

LiveSystem pro PE Builder

User Manual (v.13)

Version 1.1.0.3 and higher
© www.kare-net.de 2011-2017

Created: June 30, 2017

Table of Contents

TABLE OF CONTENTS	2
1. FIRST STEPS	4
1.1. PURPOSE OF WINDOWSPE	4
1.2. BUILDING WINDOWSPE	4
1.3. SUPPORTED HOST COMPUTER SYSTEMS	4
1.4. SUPPORTED PE SYSTEMS	4
2. SYNTAX OF COMMANDS	5
2.1. SYSTEM	5
Convert	5
Echo	5
Exit	6
Extract	6
Halt	6
Math	6
Message	7
Pack	7
Return	7
Run	8
Set	8
Setmacro	8
SetMvar	9
SYSTEM	9
Wait	11
Wim Commands	11
Update	12
2.2. PROGRAMMABLE CONDITIONS (IF CONDITIONS)	12
If,...,Then	12
If,...,Then,...,Else-Block	13
2.3. CONDITIONS FOR DIRECTORIES, FILES AND INTERFACES	14
2.3.1. Directories	14
DirCopy	14
DirRemove	14
DirMove	14
DirMake	15
2.3.2. Files	15
2.3.2.1. <i>Conditions for Files</i>	15
FileCopy	15
FileDelete	15
FileRename	15
FileCreateBlank	16
FileRead	16
FileSetAttrib	16
CopyFile	17
CopyOrExpand	17
Expand	18
GetDistLang	18
ReqFile	18
Require_File	19
Retrieve	19
Search File	20
SetReqFile	20
ShellExecute	20
Start	21
2.3.2.2. Word Processing	21
TXTAddLine	21
TXTReplace	22

StrFormat	22
2.3.2.3. INI Processing	22
IniWrite	22
IniWriteSection	23
IniDeleteSection	23
IniRead	23
IniReadPos	23
IniRenameSection	24
2.3.3. Interface Processing	24
Interface,Get	24
Interface,Set	25
Interface,State	25
Interface,Update	25
2.4. REGISTRY PROCESSING	26
RegLoad	26
RegUnLoad	26
RegRead	26
RegWrite	27
RegDelete	28
2.5. STRING PROCESSING	28
String LEFT/RIGHT	28
String TLeft/TRight	29
String FIND	29
String LEN	29
String MID	30
String REPLACE	30
String SPLIT	30
2.6. NETWORK COMMANDS	31
WebGet	31
2.7. LOOP COMMANDS	31
2.7.1. For / Next - loop	32
2.7.2. ExitLoop	32
2.8. VARIABLES (INTERN)	33
2.9. VARIABLES (PROJECT)	33
2.10. SYSTEM,SETPEINIT	35
2.11. SPECIAL CHARACTERS	37
2.12. SHORTCUTS	37
2.13. MACROS	38
3. SCRIPT CONTROL	39
3.1. STANDARD SECTIONS OF SCRIPTS	40
3.1.1. Section [Main]	40
3.1.2. Section [Interface]	40
3.1.3. Section [Process]	41
3.2. ACTIVE SCRIPT CONTROL	41
3.2.1. Choosing Windows Source	41
3.2.2. Script-Select	42

Comment:

Optional commands are written as

[Command] resp. *[<Command>]*

1. First Steps

1.1. Purpose of WindowsPE

Windows PE (Windows PreinstallationEnvironment) is excellent suitable for following range of applications:

- ⇒ Repairing Computers
- ⇒ Performing Data Recovery
- ⇒ Preparing a reinstallation of Operating System
- ⇒ Removing a virus
- ⇒ Rebuilding private data (documents and pictures)
- ⇒ Restoration of images files

[Back to Table of Contents](#)

1.2. Building WindowsPE

Video manual in forum **LiveSystem-pro.de**:

<http://www.livesystem-pro.de/showthread.php?92-Videolanleitung-1-quot-LiveSystem-pro-quot-PE-Builder>

[Back to Table of Contents](#)

1.3. Supported Host Computer Systems

At moment following systems will be supported:

- | | | |
|---|-----------------|-----------------------------------|
| ⇒ | Windows XP/2003 | ServicePack 3 (32/64-bit) |
| ⇒ | Windows Vista | with all ServicePacks (32/64-bit) |
| ⇒ | Windows 7 | with all ServicePacks (32/64-bit) |
| ⇒ | Windows 8/8.1 | (32/64-bit w/o ServicePack) |
| ⇒ | Windows 10 | (32/64-bit w/o ServicePack) |

[Back to Table of Contents](#)

1.4. Supported PE Systems

LiveSystempro is available with 4 projects:

- | | | |
|---|--------|---|
| ⇒ | WinXPE | Windows XP, Windows 2003 (32/64-bit) |
| ⇒ | Win7PE | Windows Vista (32/64-bit)*
Windows 2008 (32/64-bit)*
Windows 7 (32/64-bit)* |

		Windows 2010 (32/64-bit)* * = all ServicePacks
⇒	Win8PE	Windows 8 (32/64-bit) Windows 8.1 (32/64-bit) Windows 2010 R2 (32/64-bit), Windows 8.1 Update 1 (32/64-bit)
⇒	Win10PE	Windows 10 (32/64-bit)

All of them are integrated in LiveSystempro and selectable separately.

[Back to Table of Contents](#)

2. Syntax of Commands

2.1. System

Convert

Converts decimal and hexadecimal values

Syntax:

Convert,<%Variable%>,<Method>,<Value>

Method:	-	<i>Hex2Dec</i>	converts a hexadecimal value into a decimal one
		<i>Dec2Hex</i>	converts a decimal value into a hexadecimal one
		<i>Dec2chr</i>	converts a decimal value into Ascii code (Unicode will be supported)

Examples:

Convert,%Var%,Hex2Dec,401a
Allocates to variable *%Var%* the value **16410** .

Convert,%Var%,Dec2Hex,16410
Allocates to variable *%Var%* the value **401a** .

Convert,%Var%,Dec2chr,65
Allocates to variable *%Var%* the value **A** .

[Back to Table of Contents](#)

Echo

Shows an information

Syntax:

Echo,<Text>

Example:

Echo,Running script

[Back to Table of Contents](#)

Exit

Quits the running script

Syntax:

Exit

Example:

If,%Arch%,x64,Then,Exit

[Back to Table of Contents](#)

Extract

Extracts data from a script file

Syntax:

Extract.<Scriptfile>,<Attachment>,<TargetPath>

Example:

Extract,%ScriptFile%,ReBoot-x86.dll,%Target_Sys%\ReBoot.dll

Extracts from running script the file *ReBoot-x86.dll* and stores it as *ReBoot.dll* in folder *System32*.

[Back to Table of Contents](#)

Halt

Stops accomplishment of build

Syntax:

Halt,<Text>

Text - any text (optionally)

Example:

Halt,Aborted by user

[Back to Table of Contents](#)

Math

Performs a mathematic counting

Syntax:

Math,<%Variable%>,<Value>,<PARAMETER>

%Variable% - any name
Value - mathematic counting
Parameter (optional) - Round_Up / Round_Down

Examples:

*Math,%NewVar%,45+(7*4)*

Allocates to variable *%NewVar%* the value of **73** (sum of 45 + 28)

Math,%NewVar%,16.9-10,Round_Down

Allocates to variable *%NewVar%* the value of **6** in round terms

[Back to Table of Contents](#)

Message

Shows a message box

Syntax:

Message, <Text>[, <Title>], [<Timeout>]

<i>Text</i>	-	Text in message box
<i>Title</i>	-	Title of message box
<i>Information</i>		shows symbol Information
<i>Confirmation</i>		shows symbol Question mark
<i>Error</i>		shows symbol Error
<i>Question</i>		shows a button YES/NO
<i>Warning</i>		shows symbol Warning
<i>Timeout</i>	-	Display duration of box in seconds

Examples:

Message, Hallo World, Information, 10

Message, Continue?, Question

If, %Exitcode%, =, Yes, Then, Message, Button YES has been pressed

If, %Exitcode%, =, No, Then, Message, Button NO has been pressed

[Back to Table of Contents](#)

Pack

Compresses or decompresses a file

Syntax compress command:

Pack, <Compress>, <Targetfolder>, <File>

For files with extension '.zip' only

Syntax decompress command:

Pack, <Decompress>, <Filepath>, <Folder>[, <Filter>]

For files with extension '.zip' and '.7z'.

Using the char * the filter changes to partial search (similar to a Wildcard function)

Examples:

pack, compress, C:\Test, C:\Test.zip

Compresses the folder *Test* to ZIP file *Test.zip*.

pack, decompress, C:\RegConfig.7z, C:\MyFolder, Win8x86 productoptions.reg

Decompresses from 7z file *RegConfig.7z* file *Win8x86* and *productoptions.reg*.

*pack, decompress, C:\Test.7z, C:\Test, *.pdf*

Decompresses from 7z-file *Test.7z* all files with extension *pdf*.

[Back to Table of Contents](#)

Return

Quits the called section of a script

Syntax:

Return

Example:

Run, %ScriptFile%, MySection

Message, %Arch%

```
[MySection]
If,%Arch%,x64,Then,Return
Message,Fitting architecture
```

The message box **Fitting architecture** is available with Architecture x86 only.

[Back to Table of Contents](#)

Run

Performs a section in the specified script

Syntax:

```
Run,<ScriptPath>,<Section>[,<PARAMETER>]
```

<i>ScriptPath</i>	-	a.	Full path of script
		b.	Calling a section in the running script typing <i>%ScriptFile%</i> is enough
<i>Section</i>	-		Section to perform in script
<i>PARAMETER</i>	-		9 Parameters to commit (#1 up to #9)

Example:

```
Run,%ScriptFile%,MySection,Hallo
```

```
[MySection]
Message,#1,Information
```

In this example a message box will appear with text **Hallo**.

[Back to Table of Contents](#)

Set

Allocates a value to a Variable

Syntax:

```
Set,<%Variable%>,<Value>[,<PERMANENT>]
```

<i>Value</i>	-	any value or path
<i>PERMANENT</i>	-	valid for the whole build, stored in <i>project.cfg</i> in section [variables]

Examples:

```
Set,%NewVar%,myPE
Allocates to variable %NewVar% the value myPE.
```

```
Set,%NewVar%,myPE,PERMANENT
Allocates to variable %NewVar% the value myPE, valid for the whole build.
```

[Back to Table of Contents](#)

Setmacro

Creates a macro

Syntax:

```
Setmacro,<Macroname>,<Command>[,<PERMANENT>]
```

<i>Macroname</i>	-	any name without %%
<i>Command</i>	-	any valid build command

PERMANENT - valid for the whole build, stored in *project.cfg* in section [Macros]

Example:

Setmacro,require_file,Run,%ProjectDir%\Macro.srt,Process-ReqSys32,#1,PERMANENT
Creates the macro *require_file* with command
Run,%ProjectDir%\Macro.srt,Process-ReqSys32,#1,PERMANENT

[Back to Table of Contents](#)

SetMvar

Creates a macrovariable

Syntax:

SetMvar,<%Variable%>,<Value>

Example:

SetMvar,%MyVariable%,myPE
Creates a macrovariable with value *myPE*.
The macrovariable is valid in running script only!

[Back to Table of Contents](#)

SYSTEM

Command for System settings

Syntax:

System,<Systemcommand>,<Systemparameter>

Systemcommand:	<i>Error</i>	Enables / Disables error messages
	<i>FreeRam</i>	Shows free RAM
	<i>TotalRam</i>	Shows total RAM
	<i>GetDate</i>	Shows Date and Time
	<i>GetEnv</i>	reads Environment Variables
		<i>Temp</i>
		<i>ProgramFiles</i>
		<i>Number_Of_Processors</i>
	<i>GetFreeDrive</i>	Shows the next free available Drive Letter
	<i>GetFreeSpace</i>	Shows the available free Space
	<i>GetDriveType</i>	Shows medium's type
	<i>Log</i>	Enables / deactivates writing of LOG file
	<i>ProcessMode</i>	Launches the GUI in Process mode
	<i>SetPEinit</i>	Writes an entry in <i>Peinit.ini</i> (see section 2.10)

Systemcommand (x64 systems only):

<i>FileRedirect</i>	Turns FileRedirection (Wow64) for 64-bit systems on or off
<i>RegRedirect</i>	Turns RegRedirection (Wow64) for 64-bit systems on or off

Systemparameter:

concerning <i>Error</i>	On Off
concerning <i>FileRedirect</i>	On Off
concerning <i>FreeRam</i>	%Variable%
concerning <i>TotalRam</i>	%Variable%
concerning <i>GetDate</i>	Format - %yyyy - Year (4-digit) %yy - Year (2-digit) %mm - Month (2-digit) %dd - Day (2-digit) %hh - Hour (2-digit)

		%ii	- Minute (2-digit)
		%ss	- Second (2-digit)
concerning <i>GetFreeDrive</i>	%Variable%		
concerning <i>GetFreeSpace</i>	%Variable%		
concerning <i>GetDriveType</i>	Return value		
	1 -	no Root	
	2 -	portable Device	
	3 -	Harddisk	
	5 -	CDROM	
concerning <i>SetPEinit</i>	%Variable%, [<Section>]		
concerning <i>Log</i>	On Off		
	<i>Log, On, 1</i>	Enables writing of LOG file and	
		deletes an existing LOG file	
concerning <i>RegRedirect</i>	64 Off		

Examples:

System, *FILEREDIRECT*, On

System, *FILEREDIRECT*, Off

System, *REGREDIRECT*, 64

System, *REGREDIRECT*, Off

System, *GETDATE*, %yyyy, %Var%

Allocates to variable *%Var%* the current year (four-digit)

System, *GETFREEDRIVE*, %FreeLetter%

Allocates to variable *%FreeLetter%* the next available free drive letter.

System, *GETFREESPACE*, %Drive%, %Size%

Example:

Set, %Txt%,

For, %Var%, 65, 90

Convert, %Drive%, Dec2chr, %Var%

System, *GetFreeSpace*, %Drive%:\, %Size%

If, %Size%, >, 0, Then, Set, %Txt%, %Txt% %Drive% - %Size%# \$x

Next

String, %Txt%, TRight, %Txt%, 2

Allocates to variable *%Txt%* all partitions with current free space.

Notation:

65 and **90** are the ASCII values of letters **A** and **Z**.

System, *GETDRIVETYPE*, %Drive%, %Type%

Example:

Set, %Txt%,

For, %Var%, 65, 90

Convert, %Drive%, Dec2chr, %Var%

System, *GetDriveType*, %Drive%:\, %Type%

If, %Type%, =, 3, Then

System, *GetFreeSpace*, %Drive%:\, %Size%

If, %Size%, >, 0, Then, Set, %Txt%, %Txt% %Drive% - %Size% MB# \$x

End

Next

String, %Txt%, TRight, %Txt%, 2

Allocates to variable *%Txt%* the free space of the chosen type.

System, *GetEnv*, %Var%, ProgramFiles

Allocates to variable *%Var%* the full path of *%ProgramFiles%*.

System, *GetEnv*, %Var%, Temp

Allocates to variable *%Var%* the full path of folder *%Temp%*.

(e.g.: C:\Users\<Username>\AppData\Local\Temp

System, *GetEnv*, %Var%, Number_Of_Processors

Allocates to variable *%Var%* the number of processors

Wait

Pauses the script by the specified milliseconds

Syntax:

Wait,<Milliseconds>

Example:

Wait,1000

(1 second = 1000 milliseconds)

[Back to Table of Contents](#)

Wim Commands

Commands processing Image files

Syntax:

Wim,Create,<PackMethod>,<Folder>,<WimFile>

PackMethod	-	none (unpacked)
		LZMS (LZMS Compression)

Wim,Extract,<Mode>,<WimFile>,<%WimIndex%>,<Parm1>[,<...>]

Mode	-	File	extracts a file or a folder
		FileList	extracts files and folders from an existing list
		Image	extracts a complete image
%WimIndex%	-	Numeric value starting with 1 (system variable)	
Parm1	-	concerning File	<i><File/Folder>,<TargetFolder></i>
		concerning FileList	<i><FileList>,<TargetFolder></i>
		concerning Image	<i><TargetFolder></i>

Wim,Info,<WimFile>,<%WimIndex%>,<ScriptFile>

%WimIndex%	-	Numeric value starting with 1 (system variable)	
-------------------	---	---	--

Wim,InfoEdition,<WimFile>,<ScriptElement>,<ScriptFile>

ScriptElement	-	supported is ,pScrollBar' only	
----------------------	---	--------------------------------	--

Path to Image:

Win7PE, Win8PE+Win10PE: Usage of var **%Installwim%** (compare section 2.9)

Examples:

Wim,Create,lzms,%TargetDir%,%Target_CD%\sources\Boot.wim
Compresses Boot.wim using compression method **lzms**.

Wim,info,%SourceDir%\sources\install.wim,%WIM_Index%,%ProjectDir\project.cfg
Reads the **WIM Index** from install.wim and writes it in **project.cfg**.

Wim,infoEdition,,%SourceDir%\sources\install.wim,pScrollBar1,%ProjectDir%\Build\0-PreConfig.srt
Writes informations of all images from install.wim in file **0-PreConfig.srt**.

Wim,Extract,File,%SourceDir%\sources\install.wim,%WIM_Index%,#1,%InstallSRC%
Extracts the file with parameter **#1** to **%InstallSRC%**.

Wim,Extract,Image,%SourceDir%\sources\install.wim,%WIM_Index%,%InstallSRC%
Extracts the content of a determined WIM Index from Install.wim to **%InstallSRC%**.

*Wim,Extract,FileList,%SourceDir%\Sources\Install.wim,%WIM_Index%,%ProjectTemp%\Instwim.txt,%Basedir%\Wim\OS%%SourceE
dition%%arch%%SP%%Distlang%-I\%WIM_Index%*

Extracts to the target directory files based on the content of *Instwim.txt*.

Wim,Extract,Image,f:\boot.wim,2,f:\test5

Extracts boot.wim to directory *test5*.

[zurück zum Inhaltsverzeichnis](#)

Update

Refreshes the variable data memory

Syntax:

Update,<Parameter>[,ScriptDir]

Parameter:	<i>ProjectVar</i>	all variables from <i>project.cfg</i> , section <i>[variables]</i> will be loaded into the variable data memory
	<i>InterfaceVar</i>	variables from section <i>[Interface]</i> of any script will be loaded into the variable data memory
	<i>Project</i>	all scripts will be loaded again
	<i>Project,ScriptDir</i>	script directory will be loaded again

Examples:

Update,ProjectVar

Update,InterfaceVar

Update,Project

Update,Project,Apps\%pTextBox111%.srt

[Back to Table of Contents](#)

2.2. Programmable Conditions (If Conditions)

If,...,Then

Syntax:

If,<Condition>,<%Variable%>,Then,<Command>

Negative Syntax:

If,Not,<Condition>,<%Variable%>,Then,<Command>

Available Conditions:

ExeRunning
ExistFile
ExistDir
ExistSection
ExistRegKey
ExistWebFile
SrtChecked
= (equal)
< (smaller)
> (bigger)

Examples:

If,%Var%>=,1,Then,Message,Value is equal 1

This message will be shown only if value of *%Var%* is equal *1*.

If,3,>,2,Then,Message,3 is bigger than 2

This message will be shown always as *3* is bigger than *2*.

If,ExistFile,%ScriptDir%\test.exe,Then,Message,File does exist

This message will be shown only if *test.exe* does exist.

If,ExistDir,C:\Programs,Then,Message,Directory does exist
This message will be shown only if directory **C:\Programs** does exist.

If,Not,ExistDir,C:\Intel,Then,Message,Directory does not exist
This message will be shown if directory **C:\Intel** does **NOT** exist.

If,ExistRegKey,HKLM,PE-SYSTEM\ControlSet001\Services\v8042prt\Parameters,EnableWheelDetection,Then,Message,Value exists
This message will be shown if value **EnableWheelDetection** does exist in mentioned key.

If,ExistSection,%target_win%\TXTSETUP.SIF,SourceDiskFiles,Then,Message,Section does exist
This message will be shown if section **SourceDiskFiles** does exist in file **TXTSETUP.SIF**.

If,ExistRegKey,HKLM,PE-SYSTEM\ControlSet001\Services\v8042prt\Parameters,Then,Message,Key exists
This message will be shown if the complete RegKey does exist.

If,EXISTREGKEY,HKLM,PE-SYSTEM\ControlSet001\Services\v8042prt\Parameters,,Then,Message,exists
This message will be shown if value (**Default**) does exist in mentioned key.

If,SrtChecked,Build\8-HW-pnp.srt,Then,Message,Script is enabled
This condition proves if the script has been enabled.

If,ExistWebFile,http://www.xyz.com/xyz.zip,Then,Message,File is available
This message will be shown only if **xyz.zip** is available.

If,ExeRunning,vmware.exe,Then,Message,VMware is already running
This message will be shown only if **VMware** is already running.

Tip to multiline notation:

This notation is also possible:

```
If,%Var%,=,1,Then
    Message,Value is 1
End
```

Using this notation the command **END** is **ALWAYS** necessary !

[Back to Table of Contents](#)

If,....,Then,....,Else-Block

All above mentioned conditions can also be used as **EITHER – OR Block**

Syntax:

```
If,<Condition>,<%Variable%>,Then
    <Command_#1>
Else
    <Command_#2>
End
```

Examples:

```
If,ExistDir,C:\Intel,Then
    Message,Directory does exist
Else
    Message,Directory does not exist
End
```

This condition shows **either** the first **or** the second message depending on the result.

```
If,%Var%,=,1,Then
    Message,Value is equal 1
Else
    Message,Value is not equal 1
End
```

Same result as written above.

Combination with **NOT** is also possible.

Example:

```
If,Not,%Var%,=,1,Then
    Message,Value is unequal 1
Else
    Message,Value is equal 1
End
```

[Back to Table of Contents](#)

2.3. Conditions for Directories, Files and Interfaces

2.3.1. Directories

DirCopy

Copies the content of a folder into another folder

Syntax:

DirCopy,<SourceFolder>,<TargetFolder>

Example:

DirCopy,C:\MyFolderC:\MyOtherFolder
Copies the content of *C:\MyFolder* to *C:\MyOtherFolder*.
If the targetfolder does not exist it will be created.

[Back to Table of Contents](#)

DirRemove

Deletes a directory

Syntax:

DirRemove,<Files/FolderToDelete>[,recursive]

Recursive:	<i>1</i>	deletes folders and possible subfolders
	<i>0</i>	default value, deletes only the files in the quoted folder

Examples:

DirRemove,C:\MyFolder,1
Deletes the folder *C:\MyFolder* and all subfolders.

DirRemove,C:\MyFolder,0
Deletes all the files in folder *C:\MyFolder*.

DirRemove,C:\MyFolder\MySubfolder,0
Deletes all the files in subfolder *MySubfolder*.

[Back to Table of Contents](#)

DirMove

Moves a directory

Syntax:

DirMove,<FolderToMove>,<TargetFolder>

Example:

DirMove, C:\MyFolder, C:\MyOtherFolder
Moves the **content** of *C:\MyFolder* to *C:\MyOtherFolder*.
The origin folder will be deleted.

[Back to Table of Contents](#)

DirMake

Creates a folder

Syntax:
DirMake, <NewFolder>

Example:
DirMake, C:\MyFolder
Creates the new folder *C:\MyFolder*.

[Back to Table of Contents](#)

2.3.2. Files

2.3.2.1. Conditions for Files

FileCopy

Copies a single source file to a specified destination

Syntax:
FileCopy, <SourceFile>, <TargetFolder>

Example:
FileCopy, C:\MyFolder\MyFile.txt, C:\MyOtherFolder
Copies the file *MyFile.txt* from *C:\MyFolder* to *C:\MyOtherFolder*.
Wildcards (*.*) are supported.

[Back to Table of Contents](#)

FileDelete

Deletes a single file

Syntax:
FileDelete, <File>

Example:
FileDelete, C:\MyFolder\MyFile.txt
Deletes the specified file.

[Back to Table of Contents](#)

FileRename

Renames an existing file

Syntax:
FileRename, <OldFilename>, <NewFilename>

Example:

FileRename,C:\MyFolder\MyFile.txt,C:\MyFolder\MyNewFile.txt
Renames the file *MyFile.txt* to *MyNewFile.txt*

[Back to Table of Contents](#)

FileCreateBlank

Creates an empty file (an existing same-titled file will be deleted)

Syntax:

FileCreateBlank,<NewFile>[,<Encoding>]

CharacterSet (optional) - ANSI
Unicode
UTF-8

Examples:

FileCreateBlank,C:\MyFolder\test.txt
Creates in the specified folder the empty file *test.txt*

FileCreateBlank,C:\MyFolder\test.txt,utf8
Creates in the specified folder the empty file *test.txt* with charset *UTF-8*

[Back to Table of Contents](#)

FileRead

Reads the whole content of a file into a variable

Syntax:

FileRead,<%Variable%,<File>[,<Counter>]

Counter: - optional parameter (number of chars to read)

Examples:

FileRead,%Var%,C:\MyFolder\test.txt
Reads the whole content of *test.txt* into variable *%Var%*.

FileRead,%Var%,C:\MyFolder\test.txt,20
Reads 20 chars of the content of *test.txt* into variable *%Var%*.

Possible action when cursor is placed undesired at the begin of a blank line:

(*%Var%=C:\MyFolder\test.txt*)
String,%Result%,RIGHT,%Var%,2
If,%Result%,=#\$x,Then,String,%Var%,TRIGHT,%Var%,2
Sets the cursor **next** to the last existing char.

[Back to Table of Contents](#)

FileSetAttrib

Changes attributes of a file

Syntax:

FileSetAttrib,<File>,<Attribute>[,<Parameter>]

Available Attributes: *R* ReadOnly
 A Archive
 S System

<i>H</i>	Hidden
<i>N</i>	Normal
<i>O</i>	Offline
<i>T</i>	Temporary

Parameter: *1* Recursive

Using Wildcards (*) is possible.
Attributes will be changed by using **+** or **-**

Examples:

FileSetAttrib,C:\MyFolder\test.exe,+RS
FileSetAttrib,C:\MyFolder.doc,+RS*
FileSetAttrib,C:\MyFolder.doc,-H*

FileSetAttrib,C:\MyFolder\,+H
Changes the attribute of a single folder.

FileSetAttrib,C:\MyFolder\,+H,1
Changes the attribute of a folder with all included subfolders.

[Back to Table of Contents](#)

CopyFile

Copies a file from Sourcefolder to Targetfolder.
This command is suitable for projects **Win7PE**, **Win8PE** and **Win10PE**.

Note:

The command **SetReqFile** has to be used previously to define directories and parameters

Syntax:

CopyFile,<File>[,Mode]

Mode	0	-	w/o .mui file
	1	-	with .mui file + Fallback en-US
	2	-	with .mui file w/o Fallback en-US

Example:

SetReqFile,%boot_sys%,%DistLang%,%Fallback%,%target_sys%
CopyFile,drivers\acpi.sys,1
CopyFile,atmlib.dll
Copies the file *drivers\acpi.sys* to *%target_sys%*.
Copies the file *drivers\acpi.sys.mui* to *%target_sys%\%DistLang%*.
Copies the file *atmlib.dll* to *%target_sys%*.

[Back to Table of Contents](#)

CopyOrExpand

Copies a file and expands it if necessary
If the file cannot be found the builder searches for an extension with underline or for an abbreviated Extension

Syntax:

CopyOrExpand,<File>,<Targetfolder>

Example:

CopyOrExpand,%source_win%\SHELL32.DLL,%target_sys%
Copies the file *SHELL32.DLL* to *%target_sys%*.

[Back to Table of Contents](#)

Expand

Extracts files from a CAB file

Syntax:

Expand,<CAB_File>,<TargetFolder>[,<Filter>]

Filter Expands a single file or multiple files, separated using char :

The Targetfolder has to exist and will **NOT** be created automatically.

Examples:

Expand,C:\MyFolderMyFile.cab,C:\MyNewFolder
Extracts all files inside *MyFile.cab* to *C:\MyNewFolder*.

Expand,%source_sys%\%SPCabFile%,%target_sys%\drivers,aec.sys:agp440.sys
Extracts in Project WinXPE from CAB file (e.g.: *SP3.cab*) the files *aec.sys* and *agp440.sys* to *%target_sys%\drivers*.

Expand,%source_sys%\%SPCabFile%,%target_sys%
Extracts in Project WinXPE from CAB file (e.g.: *SP3.cab*) the file *bthci.dll* to *%target_sys%*.

[Back to Table of Contents](#)

GetDistLang

Reveals entering a folder or a file path the folder name with the best language file.
If there is no language file (.mui) available you will get an empty string.

Syntax:

GetDistLang,<%Variable%>,<Folder/File>,<Mui-List>[,<Mode>]

Folder/File	-	Folder or mother file
Mui-List	-	Folder name with mui files e.g. es-MX:de-DE:en-US (Entries separated by colon)
Mode	-	0 (empty issue will get ignored) 1 (empty issue results in error message)

Examples:

GetDistLang,%Var%,%boot_sys%\boot,%DistLang%:%FallBack%
Allocates to variable *%Var%* the value *de-DE* using a german source.

GetDistLang,%Var%,%boot_sys%\shell32.dll,%DistLang%:%FallBack%
Allocates to variable *%Var%* the value *de-DE* using a german source.

GetDistLang,%Var%,%boot_sys%\srms62.dat,%DistLang%:%FallBack%,1
Results in an error message as there is no .mui file available for *srms62.dat*.

[Back to Table of Contents](#)

ReqFile

Verifies if the file already exists in target. If not so the file will be copied from Sourcefolder to Targetfolder.
This command is suitable for projects **Win7PE**, **Win8PE** and **Win10PE**.

Note:

The command **SetReqFile** has to be used previously to define directories and parameters

Syntax:

ReqFile,<File>[,Mode]

Mode	0	-	no .mui file
	1	-	with .mui file + Fallback en-US
	2	-	with .mui file w/o Fallback en-US

Example:

SetReqFile,%Install_sys%,%DistLang%,%Fallback%,%target_sys%
ReqFile,accessibilitycpl.dll,1
ReqFile,control.exe
 Copies the file *accessibilitycpl.dll* to *%target_sys%*.
 Copies the file *accessibilitycpl.dll.mui* to *%target_sys%\%DistLang%*
 Copies the file *control.exe* to *%target_sys%*.

[Back to Table of Contents](#)

Require_File

Verifies if the file already exists in target. If not so the file will be copied from Sourcefolder to Targetfolder. This command is used in project **WinXPE**.

Syntax:

Require_File,<File>

Example:

Require_File,shell32.dll
 Copies the file *shell32.dll* to *%target_sys%*.

[Back to Table of Contents](#)

Retrieve

Query of special file informations

Syntax:

Retrieve,<%Variable%,<Parameter>,<FolderOrFile>,[Param2],[Param3]

Parameter	-	DirSize	
		FileSize	
		FileVersion	
		MD5	
		FileDate	
		ResStr	(Resources string)
		FileArch	(for EXE files only)
FileDate only:		Param2	Modified
			Created
ResStr only:		Param2	optional [shows LCID (locale identifiers)]
		Param3	optional [shows resource number]

Examples:

Retrieve,%Var%,DirSize,C:\MyFolder
 Allocates to variable *%Var%* the size of *MyFolder* .

Retrieve,%Var%,FileArch,C:\Windows\explorer.exe
 Allocates to variable *%Var%* the architecture of *explorer.exe* .
ONLY EXE files are supported.

Retrieve,%Var%,MD5,C:\boot.ini
 Allocates to variable *%Var%* the MD5 sum of file *boot.ini* .

Retrieve,%Var%,ResStr,C:\Windows\system32\shell32.dll, 0000409,21771

Allocates to variable **%Var%** the value *Documents and Settings* .

Retrieve, %Var%, FileDate, %BaseDir%\Temp\update.srt, Modified
Allocates to variable **%Var%** the alteration date of the mentioned file.

[Back to Table of Contents](#)

Search File

Searches for a file. Wildcards are supported

Syntax:

Search, <%Variable%>, <Parameter>, <Directory>, <Filename>

Parameter	-	File	searches in chosen directory for files with same file extension
		FileRec	recursive search for maximum 2000 files

Beispiel:

```
[Interface.07]
pScrollBar3=,1,0,0,20,270,215,20
...
[Script-Select]
search,%Var%,File,%ISODir%,*.iso
Interface,set,pScrollBar3,%Var%
Searches for an existing chosen file in %ISODir% and updates the element
pScrollBar3 e.g. with following line:
[Interface.07]
pScrollBar3=Win7[SP0]_de-DE_x86.iso,1,0,0,20,270,215,20,Win7[SP0]_de-DE_x86.iso
```

[Back to Table of Contents](#)

SetReqFile

Defines the directories and parameters for commands **Copyfile** and **Reqfile**

Syntax:

SetReqFile, <SourceFolder>, %DistLang%, %Fallback%, <TargetFolder>

SourceFolder	-	any folder
%DistLang%	-	Directory in %WinDir% with localized language file (MUI)
%Fallback%	-	Mui list (entries separated by “:”)
TargetFolder	-	any folder

Example:

```
SetReqFile,%boot_sys%,%DistLang%,%Fallback%,%target_sys%
Defines %boot_sys% as source folder and %target_sys% as target folder.
```

(See also the explanations of commands **Copyfile** and **Reqfile**)

[Back to Table of Contents](#)

ShellExecute

Starts a program or a file

Syntax:

ShellExecute, <Action>, <File>[, <Parameter>][, <WorkingDir>]

Action	-	Show	Process will be shown
		Hide	Process will be hidden

Parameter - optional set of switches

WorkingDir - optional

Return value of ShellExecute: %ExitCode%
 Exitcode 0: no error
 Exitcode 1-32: different errors

Examples:
ShellExecute,Show,C:\Windows\system32\cmd.exe
 This command opens a console window.

Note:
 The command *ShellExecute* in a script effects the script to wait for the end of the launched file / program. Afterwards the remaining lines in script will be executed.

[Back to Table of Contents](#)

Start

Starts a program without waiting

Syntax:
Start, <Action>, <File>[, <Parameter>][, <WorkingDir>]

Action - *Show* Process will be shown
Hide Process will be hidden

Parameter - optional set of switches

WorkingDir - optional

Example:
Start,Start,%WindowsDir%\Notepad.exe,%ScriptFile%
 This command opens the running script with program *Notepad*.

[Back to Table of Contents](#)

2.3.2.2. Word Processing

TXtAddLine

Adds a line inside a file

Syntax:
TXtAddLine, <File>, <Text>, <Parameter>

Parameter - *Top* adds line as first line
Bottom adds line as last line
Line number (first line = 0)

Example:
TXtAddLine,C:\MyFile.txt,My new Line,Top
 Adds to *MyFile.txt* the line *My new Line* as first line.
 If the file does not exist it will be created.

[Back to Table of Contents](#)

TXTReplace

Renames an existing text

Syntax:

TXTReplace,<File>,<OldText>,<NewText>

Example:

TXTReplace,C:\MyFile.txt,My old Line,My new Line
Replaces in *MyFile.txt* the line *My old Line* by *My new Line*.
If the file does not exist an empty file will be created.

[Back to Table of Contents](#)

StrFormat

Returns some parts of a path name

Syntax:

StrFormat,<%Variable%>,<Parameter>,<File>

Parameter	-	<i>FILENAME</i>	File name
		<i>FILENAMEEX</i>	File name w/o extension
		<i>PATH</i>	File path
		<i>EXT</i>	File extension
		<i>SHORTPATH</i>	Path in format DOS 8.3

Examples:

StrFormat,%Var%,FILENAME,C:\boot.ini
Allocates to variable *%Var%* the value *boot.ini* .

StrFormat,%Var%,FILENAMEEX,C:\boot.ini
Allocates to variable *%Var%* the value *boot* .

StrFormat,%Var%,PATH,C:\boot.ini
Allocates to variable *%Var%* the value *C:* .

StrFormat,%Var%,EXT,C:\boot.ini
Allocates to variable *%Var%* the value *ini* .

StrFormat,%Var%,SHORTPATH,C:\MyNewFile.txt
Allocates to variable *%Var%* the value *C:\MYNEWF~1.TXT* as far as the file does exist.

[Back to Table of Contents](#)

2.3.2.3. INI Processing

IniWrite

Writes a value in a section of an INI file

Syntax:

IniWrite,<File>,<Section>,<Key>,<Value>

Example:

IniWrite,C:\LiveSystempro\LiveSyspro.ini,Main,Project,Win7PE
Creates in file *LiveSyspro.ini* in section *[Main]* the entry *Project=Win7PE*

[Back to Table of Contents](#)

IniWriteSection

Creates a new section in an INI file with optional text settings.
Entries will be written one below the other.

Syntax:

IniWriteSection, <File>, <Section>[, Text]

Examples:

IniWriteSection, C:\LiveSystempro\LiveSyspro.ini, NewSection,

Creates in file *LiveSyspro.ini* the section *[NewSection]*.

The final comma is an optional comma, thus this notation is also possible:

IniWriteSection, C:\LiveSystempro\LiveSyspro.ini, NewSection

IniWriteSection, C:\LiveSystempro\LiveSyspro.ini, NewSection, MyKey

Creates in file *LiveSyspro.ini* the section *[NewSection]* with entry *MyKey*.

If the INI file does not exist it will be created.

[Back to Table of Contents](#)

IniDeleteSection

Deletes an existing section inside an INI file

Syntax:

IniDeleteSection, <File>, <Section>

Example:

IniDeleteSection, C:\LiveSystempro\LiveSyspro.ini, NewSection

Deletes in file *LiveSyspro.ini* the section *NewSection*.

[Back to Table of Contents](#)

IniRead

Reads the value of a key from a section inside an INI file

Syntax:

IniRead, <%Variable%>, <File>, <Section>, <Key>

Example:

IniRead, %Var%, C:\LiveSystempro\LiveSyspro.ini, Main, Project

Allocates to variable *%Var%* the **value** of *Project*.

[Back to Table of Contents](#)

IniReadPos

Reads either the value of a key or the key of a value

Syntax:

IniReadPos, <%Variable%>, <File>, <Section>, <Parameter>, <Position>

Parameter	<i>Count</i>	reads the number of defined entries in an INI file	
	<i>KeyPos</i>	reads the Key in a defined line	(Result=Value)
	<i>Value</i>	reads the Value	(Result=Key)

ValuePos reads the Value in a defined line (Result=Key)

Position line number (*ValuePos* and *KeyPos* only)

Examples:

IniReadPos,%Var%,C:\MyFile.ini,MySection,Value,1
Allocates to variable *%Var%* the **Key** whose value is **1** .

IniReadPos,%Var%,C:\MyFile.ini,MySection,ValuePos,2
Allocates to variable *%Var%* the **Key** from line **2** .

IniReadPos,%Var%,C:\MyFile.ini,MySection,KeyPos,2
Allocates to variable *%Var%* the **Value** of key from line **2** .

INI file *C:\Lang.ini* with following content:

```
[Section]
Key_1=german
Key_2=English
Any text
Key_3=french
```

IniReadPos,%Var%,C:\Lang.ini,Section,count
Allocates to variable *%Var%* the **value 3** (number of defined INI lines)

[Back to Table of Contents](#)

IniRenameSection

Renames an existing section inside an INI file

Syntax:

IniRenameSection,<File>,<OldSection>,<NewSection>

Example:

IniRenameSection,C:\MyFile.ini,Section_1,Section_2
Renames *Section_1* to *Section_2*.

[Back to Table of Contents](#)

2.3.3. Interface Processing

Interface,Get

Reads the value of an Interface element

Syntax:

Interface,Get,<Element>,<%Variable%>

Supported Element Types - pTextBox
pScrollBar
pCheckBox
pFileBox
pTextLabel

The element types can also be defined as variables

Example:

```
[Interface]
pCheckbox1=True,Checkbox,0,<Any_Section>,50,80,120,18
...
[Process]
Interface,Get,pCheckbox1,%Var%
Allocates to variable %Var% the value True .
```


[Back to Table of Contents](#)

Interface,Set

Sets the value of an Interface element

Syntax:

Interface,Set,<Element>,<Value>

Supported Element Types - pTextBox
pScrollBar
pCheckBox
pFileBox
pTextLabel

The element types can also be defined as variables

Examples:

[Interface]
pCheckBox1=True,Checkbox,0,<Any_Section>,50,80,120,18

...

[Process]
Interface,Set,pCheckBox1,False
Allocates to **pCheckBox1** the value **False** .

Set,%RowCount%,10
Set,%Value%,Test
Interface,Set,pTextlabel%RowCount%,%Value%
Allocates to **pTextlabel10** the text **Test** .

[Back to Table of Contents](#)

Interface,State

Changes the attribut of an Interface element

Syntax:

Interface,State,<Parameter>,<Element1>[,<Element2>][,...]

Supported Element Types - pTextBox
pScrollBar
pCheckBox
pFileBox
pTextLabel

The element types can also be defined as variables

Parameter: - hide / show

Example:

Interface,State,Hide,pScrollBar1
Hides the element **pScrollBar1** inside the Interface

[Back to Table of Contents](#)

Interface,Update

Updates the value of an Interface element

Syntax:

Interface,Update,<Element>

Example:

*IniWrite,%ScriptFile%,Interface.07,pScrollBar2,Test,1,0,0,100,190,60,20,Val1,Val2,Test
Interface,Update,pScrollBar2*

Updates the element **pScrollBar2** in Interface.07 with value **Test** .

[Back to Table of Contents](#)

2.4. Registry Processing

To handle the PE Registry the Registry Hives must be loaded, elaborated and after that unloaded.

RegLoad

Loads an external Registry Hive into the locale Registry of your system.

Syntax:

RegLoad,<HivePath>,<RegistryRoot>

Hivepath	-	Path to Hive file	
RegistryRoot	-	<i>PE-SYSTEM</i>	HKLM/SYSTEM
		<i>PE-SOFTWARE</i>	HKLM/SOFTWARE
		<i>PE-USER</i>	HKCU/Software
		<i>PE-COMPONENTS</i>	HKLM/COMPONENTS

Examples:

*RegLoad,%RegSystem%,PE-SYSTEM
RegLoad,%RegSoftware%,PE-SOFTWARE
RegLoad,%RegUser%,PE-USER
RegLoad,%target_sys%\config\components,PE-COMPONENTS*

Note:

By default HKLM\COMPONENTS will not be shown in the PE Registry.
To view this hive the following procedure is necessary:

1. Open the Registry Editor of running PE
2. Click on the hive HKEY_LOCAL_MACHINE
3. In menu "File" choose *Load Hive*
4. Choose path *X:\Windows\System32\config\components* and click on OPEN
5. In the appearing window LOAD HIVE enter **COMPONENTS**
6. HKLM\COMPONENTS is now available

[Back to Table of Contents](#)

RegUnLoad

Unloads the external Registry Hive

Syntax:

RegUnLoad,<RegistryRoot>

Example:

RegUnLoad,PE-SYSTEM

[Back to Table of Contents](#)

RegRead

Reads a Registry Key

Syntax:

RegRead,<%Variable%>,<HKEY>,<Key>,<Value>

Example:

RegRead,%Var%,HKLM,SOFTWARE\7-zip,Path

Allocates to variable *%Var%* the value of *Path* (=C:\Program Files\7-zip)

Using REG_MULTI_SZ (0x7) delimiter will be interpreted as *#\$x* (Carriage Return)

[Back to Table of Contents](#)

RegWrite

Creates or changes a Registry Key

Syntax:

RegWrite,<HKEY>,<Type>,<Key>[,ValueName][,Value]

HKEY	-	<i>HKLM</i>	-	HKEY_LOCAL_MACHINE
		<i>HKCR</i>	-	HKEY_CLASSES_ROOT
		<i>HKCU</i>	-	HKEY_CURRENT_USER
		<i>HKU</i>	-	HKEY_USERS
		<i>HKCC</i>	-	HKEY_CURRENT_CONFIG
		<i>HKLM64</i>	-	HKEY_LOCAL_MACHINE Wow64
		<i>HKCU64</i>	-	HKEY_CURRENT_USER Wow64
Type	-	<i>0x0</i>		empty key with entry (<i>Standard</i>)
		<i>0x1</i>	REG_SZ	writes a text
		<i>0x2</i>	REG_EXPAND	expanded variable in writing style <i>%%</i>
		<i>0x3</i>	REG_BINARY	writes a binary value
		<i>0x4</i>	REG_DWORD	writes a numeric value
		<i>0x7</i>	REG_MULTI_SZ	writes multiple strings separated by <i>#\$x</i>
Key	-			Name of the key
Valuename	-			optional name of the value
Value	-			optional value

Extended notation of Variables

Set,%var%,01,a2,03

RegWrite,HKLM,0x3,PE-SOFTWARE\Software,Test,%var%

Set,%var%,01,02,03

RegWrite,HKLM,0x3,PE-SOFTWARE\Software,Test,%var%,04,aa,05

Extended notation of REG_MULTI_SZ (0x7):

used delimiter is *#\$x*

Set,%Var%,10#\$x20

RegWrite,HKLM,0x7,SOFTWAREMyKey,Key0x7,%Var%,30,40

32-bit system:

Writes in REGKey *SOFTWAREMyKey* the character string *Key0x7* with following values

10 20 30 40

64-bit system:

Writes in REGKey *SOFTWAREWow6432NodeMyKey* the character string *Key0x7*

with following values *10 20 30 40*

Empty Multistring entries will get skipped

Examples:

RegWrite, HKLM, 0x0, PE-SOFTWARE\NewKey
Writes into the PE-Registry the new key *NewKey*.

RegWrite, HKLM, 0x0, PE-USER\Software\Microsoft\Windows\CurrentVersion\MyRegKey
Writes into the PE Registry in *HKCU\Software\Microsoft\Windows\CurrentVersion* the new key *MyRegKey* with string (*Standard*) and the type *REG_SZ* as well as in array *Data* the value (*Value not set*).

Set, %var%, 01, a2, 03
RegWrite, HKLM, 0x3, PE-SOFTWARE\Software, Test, %var%
Writes into the PE Registry in *HKLM\Software* the valuename *Test* with a binary value *01, a2, 03*.

Set, %var%, 01, 02, 03
RegWrite, HKLM, 0x3, PE-SOFTWARE\Software, Test, %var%, 04, aa, 05
Writes into the PE Registry in *HKLM\Software* the valuename *Test* with a binary value *01, 02, 03, 04, aa, 05*.

Advice to Type 0x4:

Decimal notation is possible as well as hexadecimal notation

RegWrite, HKLM, 0x4, PE-SYSTEM\ControlSet001\Services\WpsSvc, Start, 3
Writes into the PE Registry in *HKLM\SYSTEM\ControlSet001\Services\WpsSvc* the valuename *Start* with a numeric value *3*.

RegWrite, HKLM, 0x4, PE-SYSTEM\ControlSet001\Services\WpsSvc, Start, 0xFF
Writes into the PE-Registry in *HKLM\SYSTEM\ControlSet001\Services\WpsSvc* the valuename *Start* with a numeric value *255*.

[Back to Table of Contents](#)

RegDelete

Deletes a Registry Key

Syntax:

RegDelete, <HKEY>, <Section>[, <ValueName>]

Examples:

RegDelete, HKLM, PE-SYSTEM\ControlSet001\Control\ComputerName\ComputerName, ComputerName
Deletes in the mentioned Registry Key the **valuename** *ComputerName*.

RegDelete, HKLM, PE-SYSTEM\Test
Deletes in the mentioned Registry Key the **valuename** (*Standard*).

[Back to Table of Contents](#)

2.5. String Processing

String LEFT/RIGHT

Checks the specified number of characters in strings

Syntax:

String, <%Variable%>, <Action>, <String>, <Counter>

Action	-	<i>Left</i>	Number of characters in a string, beginning from the left
		<i>Right</i>	Number of characters in a string, beginning from the right

Counter - Number of characters

Examples:

Set,%String%,Sommerurlaub
String,%var%,Left,%String%,6
Allocates to variable **%var%** the value **Sommer** .

Set,%String%,Sommerurlaub
String,%var%,Right,%String%,6
Allocates to variable **%var%** the value **urlaub** .

[Back to Table of Contents](#)

String TLeft/TRight

Trims any characters of a string

Syntax:

String,<%Variable%>,<Action>,<String>,<Counter>

Action - **TLeft** Trims characters of a string, beginning from the left
TRight Trims characters of a string, beginning from the right

Examples:

Set,%String%,Hallo
String,%Var%,TLeft,%String%,1
Allocates to variable **%Var%** the value **allo** .

Set,%String%,Hallo
String,%Var%,TRight,%String%,1
Allocates to variable **%Var%** the value **Hall** .

[Back to Table of Contents](#)

String FIND

Shows the first position of a substring in a specified string

Syntax:

String,<%Variable%>,FIND,<String>,<SubString>

Example:

Set,%String_1%,Sommerurlaub
Set,%String_2%,url
String,%Var%,FIND,%String_1%,%String_2%
Allocates to variable **%Var%** the position **(7)** of correlation of char **u** .

Note:

If there is no correlation the position has always a value of **0**

[Back to Table of Contents](#)

String LEN

Determines the length of a string

Syntax:

String,<%Variable%>,LEN,<String>

Example:

String,%Var%,LEN,Teststring

Allocates to Variable **%Var%** the length (10) of string **Teststring** .

[Back to Table of Contents](#)

String MID

Trims any part of a string

Syntax:

String, <%Variable%>, MID, <String>, <Position>, <Counter>

Example:

Set, %String%, Sommerurlaub

String, %Var%, MID, %String%, 2, 4

Allocates to variable **%Var%** the value **omme** .

[Back to Table of Contents](#)

String REPLACE

Replaces a part of a string by another part

Syntax:

String, <%Variable%>, REPLACE, <String>, <Searchstring>, <Replacestring>[, Parameter]

Parameter	-	0	(default)	Disbandment of variables
		1		Disbandment of variables up to level 1 (1 st variable in variable)

Example:

Set, %String%, The quick brown fox jumps over the lazy dog

Set, %Searchstring%, brown

Set, %Replacestring%, black

String, %Var%, REPLACE, %String%, %Searchstring%, %Replacestring%

Allocates to variable **%Var%** the new string

The quick **black** fox jumps over the lazy dog

[Back to Table of Contents](#)

String SPLIT

Splits a string

Syntax:

String, <%Result%>, SPLIT, <%Variable%>, <Delimiter>, <Parameter>

Parameter = 0	-	Number of items of %Variable%
Parameter > 0	-	Returns the substring of %Variable% of indicated Index

Example:

RegRead, %Var%, HKLM, SYSTEM\CurrentControlSet\Services\NetBIOS\Linkage, Bind

String, %Count%, SPLIT, %Var%, #\$, 0

For, %x%, 1, %Count%

String, %Res%, SPLIT, %Var%, #\$, %x%

Message, %Res%=%x%

Next

Allocates to Variable **%Var%** the content of character string **Bind** in above mentioned REGKey, while Variable **%Count%** contents the number of found arrays.

In the following **For/Next Loop** the content of Variable **%Var%** will be split on the basis of the

particular parameters and allocated in each loop to Variable **%Res%** .

[Back to Table of Contents](#)

2.6. Network Commands

WebGet

Loads a file from the internet

Syntax:

WebGet,<FileInInternet>,<PathToStore>[,Parameter]

Parameter	-	1	shows a progress bar
		2	loads a file of unknown size

Examples:

Download without progress bar (http://)

WebGet, http://irfanview.tuwien.ac.at/iview433.zip,C:\iview433.zip

Loads the file **iview433.zip** from **http://irfanview.tuwien.ac.at** and stores it in **C:**

Download without progress bar (ftp://)

WebGet,ftp://ftp.cpubid.com/cpu-z/cpu-z_1.61-32bits-en.zip,C:\cpu-z_1.61-32bits-en.zip

Loads the file **cpu-z_1.61-32bits-en.zip** from **ftp.cpubid.com** and stores it in **C:**

Download with progress bar

Webget,ftp://wa651f4:anonymous@mh-nexus.de/HxDSetupDE.zip,C:\HxDSetupDE.zip,1

Loads the file **HxDSetupDE.zip** and shows a progress bar

Download a file of unknown size and progress bar

Webget,http://googledrive.com.../Firefox_36.0.4.zip,C:\Firefox.zip,3

Loads the file of unknown size **Firefox_36.0.4.zip** and shows a progress bar

WebGet has available a return variable **%Exitcode%**:

0	-	Ok
1	-	Error

Example:

System,Error,Off

WebGet,http://www.a43filemanager.com/download/a43.zip,c:\a43.zip,1

Message,Download: %Exitcode%

System,Error,on

Note:

Using the WebGet Parameter **3** (addition from 1 and 2) **%Exitcode%** is not necessary.

Usage of FTP Downloads with Password:

Example:

WebGet,ftp://wa651f4:anonymous@mh-nexus.de/HxDSetupDE.zip,C:\HxDSetupDE.zip,1

[Back to Table of Contents](#)

2.7. Loop commands

2.7.1. For / Next - loop

Runs a command repeatedly

Syntax:

```
For,<%Variable%>,<Start>,<End>  
  <Command>
```

Next

Start

Initial value

End

Final value

Example:

```
For,%Var%,1,5  
  Message,Loop: %Var%
```

Next

This loop will be executed five times whereas the values of *%Var%* are changing each time, showing the following messages:

```
Loop: 1  
Loop: 2  
Loop: 3  
Loop: 4  
Loop: 5
```

[Back to Table of Contents](#)

2.7.2. ExitLoop

Quits a For/Next Loop

Syntax:

```
ExitLoop
```

Example for recursive loop:

```
If,a,=,a,then  
  For,%x%,1,6  
    if,a,=,a,then  
      Run,%ScriptFile%,Load-D,mouclass.sys  
      if,%x%,=,4,then,ExitLoop  
    Else  
      Message,Message window without window title  
    End  
    Message,Loop: %x%  
  Next  
Else  
  Message,Test 1  
End  
Message,End of Loop
```

```
[Load-D]  
Message,sub %x%,#1
```

In this example a message window appears **with** window title *mouclass.sys* and the message *sub1*, afterwards a message window appears **without** window title and the message *Loop: 1*, afterwards a message window appears **with** window title *mouclass.sys* and the message *sub2*. This process repeats itself until a message window **with** window title *sub4*, afterwards a message window appears **without** window title and the message *End of Loop*

[Back to Table of Contents](#)

2.8. Variables (intern)

These variables will be set by the Builder on StartUp

%BaseDir%	-	Root directory	(e.g.: C:\LiveSystempro)
%BuildMode%	-	Process mode	
		0:	Button „Start“
		1:	Commandline
		2:	Button „>“
		3:	Interface Element [Section]
%CPUVM%	-	checks if the processor is capable for Virtualization Technology (VT-x)	
		0:	not capable
		1:	capable
%ExitCode%	-	Exit code	
%HostAdmin%	-	User's administrator rights	
		0:	No administrator right
		1:	Administrator right
%HostArch%	-	Architecture of Host	(x86, x64)
%HostLang%	-	Locale ID (hexadecimal)	(07, 09, ...)
%HostNT%	-	NT version of Host	(5.1, 5.2, 6.0, 6.1, 6.2, 6.3, 10.0)
%HostOS%	-	Operating system of Host	(WinXP, Win2003, WinVista, Win7, Win8, Win10)
%ISODir%	-	Path to ISO directory	(e.g.: %BaseDir%\Iso)
%ISOfile%	-	Path to ISO file	(e.g.: %BaseDir%\Iso\winpe3.iso)
%ISOFilename%	-	Name of ISO file	
%PEMode%	-	Win8PE only (with source Windows 8.1)	
		PE:	Build in PE-Mode
		OS:	Build in OS-Mode
%ProcessError%	-		
		0:	No error
		3:	Halt
		4:	Error
			(will be set in %Projectini%, [ONBUILDEXIT] only)
%ProgramFilesDir%	-	Program directory	(e.g.: C:\Program Files)
%Programs64%	-	64-bit program directory	
%ProjectDir%	-	Path to Project directory	(%BaseDir%\Projects\ <i><Project></i>)
%ProjectFolder%	-	Directory name of project	(Win7PE, Win8PE, Win10PE, WinXPE)
%Projectini%	-	Project file	(%BaseDir%\Projects\ <i><Project></i> \project.cfg)
%ProjectName%	-	Project name	(Win7PE, Win8PE, Win10PE, WinXPE)
%ScriptDir%	-	Folder with current scriptfile	
%ScriptFile%	-	current scriptfile	
%ScriptLvl%	-	Script level	
%SourceDir%	-	Source directory on installation media	
%TargetDir%	-	Target directory	(e.g.: %BaseDir%\Target\Win7PE)
%TempDir%	-	Path to User's Temp directory	(e.g.: C:\Users\ <i><Username></i> \AppData\Local\Temp)
%UserName%	-	Username	
%UserProfile%	-	Path to Home directory	(e.g.: C:\Users\ <i><Username></i>)
%WindowsDir%	-	Path to Windows directory	(e.g.: C:\Windows)
%WinUAC%	-	User Account Control (from Windows Vista onwards)	
		Value	0/1

[back to Table of Contents](#)

2.9. Variables (Project)

Win7PE, Win8PE, Win10PE:

%BootSRC%	-	Boot source (decompressed file boot.wim)	
		e.g.:	%BaseDir%\Wim\Win7SP0de-DE-B\2
%boot_sys%	-	System32 directory of %BootSRC%	
		e.g.:	%BaseDir%\Wim\Win7x86SP0de-DE-B\2\Windows\system32
%boot_win%	-	Windows directory of %BootSRC%	
		e.g.:	%BaseDir%\Wim\Win7x86SP0de-DE-B\2\Windows
%boot_wow%	-	SysWow64 directory in %BootSRC%	(64-bit media only)

		e.g.:	
		%BaseDir%\Wim\Win7x86SP0de-DE-B2\Windows\SysWoW64	
		Drive Y :	
		Language of source media	
		Mui list of source media (delimiter for 2 languages is “.”)	
		Install source (decompressed file install.wim)	
		e.g.: %BaseDir%\Wim\Win7-Enterprisex86SP0de-DE-I	
		System32 directory of %InstallSRC%	
		e.g.:	
		%BaseDir%\Wim\Win7Enterprisex86SP0de-DE-I\Windows\System32	
		Windows directory of %InstallSRC%	
		e.g.:	
		%BaseDir%\Wim\Win7Enterprisex86SP0de-DE-I\Windows	
		SysWoW64 directory in %InstallSRC% (64-bit media only)	
		e.g.:	
		%BaseDir%\Wim\Win7Enterprisex64SP0de-DE-I\Windows\SysWoW64	
		Path to image file (Install.wim / Install.esd)	
		e.g.: %BaseDir%\ISO\Win8[SP0]-de_DE_x86.iso	
		%TargetDir%\Windows\System32\win7pe.cfg	
		Program Files	
		Drive X :	
		Edition of Windows Source	
		SysWoW64 Target directory	(%TargetDir%\Windows\SysWoW64)
		Image Architecture	
		Image Index of boot.wim	
		Part of Microsoft file version	(e.g.: 7600 bei ,6.1.7600.16385')
		Image Index	
		Part of Microsoft file version	(e.g.: 6 bei ,6.1.7600.16385')
		Part of Microsoft file version	(e.g.: 1 bei ,6.1.7600.16385')
		Part of Microsoft file version	(e.g.: 16385 bei ,6.1.7600.16385')
		ServicePack Version	

WinXP PE:

		Administrator	
		Computer Management	
		All Users	
		Application Data	
		Cookies	
		Common Files	
		Desktop	
		Documents and Settings	
		Documents	
		Compressed driver file (in CAB format)	
		Favorites	
		History	
		Keyboard settings	
		Language of source media	
		Language Group	
		Locale Settings	
		Locale ID	
		Microsoft\Internet Explorer\Quick Launch	
		My Documents	
		Code Page	
		%Systemdrive%\%prog_files%	
		Programs	
		Drive B :	
		SendTo	
		%SourceDir%\I386\System32	
		%SourceDir%\I386	
		Version of ServicePack cabinet file	
		Start menu	
		Start menu\Programs	
		Start menu\Programs\Autostart	
		Autostart	
		%TargetDir%\I386\System32	
		%TargetDir%\I386	
		Templates	

%temp_internet_files% - Temporary Internet Files

Common Variables:

%AppsDir% - Default directory for Downloads (*%BaseDir%\Temp\Apps*)
%Arch% - Architecture of Source (*x86, x64*)
%LSPVer% - Version number of *LiveSyspro.exe*
%NT% - NT Version of Source
5.1 Windows XP
5.2 Windows XP Professional x64, Server 2003
6.0 Windows Vista, Server 2008
6.1 Windows 7, Server 2008 R2
6.2 Windows 8
6.3 Windows 8.1
10.0 Windows 10
%NTVersion% - NT Version of Source (*5, 6*)
%OS% - Operating system of Source
(*WinXP, Win2003, WinVista, Win7, Win8, Win10*)
%PE_programs% - either *X:\%prog_files%* or *Y:\%prog_files%*
%PreUserdata% - *%BaseDir%\Temp\%ProjectName%\UserData*
%ProjectTemp% - Temporary Project directory (e.g.: *%BaseDir%\Temp\WinXPE*)
%Ramboot% - True/False
%RegSoftware% - Path to Registry hive *PE-SOFTWARE*
%RegSystem% - Path to Registry hive *PE-SYSTEM*
%RegUser% - Path to Registry hive *PE-USER*
%SP% - Version number of ServicePack
%Target_CD% - Target directory (*%TargetDir%_CD*)
%Target_Prog% - *%TargetDir%\%prog_files%*
%Target_Sys% - System Target directory (*%TargetDir%\Windows\System32*)
%Target_Win% - Windows Target directory (*%TargetDir%\Windows*)
%Tools% - Tools directory (*%BaseDir%\Tools*)

[Back to Table of Contents](#)

2.10. System,SetPEInit

PEInit runs some tasks (Start of programs, opening cmd-Boxes, etc.) together with start of **WindowsPE**.

These commands are written to file *Peinit.ini*.

The range of commands reaches from **001** up to **059**, whereas **001** up to **015** are reserved for the running project.

Syntax:

System,SetPEInit,<ID>,<Section>,<Command>

ID	Entry ID (chars or digits) Valid in script only
Section	Init (before starting Explorer) Startup

Commands using Show/Hide Effect for start of programs:

CMD

Run

RunWait

Show/Hide Effect:

Show	shows the starting program (Default setting, might be ignored therefore)
Hide	hides the starting program

Example:

System,SetPEInit,A,Init,RunWait||%PE_programs%\DrvImp\DrvImpe.exe -d -i:"%UFD%\Drivers\%Archit%"

Writes in *PEInit.ini* e.g. these entries:

```
[Init]
20=RunWait||X:\Programme\Drvimp\DrvImpe.exe|-d -i:"%UFD%\Drivers\%Archit%
[Init-ID]
20=A,Apps\my - Driver Import PE\Drvimpe.srt
```

Examples (WinXPE):

```
[Init]
003=Ramdisk|25|%\imdisk.exe -a -t vm -m B: -s %Rsize%M -p "/fs:NTFS /q /y /v:Imdisk"
004=CMD|cmd "B:\Documents and Settings\Default User\Local Settings\Temp"
006=DllInstall|shell32.dll
007=DllInstall|browseui.dll
008=DllInstall|shdocvw.dll
039=BROADCAST
016=RunWait||reg.exe|add HKLM\SYSTEM\CurrentControlSet\Services\usbhub /v ImagePath /t
REG_EXPAND_SZ /d system32\DRIVERS\usbhub.sys /f
```

The numbers don't have to be arranged continuously, but existing numbers will be overwritten by identical numbers.

Command list peinit.exe:

- CMD** runs a DOS command
Examples:
 CMD|cmd #q%RamDrive%\%doc_and_sett%\Default User\%local_sett%\Temp#q
 Shows the cmd box
 CMD|Hide|cmd #q%RamDrive%\%doc_and_sett%\Default User\%local_sett%\Temp#q
 Hides the cmd box
- Run** runs a program
Example:
 Run||imdisk.exe|-a -t vm -m %RamDrive% -s %Rsize%M -p #q/fs:%Format% /q /y
 /v:%pTextBox2%#q
- Runwait** runs a program and waits for its end
Example:
 Runwait||imdisk.exe|-a -t vm -m %RamDrive% -s %Rsize%M -p #q/fs:%Format% /q /y
 /v:%pTextBox2%#q
- BROADCAST** refreshes the environment variables
Example:
 BROADCAST
- DllInstall** installs a DLL file
Example:
 DllInstall|shell32.dll
 also available with additional install parameter
Examples:
 DllInstall|shell32.dll|setup
 DllInstall|shell32.dll|InitMain
 DllInstall|themeui.dll|UserInstall
- DrvLetter** changes the drive letter
Example:
 DrvLetter|%CDDrive%|I386\%ImageFile%
- FileDelete** deletes a file
Example:
 X:\User\Desktop.ini
- Ramdisk** creates a RamDisk
Example:
 Ramdisk|%pTextBox1%|%pScrollBar1%|imdisk.exe -a -t vm -m %RamDrive% -s %
 Rsize%M -p#q/fs:%Format% /q /y /v:%pTextBox2%#q
 Value of %pTextBox1%: Size of RamDisk
 Value of %pScrollBar1%: % or MB
 %Rsize%: Result of %pTextBox1% + %pScrollBar1%
- RegDelete** deletes a Registry value
 RegDelete,<Key>,[optional Value]
Examples:
 RegDelete\HKLM\SYSTEM\CurrentControlSet\Services\AmdK8

RegDelete\HKLM\SYSTEM\CurrentControlSet\Services\AmdK8\Start

Register	registers a DLL file Example: <i>Register shell32.dll</i>
RegWrite	writes a Registry value Example: <i>RegWrite\HKLM\0x1\SYSTEM\ControlSet001\key\ka#\$-dre ;'\$-d = " "</i> For special chars you have to use the keyword \$-d = " "
SetRes	changes the screen resolution Example: <i>SetRes 1024 768</i>
Shell	loads the chosen file as Shell (only available in section Init) Example: <i>Shell Explorer.exe</i>
StartService	starts a service Example: <i>StartService Ext2fs IfsDrives</i> multiple specifications are possible (separated by)
Wait	pauses the given time (in Milliseconds) Example: <i>Wait 3000</i>

Location of Peinit.ini:

%Target_Sys% in all projects

[back to Table of Contents](#)

2.11. Special Characters

The following special chars will be used within LiveSystempro:

#\$x	-	Carriage Return + LineFeed
#\$q	-	Quotation marks (")
#\$c	-	Comma
#\$p	-	Percent sign (%)
#\$s	-	Blank space
#\$d	-	Pipe ()

[Back to Table of Contents](#)

2.12. Shortcuts

Syntax:

Add_Shortcut,[Type],[StartMenuFolder],[FullFileName],[Title],[Workfolder],[Parameters],[IconFile]

Type	-	StartMenu Desktop Quicklaunch
StartMenuFolder	-	a. Special name for "Folder\Subfolder" b. "." For a shortcut in <i>StartMenu\Programms</i> c. Empty field is also possible
FullFileName	-	Only necessary if shortcut is different to <i>%ProgramExe%</i> . Default: <i>%Pe_Programs%%ProgramFolder%%ProgramExe%</i>

Title	-	Any title Default: <i>%ProgramTitle%</i>
Workfolder	-	Any work folder
Parameters	-	To start <i>%ProgramExe%</i> with parameters
IconFile	-	Only necessary using a special icon for shortcut

Examples:

Add_Shortcut,Desktop

Creates a shortcut of the program to run on the Desktop.

Add_Shortcut,StartMenu,%pTextBox1%

Creates a shortcut of the content of *%pTextBox1%* with the StartMenu.

Add_Shortcut,StartMenu,,MyProgramEXE,,,

Creates a shortcut of the program to run in *StartMenu\Programs* (No Subfolder).

Add_Shortcut,StartMenu,,MyProgramEXE,,exe.ico#\$c7

Creates a shortcut of the program to run in *Startmenü\Programme* with icon #7 of the EXE file.

Add_Shortcut,StartMenu,MyFolderMySubfolder,MyProgramEXE,,,

Creates the folder *MyFolderMySubfolder* in *StartMenu\Programs*, within this folder *MyProgramEXE* will be placed.

Add_Shortcut,Desktop,,,IrfanView Thumbnails,./thumbs,%PE_Programs%%\%ProgramFolder%\Thumbnails.ico

Creates a shortcut of *IrfanView Thumbnails* on the Desktop.

Start parameter for *%ProgramExe%* is */thumbs*, shows the icon of file *Thumbnails.ico*

[Back to Table of Contents](#)

2.13. Macros

- Macros provide the possibility to execute often used commands automatically (compare command **SetMacro**).
- Certain files (not all are needed) are decompressed by PreConfig from the source and put into the Wim cache. These are 2 separate Wim subdirectories (*boot.wim* and *install.wim*)
The macros work with these 2 Wim caches.

Exceptions:

Add_Shortcut and *RunFromRam* (all Projects)

- Macros will be stored in *%Projectini% [Macros]*.
- **Fallback en-US:**
an English mui file will only be copied if *%Distlang% mui* does not exist.
If there are no matching mui files none will be copied. A missing mui file will **never** give an error!

Note:

The LSP projects use different macros.

All Projects:

Add_ShortCut

Creates shortcuts of programs (compare section 2.11)

RunFromRAM,<Value>

Value - False | True

Regulates program execution in *RAM* or from *CD*

Examples:

RunFromRAM,False

The program (*=%PE_programs%*) will run from *CD* (Drive Y:)

RunFromRAM,True

The program (*=%PE_programs%*) will run in *RAM* (Drive X:)

Win7PE, Win8PE, Win10PE:

FileDecBoot

Unpacks files from *boot.wim* to *%BootSRC%*

Example:

FileDecBoot,Windows\System32\Wbem\%DistLang%\Msi.mfl

Extracts the file *Msi.mfl* to e.g.:

%BaseDir%\Wim\Win7SP0de-DE-B1\Windows\System32\Wbem\%DistLang%

FileDeInst

Unpacks files from *install.wim* to *%InstallSRC%*

Example:

FileDeInst,Windows\System32\Wbem. **

Extracts all files to e.g.:

%BaseDir%\Wim\Win7Enterprisex86SP0de-DE-I\Windows\System32\Wbem

ReqSys32,<File>

File - File in System32 directory (Install Wim Cache) to copy + mui (fallback en-US)

Example:

ReqSys32,mapi32.dll

Copies *mapi32.dll* to *Windows/System32*

Copies *mapi32.dll.mui* to *Windows/System32\%DistLang%*

Wildcards are not supported.

Gives an error if the file does not exist.

ReqB32,<File>

File - File in System32 directory (Boot Wim Cache) to copy + mui (fallback en-US)

Example:

ReqB32,attrib.exe

Copies *attrib.exe* to *Windows/System32*

Copies *attrib.exe.mui* to *Windows/System32\%DistLang%*

Wildcards are not supported.

Gives an error if the file does not exist.

ReqSysWoW64,<File>

File - File in SysWow64 directory (Install Wim Cache) to copy + mui (fallback en-US)

Example:

ReqSysWoW64,activeds.dll

Copies the 32-bit file *activeds.dll* to *Windows\SysWoW64*

Copies the 32-bit file *activeds.dll.mui* to *Windows\SysWoW64\%DistLang%*

Wildcards are not supported.

Gives an error if the file does not exist.

ReqWin,<File>

File - File in Windows directory (Install Wim Cache) to copy + mui (fallback en-US)

Example:

ReqWin,explorer.exe

This will copy *explorer.exe* to *Windows*.

This will copy *explorer.exe.mui* to *Windows\%DistLang%*.

Wildcards are not supported.

Gives an error if the file does not exist.

[Back to Table of Contents](#)

3. Script Control

3.1. Standard Sections of Scripts

Normally scripts are structured in these sections

[Main]
[Interface]
[Process]

Creation of additional necessary sections is always possible.

[Back to Table of Contents](#)

3.1.1. Section [Main]

Preferentially these entries should exist:

[Main]
Title= *Program title*
Description= *Short description of program*
Selected= *true* Script is enabled in tree view of interface
False Script is disabled in tree view of interface
None Script will be executed, cannot be checked or unchecked
Noprocess Script will be shown but not executed
Level= *Script level in tree view of LiveSystem pro*
1 - Build
2 - Shells
3 - Components
4 - Customize + Drivers
5 - Apps
8 - Finish + Virtual Test
9 - Utils
Author= *Name of script author*
Version= *Progressive script version number*
Date= *Date of creation or update date*

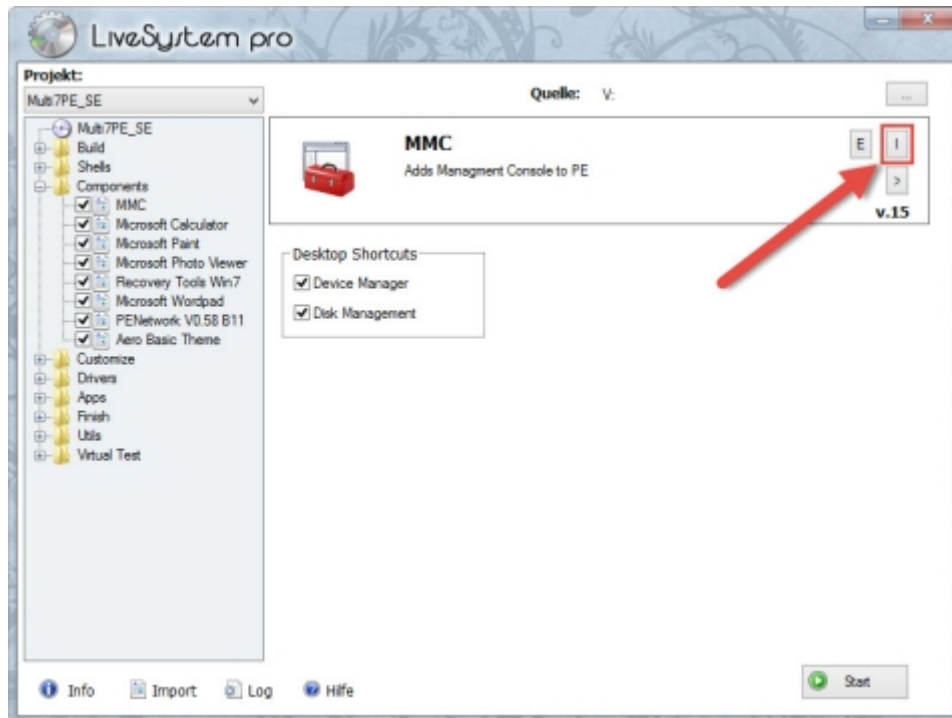
Entries in section [Main] to control other scripts:

Disable= *xyz.srt* Script will get disabled
Enable= *xyz.srt* Script will get enabled

[Back to Table of Contents](#)

3.1.2. Section [Interface]

In this section happens the optical illustration of needed texts and checkboxes.
Selection of checkboxes happens by clicking button **I** (see image):



Creation of multilingual interfaces

Additional interfaces can always get included according to this example:

[Interface]	english texts and checkboxes
[Interface.07]	german texts and checkboxes
[Interface.13]	dutch texts and checkboxes

The builder will use automatically the right version.

[Back to Table of Contents](#)

3.1.3. Section [Process]

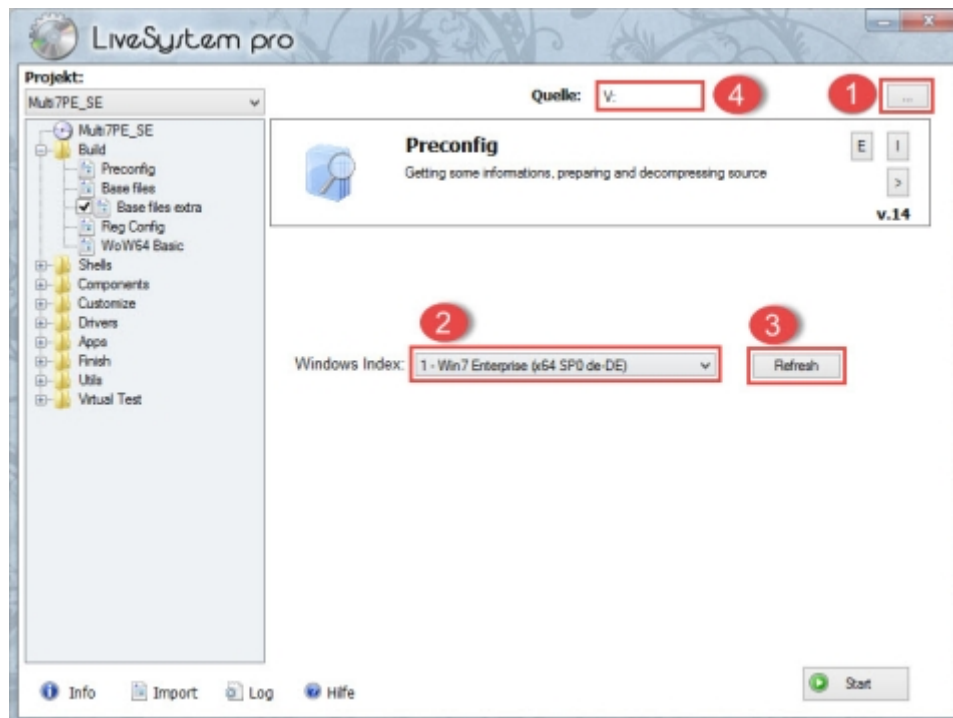
In this section all necessary commands have to take place to run a program.

[Back to Table of Contents](#)

3.2. Active Script Control

Some interface checkboxes can be used to make specific changes to the display of entries.

3.2.1. Choosing Windows Source



Before running the first build

Click on button **1**: Result: **4**: Display of source path
2: Display of Windows version

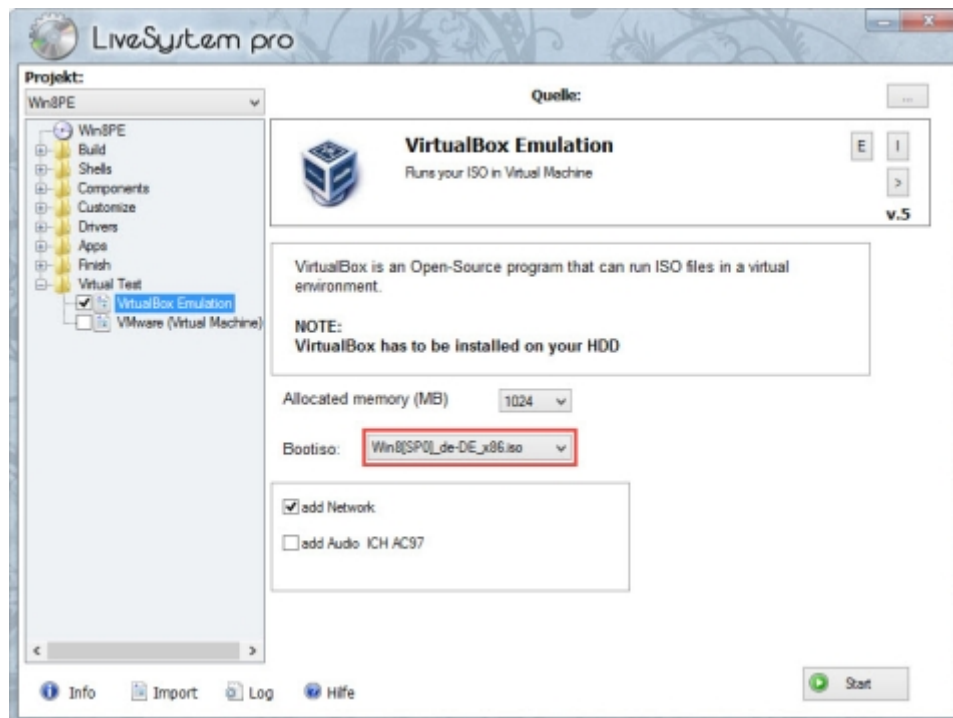
Before running an additional build using another Windows source

Click on button **3**: Result: **2**: Display of Windows version

[Back to Table of Contents](#)

3.2.2. Script-Select

Used to update boxes (example of an originally empty Scrollbox)



Example code:

```
[Interface.07]
pScrollBar3=,1,0,0,20,270,215,20
```

```
[Script-Select]
search,%Var%,File,%ISODir%,*.iso
Interface,set,pScrollBar3,%Var%
```

Result:

- Choosing the script in the tree view the Scrollbox gets updated
- pScrollBar3 gets updated:

```
[Interface.07]
pScrollBar3=Win7[SP0]_de-DE_x86.iso,1,0,0,20,270,215,20,Win7[SP0]_de-DE_x86.iso
```

[Back to Table of Contents](#)