

1. Global text constants

1.1. System informations

%AppdataDir%
C:\Dokumente und Einstellungen\
%USERNAME%\Anwendungsdaten

%AllUsersProfileDir%
C:\Dokumente und Einstellungen\All Users

%CommonStartMenuPath%
C:\Dokumente und Einstellungen\
All Users\Startmenü

%CommonAppdataDir%
C:\Dokumente und Einstellungen\
All Users\Anwendungsdaten

%ProfileDir%
C:\Dokumente und Einstellungen

%UserProfileDir% (only with /AllntUserProfiles)

%ProgramFilesDir% c:\programme

%Systemroot% c:\windows

%System% c:\winnt\system32

%Systemdrive% c:\

1.2. winst-Path and Directories

%ScriptPath%
%ScriptDrive%
%WinstDir%

1.3. Network informations

%Host% value of environment variable HOST.

%PCName%
value of environment variable PCNAME,
or if absent of COMPUTERNAME.

%Username% Name of actual user.

2. Primary Sections:

[Initial]
[Aktionen]
[sub<identifier>]
sub <file name>

3. Winst control

LogLevel = <number>
or LogLevel = <string>
LogLevel = "2"

ExitOnError = <boolean value>

Message <string> or Message = <const string>

ShowMessageFile <string>

ShowBitMap [/<number>] [<file name>] [<sub titel>]
comment <string> or comment = <const string>

LogError <string> or LogError = <const string>

LogWarning <string> or LogWarning = <const string>

Pause <string> or Pause = <const string>

Stop <string> or stop = <const string>

4. Variables

4.1. strings

DefVar <variable name>
Set <variable name> = <value>

4.2. stringlists

DefstringList <variable name>

5. Functions

5.1. string functions

GetOS
GetNtVersion
EnvVar (<environment variable>)
ParamStr
GetRegistrystringValue ("[key] var")
GetValueFromInifile (file, section, key, default value)
GetValueFromInifile ("myfile", "mysec", "mykey", "")
IniVar (<key>)
GetHostsName (<hostaddress>)
GetHostsAddr (<hostname>)
ExtractFilePath (<path>)
SubstringBefore (<string1>, <string2>)
RandomStr
getValue(<key string>, <hash string list>)
getLastExitCode

5.2. String list functions

createStringList (<string0>, <string1>, ...)
set \$list\$ = createStringList ('a', 'b')

splitString (<string1>, <string2>)
set list = splitString
("\\server\share\dir", "\\")

splitStringOnWhiteSpace (<string>)

loadTextFile (<file name>)
loadUnicodeTextFile (<file name>)
composeString (<string list>, <Link>)
takeString (<index>, <list>)
retrieveSection (<section name>)
getOutputStreamFromSection (<dos section name>)
getOutputStreamFromSection ('DosBatch_try')

getReturnListFromSection (<xml section name>)
getSubList (Startindex, Endindex, Liste)
reverse (Liste)

for %s% in List do <one statement | sub section>
for %s% in list1 do sub_test_string

GetLocaleInfoMap
getFileInfoMap(\$InterestingFile\$)

5.3. Boolean operators and functions

<string1> = <string2>
<bool1> AND <bool2>
<bool1> OR <bool2>
NOT(<bool3>)

FileExists (<file name>)
LineExistsIn (line, filename)
LineBeginning_ExistsIn (stringval, file name)
XMLAddNamespace(XMLfilename, XMLelementname,
XMLnamespace)
XMLRemoveNamespace(XMLfilename, XMLelementname,
XMLnamespace)

HasMinimumSpace (drive letter, capacity)

Example:

```
if not (HasMinimumSpace ("%SYSTEMDRIVE%",
"500 MB"))
    LogError "Neede Space of 500 MB not"
```

```
available on %SYSTEMDRIVE%"
  isFatalError
endif
```

5.4. Misc functions

```
Killtask <process name>
requiredWinstVersion <relation operator> <version>
  requiredWinstVersion >= "4.3"
```

6. Flow control

if - else - endif

Syntax:

```
if <condition>
  ;statement(s)
[else
  ;statement(s)]
endif
```

Example:

```
if $OS$ = "Windows_NT"
  Set $NTVer$ = GetNTVersion
  if ($NTVer$="NT4") or ($NTVer$="Win2k")
    sub_install_winnt
  else
    if ( $NTVersion$ = "WinXP" )
      sub_install_winXP
    else
      stop "not a supported OS-Version"
    endif
  endif
endif
```

```
isFatalError
ExitWindows /ShutdownWanted
ExitWindows /RebootWanted (depricated, acts like /Reboot)
ExitWindows /Reboot
ExitWindows /ImmediateReboot
ExitWindows /ImmediateLogout
```

```
sleepSeconds <Integer>
```

7. Secondary Sections:

[WinBatch<identifier>] [/LetThemGo]

Function: execute programs via Windows API

[DosBatch<identifier>]

[DosInAnIcon<identifier>]

Function: Execute section via cmd.exe

[Files<identifier>] [/AllNTUserProfiles or /AllNTUserSendTo]

Function: File Operations

```
checkTargetPath <destination directory>
copy [Options] <source file(s)> <destination directory>
  some options: -s rekursiv
  -V version control against targetdir
  -v version control against targetdir, %systemroot% and
  %system%
  -c continue without reboot even if it is needed
```

```
delete [Options] <path[/mask]]
  some options: -s rekursiv -f force
```

Example (don't forget the trailing Backslash):

```
delete -sf c:\delete_this_dir\
```

[Registry<identifier>] [/AllNTUserDats]

Function: edit Registry

OpenKey <Key>

```
  openkey [HKLM\Software\opsi.org]
```

Set <varname> = <registry type>:<value>

Add <varname> = <registry type>:<value>

Examples for registry types:

```
set "var1" = "my string"
set "var2" = REG_SZ:"my string"
set "var3" = REG_EXPAND_SZ:"%Systemroot%"
set "var4" = REG_DWORD:123 (Decimal)
set "var5" = REG_BINARY:00 01 02 0F 10
set "var6" = REG_MULTI_SZ:"A|BC|de"
```

Supp <varname> <list char> <supplement>

```
  supp "Path" ; "C:\utils; %JAVABIN%"
```

GetMultiSZFromFile <file name>

DeleteVar <varname>

DeleteKey <registry key> (don't work with /AllNTUserDats)

[Patches<identifier>] <file name>

Function: edit Ini-files

add [<section name>] <variable1> = <value1>

set [<section name>]<variable1> = <value1>

addnew [<section name>]<variable1> = <value1>

change [<section name>]<variable1> = <value1>

del [<section name>] <variable1> = <value1>

del [<section name>] <variable1>

delsec [<section name>]

replace <variable1>=<value1> <variable2>=<value2>

[PatchTextFile<identifier>] <file name>

Function: edit text files

Set_Netscape_User_Pref ("<key>", "<value>")

AddstringListElement_To_Netscape_User_Pref

FindLine <search string>

FindLine_StartingWith <search string>

FindLine_Containing <search string>

GoToTop

AdvanceLine [<number of lines>]

GoToBottom

DeleteTheLine

AddLine_ <line> or Add_Line_ <line>

InsertLine <line> or Insert_Line_ <line>

AppendLine <line> or Append_Line <line>

Append_File <file name>

Subtract_File <file name>

SaveToFile <file name>

Sorted

[LinkFolder<identifier>]

Function: Startmenu + Desktop Icons

set_basefolder <system folder>

set_subfolder <folder path>

set_link

```
  name: [link name]
```

```
  target: <path and name of the program>
```

```
  parameters: [command line arguments]
```

```
  working_dir: [working directory]
```

```
  icon_file: [path and name of icon file, default=target]
```

```
  icon_index: [number of icon in icon file, default=0]
```

end_link

delete_element <link name>

delete_subfolder <folder path>

The predefined virtual system folders which can be used are desktop, sendto, startmenu, startup, programs, desktopdirectory, common_startmenu, common_programs, common_startup, common_desktopdirectory

[OpsiServiceCall<identifier>]

*Function: opsi-Service access
see manual*

[IdapiConfig<identifier>]

*Function: BDE configuration
see manual*

[PatchHosts<identifier>]

*Function: hosts-files bearbeiten
see manual*

[XMLPatch<identifier>]

*Function: edit XML files
see manual*

[ExecPython<identifier>]

*Function: Execute section via python interpreter
see manual*

[ExecWith<identifier>] <interpreter>

*Function: Execute section via any interpreter
see manual*