

Sample Scripts

[{bmc_j_a.bmp}](#) [{bmc_j_b-c.bmp}](#) [{bmc_j_d.bmp}](#) [{bmc_j_e.bmp}](#) [{bmc_j_f.bmp}](#) [{bmc_j_g.bmp}](#) [{bmc_j_h-k.bmp}](#) [{bmc_j_l.bmp}](#) [{bmc_j_m.bmp}](#) [{bmc_j_n.bmp}](#) [{bmc_j_o.bmp}](#) [{bmc_j_p.bmp}](#) [{bmc_j_q.bmp}](#) [{bmc_j_r.bmp}](#) [{bmc_j_s.bmp}](#) [{bmc_j_t.bmp}](#) [{bmc_j_u-v.bmp}](#) [{bmc_j_w-z.bmp}](#)

Note: In almost all of the sample scripts, the `sub main()` and `end sub` statements are commented out because the **DCL Editor** includes these commands in each new script. These commands are not commented out in sample scripts where functions are defined or multiple subroutines used.

#4 A

[Abs](#)

[AddIni](#)

[And](#)

[AnswerBox](#)

[AppActivate](#)

[AppClose](#)

[AppFileName\\$](#)

[AppFind](#)

[AppGetActive\\$](#)

[AppGetPosition](#)

[AppGetState](#)

[AppHide](#)

[AppList](#)

[AppMaximize](#)

[AppMinimize](#)

[AppMove](#)

[AppRestore](#)

[AppSetState](#)

[AppShow](#)

[AppSize](#)

[AppType](#)

¹ Sample_Scripts

² Sample Scripts

³ Sample scripts;Scripts, sample;Example scripts

⁴ sam_toc_a

ArrayDims

ArraySort

Asc

AskBox\$

AskPassword\$

Atn

#5 **B -- C**

Beep

ButtonEnabled

ButtonExists

Call

CDbl

ChDir

ChDrive

Chr\$

CInt

Clipboard\$

ClipboardClear

CLng

CSng

CStr

CurDir\$

#6 **D**

Date\$ Statement

Date\$ Function

DateSerial

DateValue

Day

DCLHomeDir\$

⁵ sam_toc_bc

⁶ sam_toc_d

DCLOS\$

DCLVersion\$

DDE Commands

Declare

DEFtype

DesktopCascade

DesktopSetColors

DesktopSetWallpaper

DesktopTile

Dialog Commands

OK, Cancel, and Push Buttons

A Comprehensive Dialog Example

Dialog Statement and Function

Dim

DirExists

Dir\$

DiskDrives

DiskFree

Do...Loop

DoEvents

DoKeys

^{#7} **E**

EnableStopScript

End

Environ\$

Err

Error Statement and Function

Exclusive

Exit Commands

Exp

^{#8} **F**

[FileCopy](#)
[FileDateTime](#)
[FileDirs](#)
[FileExists](#)
[FileLen](#)
[FileList](#)
[FileParse](#)
[FileType](#)
[FindFile\\$](#)
[Fix](#)
[For...Next](#)
[FreeFile](#)

^{#9} **G**

[GetAttr](#)
[GetEnv](#)
[GoSub](#)
[Goto](#)

^{#10} **H -- K**

[Hex\\$](#)
[HLine](#)
[Hour](#)
[HPage](#)
[HScroll](#)
[If...Then...Else](#)
[InputBox\\$](#)
[Input/Output Commands](#)
[InStr](#)
[Int](#)

⁸ sam_toc_f

⁹ sam_toc_g

¹⁰ sam_toc_hk

Item\$ and ItemCount

Kill

#11 **L**

LBound

LCase\$

Left\$

Len

Let

Line\$

LineCount

Log

LTrim\$

#12 **M**

MCI

Menu

Message Commands

Mid\$

Minute

MkDir

Mod

Month

MsgBox

#13 **N**

Name

NetAttach

NetConnectDrive

NetDetach

¹¹ sam_toc_l

¹² sam_toc_m

¹³ sam_toc_n

NetDirectoryRights
NetDisconnectDrive
NetGetDirectoryRights
NetMemberOf
NetStationID
NetUserName
NetworkStatus
Not
Now
Null

#14 **O**

Oct\$
On Error
OpenFileName\$
Option Base
Or

#15 **P**

PI
PopupMenu
Print
Print #
PrinterGetOrientation
PrinterSetOrientation
PrintFile

#16 **Q**

Queue Commands

¹⁴ sam_toc_o

¹⁵ sam_toc_p

¹⁶ sam_toc_q

#17 **R**

Random
Randomize
ReadINI\$
ReadINISection
ReDim
REM
RefreshIni
Reset
RestoreEnv
Resume
Return
Right\$
Rmdir
Rnd
RTrim\$

#18 **S**

SaveEnv
SaveFileName\$
Second
Select...Case
SelectBox
SendKeys
SetAttr
SetEnv
Sgn
Shell
Sin
Sleep
SleepUntil

¹⁷ sam_toc_r

¹⁸ sam_toc_s

Snapshot
Space\$
Sqr
Stop
StrComp
Str\$
String\$
SystemFreeMemory
SystemFreeResources
SystemMouseTrails
SystemRestart
SystemTotalMemory
SystemWindowsDirectory\$
SystemWindowsVersion\$

#19 **T**

Tan
Time Statement
Time Function
Timer
TimeSerial
TimeValue
Trim\$

#20 **U -- V**

UBound
UCase\$
Val
Viewport Commands
VLine
VPage

¹⁹ sam_toc_t

²⁰ sam_toc_uv

VScroll

#21 **W -- Z**

WaitForTaskCompletion

Weekday

While...Wend

WinActivate

WinClose

WinFind

WinList

WinMaximize

WinMinimize

WinMove

WinRestore

WinSize

Word\$

WordCount

WriteINI

Xor

Year

#22 S23 'DDE Example {bmc no_dos.bmp}

{bmc xc_copy2.bmp} {bmc xc_copy.bmp} {bmc xc_print.bmp} {bmc xc_close.bmp}

```
'sub main()
'   This script illustrates the use of the DDE commands.
'   The current shell is used to demonstrate this.
'   NetTools Applications Manager is assumed to be the
'   shell.
'
'   Demonstrated in this script are:
'   DDEInitiate, DDETimeout, DDERequest, DDETerminate,
'   DDEExecute, and DDETerminateAll

dim channel as integer
dim winshell as string
dim topic as string
dim regstr as string
dim grouplist() as string
dim i as integer
dim nl as integer
dim selitem as integer
dim groupname as string

winshell = "AppMan"
topic = "AppMan"

'   initiate a DDE conversation with the shell
channel = DDEInitiate(winshell,topic)
DDETimeout 5000           'set time out to 5 seconds (default is 10000 milliseconds)
if channel = 0 then
```

²² CMD_DDE_Example

²³ DDE_Example

```
        msgbox "Unable to initiate DDE link with "+winshell+      ". Script will end now."
    end
end if
```

```
'    Get a list of the groups in the shell
reqstr = DDERequest$(channel,"Groups")
'    Now the group names have to be placed into an array
'    for use in a Selection Box
nl = LineCount(reqstr) - 1
redim grouplist(1 to nl)
for i = 1 to nl
    grouplist(i) = Line$(reqstr,i)
next i
```

```
'    Now that we have the groups in a usable list,
'    allow the user to make selections to
'    view the items in the group
```

viewgroups:

```
    selitem = SelectBox("Groups",winshell+" Groups:",grouplist)
    if selitem = 0 then goto shutdown
```

```
'    Get the group name, and enumerate the item names
groupname = grouplist(selitem)
reqstr = DDERequest$(channel,groupname)
'    Display the items in a message box
msgbox reqstr
goto viewgroups
```

shutdown:

```
    DDETerminate channel
    msgbox "Now we'll create a group, and then delete it."
    channel = DDEInitiate(winshell,topic)
    DDEExecute channel,"[CreateGroup(TestGrp)]"
```

```
DDEExecute channel,"[ShowGroup(TestGrp,1)]"  
msgbox "Look at your shell now and see the new group before we delete it."  
DDEExecute channel,"[DeleteGroup(TestGrp)]"  
DDETerminateAll  
end  
'end sub
```

Dialog Examples

{bml j_bullet.bmp}OK, Cancel, and Push Buttons

{bml j_bullet.bmp}A Comprehensive Dialog Example

{bml j_bullet.bmp}Dialog Statement and Function Example

²⁴ CMD_Dialog_Examples

²⁵ Dialog_Examples

²⁶ Dialog_examples

#27 #28 'OK, Cancel, and Push Buttons

```

    {bmc_xc_copy2.bmp}  {bmc_xc_copy.bmp}  {bmc_xc_print.bmp}  {bmc_xc_close.bmp}

'sub main()
    'Example of a simple Dialog box
    Begin Dialog UserDialog 16,32,233,105, "Title"
        OKButton 176,7,43,14
        CancelButton 176,30,44,14
        Text 6,34,158,8, "This is a sample text field."
        PushButton 176,52,44,14, "Other #1"
        PushButton 177,76,43,14, "Other #2"
    End Dialog

    dim adialog as UserDialog

    returncode = Dialog(adialog)

    '   returncode contains the value of the button selected
    '   OK = -1
    '   Cancel = 0
    '   Other buttons are numbered 1 to N where N is the number
    '   of other buttons on the dialog
    select case returncode
        case -1
            msgbox "OK pressed."
        case 0
            msgbox "Cancel pressed."
        case else
            msgbox "Other button #" + str$(returncode) + " pressed."
    end select
'end sub
```

²⁷ CMD_Buttons_Example

²⁸ OK, Cancel, and Push Buttons

#29 S30 'A Comprehensive Dialog Example {bmc no_dos.bmp}

{bmc xc_copy2.bmp} {bmc xc_copy.bmp} {bmc xc_print.bmp} {bmc xc_close.bmp}

```
'sub main()
'Example showing the entire process of defining, displaying,
'and interpreting the input from a dialog box.

'Initialization
'-----
'ListBox$ and ComboBox are single-dimensional arrays to hold the
'contents of the list and combo box in the dialog box.
Dim ListBox1$() as string
Dim ComboBox1$() as string

Dim stf$ as string      ' static text variable
stf$ = "123456789012345678901234567890"

'In this example, the Text field uses the str$
'variable to get its field name.  This can
'be done for any of the dialog control types except:
'OKButton, CancelButton, TextBox
,
'Note that the str$ variable must have a
'value assigned BEFORE you can define the dialog box.

'Define dialog box
'-----
Begin Dialog UserDialog 16,32,304,168, "Sample Dialog Box"
    OKButton 251,9,44,14
    CancelButton 252,30,44,14
```

²⁹ CMD_Dialog_Comprehensive_Example

³⁰ A Comprehensive Dialog Example


```

    PushButton 252,51,44,14, "Pushbutton1"
    PushButton 252,73,44,14, "PushButton2"
    GroupBox 13,9,84,59, "Sample Group Box"
    OptionGroup .OptionGroup1
        OptionButton 21,24,65,14, "Option 1"
        OptionButton 21,44,66,14, "Option 2"
    CheckBox 15,78,79,14, "Sample Checkbox", .CheckBox1
    Text 14,105,79,8, stf$
    TextBox 16,120,81,12, .TextBox1
    ListBox 114,14,120,48, ListBox1$, .ListBox1
    ComboBox 113,68,120,84, ComboBox1$, .ComboBox1
End Dialog

'Prepare to display dialog box.
'-----

'Declare the dialog type using Dim ... as UserDialog command
Dim aSampleDialog as UserDialog

'Load the list box and combo box arrays with app window names
AppList ComboBox1$
AppList ListBox1$

'Load the text box with an initial value
aSampleDialog.TextBox1 = "123456789012345678901234567890"
'Note that you reference a dialog box field as follllows:
'    <DialogBoxName>.<FieldName>

'Display the Dialog
'-----

a% = Dialog(aSampleDialog)      'Returns integer indicating button chosen

'Interpret user input
'-----

```

```

'This example builds a string describing the contents of the
'dialog box when the user finished making changes.
crlf$ = chr$(13)+chr$(10)
dlgstr$ = "Button pushed = "
'Determine button pushed.
select case a%
    case -1
        dlgstr$ = dlgstr$ + "OK"
    case 0
        dlgstr$ = dlgstr$ + "Cancel"
    case 1
        dlgstr$ = dlgstr$ + "PushButton1"
    case 2
        dlgstr$ = dlgstr$ + "PushButton2"
end select
dlgstr$ = dlgstr$ + crlf$
'Determine new value of each dialog box component
dlgstr$ = dlgstr$ + "Option = " + str$(aSampleDialog.OptionGroup1) + crlf$
dlgstr$ = dlgstr$ + "Checkbox = " + str$(aSampleDialog.CheckBox1) + crlf$
dlgstr$ = dlgstr$ + "TextBox = " + aSampleDialog.TextBox1 + crlf$
dlgstr$ = dlgstr$ + "ListBox = " + ListBox1$(aSampleDialog.ListBox1) + crlf$
dlgstr$ = dlgstr$ + "ComboBox = " + aSampleDialog.ComboBox1
'Display the dlgstr$ string in a message box.
msgbox dlgstr$

'end sub

```

'Exit Statement Examples

{bmc xc_copy2.bmp} {bmc xc_copy.bmp} {bmc xc_print.bmp} {bmc xc_close.bmp}

```
function testfunc (a as integer) as integer
    testfunc = 0
    if a = 2 then
        Exit Function
    end if
    msgbox "no premature exit from testfunc"
    testfunc = 100
end function
```

```
sub testsub (a as integer)
    if a = 4 then
        Exit Sub
    end if
    msgbox "no premature exit from testsub"
end sub
```

```
sub main()
    'examples of EXIT statements
    for i% = 1 to 10
        if i% >= 1 and i% <= 2 then
            b% = testfunc(i%)
            msgbox str$(b%)
        end if
        if i% >= 3 and i% <= 4 then
            testsub(i%)
        end if
        if i% = 7 then
```

³¹ CMD_Exit_Statement_Examples

³² Exit Examples

```
        Exit For
    else
        msgbox "no exit yet "+str$(i%)
    end if
next i%
msgbox "we just exited from the for loop"
end sub
```

#33 S34 'Input/Output Example {bmc no_dos.bmp}

{bmc xc_copy2.bmp} {bmc xc_copy.bmp} {bmc xc_print.bmp} {bmc xc_close.bmp}

```
'sub main()
'Example of file input/output functions
'Open, Close, Eof, FileAttr, Line Input#, Lof
'Because of the use of ViewPort... commands,
'this script should not be run under DOS.

dim fileno as integer
dim flen as integer
dim fileatr as integer
dim osfile as integer
dim inline as string
dim linedesc as string
dim linemult as integer

ViewPortOpen "AUTOEXEC.BAT"
ViewPortClear
fileno = FreeFile()           'get next available file number
Open "C:\AUTOEXEC.BAT" for input as fileno
flen = Lof(fileno)           'get length of file
Print "File Autoexec.Bat - Length"+str$(flen)
fileatr = FileAttr(fileno,1)
osfile = FileAttr(fileno,2)
Print "File is opened for ";
Select Case fileatr
    Case 1
        Print "Input";
    Case 2
```

³³ CMD_Input_Output_Example

³⁴ Input/Output Example

```

        Print "Output";
    Case 8
        Print "Append";
End Select
Print " and has an Operating System Filehandle of"+str$(osfile)
Print string$(50,"-")
while not eof(fileno)
    Line Input #fileno, inline
    Print inline
wend
Close fileno
msgbox "Click OK when you're finished viewing the file."

ViewPortClear
Print "Creating a simple output file."
fileno = FreeFile()
Open "C:\JUNK.TXT" for output as fileno
for i% = 1 to 10
    Write #fileno,"Line"+str$(i%),i% * 10
next i%
Close fileno
Print "Reading created file."
Print string$(50,"-")
fileno = FreeFile()
Open "C:\JUNK.TXT" for input as fileno
while not eof(fileno)
    print "Seek Position"+str$(Seek(fileno))
    print "File Position"+str$(loc(fileno))  'print the file position
    Input #fileno,linedesc,linemult
    Print linedesc+", Value ="&str$(linemult)
wend
Close fileno
msgbox "Click OK when you're finished viewing the file."

```

```

ViewPortClear
Print "Display Autoexec.Bat again...this time in a different way."
fileno = FreeFile()
Open "C:\AUTOEXEC.BAT" for input as fileno
flen = lof(fileno)
aexec$ = Input$(flen,fileno) 'read the entire file at once
print aexec$
Close fileno
msgbox "Click OK when you're finished viewing the file."

ViewPortClear
Print "Now create Junk.Txt using Print# instead of Write#"
fileno = FreeFile()
Open "C:\JUNK.TXT" for output as fileno
for i% = 1 to 10
    Print #fileno,i%,"test","more"
next i%
Close fileno
fileno = FreeFile()
Open "C:\JUNK.TXT" for input as fileno
stuff$ = Input$(lof(fileno),fileno)
Close fileno
print stuff$
msgbox "Click OK when you're finished viewing the file."
ViewPortClose
'end sub

```

#35 S36 'Message Example {bmc no_dos.bmp}

{bmc xc_copy2.bmp} {bmc xc_copy.bmp} {bmc xc_print.bmp} {bmc xc_close.bmp}

```
'sub main()
  'Example of MsgOpen, MsgSetText, MsgSetThermometer, and MsgClose

  MsgOpen "Message Box - 0% complete.",0,TRUE,TRUE
  on error goto cancelpressed
  for i% = 1 to 100
    sleep 100
    msg$ = "Message Box -"+str$(i%)+"% complete."
    MsgSetText msg$
    MsgSetThermometer i%
  next i%
  MsgClose
  msgbox "Finished Normally"
end

cancelpressed:
  MsgClose
  on error goto 0
  msgbox "Cancel was selected."
end

'end sub
```

³⁵ CMD_Message_Example

³⁶ Message Example

'Queue Example {bmc no_dos.bmp}

{bmc xc_copy2.bmp} {bmc xc_copy.bmp} {bmc xc_print.bmp} {bmc xc_close.bmp}

```
'sub main()
    'Example of QUE commands
    dim npWnd as integer
    dim npname as string

    npname = AppFind$("Notepad")
    AppActivate npname
    npWnd = WinFind(npname)
    QueEmpty      'empty the queue
    QueKeyDn "D"
    QueKeyUp "a"
    QueKeys "vid"
    QueMouseClicked VK_RBUTTON,1,1
    QueMouseDown VK_RBUTTON,1,1
    QueMouseDblClk VK_RBUTTON,1,1
    QueMouseDblDn VK_RBUTTON,1,1
    QueMouseDn VK_RBUTTON,1,1

    QueMouseMove 100,100
    QueMouseUp VK_RBUTTON,100,100
    QueSetRelativeWindow npWnd
    QueFlush TRUE
    QueEmpty
'end sub
```

