

Inhaltsverzeichnis

- [1 bCreateShortcut](#)
- [2 bFileCopy](#)
- [3 bFileDelete](#)
- [4 bFileDownload](#)
- [5 bFileExists](#)
- [6 bFileIsReadOnly](#)
- [7 bFileMove](#)
- [8 bFileSetCreationTime](#)
- [9 bIs64BitOperatingSystem](#)
- [10 bKillTask](#)
- [11 bPathCreate](#)
- [12 bPathDelete](#)
- [13 bPathExists](#)
- [14 bShellExecute](#)
- [15 dtFileGetCreationTime](#)
- [16 sCommandLine](#)
- [17 sComputerDomain](#)
- [18 sComputerName](#)
- [19 sComputerUser](#)
- [20 sDirectoryApplicationData](#)
- [21 sDirectoryCommonApplicationData](#)
- [22 sDirectoryCommonDesktop](#)
- [23 sDirectoryCommonDocuments](#)
- [24 sDirectoryCommonProgramFiles](#)
- [25 sDirectoryCommonPrograms](#)
- [26 sDirectoryCurrent](#)
- [27 sDirectoryDesktop](#)
- [28 sDirectoryLocalApplicationData](#)
- [29 sDirectoryMyDocuments](#)
- [30 sDirectoryProgramFiles](#)
- [31 sDirectoryPrograms](#)
- [32 sDirectoryStartMenu](#)
- [33 sDirectoryStartup](#)
- [34 sDirectorySystem](#)
- [35 sDirectoryUserProfile](#)
- [36 sDirectoryWindows](#)
- [37 sFileMD5](#)
- [38 sFileVersion](#)
- [39 sSystemName](#)
- [40 sSystemVersion](#)

Das Aufgaben-Center Objekt goSystem

1 bCreateShortcut

Die Funktion **bCreateShortcut** erstellt eine Verknüpfung zu einer Datei *sFilename*. Mit dem Argument *sArguments* können Parameter für den Aufruf der Datei angegeben werden. Über das Argument

sShortcutPath gibt man das Verzeichnis und mit *sName* den Namen für die Verknüpfung an. *sWorkDirectory* definiert das Arbeitsverzeichnis der Datei und *sIcon* das Symbol für die Verknüpfung. Ist die Funktion erfolgreich liefert die Funktion *True* zurück. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

bCreateShortcut(sFilename As String, sArguments As String, sShortcutPath As String, sName As String, sWorkDirectory As String, Optional sIcon As String = "", Optional blgnoreError As Boolean = False) As Boolean

2 bFileCopy

Die Funktion **bFileCopy** kopiert die Datei *sSource* an das angegebene Ziel *sDestination*. Mit dem Argument *bOverwrite* kann bestimmt werden ob eine bereits existierende Datei überschrieben werden soll. Ist die Funktion erfolgreich liefert diese *True* zurück. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

bFileCopy(sSource As String, sDestination As String, Optional bOverwrite As Boolean = False, Optional blgnoreError As Boolean = False) As Boolean

3 bFileDelete

Die Funktion **bFileDelete** löscht die Datei *sFilename*, ist dies erfolgreich liefert die Funktion *True* zurück. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

bFileDelete(sFilename As String, Optional blgnoreError As Boolean = False) As Boolean

4 bFileDownload

Die Funktion **bFileDownload** lädt eine Datei über eine Http-Adresse *sUrl* runter und speichert diese unter dem angegebenen Namen *sDestination* ab, ist dies erfolgreich liefert die Funktion *True* zurück. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

bFileDownload(sUrl As String, sDestination As String, Optional blgnoreError As Boolean = False) As Boolean

5 bFileExists

Die Funktion **bFileExists** prüft ob die Datei *sFilename* existiert ist und liefert dann *True* zurück. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

bFileExists(sFilename As String, Optional blgnoreError As Boolean = False) As Boolean

6 bFileIsReadOnly

Die Funktion **bFileIsReadOnly** prüft ob die Datei *sFilename* schreibgeschützt ist und liefert dann *True* zurück. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

bFileIsReadOnly(sFilename As String, Optional blgnoreError As Boolean = False) As Boolean

7 bFileMove

Die Funktion **bFileMove** verschiebt eine Datei *sSource* in ein anderes Verzeichnis *sDestination*, ist dies erfolgreich liefert die Funktion *True* zurück. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

bFileMove(sSource As String, sDestination As String, Optional blgnoreError As Boolean = False)

8 bFileSetCreationTime

Mit der Funktion **bFileSetCreationTime** kann das Erstellungsdatum einer Datei *sFilename* mit einem neuen Zeitstempel *dtTime* versehen werden. Ist dies erfolgreich liefert die Funktion *True* zurück. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

bFileSetCreationTime(sFilename As String, dtTime As Date, Optional blgnoreError As Boolean = False) As Boolean

9 bls64BitOperatingSystem

Die Funktion **bls64BitOperatingSystem** prüft ob es sich um ein 64-Bit Betriebssystem handelt und liefert dann *True* zurück.

bIs64BitOperatingSystem As Boolean

10 bKillTask

Die Funktion **bKillTask** beendet eine laufende Anwendung *sTaskname*, ist dies erfolgreich liefert die Funktion *True* zurück. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

bKillTask(sTaskname As String, Optional blgnoreError As Boolean = False) As Boolean

11 bPathCreate

Die Funktion **bPathCreate** erstellt das Verzeichnis *sPath*, ist dies erfolgreich liefert die Funktion *True* zurück. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

bPathCreate(sPath As String, Optional blgnoreError As Boolean = False) As Boolean

12 bPathDelete

Die Funktion **bPathDelete** löscht das Verzeichnis *sPath*, ist dies erfolgreich liefert die Funktion *True* zurück. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

bPathDelete(sPath As String, Optional blgnoreError As Boolean = False) As Boolean

13 bPathExists

Die Funktion **bPathExists** prüft, ob das Verzeichnis *sPath* existiert und liefert dann *True* zurück. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

bPathExists(sPath As String, Optional blgnoreError As Boolean = False) As Boolean

14 bShellExecute

Die Funktion **bShellExecute** führt eine Datei *sFilename* mit den übergebenen Aufruf-Parametern *sArguments* aus. Über das optionale Argument *bWaitForExit* kann bestimmt werden, ob die Programmausführung auf das Beenden der auszuführenden Datei warten soll. Ist die Funktion erfolgreich liefert die Funktion *True* zurück. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

`bShellExecute(sFilename As String, sArguments As String, Optional bWaitForExit As Boolean = True, Optional blgnoreError As Boolean = False) As Boolean`

15 dtFileGetCreationTime

Die Funktion **dtFileGetCreationTime** ermittelt zu einer Datei *sFilename* das Erstellungsdatum. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

`dtFileGetCreationTime(sFilename As String, Optional blgnoreError As Boolean = False) As Date`

16 sCommandLine

Die Eigenschaft **sCommandLine** liefert die Kommandozeilenparameter die beim Start der Applikation angegeben wurden.

`sCommandLine As String`

17 sComputerDomain

Die Eigenschaft **sComputerDomain** liefert den Namen der Domäne bzw. Arbeitsgruppe des angemeldeten Benutzers.

`sComputerDomain As String`

18 sComputerName

Die Eigenschaft **sComputerName** liefert den Namen des Computers.

sComputerName As String

19 sComputerUser

Die Eigenschaft **sComputerUser** liefert den Namen des angemeldeten Benutzers.

sComputerUser As String

20 sDirectoryApplicationData

Die Eigenschaft **sDirectoryApplicationData** liefert das Anwendungsdaten-Verzeichnis des angemeldeten Benutzers.

C:\Users\%Username%\AppData\Roaming\

sDirectoryApplicationData As String

21 sDirectoryCommonApplicationData

Die Eigenschaft **sDirectoryCommonApplicationData** liefert das gemeinsame Anwendungsdaten-Verzeichnis.

C:\ProgramData\

sDirectoryCommonApplicationData As String

22 sDirectoryCommonDesktop

Die Eigenschaft **sDirectoryCommonDesktop** liefert das gemeinsame Desktop-Verzeichnis.

C:\Users\Public\Desktop\

sDirectoryCommonDesktop As String

23 sDirectoryCommonDocuments

Die Eigenschaft **sDirectoryCommonDocuments** liefert das gemeinsame Dokumente-Verzeichnis.
C:\Users\Public\Documents\

sDirectoryCommonDocuments As String

24 sDirectoryCommonProgramFiles

Die Eigenschaft **sDirectoryCommonProgramFiles** liefert das gemeinsame Dateien-Verzeichnis.
C:\Program Files (x86)\Common Files\

sDirectoryCommonProgramFiles As String

25 sDirectoryCommonPrograms

Die Eigenschaft **sDirectoryCommonPrograms** liefert das gemeinsame Programme-Verzeichnis des Start-Menüs.
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\

sDirectoryCommonPrograms As String

26 sDirectoryCurrent

Die Eigenschaft **sDirectoryCurrent** liefert das Arbeitsverzeichnis der aktuell laufenden Anwendung.
C:\Program Files (x86)\Sage\Office Line\6.2\LogiSoft\

sDirectoryCurrent As String

27 sDirectoryDesktop

Die Eigenschaft **sDirectoryDesktop** liefert das Desktop-Verzeichnis des angemeldeten Benutzers.
C:\Users\%Username%\Desktop\

sDirectoryDesktop As String

28 sDirectoryLocalApplicationData

Die Eigenschaft **sDirectoryLocalApplicationData** liefert das lokale Anwendungsdaten-Verzeichnis des angemeldeten Benutzers.

C:\Users\%Username%\AppData\Local\

sDirectoryLocalApplicationData As String

29 sDirectoryMyDocuments

Die Eigenschaft **sDirectoryMyDocuments** liefert das Dokumente-Verzeichnis des angemeldeten Benutzers.

C:\Users\%Username%\Documents\

sDirectoryMyDocuments As String

30 sDirectoryProgramFiles

Die Eigenschaft **sDirectoryProgramFiles** liefert das Programme-Verzeichnis.

C:\Program Files (x86)\

sDirectoryProgramFiles As String

31 sDirectoryPrograms

Die Eigenschaft **sDirectoryPrograms** liefert das Programme-Verzeichnis des Start-Menüs für den angemeldeten Benutzer.

C:\Users\%Username%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\

sDirectoryPrograms As String

32 sDirectoryStartMenu

Die Eigenschaft **sDirectoryStartMenu** liefert das Verzeichnis des Start-Menüs für den angemeldeten Benutzer.

C:\Users\%Username%\AppData\Roaming\Microsoft\Windows\Start Menu\

sDirectoryStartMenu As String

33 sDirectoryStartup

Die Eigenschaft **sDirectoryStartup** liefert das Autostart-Verzeichnis des Start-Menüs für den angemeldeten Benutzer.

C:\Users\%Username%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\

sDirectoryStartup As String

34 sDirectorySystem

Die Eigenschaft **sDirectorySystem** liefert das System-Verzeichnis des Betriebssystems.

C:\Windows\System32\

sDirectorySystem As String

35 sDirectoryUserProfile

Die Eigenschaft **sDirectoryUserProfile** liefert das Profil-Verzeichnis für den angemeldeten Benutzer.

C:\Users\%Username\

sDirectoryUserProfile As String

36 sDirectoryWindows

Die Eigenschaft **sDirectorySystem** liefert das Verzeichnis des Betriebssystems.

C:\Windows\

sDirectoryWindows As String

37 sFileMD5

Die Funktion **sFileMD5** ermittelt zu einer Datei *sFilename* einen MD5 Hashwert. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

sFileMD5(sFilename As String, Optional blgnoreError As Boolean = False) As String

38 sFileVersion

Die Funktion **sFileVersion** ermittelt zu einer Datei *sFilename* die Dateiversion. Ist das optionale Argument *bProductVersion* gesetzt, liefert die Funktion die Produktversion der Datei. Mit dem optionalen Argument *blgnoreError* lassen sich Fehler beim Ausführen der Funktion ignorieren.

sFileVersion(sFilename As String, Optional bProductVersion As Boolean = False, Optional blgnoreError As Boolean = False) As String

39 sSystemName

Die Eigenschaft **sSystemName** gibt den Namen des Betriebssystems zurück.

sSystemName As String

40 sSystemVersion

Die Eigenschaft **sSystemVersion** gibt die Version des Betriebssystems zurück.

sSystemVersion As String