Task Language Statements

ARRAY *user_var1#* [*size1*]
[...,*user_varN#* [*sizeN*]]
Declares one or more array variables. The brackets around size are required.

BEEP [*duration, pitch*]
Sounds a tone.

CLEARWINDOW [*window_id*]
Clears a conversational window.

CLOSEWINDOW [*window_id*]
Closes a conversational window.

DEFINEWINDOW *window_id size_region*
{ **CBT** | **POPUP** [*title_str*] }
Defines a conversational window.

DO *procedure_name*
Moves task execution to the named procedure.

DO_KEYSTROKES *"key_str"*
Sends keystrokes to the object that currently has the Agent's FOCUS.

EDITBOX *window_id user_var#*
length_int **AT** *origin_point*
Displays a text box in a conversational window.

END
Ends task execution.

FOCUS *object_type title_str*
Directs the Agent to focus on an object.

GOTO *label_name*
Moves task execution to the named label.

IF *conditional_statement*
if-statements
[ELSE
else-statements]
ENDIF
Conditional IF statement with optional ELSE clause.

INPUT *user_var#* [**LENGTH**
length_int][*prompt_string*]
Displays an input conversational window.

LABEL *label_name*
Defines a label to be referenced by a GOTO statement.

LOCATE [*window_id*] [*col_int row_int*]
Positions the pointer in a conversational window.

MENU_COMMAND *"command_name_str"*
[*command_id*]
Sends a menu command to the object that currently has the Agent's FOCUS.

MESSAGE *user_var#* [*prompt_str*] [**HAND** | **EXCLAMATION_POINT** | **QUESTION** | **STOP**] {**OK** | **OKCANCEL** | **RETRYCANCEL** | **ABORTRETRYIGNORE** | **YESNO** | **YESNOCANCEL** | **USER** *button_str*]
Displays a message conversational window.

ON ERROR DO *procedure_name*
Sets a trap for task execution errors.

ON ESCAPE DO *procedure_name*
Sets a trap for interrupting a running task.

ON TIMEOUT DO *procedure_name*
Sets a trap for processing timeout errors.

OPENWINDOW [*window_id*]
Opens a conversational window.

OUTPUT [*window_id*] [*output_str1; output_str2; output_str* [;]]
Displays lines of text in a conversational window.

PAUSE *time_int*
Pauses task execution.

PERFORM_TASK *task_id* [*parameter1, ..., parameterN*] [**RECEIVE** *user_var1#, ..., user_varN#*]
Performs an Agent task with internal object ID *task_id*. Parameters are passed to the called task. Variables specified after RECEIVE get the return values from the called task.

PROCEDURE *procedure_name*
statements
ENDPROC
Groups a set of commands into a procedure.

PUSHBUTTON *window_id user_var#*
text_str **AT** *location_point*
[DEFAULT][ESCAPE]
Displays a button in a conversational window.

RETURN [*return_value1, ..., return_valueN*]
Returns task execution to the line immediately following the call to a procedure or to a task. Return values cannot be specified inside procedures.

SCREEN *width_int height_int*
Switches the mapping of conversational window coordinates from physical to logical coordinates.

SET ERROR {**ON** | **OFF**}
Sets error trapping on or off.

SET ESCAPE {{[**TO**] [**ALT**] [**CTRL**] *key_str*} | **OFF**}
Sets escape trapping on or off. Defines the escape key.

SET RATE {{[**TO**] *time_int*} | **OFF**}
Sets the rate of task execution.

SET TIMEOUT {{[**TO**] *time_int*} | **OFF**}
Sets the length of time a task will wait for a command to execute. Default is 120 seconds.

TASK [*user_var1#* [['=' *literal_val1*] | [*size1*]] ... [,*user_varN#* [['=' *literal_valN*] | [*sizeN*]]]
statements
ENDTASK
Defines the boundaries of a task. TASK command may optionally include formal parameters. The brackets around size are required.

TITLEWINDOW [*window_id*] *title_str*
Changes the title of a defined conversational window.

WAIT
Suspends task execution until an event is trapped (e.g. clicking a button in a conversational window).

WHILE *conditional_statement*
statements
ENDWHILE
Conditional while statement.

Task Language Functions

ABS (*numeric*)
Returns the absolute value of *numeric*.

ASC (*string*)
Returns an int containing the ASCII value of the first character in *string*.

CENTER (*region*)
Returns the center point of the supplied *region*.

CHR (*int*)
Returns a one-character string whose ASCII value is *int*.

DDEGETDATA (*item_str, application_str, topic_str*)
Returns a string containing data received from an application via DDE. Returns a null string if an error occurred.

DDESENDCOMMAND (*command_str, application_str, topic_str*)
Sends a command to an application via DDE. Returns 1 if successful, 0 otherwise.

DDESENDDATA (*item_str, value_str, application_str, topic_str*)
Sends data to an application via DDE. Returns 1 if successful, 0 otherwise.

DLLCALL (*dll_id, function_name_str, return_type* [, *parm1_type, parm1, parm2_type, ...*])
Calls a function in the specified DLL. Returns the return value from the DLL's function. *return_type#* should be VOID! (0) for functions that don't return a value.

DLLFREE (*dll_id*)
Frees a previously loaded DLL. Always returns 0.

DLLLOAD (*dll_filename_str*)
Returns an internal ID that identifies the DLL library just loaded. Returns 0 if the file is not a library, and returns numbers less than 32 for other errors.

EDITBOX (*user_var#*)
Returns a string containing the contents of the editbox identified by *user_var#*.

FIND (*str1, str2, index*)
Returns an int containing the number of characters beyond *index* (3rd parm) where *str1* starts in *str2*. If *str1* is not found then -1 is returned.

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

FINDNEXTOBJECT ( )
Returns an internal ID that identifies the next object that was found by the previously called FINDOBJECT function. Returns O when there are no more found objects.

FINDOBJECT ([*text_id, object_title*])
Returns an internal ID that identifies the first object in the system that is found that matches the specified *text_id* and *object_title*. The ID for any additional objects found by this function call may be retrieved via FINDNEXTOBJECT. Returns 0 if no objects are found. If no parameters are specified, the ID for the object that currently has the Agent's FOCUS will be returned. If *object_title* = " " then all objects of type *text_id* will be found.

GETCLIPBOARDTEXT ( )
Returns a string containing the TEXT data on the clipboard.

GETFIRSTOBJECT ([*text_id, object_title*])
Returns an internal ID that identifies the first object found in the specified container object. If no parameters are provided, the object with the Agent's FOCUS is searched. Zero is returned if no objects are found.

GETNEXTOBJECT ( )
Returns an internal ID that identifies the next object found; used in conjunction with **GETFIRSTOBJECT** and **GETOBJECT**. Zero is returned when all objects have been enumerated.

GETOBJECT ([*object_id, get_type*])
Enumerates a specified set of objects. Returns an internal ID that identifies the first object in the set. The internal ID for any additional objects enumerated by this function may be retrieved via GETNEXTOBJECT. Returns 0 if no objects are found in the set. *object_id* is the internal ID of the object whose parents or children are enumerated. *get_type* is the type of enumeration to perform and is either TOOLS!, CHILDREN!, PARENTS!, HIDDENCHILDREN!, or HIDDENPARENTS! If no parameters are specified, the children of the object that currently has the Agent's focus are enumerated. If only the *object_id* parameter is specified, the children of that object are enumerated.

GETPTRVALUE (*pointer_name, byte_offset, value_type*)
Returns data referenced by *pointer_name* starting from *byte_offset* bytes. The returned value is of *value_type* type.

GETTASKOBJECT ( )
Returns the internal ID that corresponds to the performing task itself.

HEIGHT (*region*)
Returns the height of the supplied *region*.

IN (*point, region*)
Returns a 1 if *point* is in *region*, else returns 0.

INT (*real*)
Returns an int containing the integer portion of *real*.

LEFT (*string, int*)
Returns *int* chars from the left of *string*.

LEN (*string*)
Returns an int containing the number of chars in *string*.

MAKEBYTE (*number*) **MAKEINT** (*number*)
MAKEDOUBLE (*number*) **MAKELONG** (*number*)
MAKEFLOAT (*number*)
Returns the number as the specified type, doing any necessary truncation.

MID (*string, int1, int2*)
Returns *int2* chars from position *int1* in *string*.

MOD (*int1, int2*)
Returns the modulus of *int1* divided by *int2*.

OPENOBJECT (*object_id, open_type*)
Opens the object specified by internal ID *object_id*. Returns 1 if the open is successful, else it returns 0. *open_type* is either OPEN! or OPEN_SHIFTED!.

SETPTRVALUE (*pointer_name, byte_offset, value_type, newvalue*)
Places *newvalue* into the memory buffer referenced by *pointer_name* starting from *byte_offset* bytes. The data is of *value_type* type. Returns 1 if successful, 0 otherwise.

VAR (*variable#*)
Returns the variable ID of variable. This variable ID can be used in DLLCALL() to allow the Agent to modify a variable.

Object Attribute Functions

OBJxxx (*object_id*)
where ***xxx*** is described below and *object_id* identifies the target object. *object_id* should have been retrieved via GETFIRSTOBJECT, GETNEXTOBJECT, GETOBJECT, FINDOBJECT, FINDNEXTOBJECT, or GETTASKOBJECT. If no parameter is specified then the object with the Agent's FOCUS is assumed.

Function	Return Value
OBJCLASS	string, class name
OBJCOMMENTS	string, comments
OBJCREATED	int, creation date
OBJCREATOR	string, creator's name
OBJCRMAILADDR	string, creator's mail address
OBJLASTWRITER	string, last writer's name
OBJLINKSDOWN	int, # of child objects
OBJLINKSUP	int, # of parent objects
OBJLWMAILADDR	string, last writer's mail address
OBJMODIFIED	int, modification date
OBJTITLE	string, object's title
OBJTYPE	string, object's type or TEXT_ID

ORIGIN (*region*)
Returns the origin point of *region*.

PUTCLIPBOARDTEXT (*string*)
Puts *string* on the clipboard in TEXT format. Returns 1 if successful, else returns 0.

RIGHT (*string, int*)
Returns *int* characters from the right side of *string*.

STR (*numeric*)
Returns a string representing *numeric*.

SYSDATE ( )
Returns an int that is the current date in the internal NewWave system date format.

SYSDATETOSTR (*date_int*)
Returns a string representation of *date_int*. If no parameter is supplied then the string for the current date is returned.

SYS_ERROR ( )
Returns an int containing the error number of the last task execution error.

SYSTIME ( )
Returns an int that is the current time in the internal NewWave system time format.

SYSTIMETOSTR (*time_int*)
Returns a string representation of *time_int*. If no parameter is supplied then the string for the current time is returned.

STRTOSYSDATE (*date_str*)
Returns an int that corresponds to *date_str*.

STRTOSYSTIME (*time_str*)
Returns an int that corresponds to *time_str*.

TESTBUTTON (*button_id*)
Returns 1 if the button has been pushed, else 0.

VAL (*string*)
Returns a numeric for *string*. Returns 0 if *string* does not represent a valid numeric.

WIDTH (*region*)
Returns the width of *region*.

X (*point*)
Returns the X coordinate of *point*.

Y (*point*)
Returns the Y coordinate of *point*.

Syntax Key:

[ ] = optional keyword or parameter

{ } = required choice of one or another keyword or parameter

{ [= required choice of one or more keywords or parameters

P/N D1704-90128