Start parameters of CPDev.exe and STComp05.exe programs and names of selected configuration fields.

CPDev.exe

The following parameters can be used by CPDev.exe application during start up:

Name of switch and arguemnts	Description
`-op Arg0` *`Arg1 Arg*`* `--openproject` (as above)	If `Arg0` is different than double minus (`--`), then **only** `Arg0` is open as project file by CPDev package. Otherwise, `Arg0` is neglected and **all following arguments** are treated as names of files to be open during start up. `--openproject` is equivalent longer name.
`--help`	Displays brief information on some switches and refers to this documentation for further details.
`--resetconfig`	Set default settings for this installation (program, global settings) and terminates the application.
`--set-save-exit`	Saves updated configuration settings and terminates the application.
`--get-conf Arg0`	Sends configuration parameters (in text form) to `stdout` (but not to console); `Arg0` as in `--set-conf` below, but without `value` (type T must follow the sign =).
`--exit`	Terminates the application; useful while using `--get-conf` (and similar) in scripts. Contrary to `--set-save-exit`, `--exit` does not save updated configuration.
`--set-conf Arg0`	Sets value of configuration parameter. `Arg0` has the form: `SettingName=TValue` where: `SettingName` denotes configuration field (further sections), `T` – conversion type (see below), `Value` – Text being written. Due to limitations of operating systems, Value must be in **mini** para-HTML format: _[see table below]_

Notation	Meaning
`&`	`&`
`<`	`<`
`>`	`>`
`&qt;`	`'`
`&dq;`	`"`
`&#`*`hex`*`;`	ASCII character of *`hex`* number
` `	'new line' character

Conversion types of `Value` into type `T`

Character	Type description
`s`	String (no conversion after decoding from para-HTML format)
`i`	Int32 in decimal form

	f	float – in decimal form, with **regional** separator.
	F	float – in decimal form, with constant **dot** ('.') separator.
	d	double – in decimal form with **regional** separator
	D	double – in decimal form with constant **dot** ('.') separator
	c	char – Unicode character
	x	Int32 in pure hexadecimal form (without prefix $0x$, $\$$ or suffix h
	a[1]	AND bit product of actual setting with hexadecimal value (as x above); result written back into the setting.
	o[1]	OR bit sum of actual setting with hexadecimal value (as x above); result written back into the setting.
	b	Boolean value of a parameter. Admissible values: Int32 number (0 means false, otherwise true), or True, False.
	p[2]	Value of the argument is treated as path stored with application variables (they starts from dolar ('$\$$') sign). The value is unfolded, and next full path is stored.
	Example of using the switch: `cpdev.exe --set-conf UseAutoUnlockEdit=bFalse` `--set-conf` parameter can be repeated many times as argument. Error windows appear in case of lack of such field or faulty conversions.	
--diagnose-config	Sends current configuration with additional information to stdout (not to console); useful for diagnosing with configuration problem.	
--gu-get-conf Arg0 and --gu-set-conf Arg0	Operate similarly to --get-conf and –set-conf described above, except that they accept global settings (gu prefix). Global configuration fields are described at the end.	

--execute-script Arg0	Because of system limitations (8kB) for the longest command line, there is a possibility to "execute script" with command line arguments stored in file Arg0, with following format:
	  – every line is a separate argument,
	  – if line starts with '"' character then currently error is signalized with currently unsupported string,
	  – if line starts with '-' character then argument is splited into two parts where first space occurs.	

--quiet	Allows to start CPDev without splash screen.	

---

1   Parameter of --set-conf only (not for --get-conf).
2   In case of --get-conf the value from settings is folded to path with application variable.

STComp05.exe

Automatic compilation of library and other projects is executed by STComp05.exe compiler using the following parameters:

Switch	Description
`--opt Arg0`	Optimization level `Arg0` must be decimal
`--lcf Arg0` `-l Arg0`	`Arg0` denotes virtual machine specification file.

`--help`
`-?`
`/?`
`/help`
`-h`
	Displays information on all switches and parameters, and terminates compiler operation.

`--dcp Arg0` `-d Arg0`	`Arg0` denotes name of output file with information for simulator, etc.
`--out Arg0` `-o Arg0`	`Arg0` – file with compiled output code in binary form.
`--library`	Switches into library compilation mode; `--out` parameter (above) defines library output file.
`--lib-copyright Arg0`	Set library copyrights to `Arg0`.
`--lib-menu Arg0`	Sets menu path (library location) to `Arg0`.
`--lib-ver Arg0`	Sets library version number to `Arg0`. The argument must be pure hexadecimal, without prefixes (`0x`, `$`) or suffixes (`h`).
`--exportlist Arg0`	Sets list of names of POU objects exported to library. The list consists of POUs' short identifiers separated by commas. Spaces before or after the commas are not allowed. However, if `Arg0` begins with the sign '`@`' (at), then the rest of `Arg0` after '`@`' is treated as name of a file with **single line** which contains list of POU identifiers formatted as above (without spaces). If `Arg0` is the sign '`*`' (star), then all objects from compiled files are exported.
`--`	Terminates interpretation of switches (as in Linux/Unix). **All** remaining arguments are interpreted as files for compilation. <u>Notice !</u> Compilation of a number of files with source codes is possible, since they are temporarily collected in one big file. However, error reporting may be ambiguous sometimes in such case.
`--cf Arg0`	Compilation flags. `Arg0` must be pure hexadecimal (like in `--lib-ver`). Meaning of the bits: <table><tr><th>Bit mask</th><th>Meaning</th></tr><tr><td>0x00000001</td><td>Skips code section with errors (continues compilation despite the errors; report all errors at the end)</td></tr><tr><td>0x00000002</td><td>Adds supplementary information to Fenixile debugger.</td></tr><tr><td>0x00000004</td><td>Turns on Big-Endian mode (not supported yet)</td></tr></table>

	0x00000008	Disable automatic aligning of global variable addresses depending on variable sizes.
	0x00000010	Enable automatic aligning of local variable addresses depending on variable sizes.
	0x00000020	Appends object map (in old format).
	0x00000040	Detects unused variables (compiler indicates unused variables declared by the user and automatically removes variables without names created in VMASM code). Sometimes, however, it may present misleading hints (ignore them).
	0x00000080	Enable modifications of code for FPGA platforms (it causes limitations in portability of universal executable code).
	0x00000100	Enable variable merging when reading from array.
	0x00000200	Generate code for dynamic checking of verification conditions (not implemented).
	0x00000400	Enable new processing of global variable clauses (to conform IEC 61131-3), declared everywhere.

`--sf Arg0`	Scanner flags (compiler module). `Arg0` must be pure hexadecimal (see `--lib-ver`). Meaning of the bits: | **Bit mask**	**Meaning**
0x0000000F	Used internally. Do not set bits in range 0x00000001 ÷ 0x00000008
0x00000010	Allows // comments (C++) in ST code
0x00000020	Allows nested comments in ST code
0x00000040	Allow creating tokens with markers for verification correctness.
0x00000080	Converts all identifiers to upper case (enable for compatibility with versions prior 1.1.1.5)	
`--project`	Files with source code (normally *.cst for STComp05) will be treated as CPDev projects (*.xml) and converted internally into *.cst format before compilation.
`--WaitIfErrors` `-wie`	Waits for pressing a key after displaying compiler messages such as errors, warnings, hints.
`--force-lcid` `LCIDcode`	Forces language for compiler messages. LCID indicates language and country according to Microsoft table. It may be either decimal or hexadecimal (with `0x` or `$`).
(remaining)	Leading '-' (minus) implies that no such switch exists (all permitted switches are above). Source file can be preceded by '--'. Naturally, source files are at the end of command line. Leading sign '/' (slash) makes the compiler to check whether slashes

	separates folders further down. If yes (Unix systems), the argument is treated as name of the file to be compiled (rooted path). Otherwise, a message is displayed that parameters are passed according to Unix convention, with the sign '-' (minus), and not with '/'.

Examples of using STComp05 compiler (all operating modes) in EBNF notation:

1. `STComp05.exe [/?] [/help] [-?] [{-h | --help}]`
   Displays version numbers of command interpreter and compiler, together with help.

2. `STComp05.exe [--opt OptLevel] [--cf CompilerFlags] [--sf ScannerFlags] {{--lcf | -l} LibConfFile} [{--dcp | -d} DebugConfFile] [{--out | -o} OutputCode] [--project] [--] CompileFileList`
   Compiles files to universal executable code (*.xcp files).

3. `STComp05.exe {{--lcf | -l} LibConfFile} --library [--lib-copyright CopyInfo] [--lib-menu MenuInfo] [--lib-ver VerInfo] {--exportlist CommaIdList} [{--out | -o} OutputFile] [--] CompileFileList`
   Compiles files to library form (*.lcp files).

Parameters of configuration settings (selected)

CPDev IDE versions 1.0.2.* uses system of application settings (available from `--get-conf` and `--set-conf` arguments) provided by .NET. Common configuration for all programs in package is stored in global configuration module (available from `--gu-get-conf` and `--gu-set-conf` arguments).

Name	Type	Description
MW_ConfigFile	string	Default specification file of virtual machine ('?' (question mark) resets to default during start up).
~~COMM_Port~~	string	Deleted (COM port number)
~~COMM_Rate~~	string	Deleted (COM baud rate)
~~COMM_Bits~~	string	Deleted (COM data bits)
~~COMM_StopBits~~	string	Deleted (COM stop bits)
UseColorizedST	int	Number of ST editor being used.
UseAutoPreFillST	bool	Fills in new POUs with ST templates.
UseAutoUpdatePropName	bool	Automatic synchronization (update) of names in properties window
STHighLightSet	string	Set of records to color (highlight) ST syntax
RFL_Entries	string	Entries of recent file list
UserLangUICode	int	LCID code of regional settings (if 0, acquired automatically from control panel)
RFL_MaxEntries	int	Maximum number of entries in recent file list.
MW_SimulTool	string	Tool for simulation of universal executable code. If '?', returns to default.

`MW_SendTool`	string	Tool for downloading configuration to controller. If '?', displays "Not defined tool" message.
`UseAutoUnlock Edit`	bool	Automatically unlocks component for editing.
`UpdateConnTim eout`	int	Time-out for internet connections.
`UseAutoInclud eLibs`	bool	Automatically includes libraries into new projects.
`AutoIncludeLi brary`	string	List of automatically included libraries. If '?' replaced by default.
`OpimizationLe vel`	int	Optimization level: 1 – recommended, 2 – with detection of unused variables
`UseAutoEdVal`	bool	Automatically synchronizes name in editor window with name in project tree.
`UseAlwaysAskC lose`	bool	Always asks "Save change in the project" before closing the project.
`HelpPathRoot`	string	Path to help folder. If '?', returns to default.
`STComp05DefOp t`	int	Default flag settings for STComp05 compiler (parser).
`ShowSMCModbus AddressInRepo rt`	bool	Show Modbus SMC addresses in compilation report.
`ShowIntouchMo dbusAddressIn Report`	bool	Show Modbus Intouch addresses in compilation report.
`UseBoldFontFo rCurrentProje ct`	bool	Uses bold font for project name when object selected in project tree.
`AdvancedUserM ode`	bool	Switches configuration window into service mode (advanced).
`AllowUseCppCo mmentsInST`	bool	Allows // comments in ST code
`AllowUseNeste dComments`	bool	Allows nested comments (displayed as non nested due to technical reasons)
`STPrimePropos alWidth`	int	Width of basic autocomplete window for identifiers
`STSecondPropo salWidth`	int	Width of additional autocomplete window for keywords, types, etc.
`MaxNumTask`	int	Maximum number of tasks in the project, if not set in global configuration.
`ShouldAskToOp enSavedReport`	bool	Asks whether to open file with saved report of the project.
`OverrideMWSFro mProject`	bool	Replaces (overrides) virtual machine specification file written in the project by default specification file from application settings.

VarContainerFlags	int	Flags to display unfolded variables in project tree: 1 – type of variable; 2 – physical address; 4 – logical address; 8 – comment;
CodeGeneratorFlags	int	Code generator flags: 1 – use global variables optimization for projects with one task.
ShowHelpAsMDIChild	bool	If `true`, displays help window as MDI child; if `false`, help becomes main window and appears in the task bar.
STColorizedColorizeSpecProc	bool	Displays special parsing section (`*# ...*`) by STColorized editor.
STColorizedColorizeVerifCond	bool	Displays section (`*@ ...*`) for ST language verifier by STColorized editor.
SettingsLevel	int	Level of settings applied for the user. If 0, provides all default settings during start up.
STPrimeProposalAutoDelay	int	Delay time (in milliseconds) before showing main list of autocomplete objects. (The list will not appear if time is too large).
FollowReportList	bool	Flag to follow new notifiers in bottom message list.
Debug_DisplayStartedProcess	bool	Diagnostics flag to display newly running process (in bottom message list). Since v. 1.0.2.18.
STColorized_TabSize	int	Number of space characters inserted when tab key is pressed in single colorable (STColorized in previous editions) editor. Since v. 1.0.2.20.
STColorized_SynEditOptions	string	Set of options for single colorable editor (STColorized). It have form of a string with text values separated by comma. Since v. 1.0.2.20.
DefaultCompiler_AssemblyName	string	Default .NET assembly name for compiler (in fact parser) (since v. 1.0.2.20).
DefaultCompiler_ClassName	string	Default .NET class name for compiler (in fact parser) (since v. 1.0.2.20).
DefaultGenerator_AssemblyName	string	Default .NET assembly name for code generator (since v. 1.0.2.20).
DefaultGenerator_ClassName	string	Default .NET class name for code generator (since v. 1.0.2.20).
ShowFullNameInReport	bool	Flag for column visibility of Full variable name in compilation report (since v. 1.0.2.22).
ShowCommentInReport	bool	Flag of column visibility of variable comment in compilation report (since v. 1.0.2.22).
CheckForEmptyDirWhenSaving	bool	If True enables checking for empty destination directory before saving project (since v. 1.0.2.29)
SumatraExecutable	string	Path to executable file of Sumatra PDF Browser

`UseSumatraPDFBrowser`	bool	Main setting for choosing between Sumatra and Adobe Reader PDF displayer.
`UseFINTinSTSyntax`	bool	Enables displaying FINT type in colour when coding in ST language.
`CPVisPath`	string	Contains a string where CPVis path is be stored.
`SaveBeforeCompilation`	bool	True – enables automatic saving before compilation
`AutoSaveAfterTime`	bool	True – enables automatic saving after minutes defined in `AutoSaveInterval` field.
`AutoSaveInterval`	int	Amount of minutes to automatically save projects on the disk.
`KeepIdentsInUpCase`	bool	Allows compiler to keep all identifiers in upper case (enable for compatibility with versions prior to 1.1.1.5)

Global configuration settings

The settings listed below configure all programs of CPDev package (so they are global). All fields are in text format and applied to current user. In some cases values can be interpreted as int or double. Setting name is case sensitive (upper or lower). Global configuration is handled by switches `--gu-set-conf Arg0` and `–gu-get-conf Arg0`, with structure of `Arg0` described at the beginning (see `--set-conf Arg0`). It is recommended to use `int` instead of `bool` while deailing with Boolean value.

Name	Type	Description
`Global.Comm.Name`	s	COM port name; ("COM1", "COM2" or other selected or entered by the user.
`Global.Comm.Bits`	s/i	Number of data bits; INDEX in combobox. Values: {4,5,6,7,8}. -1 if none selected.
`Global.Comm.Parity`	s/i	Parity bit; INDEX in combobox. Values: {Even, Odd, None, Mark, Space}. -1 as above.
`Global.Comm.Bauds`	s/i	Baud rate; INDEX in combobox. Values: {75,110,134,150,300, 600,1200,1800,2400,4800,7200,9600,14400,19200,38400, 57600,115200,128000,230400}. -1 as above.
`Global.Comm.BStop`	s/i	Stop bits; INDEX in combobox. Values: {1,1.5,2}. -1 as above.
`Global.Comm.CtrFlow`	s/i	Flow control; INDEX in combobox. Values: {None, Hardware, Xon / Xoff}. -1 as above.
`Global.SMC.DevNo`	s/i	SMC controller number. INT number as text.
`Global.SMC.VertSpeed`	s/i	Speed (baud rate) of horizontal communication (with distributed I/O modules); INDEX in combobox. Values: {2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 1200}.
`Global.SMC.VertMode`	s/i	Horizontal communication mode; INDEX in combobox. Values: {8N1,8N2,8E1,8O1,7N2,7E1,7O1}.
`Global.UserLangUICode`	s/i	Language used for application. LCID as INT number.

`Global.SMC.TimeOut`	s/i	TimeOut for SMC controller
`Global.SMC.SilentTime`	s/i	Time of silence for SMC controller
`CPDev.Update.NotifyURL`	s	Address of server with information script on last CPDev version.
`CPDev.Update.NotifyAccUsr`	s	Name of the user authorized to read the script.
`CPDev.Update.NotifyAccPass`	s	Password of the user [`CPDev.Update.NotifyAccUsr`] to web page.
`CPDev.Update.WebProxy`	s/i/b	Use http proxy for connection? INT number (<> 0 denotes TRUE)
`CPDev.Update.ProxyAddress`	s	Proxy server address.
`CPDev.Update.User.ProxyUser`	s	Proxy server user.
`CPDev.Update.User.ProxyPassword`	s	Password of the proxy server user [`CPDev.Update.User.-ProxyUser`].
`Global.Comm.NumsOfCom`	s/i	Number of COM ports displayed in global settings window.
`Global.Comm.AllowAnyCom`	s/i/b	Allows to enter any text as communication port.
`CPDev.ST.STColorized.TemplateFile`	s	Location of file with templates for STColorized editor.
`CPDev.Custom.MaxTasks`	s/i	Maximum number of tasks in a project.
`Global.SimulatorExe`	s	Name of default simulator (can be written with application variables e.g. `$AppDir\CPSim.exe`).
`Global.ConfigurerExe`	s	Name of default configurer(can be written with application variables e.g. `$AppDir\CPCon.exe`).
`Global.DefaultVMSpec`	s	Name of default file that specifies capabilities of virtual machine (e.g. `$VMsDir\VM-SMC.xml`)
`CPDev.Extensions.ModuleCount`	s/i	Number of user external modules
`CPDev.Extensions.Module.0.AssemblyName`	s	Name of first user module (other modules are numbered consequtively).
`CPDev.Extensions.Module.0.StartupClassName`	s	Name of start up class that initializes the first user module (other modules are numbered consequtively).
`CPDev.ST.EnableBackgroundCompilation`	s/i/b	Enables (value `1`) or disables (value `0`) project compilation in background for code editors.
`CPDev.Run.ConfigurerArguments`	s	Set up arguments for configurer. `{0}` is replaced to current language code, `{1}` is replaced to full path for *.dcp file with current project.

Application variables

The table below displays application variables that are created by installer. It also describes their meaning. Values can be changed by global configuration window opened in advanced mode.

Name	Description
AppDir	Application base directory
HlpDir	Base directory for help system
LibDir	Base directory for libraries
ExamplesDir	Directory with examples
CustomTemplDir	Directory with user code templates
TechDocDir	Directory with technical documents
VMsDir	Directory with Virtual Machine Description files (LCFs).