

SPI Info Structure

Any application which ships an SPI file (Stupsi Processing Information) along with the new application files can be updated automatically from a client if Stupsi is installed on the client. It will check on a server for software updates.

The following deals with the standard format of SPI files.

The documentation has been applied to Stupsi v1.6.7.

Overview

An SPI file is a simple ASCII file containing plain text information. It is performed line by line. Lines have to be separated by CR + LF (Character 13 + Character 10).

Options, commands and %-variables are case insensitive. %-Variables can be used within any line. Style of lines is

```
<Option/Command>=["<Parameter>["<Space>["<Parameter>["]]  
or  
<Option/Command>="["<Parameter>["<Space>["<Parameter>["]]"
```

If a parameter contains spaces then the "-signs are required otherwise they are optional. Lines beginning with ; are ignored during processing. If your SPI line contains an " sign at the beginning and a " sign at the end of the line, then you may have to enclose the whole line with additional leading and ending " sign.

In cause of line by line processing, settings like SrcPrefix, DstPrefix and Application must be set before they will be used later.

Available Options/Commands

Var1=<Text>	User defined variable 1
Var2=<Text>	User defined variable 2
Var3=<Text>	User defined variable 3
Var4=<Text>	User defined variable 4
Var5=<Text>	User defined variable 5
SrcPrefix=<Path>	Sets prefix for all given source paths/files.
DstPrefix=<Path>	Sets prefix for all given destination paths/files.
Silent=1 0	Decide whether use client configuration or silent install. If 1, "Confirm" and "Display messages" is unchecked (within client configuration).
ConfirmOverwrite=1 0	If 1 User will be asked to confirm overwriting of files. Default is 0.
DeleteSPI=1 0	Delete SPI file after processing. Default is 1.



MsgBox=[<Text>]	<Text> will be displayed. If Text is missing, the Message parameter is used.
Message= <Text>	Message or question to display with next ConfirmContinue statement.
ConfirmContinue= <Text>	User will be asked if processing should be continued. <Text> will be displayed asking the user if the processing should be continued. If <Text> is missing the Message parameter is used.
Continue	If User choosed "No" at ConfirmContinue, Processing will be continued after the Continue statement if there is one in the SPI file.
Application= <ExeName>	Sets the application exename. Registry will be searched for the application within HKLM\Software\Microsoft\Windows\CurrentVersion\App Paths. This option must be used before the variable %APPATH% is available. If "ExeName" could not be found, SPI performing is canceled.
FindExe= <ExeName>	ExeName must be *.exe filename. It is searched within system path settings. Result is available using %FOUNDFILE% and %FOUNDPATH%. If "ExeName" could not be found, SPI performing is canceled.
CreateDir Mkdir= <Dst>	Create the folder <Dst>. All missing subfolders are created.
UpdateFile= <Src> <Dst>	Copies source file to destination folder/file. The file is only copied if the source file is a later version than the destination or if the destination doesn't exist. If no version information is available, the file date and time of the source and destination file will be compared and the newer one will be copied.
CopyFile= <Src> <Dst>	Copies source file to destination folder/file. The destination file will be overwritten if it exists.
RenameFile= <Src> <Dst> RenameFolder= <Src> <Dst>	Renames or moves source file/folder to destination folder/file. The destination must be on the same drive.
UpdateXDLL= <Src> <Dst> UpdateOCX= <Src> <Dst>	Similar to "UpdateFile" but before updating the file "Unregister" will be executed on destination file and after update "Register" will be executed on destination file.



UpdateXExe= <Src> <Dst>	Similar to "UpdateFile" but before updating the activex server component will be unregistered using the /UNREGSERVER parameter. After the update the exe will be executed with /REGSERVER parameter to reregister the activex server component.
InUseUpdate= <Src> <Dst>	Delay file update until reboot.
LaterUpdate= <Src> <Dst> [<Pause> <Attempts>]	Update file in <Pause> seconds. If error then <Attempts> attempts will be done (each after waiting <Pause> seconds again).
DeleteFile= <Fullname>	Deletes <Fullname>
DeleteTree= <Path>	Delete all files in <Path>. Also files in and the subdirectories are deleted. The folder specified is not deleted but will be empty afterwards. Attention with this command!
RemoveDir Rmdir= <Dst>	Remove/delete the folder <Dst>. Folder must be already empty.
Open OpenASync	Opens File with standard application. Only works asynchronously.
Exec ExecSync= <CmdLine>	Execute commandline synchronously (wait for process to finish).
ExecAsync= <CmdLine>	Execute commandline asynchronously (don't wait for process to finish).
ExecHiddenSync= <CmdLine>	Like ExecSync, but execute without showing a window.
ExecHiddenASync= <CmdLine>	Like ExecASync, but execute without showing a window.
AdminExecSync= <CmdLine>	Pass job to StupsiSvc Service to run the executable under the service's user account. The commandline is executed synchronously. Required precondition is that the StupsiSvc service is running on the workstation.
AdminExecASync= <CmdLine>	Pass job to StupsiSvc Service to run the executable under the service's user account. The commandline is executed asynchronously. Required precondition is that the StupsiSvc service is running on the workstation.
Register= <Fullname> UnRegister= <Fullname>	Register OCX, DLL using RegSvr32. Unregister OCX, DLL using RegSvr32.
IniFile= <Fullname> SetIni= <Section> <Key>	Set default inifile name



<Value> [<IniFile Name>]	Set/add entry into a specified Inifile. If the inifile parameter is missing the IniFile parameter is used.
RegAddStr=<HKey> <Hive> <Key> <Value>	Set registry entry. <Hkey> must be "HKLM", "HKCU" or "HKCR". Registry hive must already exist. Supports strings only.
RegAddKey=<HKey> <Hive> <SubKey>	Add registry key. <Hkey> must be "HKLM", "HKCU" or "HKCR". Registry hive must exist.
RegDelStr=<HKey> <Hive> <Key> <Value>	Delete existing registry entry. <Hkey> must be "HKLM", "HKCU" or "HKCR".
RegDelKey=<HKey> <Hive> <SubKey>	Delete registry key. <Hkey> must be "HKLM", "HKCU" or "HKCR". Registry Key must be empty.
AddRegRun=<CmdLine>	Adds an entry to the registry autorun key (HKLM\Software\Microsoft\Windows\Run)
DelRegRun=<CmdLine> <Exe>	Delete entry from registry autorun key (HKLM\Software\Microsoft\Windows\Run)
RegRunOnce=<CmdLine>	Adds an entry to the registry autorunonce key. (HKLM\Software\Microsoft\Windows\RunOnce).
GetDir="<Pattern> [/s]" "<name>"	Saves the directory information in a .txt file with the specified name in the computers folder on the network with the filename <Name>.
RestartWindows, RestartWin, Reboot[=0 1]	Reboot machine where 1 stands for forced shutdown (don't wait for applications to quit). 0 is default.
CloseApps[=t{;t}[s{s}]]	Informs the user to close all open applications if t and s strings both missing. If t strings are given then Stupsi searches for open applications with t as part of the window title. If a window is found then the user is asked to close this particular application. If s strings are given then windows containing this strings in the title are not processed.
AutoCloseApps[=t{;t}[s{s}]]	Informs the user to close all open applications if t and s strings both missing. If t strings are given then Stupsi searches for open applications with t as part of the window title. If a window is found then a close message is sent to the application window automatically. If s strings are given then windows containing this strings in the title are not processed.
KillApps[=t{;t}[s{s}]]	Same as AutoCloseApps but processes are killed.



Tries=<Number of Tries>	Because in some cases the user leaves applications open etc. so that an update cannot be completely performed, this option can be used to do some retries of executing the spi file. The spi file will not be moved immediately to the error folder but after <number of tries> attempts had failed.
RestartStupsi[=<Seconds>]	Should be used if Stupsi itself will be updated or if components should be updated that are used by Stupsi itself. This feature executes the LaterExe.exe in the Stupsi path and then quits Stupsi. This causes Stupsi be relaunched after waiting <Seconds>. Default is 3 Minutes. This option only is necessary if Stupsi is configured to remain loaded and keep searching for new software while the computer is running.
QuitStupsi	Quits Stupsi (and turns off searching for new software)
StartService=s	Starts the service s. It must be the display name of the service in the service list (within Control Panel).
StopService=s	Stops the service s. It must be the display name of the service in the service list (within Control Panel).
PauseService=s	Pauses the service s. It must be the display name of the service in the service list (within Control Panel).
ContinueService=s	Stops the service s. It must be the display name of the service in the service list (within Control Panel).
Wait=s	Just waits s seconds.
RunScript=s	Runs the script file s (vbs extension for vb script and js extension for java scripts). Available objects are "Glob" (wmGlob0.cGlob1) and "Config" (wmGlob0.cVarCol).
RegisterFile=s	Add a file to the file registration list. The current file version/date will be registered in the softvers.ini. If the file does not exist an appropriate information is placed in the version information. s must specify the full path and filename.
RequiresFile=<FileName>	Cancels SPI processing with an error if the specified file does not exist on the destination machine (full path and filename required). If you use this to ensure another SPI has been previously installed properly then you should also use the TRIES parameter.
RequiresOccurrence=<File>;<s>	Cancels SPI processing with an error if the specified string s cannot be found in the file <File> (full filename required) on the destination machine. You can use placeholders * and ? in the string s. If you use this to ensure another SPI has been previously installed properly then you should also use the TRIES parameter.
RequiresSPI=<SPI>;<Date>	Cancels SPI processing with an error if the specified SPI (<SPIName>.spi) has not yet been in-



stalled or installed last before date <Date>.
<Date> must be in format "yyyy-mm-dd". If you use this to ensure another SPI has been previously installed properly then you should also use the TRIES parameter.

AppendStrToFile= <File> <Str> Appends string <Str> to file named <File>.

Variables

%SYSTEMROOT% %WINDIR%	Windows folder (C:\WINNT, C:\WINDOWS for example).
%SYSDIR%	Windows system folder (C:\WINNT\SYSTEM32 for example).
%TEMPDIR%	Windows temporary folder (C:\TEMP for example)
%APPATH%	Application path (read out of the registry searching for the application exe which was set by the "Application" option above.
%COMPUTERNAME%	Computer name
%SERVERFOLDER%	Server folder configured within Stupsi (\\SERV1\Stupsi for example).
%SPIFOLDER%	Folder from where the SPI file was loaded (\\SERV1\Stupsi\Station1 for example).
%COMMONFILES%	Common Files folder
%COMMONDESKTOP%	Common Desktop folder
%COMMONSTARTUP%	Common Startup folder
%PROGRAMFILES%	Program Files folder
%USERSENDTO%	Users SendTo folder
%USERDESKTOP%	Users Desktop folder
%USERSTARTUP%	Users Startup folder
%VAR1%	User defined variable 1
%VAR2%	User defined variable 2
%VAR3%	User defined variable 3
%VAR4%	User defined variable 4
%VAR5%	User defined variable 5
%FOUNDFILE%	Full filename found with last "FindExe"
%FOUNDPATH%	Full path found with last "FindExe"
%WORKSTATION%, %HOST%	Name of the workstation (Computer name)
%NETUSER%, %USER%	Current user name
%COMPANY%	Company name (info from registry)
%OWNER%	Owner name (info from registry)
%INIFILE%	Assigned inifile