ODT Vision ®

3rd Parry Program Calls

6.0

ODT Vision ®

3rd Party Program Calls Guide

Table of Contents

Calling other programs

val = InitializeProgram "ODTVISIONCall.ODTVISIONClass" [,Server]
InitializeProgram "ODTVISIONCall.ODTVISIONClass" [, Server]

Initializes an externally called program.

Used if the externally called program needs parameters or has long running functions

Only one externally called program can be active at a time. Use the TerminateProgram command to end the program.

If a variable is to be returned, it will return one of the following:
0-Called
n-Error

See the chapter on creating an externally callable program for more information.

Examples:

```
; This first example would execute the "DATE"
; command in the VB6 program and returns the
; current date in the "val" variable.

InitializeProgram "ODTVISIONCall.ODTVISIONClass"
val = CallProgram "DATE"
TerminateProgram
```

```
                    ; This next example shows how parameters are
                    ; passed to the VB6 program.

                    ; The variable "CustNo" is passed as the
                    ; "Customer" parameter and the variable
                    ; "OrderTot" is passed as the "OrderAmount"
                    ; parameter.

                    ; Then the "CreditLimit" command is called
                    ; in the VB6 program. The VB6 program then
                    ; takes the Customer parameter checks it
                    ; against a database or another system and
                    ; returns the credit limit of the customer
                    ; in the "CrLim" variable of the script.

                    ; Then the "CreditApproval" command is
                    ; called in the VB6 program.  The program
                    ; takes the Customer number and the order
                    ; amount and checks an Accounts Receivable
                    ; system to determine if the customer
                    ; order will be held due to a credit
                    ; limit problem.

                    ; The script then speaks either "Your
                    ; order was approved", or "Your order
                    ; was not approved, you are over your
                    ; credit limit of $20,000.00"

InitializeProgram "ODTVISIONCall.ODTVISIONClass"

                    ProgramParameter "Customer", CustNo
                    ProgramParameter "OrderAmount", OrderTot

                    CrLim = CallProgram "CreditLimit"
                    Approved = CallProgram "CreditApproval"
                    TerminateProgram

                    If Approved then
                       Speak "Approved.voc"
                    else
                       Speak "NotApprove.voc"
                       Speak "OverCredit.voc"
                       SpeakDollars CrLim
                     Endif
```

Example of long running code

```
; This last example shows how long running
; code in the VB6 program can be executed
; without tying up the script. The script
; can check periodically to see if the
; code has completed.

; The variable "Cardno" is passed as the
; "CreditCard" parameter and the variable
; "ExpDate" is passed as the "Expiration"
; parameter.

; Then the "CCStart" command is called
; in the VB6 program. The VB6 program then
; takes the Credit card information and
; validates it to a credit card system.

; The script waits 5 seconds, speaks
; "Please wait...checking credit card"
; then executes the "CCDone" command
; to the if the processing is done.

; If the "CCDone" command returns 0,
; credit card processing is not complete
; so look back and wait again.

; If the "CCDone" command returns 1,
; the credit card was approved. If 2
; is returned, the charge was rejected.

; The script then speaks either
; "Your card was approved", or
; "Your card was declined"


InitializeProgram "ODTVISIONCall.ODTVISIONClass"

ProgramParameter "CreditCard", CardNo
ProgramParameter "Expiration", ExpDate

CallProgram "CCStart"

LOOP:
Wait 5
```

```
Done = CallProgram "CCDone"

IF Done = 0 then
   Play "WAIT.VOC"
   Goto Loop
ENDIF

TerminateProgram
If Done = 1 then
   Speak "Approved.voc"
ELSE
   Speak "NotApprove.voc"
Endif
```

val = CallProgram command
CallProgram command

Performs a command in an externally called program.

You must first initialize the externally called program using the InitializeProgram command.

Used if the externally called program needs parameters or has long running functions

A variable can be returned from a CallProgram command.

See the chapter on creating an externally callable program for more information.

val = TerminateProgram
TerminateProgram

Terminates an externally called program.

You must first initialize the externally called program using the InitializeProgram command.

Used if the externally called program needs parameters or has long running functions

If a variable is to be returned, it will return one of the following:
0-Called
n-Error

See the chapter on creating an externally callable program for more information.

ProgramParameter parameter, data

Passes a parameter to an externally called program.

You must first initialize the externally called program using the InitializeProgram command.

You must pass the parameter name and the parameter value

See the chapter on creating an externally callable program for more information.

I

```
val = CallProgram command, "ODTVISIONCall.ODTVISIONClass" [,Server]
CallProgram command, "ODTVISIONCall.ODTVISIONClass" [,Server]
```

Calls an externally called program, executes a command without parameters, and terminates the program.

Used if the externally called program does not need parameters and does not have long running functions

If a variable is to be returned, it will return either an error or a return parameter.

See the chapter on creating an externally callable program for more information.

Example:

```
; This example shows how to use the shortcut
; version of the Execute command.  It can be
; used if the VB6 program can return the data
; quickly, and no parameters are needed.

; This example executes the "TIME" command
; in the VB6 program and returns the time
; in the "val" variable.

val = CallProgram "TIME", ODTVISIONCall.ODTVISIONClass"
```

val = ExecuteProgram program [, NOWAIT/timeout] [, focus]
ExecuteProgram program [, NOWAIT/timeout] [, focus]

Calls an external program.

“program” is a string value that contains the program name to call, and parameters required by the program.

The second parameter is optional. If a timeout number is specified, the system will wait for the externally called program for the number of seconds specified. If the NOWAIT parameter allows the script to continue to run even if the called external program has not completed. If the second parameter is not specified, the script will wait indefinitely for the called program to complete.

If a variable is to be returned, it will return either an error or a return parameter.

Possible Errors returned are:

-1	Program call ended in error.
-2	Program call timed-out.

See the chapter on creating an externally callable programs for more information.

Focus is the optional third parameter that controls how the externally called program is displayed (if it has a user interface). If not specified, focus style 1 will be used.

Example:

```
; Call NOTEPAD and wait for it to complete.

ExecuteProgram “Notepad.exe”


; Call FIXFILE passing the file to fix as a
; parameter and do not wait for it to complete.

ExecuteProgram “FixFile.exe e:\File1.dat”, NOWAIT
```

Focus number	Effect on called program
0	Externally called program window is hidden and focus is passed to the hidden window.
1	Externally called program window has focus and is restored to its normal size and position.
2	Externally called program window is displayed as an icon with focus.
3	Externally called program window is maximized with focus.
4	Externally called program window is restored to its normal size and position. The currently active window remains active.
6	Externally called program window is displayed as an icon. The currently active window remains active.