

StartProcess

%StartProcess function

Function	Function creates a new process. The new process then runs the specified executable file in the security context of the specified credentials (user, domain, and password).										
Declaration	<pre>INT %StartProcess(TEXT in userName, TEXT in domain, TEXT in password, TEXT in applName, TEXT in commandLine, TEXT in currDir, BOOL in bSync, INT in timeout, BOOL in bHide)</pre>										
Note	<table><tr><td>user name</td><td>String that specifies the name of the user. This is the name of the user account to log on to. If you use the UPN format, <i>user@DNS.domain.name</i>, domain parameter must be empty string. The user account must have the Log On Locally permission on the local computer. This permission is granted to all users on workstations and servers, but only to administrators on domain controllers. It only can be used for the Windows XP, Windows 2000 Professional, Windows Server 2003 or Windows 2000 Server platforms. For other platform, the function returns an invalid value. If the parameter is empty string, the process is started in the context of the user, who opens the calling process. Such a call can be also used for another MS Windows platforms.</td></tr><tr><td>domain</td><td>String that specifies the name of the domain or server whose account database contains the <i>userName</i> account. If this parameter is empty string, the user name must be specified in UPN format.</td></tr><tr><td>password</td><td>String that specifies the clear-text password for the <i>userName</i> account.</td></tr><tr><td>applName</td><td>String that specifies the module to execute. The specified module can be a Windows-based application. It can be some other type of module (for example, MS-DOS or OS/2) if the appropriate subsystem is available on the local computer. The string can specify the full path and file name of the module to execute. The function will not use the search path. If the file name does not contain an extension, <i>.exe</i> is assumed. Therefore, if the file name extension is <i>.com</i>, this parameter must include the <i>.com</i> extension. The <i>applName</i> parameter can be empty string. In that case, the module name must be the first white space-delimited token in the <i>commandLine</i> string. If you are using a long file name that contains a space, use quoted strings to indicate where the file name ends and the arguments begin; otherwise, the file name is ambiguous. For example, consider the string "c:\program files\sub dir\program name". This string can be interpreted in a number of ways. The system tries to interpret the possibilities in the following order: <ol style="list-style-type: none">1. c:\program.exe files\sub dir\program name2. c:\program files\sub.exe dir\program name3. c:\program files\sub dir\program.exe name4. c:\program files\sub dir\program name If the executable module is a 16-bit application, <i>applName</i> should be empty string, and the string <i>commandLine</i> should specify the executable module as well as its arguments.</td></tr><tr><td>commandLine</td><td>String that specifies the command line to execute. The <i>commandLine</i> parameter can be empty string. In that case, the function uses the string <i>applName</i> as the command line. If both <i>applName</i> and <i>commandLine</i> are non-empty strings, <i>applName</i> specifies the module to execute, and <i>commandLine</i> specifies the command line. The new process can use GetCommandLine (Win32 API) to retrieve the entire command line. Console processes written in C can use the <i>argc</i> and <i>argv</i> arguments to parse the command line. Because <i>argv[0]</i> is the module name, C programmers generally repeat the module name as the first token in the command line. If <i>applName</i> is empty string, the first white-space – delimited token of the command line specifies the module name. If you are using a long file name that contains a space, use quoted strings to indicate where the file name ends and the arguments begin (see the explanation for the <i>applName</i> parameter). If the file name does not contain an extension, <i>.exe</i> is appended. Therefore, if the file name extension is <i>.com</i>, this parameter must include the <i>.com</i> extension. If the file name ends in a period with no extension, or if the file name contains a path, <i>.exe</i> is not appended. If the file name does not contain a directory path, the system searches for the executable file in the following sequence: <ol style="list-style-type: none">1. The directory from which the application loaded.2. The current directory for the process - current directory is not explicitly defined and therefore can be optional.3. The 32-bit Windows system directory.4. The 16-bit Windows system directory.5. The Windows directory.6. The directories that are listed in the PATH environment variable.</td></tr></table>	user name	String that specifies the name of the user. This is the name of the user account to log on to. If you use the UPN format, <i>user@DNS.domain.name</i> , domain parameter must be empty string. The user account must have the Log On Locally permission on the local computer. This permission is granted to all users on workstations and servers, but only to administrators on domain controllers. It only can be used for the Windows XP, Windows 2000 Professional, Windows Server 2003 or Windows 2000 Server platforms. For other platform, the function returns an invalid value. If the parameter is empty string, the process is started in the context of the user, who opens the calling process. Such a call can be also used for another MS Windows platforms.	domain	String that specifies the name of the domain or server whose account database contains the <i>userName</i> account. If this parameter is empty string, the user name must be specified in UPN format.	password	String that specifies the clear-text password for the <i>userName</i> account.	applName	String that specifies the module to execute. The specified module can be a Windows-based application. It can be some other type of module (for example, MS-DOS or OS/2) if the appropriate subsystem is available on the local computer. The string can specify the full path and file name of the module to execute. The function will not use the search path. If the file name does not contain an extension, <i>.exe</i> is assumed. Therefore, if the file name extension is <i>.com</i> , this parameter must include the <i>.com</i> extension. The <i>applName</i> parameter can be empty string. In that case, the module name must be the first white space-delimited token in the <i>commandLine</i> string. If you are using a long file name that contains a space, use quoted strings to indicate where the file name ends and the arguments begin; otherwise, the file name is ambiguous. For example, consider the string "c:\program files\sub dir\program name". This string can be interpreted in a number of ways. The system tries to interpret the possibilities in the following order: <ol style="list-style-type: none">1. c:\program.exe files\sub dir\program name2. c:\program files\sub.exe dir\program name3. c:\program files\sub dir\program.exe name4. c:\program files\sub dir\program name If the executable module is a 16-bit application, <i>applName</i> should be empty string, and the string <i>commandLine</i> should specify the executable module as well as its arguments.	commandLine	String that specifies the command line to execute. The <i>commandLine</i> parameter can be empty string. In that case, the function uses the string <i>applName</i> as the command line. If both <i>applName</i> and <i>commandLine</i> are non-empty strings, <i>applName</i> specifies the module to execute, and <i>commandLine</i> specifies the command line. The new process can use GetCommandLine (Win32 API) to retrieve the entire command line. Console processes written in C can use the <i>argc</i> and <i>argv</i> arguments to parse the command line. Because <i>argv[0]</i> is the module name, C programmers generally repeat the module name as the first token in the command line. If <i>applName</i> is empty string, the first white-space – delimited token of the command line specifies the module name. If you are using a long file name that contains a space, use quoted strings to indicate where the file name ends and the arguments begin (see the explanation for the <i>applName</i> parameter). If the file name does not contain an extension, <i>.exe</i> is appended. Therefore, if the file name extension is <i>.com</i> , this parameter must include the <i>.com</i> extension. If the file name ends in a period with no extension, or if the file name contains a path, <i>.exe</i> is not appended. If the file name does not contain a directory path, the system searches for the executable file in the following sequence: <ol style="list-style-type: none">1. The directory from which the application loaded.2. The current directory for the process - current directory is not explicitly defined and therefore can be optional.3. The 32-bit Windows system directory.4. The 16-bit Windows system directory.5. The Windows directory.6. The directories that are listed in the PATH environment variable.
user name	String that specifies the name of the user. This is the name of the user account to log on to. If you use the UPN format, <i>user@DNS.domain.name</i> , domain parameter must be empty string. The user account must have the Log On Locally permission on the local computer. This permission is granted to all users on workstations and servers, but only to administrators on domain controllers. It only can be used for the Windows XP, Windows 2000 Professional, Windows Server 2003 or Windows 2000 Server platforms. For other platform, the function returns an invalid value. If the parameter is empty string, the process is started in the context of the user, who opens the calling process. Such a call can be also used for another MS Windows platforms.										
domain	String that specifies the name of the domain or server whose account database contains the <i>userName</i> account. If this parameter is empty string, the user name must be specified in UPN format.										
password	String that specifies the clear-text password for the <i>userName</i> account.										
applName	String that specifies the module to execute. The specified module can be a Windows-based application. It can be some other type of module (for example, MS-DOS or OS/2) if the appropriate subsystem is available on the local computer. The string can specify the full path and file name of the module to execute. The function will not use the search path. If the file name does not contain an extension, <i>.exe</i> is assumed. Therefore, if the file name extension is <i>.com</i> , this parameter must include the <i>.com</i> extension. The <i>applName</i> parameter can be empty string. In that case, the module name must be the first white space-delimited token in the <i>commandLine</i> string. If you are using a long file name that contains a space, use quoted strings to indicate where the file name ends and the arguments begin; otherwise, the file name is ambiguous. For example, consider the string "c:\program files\sub dir\program name". This string can be interpreted in a number of ways. The system tries to interpret the possibilities in the following order: <ol style="list-style-type: none">1. c:\program.exe files\sub dir\program name2. c:\program files\sub.exe dir\program name3. c:\program files\sub dir\program.exe name4. c:\program files\sub dir\program name If the executable module is a 16-bit application, <i>applName</i> should be empty string, and the string <i>commandLine</i> should specify the executable module as well as its arguments.										
commandLine	String that specifies the command line to execute. The <i>commandLine</i> parameter can be empty string. In that case, the function uses the string <i>applName</i> as the command line. If both <i>applName</i> and <i>commandLine</i> are non-empty strings, <i>applName</i> specifies the module to execute, and <i>commandLine</i> specifies the command line. The new process can use GetCommandLine (Win32 API) to retrieve the entire command line. Console processes written in C can use the <i>argc</i> and <i>argv</i> arguments to parse the command line. Because <i>argv[0]</i> is the module name, C programmers generally repeat the module name as the first token in the command line. If <i>applName</i> is empty string, the first white-space – delimited token of the command line specifies the module name. If you are using a long file name that contains a space, use quoted strings to indicate where the file name ends and the arguments begin (see the explanation for the <i>applName</i> parameter). If the file name does not contain an extension, <i>.exe</i> is appended. Therefore, if the file name extension is <i>.com</i> , this parameter must include the <i>.com</i> extension. If the file name ends in a period with no extension, or if the file name contains a path, <i>.exe</i> is not appended. If the file name does not contain a directory path, the system searches for the executable file in the following sequence: <ol style="list-style-type: none">1. The directory from which the application loaded.2. The current directory for the process - current directory is not explicitly defined and therefore can be optional.3. The 32-bit Windows system directory.4. The 16-bit Windows system directory.5. The Windows directory.6. The directories that are listed in the PATH environment variable.										

c u r r e n t d i r	String that specifies the current drive and directory for the new process. The string must be a full path that includes a drive letter. If this parameter is empty string, the new process has the same current drive as the system service that creates the process. (This feature is provided primarily for shells that need to start an application and specify its initial drive and working directory.)
b s y n c	The parameter specifies, whether the script waits for termination of the started process. <ul style="list-style-type: none">• @True - the function starts the process and waits for its termination. The parameter <i>timeout</i> specifies maximum program runtime in seconds. If the time is exceeded, the process is enforced to close and the return code of the function is _ERR_TIME_OUT. If the process is closed before expiration of the time limit, the return code of the function is the return code of the process.• @False - the function starts the process and continues executing the next action. The parameter <i>timeout</i> is to be ignored. Return code is 0, is the process has been successfully started. If not, the return code is an invalid value.
t i m e o u t	Maximum runtime of the started process. The parameter can be used only when <i>bSync</i> =@TRUE.
b h i d e	Boolean type parameter - possible values: <ul style="list-style-type: none">• @True - hide the window of the started application• @False - show the window of the started application

Return value

Return value is Int type and

1. is invalid if were unable to run the process
2. is equal to the value of _ERR_TIME_OUT, if *bSync*=@TRUE and the timeout was exceeded
3. is equal to 0, if *bSync*=@FALSE and the process has been started
4. otherwise it get the value of the return code of the process

Note

If unsuccessful, the function writes an error log to the file **event.log**.

Example

Starting the MS Excel program with opening the file *C:\TEMP\WorkBook1.xls*.
MS Excel has been installed into *C:\Program Files\Microsoft Office\Office\excel.exe*.
Name of the local user is *Adm* and the password is *pwd*.

```
BEGIN
TEXT _cmdLine
INT _retCode

_cmdLine := %Chr(34) + "c:\Program Files\Microsoft Office\Office10\EXCEL.
EXE" + %Chr(34) + "c:\temp\Workbook1.xls"
_retCode := %StartProcess("Adm", "", "pwd", "", _cmdLine, "c:\temp",
@TRUE, 30, @FALSE)
IF _retCode\VLd THEN
IF _retCode = _ERR_TIME_OUT THEN
; MS Excel was not terminated in the timeout of 30 seconds and was
enforced to close
ELSE
; MS Excel was terminated in the timeout of 30 seconds and _retCode
contains the return code
ENDIF
ELSE
; cannot to start the program
; the path c:\Program Files\Microsoft Office\Office10\EXCEL.EXE doesn't
exist, or cannot log on the user
ENDIF
END
```



Related pages:

- [ESL extended functions](#)
- [Function arguments - types](#)