CHAPTER 1 THE SCRIPT WINDOW

The Script window is really a window for Tcl commands, but ASCEND provides a set of our own commands so that you need only learn a little Tcl to get along fairly efficiently. Tcl borrows a lot from other languages, particularly C and lisp, and a little from the normal UNIX operating system. Users who want to develop design and optimization algorithms for specific problems can prototype these algorithms using Tcl and the tools already in ASCEND.

The best way to think of a script is as a METHOD container which performs operations at the universal scope, where the universe is really your entire computer and not just the universe of ASCEND objects.

The easiest way to learn the ASCEND Script commands is to turn on Record and watch what appears as you push buttons in the interface. This is how we usually write script files.

1.1 ASCEND SCRIPT KEYWORDS

Script keywords are Tcl functions

Script keywords are commands defined for ASCEND (in CAPS) which may be used on the commandline or in the Script. Keywords are actually Tcl functions which encapsulate 1 or more of the C primitives and other Tcl procedures so the user can conveniently emulate button presses.

Tcl is cAsE sensitive

Tcl is a case sensitive language, so capitalization of the keywords is important. E.g. read is a Tcl operating system call while READ is an ASCEND keyword. OF, WITH, TO, and other arguments in all CAPS are modifiers to the keyword which make it do different things.

1.1.1 ARGUMENTS

Each keyword takes zero or more arguments:

<arg> <arg> indicates the use of arg is NOT optional.

<a1,a2> <a1,a2> indicates that the use of either a1 or a2 is required

<a1 a2> <a1 a2> indicates use of both a1 and a2 required

[a1] [a1] indicates the use of a1 is optional.

[a,b] [a,b] indicates that either a or b is optional but not both.

qlfdid qlfdid is short for '**Q**ua**L**i**F**ie**D** **ID**entifier'

qlfpid qlfpid is short for '**Q**ua**L**i**F**ied **P**rocedure **ID**entifier'

qlfname qlfname is short for '**Q**ua**L**i**F**ied **NAME**'

{ } It is generally best to **enclose all object names and units in {**braces**}** to prevent Tcl from performing string substitution or otherwise operating on the arguments before passing them to the keyword function. This is particularly important for ASCEND array objects: sim.a['benzene'] is an array element to ASCEND, but Tcl will try to execute the string inside the []s as a Tcl command before passing control to the keyword unless {sim.a['benzene']} is protected by braces.

1.1.2 SCRIPT KEYWORDS

An explanation of the Script commands is given in this section. A quick reference is also given at the end of this chapter.

ASSIGN

ASSIGN <qlfdid> <value> [units]
Set the value of atom 'qlfdid' from the script. If value is real, give a

set of units compatible with the dimensions of the variable. If the variable has no dimensions yet, ASSIGN will fix the dimensions.

BROWSE

BROWSE <qlfdid>
Exports qlfdid to the browser.

COMPILE

COMPILE <simname> [OF] <type>
Build a simulation of the type given with name simname. You can get away with leaving out OF or spelling it wrong.

DELETE

DELETE <TYPES,simname>
Delete all types or delete specified simulation. If you name a simulation TYPES you get what you deserve.

DISPLAY

DISPLAY <kind> [OF] <qlfdid>
Shows something in the display window. How qlfdid is displayed varies with kind. Kinds are: VALUE, ATTRIBUTES, CODE, ANCESTRY. Not completely implemented yet. See the DISPVAL keyword.

DISPVAL

DISPVAL <qlfname>
Prints the instance part of qlfname

EXIT

EXIT [NOCONFIRM]
Quits ASCEND. A confirmation is requested unless NOCONFIRM is given. If any argument other than NOCONFIRM is given, an error occurs. Note exit (lowercase) is the standard Tcl command and using it will cause us to skip normal shutdown housekeeping.

INTEGRATE

INTEGRATE qlfdid (assumes BLSODE and entire range)
INTEGRATE qlfdid WITH integrator (assumes entire range)
INTEGRATE qlfdid FROM n1 TO n2 (assumes BLSODE)
INTEGRATE qlfdid FROM n1 TO n2 WITH integrator
Run an integrator on qlfdid. It is best to have solved qlfdid before hand to have good initial values. Requires: n1 < n2, qlfdid be of an integrable type (a refinement of ivp.)

MERGE

MERGE <qlfdid1> [WITH] <qlfdid2>
Execute ARE_THE_SAME on qlfdid1 and qlfdid2 if possible.

OBJECTIVE

OBJECTIVE <qlfid>
Not implemented yet. Semantics of OBJECTIVE that will be

supported are unclear as no OBJECTIVE other than the declarative one is yet supported.

PATCH

PATCH <simname> [OF] <patch_type>
Compiles a simulation PATCH for the given type.

PLOT

PLOT <qlfdid> [filename]
Writes plot data from qlfdid, which must be a plottable instance, to filename. If filename is not given, a file will be written to the Scratch Directory specified in the Utilities window. The file format is controlled by Plot Type in the Utilities window.

PRINT

PRINT <PROBE,DISPLAY>
Prints out the Probe or Display text in view.

PROBE

PROBE qlfdid Is as PROBE ALL qlfid
PROBE ONE qlfdid Exports the item qlfdid to the Probe.
PROBE ALL qlfdid Exports items found in qlfdid matching the current specifications of Visit under the Browser grill. By default, all variables and relations. All exporting is done to the current probe context.

PROTOTYPE

PROTOTYPE <simname>
Creates a prototype of the given simulation name.

READ

READ [FILE,<VALUES,SCRIPT>] <filename>
Loads a file from disk.We search for files in directories (WorkingDirectory):.:ASCENDLIBRARY unless a full path name is given for filename. FILE indicates ASCEND source code (.asc usually.) VALUES indicates variable data written by WRITE VALUES (.values usually.) SCRIPT indicates a file to load at the end of the Script window (.s usually.) If neither VALUES nor SCRIPT found, FILE will be assumed. *Note:* You will get **quite** a spew from the parser if you leave out the SCRIPT or VALUES modifier by accident.

READ_VIRTUAL

READ_VIRTUAL filename
Restore the values from the virtual file back to the instance. See the notes concerning WRITE_VIRTUAL.

REFINE

REFINE <qlfdid> [TO] <type>
Refine qlfdid to the given type if they are conformable.

RESUME

RESUME
RESUME <simname>
Reinvoke compiler on Browser model or on simname to attempt completion pending statements.

RUN

RUN <qlfpid>
Runs the procedure qlfpid as if from the Browser Initialize button.

SHOW

SHOW <filename,LAST>
Invokes the plotter program on the filename given or on the file LAST generated by PLOT. The program to invoke is specified on the Utilities page Plot Program line.

SOLVE

SOLVE <qlfdid> [WITH solvername]
Fires off current solver unless another is specified. Whatever is in the solver window gets displaced by qlfdid. Solvername must be given as it appears on the menu buttons.

WRITE

WRITE <kind> <qlfdid> <file> [args]
WRITE VALUES qlfdid filename.
Write something (what sort of thing indicated by kind) about qlfdid to a file. Args may modify as determined by kind. At present only VALUES is supported. SYSTEM (for solver dump) would be nice.

WRITE_VIRTUAL

WRITE_VIRTUAL qlfid filename
Save the information from the specified qlfid into a virtual file with the give name. At the moment, it only saves the information associated with reals.

set

set tcl_variable value
Set is a Tcl command to assign a value to a Tcl string variable. The value of the variable can be seen with a *puts $tcl_variable* and the value can be passed to other Tcl functions. The primary use of set in ASCEND scripts is to set the base directory for the filenames used with the READ command.

1.1.3 SCRIPT COMMANDS QUICK REFERENCE

ASSIGN set the value of something atomic.

BROWSE export an object to the browser.

COMPILE compile a simulation of a given type.

DELETE	delete a simulation or the type library
DISPLAY*	display something.
DISPVAL	print the instance part of a given simulation.
EXIT	quit ASCEND.
INTEGRATE	run an IVP integrator.
MERGE	perform an ARE_THE_SAME.
OBJECTIVE*	semantics are still unclear.
PATCH	compiles a simulation patch of a given type.
PLOT	create a plot file.
PRINT	print one of the printable windows.
PROBE	export an object to the probe.
PROTOTYPE	Create a prototype of a given simulation.
READ	read in a model, script, or values file.
READ_VIRTUAL	restore the values from the virtual file back to the instance
REFINE	perform an IS_REFINED_TO.
RESTORE*	read a simulation from disk.
RESUME	resume compiling a simulation.
RUN	run a procedure.
SAVE*	write a simulation to disk.
SHOW	call a unix plot program on a file from PLOT.
SOLVE	run the solver.
WRITE	write values in Tcl format to disk.
WRITE_VIRTUAL	save information into a virtual file
set	set a Tcl string variable.
	* This function is not currently implemented.