will prompt the user for a filename.

Command Name: **ExpType**
Parameters: an index into the list of available data types starting with E as found in the Exp|ChangeDataType option e.g. 0=E, 1=Eb, 2=Er, etc.
Return String: OK
Description: Changes data type of currently selected experimental data.

Fit Commands

Command Name: **FitDefaults**
Parameters:

MinMSE=
ChgMSE=
ReflWeight=
TransWeight=
FitIncr=

FitWeight is an index into the option list in the FitDefaults dialog box starting with 0.

MaxIter=
FitWeight=
GraphFit

Return String: OK
Description: Set default parameters for data fitting.

Example: FitDefaults(FitWeight=0, MinMSE=1e-3, ChgMSE=1e-6, MaxIter=3, GraphFit)

Command Name: **FitGet**
Parameters: none
Return String: Lists the MSE, fit parameter names, and final fit values from the most recent fit.
Description: Get the fit data from the most recent fit.

Command Name: **FitGlobal**
Parameters: iter=, parmeter={??}
Return String: Lists the MSE from the best fit.
Description: Initiates a global fit.

Command Name: **FitLastPoint**
Parameters: none
Return String: OK
Description: Initiates a 'point by point' fit for dynamic data, using only the last data points.

Command Name: **FitNormal**
Parameters: none
Return String: MSE convergence message.
Description: Initiates a normal fit.

Command Name: **FitParms**
Parameters: list of the desired fit parms, of the format(Square brackets [ ] can be used to specify upper and lower bounds on the fit parameters.):
Parmname1.Layer#1, Parmname2.Layer#2, etc.
Valid Parameter Names (depends on the current layer type)

All Layers	thick, OptRe, OptIm (optical constants), angle, rate (growth rate)
CAUCHY	An, Bn, Cn
ALLOY	Alloy
TEMP	Temp={}
ALLOY-TEMP	Alloy, Temp
PSEMI	A*m* , B*n*, En *n*, Disc*n*, PolePos, PoleMag, PolePos2, PoleMag2, Lpos*n*, Lamp*n*, L2nd*n*, Rpos*n*, Ramp*n*, R2nd*n*

Return String: OK
Description: Initiates a normal fit.
Example: FitParms(Thick.1[100 120],An.1,Bn.1,Angle.0[74.5 75.5])

Command Name: **FitPtByPt**
Parameters: shortwvl fit starts at shortest wavelength, else starts at longest
Return String: MSE convergence message.
Description: Defines the list of active fitting parameters.

Command Name: **ListParms**
Parameters: none
Return String: NumOfParms ParmName1 ParmName2 ParmName3
Description: Lists the names of the currently defined fit parameters.

Command Name: **SetParmBounds**
Parameters: list of bounds for all fit parameters separated by spaces
Return String: OK
Description: Defines lower and upper bounds for the currently defined fit parameters. This command also sets the current parameter value to the average of the lower and uper bounds. Call the ListParms function to determine the number,

names, and order for the WVASE fit parms.
Example: SetParmBounds(100 200 1.5 1.6 74.5 75.5)

Command Name: **SetParmValues**
Parameters: list of parameter values for all fit parameters separated by spaces
Return String: OK
Description: Defines parameter values for the currently defined fit parameters. Call the ListParms function to determine the number, names, and order for the WVASE fit parms.
Example: SetParmValues(120 1.57 75.3)

Command Name: **CalcMSE**
Parameters: ??
Return String: ??
Description: ??.

Generate Data Commands

Command Name: **GenOpen**
Parameters: full path name of file including .dat extension.
Return String: OK
Description: Opens a data file in the Generated window for the current model.

Command Name: **GenData**
Parameters: none
Return String: OK if successful
Description: fills the Generated window as if the Generate_Data menu item had been used..

Command Name: **GenDelete**
Parameters: none
Return String: OK
Description: Deletes all data in the Generated window for the current model.

Command Name: **GenSave**
Parameters: full path name of file including .dat extension.
Return String: OK
Description: Saves the data in the Generated window for the current model.
Important Note: If a filename is not specified, the file SaveAs box will prompt the user for a filename.

Graph Commands

Command Name: **GraphCopy**
Parameters: graphwnd= integer specifying a Windows device context to draw into, else to clipboard.
Return String: OK
Description: Exports graph to clipboard or specified Windows handle.

Command Name: **GraphStyle**
Parameters:

2d	Directive to show 2d plots.
3d	Directive to show 3d plots.

dblyaxis={0,1}	If 1 then double y-axis graphs created, if not present existing setting remains.
trajplot={0,1}	If 1 then trajectory plot is grpahed (e.g. Psi v time).
derivplot={0,1}	If 1 then derivatives are plotted, if not present existing setting remains.
derivorder=	Sets the derivative order to plot.
derivpts=	Sets the number of data points to work with when calculating derivatives.
derivpolyorder=	Sets the polynomial order for plotting derivatives.
3delevation=	Set the elevation angle for view a 3d plot.
3drevx={0,1}	If 1 then x-axis reversed, if not present existing setting remains.
3drevy={0,1}	If 1 then y-axis reversed, if not present existing setting remains.
3dcolor={0..15}	Index into color pallete.

Return String: OK
Description: Change the graph style.

Command Name: **GraphType**
Parameters: type number as indexed according the Graph|Type menu, currently {0..15} (e.g. 0=Psi, 1=Del, 2=TanPsi, …)
Return String: OK
Description: Sets type of graph to display.

Command Name: **GraphSetupSave**
Parameters: ??
Return String: ??
Description: Saves current graph settings.

Command Name: **GraphSetupApply**
Parameters: ??
Return String: ??
Description: ??.

Command Name: **GraphSetupFlags**
Parameters: ??
Return String: ??
Description: ??.

Miscellaneous Commands

Command Name: **Exit**
Parameters: none
Return String: OK
Description: Exits the WVASE program.

Command Name: **GetVersion**
Parameters: none
Return String: WVASE version number
Description: Retrieves the version of the WVASE being talked to..

Command Name: **LoadEnv**
Parameters: full path name of file including .env extension
Return String: OK
Description: Loads environment file.

Command Name: **LogFile**
Parameters: text string
Return String: OK
Description: Logs a string of text to the WVASE log file.

Command Name: **Message**
Parameters: message string
Return String: *message string*
Description: Echos a command back through the DDE return string.

Command Name: **MenuDisable**
Parameters: none
Return String: OK
Description: Removes menus from WVASE screen, preventing direct user access.

Command Name: **MenuEnable**
Parameters: none
Return String: OK
Description: Restores menus to WVASE screen, allowing direct user access.

Command Name: **SaveEnv**
Parameters: full path name of file including .env extension
Return String: OK
Description: Saves environment to file.

Command Name: **Window**
Parameters: *name of window*, *size for window*
Window names: Hard, Exp, Gen, Graph, Fit, Model
Window sizes: Maximize, Minimize, Restore, Hide, Size=*x*,*y*,*width*,*height*
Return String: OK
Description: Selects current window and specifies its size.

Command Name: **CmosTime**
Parameters: ??
Return String: ??
Description: ??.

Command Name: **Delay**
Parameters: none
Return String: OK
Description: Inserts a 1 millisecond delay.

Command Name: **MsgBox**
Parameters: ??
Return String: ??
Description: ??.

Command Name: **Input**
Parameters: ??
Return String: ??
Description: ??.

Command Name: **Let**
Parameters: ??
Return String: ??
Description: ??.

Advanced WVASE EPI Commands for use in Fit Strategies

Command Name: **SetVars**
Parameters: VarName1=DefaultValue1, VarName2=DefaultValue2,…
Return String: OK
Description: This optional command should be the first line in the fit strategy, as it is used to specify variables for use throughout the remaining fit strategy text. Variable names *can*include spaces. To access the value of a variable, simply embed the variable name enclosed in {} within the fit strategy text, i.e. EditLayer(Thick={*VarName1*}). You can also perform simple arithmetic with fit strategy variables: multiplication by a constant, and addition by a constant. The syntax for the variable arithmetic is {*ConstantValue*VarName1*} or {*ConstantValue+VarName1*}. When running the ADAP program in the 'user interface mode', the user will be prompted to enter values for each variable defined in the SetVars statement; in the 'hidden mode' the variable values are parsed from the command string.

Command Name: **Message**
Parameters: message text
Return String: message text
Description: This command will echo the specified message text back in the return string..

Command Name: **#If, #Else, #EndIf**
Usage: #If(*Expression1 ComparisionOperator Expression2*)
Description: This command is used to provide basic conditional branching capability within fit strategies. The *ComparisionOperator* must be <, >, =, <=, >=, or <>. *Expression1* and *Expression2* can be constant values, the word 'MSE', or the exact WVASE name of any currently defined fit parameter (for example, Thick.1). If the '#If(...)' expression is TRUE, the EPI commands on the subsequent lines will be executed; if the expression is FALSE, WVASE will jump to the next line containing the #Else or #EndIf keyword. *Unlike other EPI commands, the #Else and #EndIf keywords are case sensitive!*

Command Name: **Goto, Label**
Usage: Goto(*LabelName*)
Label(*LabelName*)
Description: These commands allow jumps when implementing fit strategies.

Example Fit Strategy EPI Commands

```
SetVars(Nominal Oxide Thickness=100)
modeldelete
defaults(thick-nm)
defaults(nm)
addlayer(si)
addlayer(sio2-c)
fitparms(thick.1[{.5*Nominal Oxide Thickness} {1.5*Nominal Oxide Thickness}])
exprange(wvlstart=400, wvlend=800,skip=30)
fitglobal(thick.1[10],iter=1)
exprange(wvlstart=300, wvlend=900,skip=5)
fitparms(thick.1[{.5*Nominal Oxide Thickness} {1.5*Nominal Oxide
        Thickness}],an.1[1.4 1.6],bn.1[0 .1])
fitdefaults(chgmse=1e-4,maxiter=8)
fitnormal
#If(MSE<2)
```

```
  fitget(Oxide Thickness=thick.1)
  goto(ExitStrategy)
#Else
 message(Error: Data fit was not acceptable)
#EndIf
Label(ExitStrategy)
```

Examples

Looping in the WVASE EPI

This example illustrates how looping can be performed using the WVASE EPI commands.

List of WVASE EPI Commands:

```
SetVars(start=6, end=10)
Let(i = {start})
  Label(loop)
    Message(Message:  i = {i})
    Let(i = {i + 1})
  #If({i} < {end})
    Goto(loop)
  #EndIf
Message(The End)
```

WVASE EPI Response:

```
I = 6
Message:  I = 6
I = 7.000000
#If(7.000000 < 10) is TRUE
Goto:  loop
Message:  I = 7.000000
I = 8.000000
#If(8.000000 < 10) is TRUE
Goto:  loop
Message:  I = 8.000000
I = 9.000000
#If(9.000000 < 10) is TRUE
Goto:  loop
Message:  I = 9.000000
I = 10.000000
#If(10.000000 < 10) is FALSE
The End
```

Index

O

R

S

W