

## Informationen zur Call.Dll

Die Call.Dll ist entwickelt als Zusatz zu Profan vor Version7.x

Die 2 Funktionen der Dll erweitern Profan ab Version5.0 -32Bit um den Call-Befehl.

Es koennen sowohl OLE-Objekte angesprochen sowie auch API-Funktionen aufgerufen werden.

Es koennen bis zu 15 Parameter an Objekte und Funktionen uebergeben werden.

[Funktionen](#)

[Profan-Beispiele](#)

[Lizenz](#)

# Funktionen

- 1 CallFunktion
- 1 CallObj

*Deklaration für Profan :*

```
DEF CallFunktion(4) ! "Call", "CallFunktion"  
DEF CallObj(4) ! "Call", "CallObj"
```

## CallFunktion

Parameter 1 = Funktionsname (Zeiger auf String -> Bereichsvariable)

Ist dem String eine Raute # vorgestellt wird, so ist der Funktionsname eine Nummer

Parameter 2 = DLL (Zeiger auf String -> Bereichsvariable)

Parameter 3 = Anzahl der Funktionsparameter (LongInt) max. 15

Parameter 4 = Funktionsparameter ( Bereichsvariable )

BEACHTTE :

Jeder Funktionsparameter ist ein LongInt, dass heisst soll ein String übergeben werden, so kann hier nur ein Zeiger auf diesen übergeben werden ( Bereichsvariable) !

## CallObj

Parameter 1 = Zeiger auf ein Objekt das mit CoCreateInstance(..) erzeugt wurde (Bereichsvariable)

Parameter 2 = Befehl ( LongInt )

Parameter 3 = Objektparameter ( Bereichsvariable )

BEACHTTE :

Jeder Objectparameter ist ein LongInt, dass heisst soll ein String übergeben werden, so kann hier nur ein Zeiger auf diesen übergeben werden ( Bereichsvariable) !

Parameter 4 = ParameterAnzahl der Objektparameter (LongInt) max. 15

## Beispiel 1

*Das eigene Fenster aus der Taskbar entfernen oder anzeigen*

```
DEF CallObj(4) ! "Call", "CallObj"
```

```
DEF CoInitialize(1) ! "OLE32", "CoInitialize"
```

```
DEF CoUnInitialize(0) ! "OLE32", "CoUninitialize"
```

```
DEF CoCreateInstance(5) ! "OLE32", "CoCreateInstance"
```

```
Declare HrInit&, AddTab&, DeleteTab&, Release&
```

```
Let HrInit& = 12
```

```
Let AddTab& = 16
```

```
Let DeleteTab& = 20
```

```
Let Release& = 8
```

```
Declare Param#
```

```
Declare TbObj#
```

```
DECLARE CLSID_TaskbarList#
```

```
DECLARE IID_ITaskbarList#
```

```
Proc HideMyWindow
```

```
Parameters Flag&
```

```
DIM CLSID_TaskbarList#, 16
```

```
DIM IID_ITaskbarList#, 16
```

```
'GUID CLSID_TaskBarlist
```

```
LONG CLSID_TaskbarList#, 0 = $56FDF344
```

```
WORD CLSID_TaskbarList#, 4 = $FD6D
```

```
WORD CLSID_TaskbarList#, 6 = $11D0
```

```
BYTE CLSID_TaskbarList#, 8 = $95
```

```
BYTE CLSID_TaskbarList#, 9 = $8A
```

```
BYTE CLSID_TaskbarList#, 10 = $00
```

```
BYTE CLSID_TaskbarList#, 11 = $60
```

```
BYTE CLSID_TaskbarList#, 12 = $97
```

```
BYTE CLSID_TaskbarList#, 13 = $C9
```

```
BYTE CLSID_TaskbarList#, 14 = $A0
```

```
BYTE CLSID_TaskbarList#, 15 = $90
```

```
'GUID IDD_TaskBarlist
```

```
LONG IID_ITaskbarList#, 0 = $56FDF342
```

```
WORD IID_ITaskbarList#, 4 = $FD6D
```

```
WORD IID_ITaskbarList#, 6 = $11D0
```

```
BYTE IID_ITaskbarList#, 8 = $95
```

```
BYTE IID_ITaskbarList#, 9 = $8A
```

```
BYTE IID_ITaskbarList#, 10 = $00
```

```
BYTE IID_ITaskbarList#, 11 = $60
```

```
BYTE IID_ITaskbarList#, 12 = $97
```

```
BYTE IID_ITaskbarList#, 13 = $C9
```

```
BYTE IID_ITaskbarList#, 14 = $A0
```

```
BYTE IID_ITaskbarList#, 15 = $90
```

```
Dim TbObj#, 4
```

```
Dim Param#, 4
```

```
Long Param#, 0 = %hwnd
```

```
CoInitialize(0)
If CoCreateInstance(CLSID_TaskbarList#,0,1,IID_ITaskbarList#,TbObj#) = 0
    CallObj(TbObj#,HrInIt&,0,0)
    If Equ(Flag&,0)
        CallObj(TbObj#,DeleteTab&,Param#,1)
    ElseIf Equ(Flag&,1)
        CallObj(TbObj#,AddTab&,Param#,1)
    Endif
    CallObj(TbObj#,Release&,0,0)
Endif
DISPOSE CLSID_TaskbarList#
DISPOSE IID_ITaskbarList#
DISPOSE TbObj#
DISPOSE Param#
CoUnInitialize()
EndProc
```

```
cls
HideMyWindow 0
waitinput
HideMyWindow 1
waitinput
end
```

## Beispiel2

### Einige API-Funktionen

```
DEF CallFunktion(4) ! "Call", "CallFunktion"
```

```
CLS
```

```
Declare HDll&
```

```
Declare Lib#,Funktion#,Param#
```

```
Declare Filename#
```

```
Declare Caption#,Text#
```

```
Let HDll& = UseDll("Call")
```

```
Proc FileExists
```

```
'Prüft ob die Datei existiert
```

```
Parameters Filename$
```

```
Declare Lib$,Funktion$,RetVal&
```

```
Let Lib$ = "Shell32"
```

```
Let Funktion$ = "#45"FileExists
```

```
Dim Filename#,Len(Filename$)
```

```
Dim Funktion#,Len(Funktion$)
```

```
Dim Lib#,Len(Lib$)
```

```
Dim Param#,4
```

```
String Funktion#,0=Funktion$
```

```
String Lib#,0=Lib$
```

```
String Filename#,0 = Filename$
```

```
Long Param#,0 =Filename#
```

```
Let RetVal& = CallFunktion(Funktion#,Lib#,1,Param#)
```

```
Dispose Param#
```

```
Dispose Lib#
```

```
Dispose Funktion#
```

```
Dispose Filename#
```

```
Return RetVal&
```

```
EndProc
```

```
Proc WinRestart
```

```
'Ruft den Beenden-Dialog auf
```

```
Parameters Wnd&
```

```
Declare Lib$,Funktion$,RetVal&
```

```
Let Lib$ = "Shell32"
```

```
Let Funktion$ = "#60" 'Ende-Dialog
```

```
Dim Funktion#,Len(Funktion$)
```

```
Dim Lib#,Len(Lib$)
```

```
Dim Param#,4
```

```
String Funktion#,0=Funktion$
```

```
String Lib#,0=Lib$
```

```
Long Param#,0 = Wnd&
```

```
Let RetVal& = CallFunktion(Funktion#,Lib#,1,Param#)
```

```
Dispose Param#
```

```
Dispose Lib#
```

```
Dispose Funktion#
```

```
EndProc
```

```
Proc ShowMessage
```

```
'Das ist eigentlich unnötig, zeigt aber wie es geht !
```

```
Parameters Wnd&,Caption$,Text$,Style&
```

```
Declare Lib$,Funktion$,RetVal&
```

```
Let Lib$ = "User32"
Let Funktion$ = "MessageBoxA"
Dim Funktion#,Len(Funktion$)
Dim Lib#,Len(Lib$)
Dim Caption#,Len(Caption$)
Dim Text#,Len(Text$)
Dim Param#,16
String Funktion#,0=Funktion$
String Lib#,0=Lib$
String Caption#,0 =Caption$
String Text#,0 =Text$
Long Param#,0 = wnd&
Long Param#,4 = Text#
Long Param#,8 = Caption#
Long Param#,12 = Style&
Let RetVal& = CallFunktion(Funktion#,Lib#,4,Param#)
Dispose Param#
Dispose Funktion#
Dispose Lib#
Dispose Caption#
Dispose Text#
Return RetVal&
EndProc

cls
FileExists "C:\Autoexec.bit"
Print "C:\Autoexec.bit",If(EQU(@&(0),1),"existiert", "existiert nicht")
ShowMessage %Hwnd,"Meldung", "Das ist der Text",64
WinRestart %hwnd
Waitinput
Usedll("")
END
```

## Lizenz

Die CallObj.Dll ist **Freeware** , sowohl für private als auch für kommerzielle Anwendungen !  
Es gibt keinerlei Einschränkungen !

Der Autor übernimmt keinerlei Haftung für Schäden, die durch die Verwendung der Dll auftreten !

Bielefeld, im November 2002  
[www.andreas-miethe.de](http://www.andreas-miethe.de)