ICP Family Programmers

Command Line Interface

IMPORTANT NOTE:

- Starting from Mar-2019 Softlog Systems provides “Advanced Command Line” (IcpWin**Advanced**ComLine.exe) additionally to existing (legacy) “Command Line” (IcpWinComLine.exe)
- “Advanced Command Line” is supposed to be down-compatible with the existing “Command Line”
- “Advanced Command Line” additionally supports secure programming operations and multiple parameters in INI file

Contents

1 Known Issues

	Issue	*Workaround*
1	G3 programmers: errorcode 110 will be reported for environment file (PJ2) with size more than 1.3MByte when /q switch is not specified	Use switch /q as shown in example 7.5

2 Installation

- Run DLL installation file “IcpDll_setup_dll_XXX.exe”.

3 Files

NOTE: 64-bit version is available in “x64_Package”

- IcpWinComLine.exe
- IcpWinAdvancedComLine.exe
- IcpDll.dll
- GetDllVersion.bat: run it to check versions of “IcpWinComLine.exe” and “IcpDll.dll”
- GetDllAdvancedVersion.bat: run it to check versions of “IcpWinAdvancedComLine.exe” and “IcpDll.dll”
- c_icpexp.h (not required for command line)
- fr_exp.h (not required for command line)
- fr_ftb9.h (not required for command line)
- SenFromHexTemplate1.ini: template INI file for SEN generation from HEX
- SenFromPj2Template2.ini: template INI file for SEN generation from PJ2

4 ICP Firmware Options

DLL/Command Line Activation (D) should be "Yes" to use ICP family command line interface: see availability in "ICP for Windows" - Help - About - Details

5 Command Line Switches

NOTE: all switches are **not** case sensitive

###	*Switch*	*Description*	*Example*	*Minimum DLL version*	*Comments*
1.	/c	Load configuration file	/cicp01.cfg	-	Mandatory excluding /n,/h,/?,/k
2.	No switch	Hex file to be loaded	Hex1.hex	-	-
3.	/s	Serialization file to be loaded	/sser1.ser	-	-
4.	/i	Programming (standalone mode)	/i	4.4.1	**Always use for multi-channel parallel programming**
5.	/n<channel>	Get result from selected GANG channel	/n1	4.4.2	Use after GANG programming is done
6.	/a<space>	Programming (PC-driven mode), <space> is a sum (**decimal**) of the following memory spaces: 1 - PM (program memory) 2 - ID 4 - DM (EEPROM) 8 - reserved 16 - CB (configuration bits) 32 - BOOT memory 64 - OTP	/a23	-	23 means “all spaces” for the most devices. PC-driven mode is not available for GANG multi-channel programming
7.	/r<space>	Read (PC-driven mode only)	/r23	-	-
8.	/v<space>	Verify (PC-driven mode only)	/v23	-	-
9.	/l<space>	Blank check (PC-driven mode only)	/l23	-	-
10.	/f<filename>	File to be saved after read	/fRead1.hex	-	-
11.	/g<value>	Save errorcode(s) in file “auto01.res”	/g0 - disable /g, /g1-enable /g2-extended	4.4.2 for /g2	**Always set to /g2 for multi-channel programmer**
12.	/b<value>	PM beginning **(hex)** , valid if both /b and /e are specified	/b50	-	-
13.	/e<value>	PM end **(hex),** valid if both /b and /e are specified	/e1FFE	-	-
14.	/d<value>	Enable/disable debug message box after an operation	/d0 (disable) /d1 (enable) /d2 (on error)	4.4.1 for /d1	Default: disabled
15.	/w<value>	Enable/disable progress window	/w0 (disable) /w1 (enable)	-	Default: enabled
16.	/h	Display command line and DLL version string	/h	-	Use with /d0 to disable message box
17.	/t<filename>	Transfer environment file to programmer	/tenv1.pj2	4.4.1	-
18.	/u<filename>	Transfer secure environment file to a **single channel** of programmer	/usecenv1.sen	4.8.2	-
19.	/x	Enable/disable PC “sleep” NOTE: - 0: faster ICP operation in PC-driven mode - 1: less PC load	/x0 (faster ICP operation)		Default: 1 (less PC load)
20.	/o	Enable/disable gap eliminator for: 1 - PM (program memory) 4 - DM (EEPROM) NOTE: gap eliminator settings are **NOT** got from icp01.cfg (disabled)	/o5 (enable for PM and DM) /o0 (disable)	4.12.1	Default: disabled **Enable for faster programming**
21.	/q	Select environment (use with /I, /t or /u)	/q6 (select environment 6)	4.13.1	Default: last used
22.	/y	Load security bit to programming buffer (PC-driven mode) which is not a part of HEX file. Valid for ATSAMC21 and similar families	/y1 (enable)	8.1.3a	Default: 0 (no security or taken from HEX file)
23.	/zc	Select COM port number (overwrite COM in icp01.cfg file)	/zc0 Select COM1	3.1.2	DOS-style COM number (0=COM1, 1=COM2,…)
24.	/zr	Reset all enabled channels of ICP programmer	/zr	3.1.4	Minimum firmware 36.4
25.	/k	Advanced Command Line **only**: Load INI file	/kMyIni.ini	8.11.1a	-
26.	/m	Advanced Command Line **only**: Generate secure environment file (SEN)	/m0 (delete/counter) /m1 (from HEX) /m2 (from PJ2)	8.11.1a	-

6 Return Values (Errorcodes)

See document ***"DLL Description.pdf"*** for a list of error codes

7 Examples

7.1 Program Device in PC-driven Mode, Gap Eliminator = ON

- Configuration file: icp01.cfg
- Hex file: hex1.hex
- Space for programming: all memories (PM, EEPROM, ID, Configuration bits)
- NOTE: gap eliminator settings are **NOT** got from icp01.cfg therefore they are set here (/o5)

```
IcpWinComLine.exe   /cicp01.cfg      hex1.hex   /a23   /o5
```

7.2 Program Device in PC-driven Mode + Serialization + Message Box

- Configuration file: icp01.cfg
- Hex file: hex1.hex
- Serialization file: ser1.ser
- Space for programming: all memories (PM, EEPROM, ID, Configuration bits)

```
IcpWinComLine.exe   /cicp01.cfg      hex1.hex   /sser1.ser /a23 /d1
```

7.3 Erase Device (Using Program Operation)

- Configuration file: icp01.cfg
- Hex file: Blank1.hex (file contains blank values only)
- Space for programming: all memories (PM, EEPROM, ID, Configuration bits)
- PM start: 0
- PM end: 3E (valid for any PIC10/12/16/18)

```
IcpWinComLine.exe   /cicp01.cfg      Blank.hex  /a23    /b0     /e3E
```

7.4 Program 4 Devices using ICP2-GANG (Standalone Mode)

- Configuration file: icp01.cfg, contains GANG channels 1-4

```
Step 1 - gang programming:        <path to IcpWinComLine.exe>   /cicp01.cfg     /i      /g2
Step 2 - get result for channel 1: <path to IcpWinComLine.exe>  /n1     /g
Step 3 - get result for channel 2: <path to IcpWinComLine.exe>  /n2     /g
Step 4 - get result for channel 3: <path to IcpWinComLine.exe>  /n3     /g
Step 5 - get result for channel 4: <path to IcpWinComLine.exe>  /n4     /g
```

NOTE: steps 2-5 are not required if return code for step 1 is zero (all channels passed)

7.5 Transfer Environment File to Programmer

- Configuration file: icp01.cfg
- Destination environment: 2
- Environment file: env1.pj2

```
 path to IcpWinComLine.exe   /cicp01.cfg     /q2     /tenv1.pj2
```

IMPORTANT: Environment is saved in **non**-volatile flash memory of ICP2 programmer; therefore avoid multiple transfers of the same environment to keep endurance of the flash memory

8 Getting Started

8.1 Create a folder with your project

- Create a folder for your project, for example: C:\Projects\Project1
- Place your hex file into the folder above, for example: C:\Projects\Project1\Hex1.hex

8.2 Install ICP GUI software

- Install ICP GUI software by running setup file “IcpWin_setup_X_XX.exe”

8.3 Create ICP GUI shortcut to your project

- Copy created shortcut “ICP for Windows” and rename the copy, for example to “My ICP Project”
- Open the shortcut properties and change “Start in:” directory to “C:\Projects\Project1”

8.4 Run ICP GUI software

- Press on “My ICP Project” to run ICP GUI software
- Configure ICP family programmer for operation with your device
- Create serialization file in the project directory (if required), for example “ser1.ser”
- Save configuration: select “File/Save Configuration” and exit the software
- In the project directory rename “icp01.cfg” to “MyProject.cfg”

8.5 Create command line shortcut (PC-driven programming)

- Create shortcut to “IcpWinComLine.exe”
- Open the shortcut properties and edit as follows:
  Target: “…..\IcpWinComLine.exe” /cMyProject.cfg Hex1.hex /sser1.ser /a23 /d1
  Start in: “C:\Projects\Project1”
- Press on the created shortcut to validate the operation

9 Advanced Command Line

9.1 Secure Programming

- Read "Secure Programming Utility User's Manual.pdf" to be familiar with secure programming

9.2 Advanced Command Line Examples

- Generate SEN file from HEX (/d1 is optional): IcpWin**Advanced**ComLine.exe **/k**SenFromHexTemplate1.ini **/m1** /d1
- Generate SEN file from PJ2 (/d1 is optional): IcpWin**Advanced**ComLine.exe **/k**SenFromPj2Template2.ini **/m2** /d1
- Legacy operation: all legacy command line operations are supported

9.3 File Path/Name

- INI file in **/k** switch: name of the INI file should be either **absolute** or **relative** to the current working directory
- File names specified inside INI file should be either absolute or relative to the current working directory

9.4 INI File Parameters

- INI file is primarily intended for secure programming operations (SEN file generation)
- SEN file can be generated from 3 sources (none, HEX or PJ2):
  - none (neither HEX nor PJ2): applicable to "update counter only" or "delete secure environment and counter inside programmer"
  - from HEX file. NOTE: intermediate PJ2 file is generated
  - from PJ2 file

	Section/Name	***Value***	***Description***	***Default Value***	***Usage: Source = HEX (/m1)***	***Usage: Source = PJ2 (/m2)***	***Usage: Source = None (/m0)***
	[Input]						
1.	CfgFileIn	Path/name	ICP CFG file	-	Mandatory		
2.	SidFileIn	Path/name	Security ID file (SID), created by ICP Secure Programming Utility	-	Mandatory		
3.	Pj2FileIn	Path/name	Environment file (PJ2)	-	-	Not required if counter only is updated	-
4.	HexFileIn	Path/name	Original HEX file	-	Mandatory	-	-
5.	UniversalHexLoader	Path/name	Defines specific offsets to HexFileIn, see 9.7	-	-	-	-
6.	SecurityBit	0=no security 1=with security	Security bit settings for devices like ATSAMC/D	0	Required for ATSAMC/D	-	-
7.	SerFileIn	Path/name	Serialization file (SER)	-	Optional	-	-
8.	ExtractSecBufFromHex	0=don't modify HEX, 1=fill secure buffer area with random values 2=fill with SecBufReplaceVal	Replace data in PM or DM area according to SID file settings	0	Optional	-	-
9.	SecBufReplaceVal	Value 0-FF	Fixed value to fill secure buffer area	-	Required if ExtractSecBufFromHex=2	-	-
10.	GapEliminator	Sum of memory spaces: 0=no gap eliminator 1=PM(flash),2=DM(EEPROM)	Enable/disable Gap Eliminator. NOTE: gap eliminator settings are **NOT** got from icp01.cfg	0	Optional	-	-
11.	MemorySpace	Sum of spaces: 1=PM, 2=User ID, 4=DM, 16=FU, 32=boot, 64=OTP Usual value: 23	Memory areas to be programmed. NOTE: ICP DLL removes unexisting spaces	-	Mandatory	-	-
	[Update_byte_XXX]	XXX = A, B, …Z, for example: **[Update_byte_A]**	This section can be applied to the following operations only: **/m1, /a or /v**		Optional section		
12.	Space	One space: 1=PM, 2=User ID, 4=DM, 16=FU, 32=boot, 64=OTP	Space to be modified, see enum MEMORY_SPACES	-	Required for this section	-	-
13.	Offset	**Offset** in the programming buffer in **[bytes]**	Offset in [bytes], **not** absolute address	-	Required for this section	-	-
14.	Value	Value (byte)	Explicit value to be written to the specified offset	-	At least one of those keywords is required for this section, otherwise the section is ignored	-	-
15.	Modify_bits_to_1	OR mask	Reads existing byte from the buffer and sets bits specified as "1" to "1" Does **nothing** if keyword "Value" above is filled	-		-	-
16.	Modify_bits_to_0	AND mask	Reads existing byte from the buffer and sets bits specified as "1" to **"0"** Does **nothing** if keyword "Value" above is filled	-		-	-
	[Operation]						
17.	BatchNumber	0=[use value+1] from SID 1…65535=use this value	Batch number value. NOTE: SID file is always updated with used batch value	-	Mandatory		
18.	UpdateMask	SEN properties: • 0x28=update	Sum of properties as in enum SEC_ID_UPD_MASK	-	Mandatory • One of 4 combinations should present		

	Section/Name	*Value*	*Description*	*Default Value*	*Usage: Source = HEX (/m1)*	*Usage: Source = PJ2 (/m2)*	*Usage: Source = None (/m0)*
		environment and counter • 0x08=update counter only • 0x20=update environment only • 0x40=delete secure environment and counter inside programmer. NOTE: use with /m0 only	in "c_icpexp.h" file: maskCNT maskENV maskDEL_ENV_CNT Other masks are unused		• If "Delete secure environment and counter" is set, then other masks are ignored **IMPORTANT**: in order to use **permanent** batch control **without** counter (unlimited counter) set as follows: UpdateMask=0x08 (update counter) **and** CounterValue= 0xFFFFFF (unlimited)		
19.	CounterValue	Counter value: ICP2-Portable: 1-9999(dec) Other: 1-999999(dec) 0xFFFFFF=unlimited	Counter value	-	Mandatory if mask "update counter" is set		-
	[Output]						
20.	SenFileOut	Path/name	Generated SEN file		Mandatory		
21.	Pj2FileOut	Path/name	Generated PJ2 file (from HEX)		Mandatory	-	-
22.	HexFileUnchangedOut	Path/name	Original HEX file saved by ICP DLL		Optional	-	-
23.	HexFileChangedOut	Path/name	Modified HEX file saved by ICP DLL		Optional	-	-
	[Portable]						
24.	PortableCntLimitEnv1	1…9999, 0xFFFF=unlimited	Non-secure counter value for ICP2-Portable, used during environment file (PJ2) transfer. Not applicable to secure programming	0xFFFF (unlimited)	N/A	N/A	N/A
25.	PortableCntLimitEnv2						
26.	PortableCntLimitEnv3						
27.	PortableCntLimitEnv4						
28.	PortableCntLimitEnv5						
29.	PortableCntLimitEnv6						
30.	PortableButLockEnv	0=not locked, 1=locked	Defines state of ENV button used during environment file (PJ2) transfer. Not applicable to secure programming	0 (not locked)	N/A	N/A	N/A
	[Preserve]		See example 9.6				
31.	PreserveSpace	Space to be preserved according to enum MEMORY_SPACES	Only one memory space can be specified, for example 1 (PM_SPACE)	0 (normal operation)	N/A	N/A	N/A
32.	ReadSpace	Space for read operation according to enum MEMORY_SPACES	Space for read operation before re-programming	0	N/A	N/A	N/A
33.	ReadPmAddrBeg	PM start address for read operation	PM start and end addresses for read operation before re-programming. Full range will be read if not specified	0	N/A	N/A	N/A
34.	ReadPmAddrEnd	PM end address for read operation		0	N/A	N/A	N/A
35.	ReadDmAddrBeg	DM start address for read operation	DM start and end addresses for read operation before re-programming. Full range will be read if not specified	0	N/A	N/A	N/A
36.	ReadDmAddrEnd	DM end address for read operation		0	N/A	N/A	N/A
37.	PreserveOffset	Offset in programming buffer ***[bytes]***	Start offset of bytes to be "preserved"	0	N/A	N/A	N/A
38.	PreserveSize	Size to be preserved ***[bytes]***	Number of bytes to be "preserved"	0	N/A	N/A	N/A
39.	PreserveSanityCheck	0=no sanity check 1=sanity check	Sanity check: if all bytes to be preserved = 0x00 or all bytes to be preserved = 0xFF then error	0	N/A	N/A	N/A
	[Gang]		Defines number of active channels of ICP2-GANG, ICP2-COMBO or ICP2-ISO/LAN(multi)				
40.	GangNumberBoxesSet	1…16	Number of "boxes", 1 box = 4 channels	1	N/A	N/A	N/A
41.	GangChannelsBox1, GangChannelsBox2, … GangChannelsBox16	0…15	Binary mask of enables channels inside the box: 1 = channel 1 2 = channel 2 4 = channel 3 8 = channel 4	15	N/A	N/A	N/A

9.5 INI File Example 1 (SenFromHexTemplate1.ini)

```
; Run command line as: IcpWinAdvancedComLine.exe /kSenFromHexTemplate1.ini /m1 /d1
; This INI file is used to generate SEN from HEX as follows:
; - file locations:
;   - c:\Softlog\Product_1: files for product "Product_1" (unsafe location)
;   - c:\Softlog\SafeLocationForSid: safe location for SID files
;   - c:\Softlog\SafeLocationForHex: safe location for HEX files
; - no serialization
; - STM32G071C8: set read protection level 1 and change "PCROP zone preserved when RDP level decreased"
;   to "Erased"
; - fill secure buffer area with random values
; - enable Gap Eliminator for PM
; - program memory spaces: PM+User ID+DM+FU. NOTE: ICP DLL removes non-existing spaces
; - get batch number from SID file
; - update SID file with latest batch number
; - update environment and secure counter
; - unlimited counter

[Input]
CfgFileIn   = c:\Softlog\Product_1\Product_1.cfg          ; icp01.cfg file (CFG)
SidFileIn   = c:\Softlog\SafeLocationForSid\Sid_A.sid     ; Security ID file (SID)
Pj2FileIn   =                                             ; Input Environment file (PJ2)
HexFileIn   = c:\Softlog\SafeLocationForHex\Hex_1.hex     ; Original HEX file (HEX). ***IMPORTANT***-validate
                                                          ; that HEX contains code protection/security
SecurityBit = 1  ; ATSAMC/ATSAMD ***only***: 1=set security; 0=no security
SerFileIn   =    ; optional serialization file (SER)

ExtractSecBufFromHex = 1  ; 0=don't modify HEX, 1=fill secure buffer area with random values,
                          ; 2=fill with SecBufReplaceVal value
SecBufReplaceVal     =    ; 0-255, see description above for ExtractSecBufFromHex
GapEliminator        = 1  ; 0=no gap eliminator, 1=PM(flash), 4=DM(EEPROM), 5=PM+DM
MemorySpace          = 23 ; sum of memory spaces for operation: 1=PM, 2=User ID, 4=DM, 16=FU, 32=boot, 64=OTP

[Update_byte_A]
Space = 16 ; FU_SPACE – set read protection (RDP) to level 1
Offset = 0 ; fuse offset, LSB byte of fuse word of address 0x1FFF7800 which occupies offsets 0...3
Value = 0xFF ; protection level 1 (any value different from 0xAA and 0xCC)

[Update_byte_B]
Space = 16 ; FU_SPACE - change "PCROP zone preserved when RDP level decreased" to "Erased"
Offset = 19 ; fuse offset, MSB byte of fuse word of address 0x1FFF7810 which occupies offsets 16...19dec
Modify_bits_to_1 = 0x80 ; set MSB bit to 1, keep other bits unchanged

[Operation]
BatchNumber          = 0        ; 0=[use value+1] from SID, 1…65535=use this value
UpdateMask           = 0x28     ; 0x28= update environment and counter, 0x20=update environment,
                                ; 0x08=update counter, 0x40=delete environments and secure counter
CounterValue         = 0xFFFFFF ; 0xFFFFFF=unlimited ; ICP2-Portable: 1-9,999 ; other programmers: 1-999,999

[Output]
SenFileOut           = c:\Softlog\Product_1\Product_1.sen           ; output SEN file
Pj2FileOut           = c:\Softlog\SafeLocationForHex\Product_1.pj2  ; output PJ2 file
HexFileUnchangedOut  = c:\Softlog\SafeLocationForHex\Unchanged.hex  ; original HEX file saved by ICP DLL (can be
                                                                    ; used for visual compare)
HexFileChangedOut    = c:\Softlog\SafeLocationForHex\Changed.hex    ; modified HEX file saved by ICP DLL (can be
                                                                    ; used for visual compare)
```

9.6 Example: “Preserve” Memory

9.6.1 Task

Device:	PIC18F85J10 (full PM range 0x0000-0x7FF6, fuses located at 0x7FF8-0x7FFE, PM page size 0x40 = 64 bytes)
Data to be preserved:	6 bytes of PM (flash) starting from address 0x7FF0
CFG file:	“icp01.cfg”
HEX file:	“Hex1.hex”

9.6.2 INI file “Example-pm.ini”

```
[Input]
CfgFileIn                 = icp01.cfg      ; icp01.cfg file (CFG)
HexFileIn                 = Hex1.hex       ; HEX file (HEX)

[Preserve]
ReadSpace                 = 17             ; space for read operation (PM_SPACE + FU_SPACE)
ReadPmAddrBeg             = 0x7FC0         ; start address of area containing data to be preserved (full page)
ReadPmAddrEnd             = 0x7FF6         ; end address
ReadDmAddrBeg             =                ; not used
ReadDmAddrEnd             =                ; not used

PreserveSpace             = 1              ; space with preserved data (PM_SPACE)
PreserveOffset            = 0x7FF0         ; offset in programming buffer ***[bytes]***
PreserveSize              = 6              ; size to be preserved ***[bytes]***
PreserveSanityCheck       = 1              ; 1=sanity check, 0=no check. If all bytes 0x00 or all 0xFF -> fail
```

9.6.3 Command Line

IcpWinAdvancedComLine.exe /k**Example-pm.ini** /a23 /d1

9.6.4 How it works

- Step 1: read PM and FU (ReadSpace):
  - PM (ReadSpace contains 1): 0x7FC0 (ReadPmAddrBeg) to 0x7FF6 (ReadPmAddrEnd)
  - fuses (ReadSpace contains 16)
- Step 2: internally save 6 bytes (PreserveSize) of PM (PreserveSpace) starting from offset 0x7FF0 (PreserveOffset)
- Step 3: load Hex1.hex (HexFileIn)
- Step 4: replace 6 preserved bytes (PreserveSize)
- Step 5: execute PC-driven programming for entire chip (/a23). NOTE: /a17 is also OK for PIC18F85J10

9.7 Example: Universal HEX Loader

```
[Input]
CfgFileIn                 = icp01.cfg
HexFileIn                 = ATtiny816_with_810000_820000_830000.hex
UniversalHexLoader        = "PM_MAIN=0x0; DM_MAIN=0x810000; CB_REAL_FUSE=0x820000; CB_LOCK_BITS=0x830000"
```

9.8 Advanced Command Line Return Values (Errorcodes)

- No error: 0

- DLL-provided errorcodes: see document ***"DLL Description.pdf"*** for a list of error codes

- Advanced Command Line errorcodes:

```
enum ERROR_CODES {
  ERR_NO_RECORD_FOR_CH      = -1,     // No record for specified channel in file auto01.res
  ERR_CH_NUM_OUT_OF_RANGE   = -2,     // Channel number is out of range
  ERR_CANT_OPEN_RES_FILE    = -3,     // Can't open file auto01.res
  ERR_CANT_OPEN_CH_FILE     = -4,     // Can't open file ch_i.res
  ERR_INVALID_SID_FILE      = -5,     // Invalid or corrupted SID file
  ERR_EMPTY_SEC_BUFFER      = -6,     // Empty secure buffer in SID file
  ERR_CREATE_SEN_FILE       = -7,     // Can't create secure environment
  ERR_SAVE_SID_FILE         = -8,     // Can't save SID file
  ERR_BATCH_NUM_OVERFLOW    = -9,     // Batch number overflow
  ERR_SEC_BUF_RANGE         = -10,    // Secure buffer address range does not fit into the target buffer
  ERR_INI_PRESERVE_SECTION   = -11,    // Inconsistent data in [Preserve] section of INI-file
  ERR_PRESERVE_INVALID_RANGE = -12,   // Preserve range does not fit into the buffer
  ERR_PRESERVE_SANITY_CHECK  = -13,   // Preserve sanity check failed
  ERR_INSUFFICIENT_MEMORY    = -14,   // Insufficient memory
  ERR_NUM_BOXES_OUT_OF_RANGE = -15,   // Number of boxes is out of range
  ERR_UPDBYTE_SANITY_CHECK   = -16,   // Update byte sanity check failed

  //command line
  ERR_CL_HELP               = -99,    // --> the message may vary
  ERR_CL_NOACTION           = -100,   // Command line: no action specified
  ERR_CL_MEMSPACE           = -101,   // Invalid memory space specified
  ERR_CL_REP_ACTION         = -102,   // Repeated specification of action
  ERR_CL_FILE_NOT_SPECIFIED = -103,   // File not specified
  ERR_CL_FILE_NOT_FOUND     = -104,   // File not found
  ERR_CL_DUPL_HEX_FILE      = -105,   // Duplicate HEX file specified
  ERR_CL_INVALID_PM_BEGIN   = -106,   // Program memory begin address not specified on invalid
  ERR_CL_INVALID_PM_END     = -107,   // Program memory end address not specified on invalid
  ERR_CL_READ_FILE_NOT_SPEC = -108,   // READ file not specified
  ERR_CL_INVALID_G_PARAM    = -109,   // Invalid /G parameter
  ERR_CL_INVALID_M_PARAM    = -110,   // Invalid or repeated /M parameter
  ERR_CL_INVALID_N_PARAM    = -111,   // Invalid /N parameter
  ERR_CL_UNKNOWN_SWITCH     = -112,   // Unrecognized switch
  ERR_CL_MEMSPACE_POST      = -113,   // 'MemorySpace' not specified or invalid
  ERR_CL_EXTRACT_SEC_BUF    = -114,   // 'ExtractSecBufFromHex' is invalid
  ERR_CL_BATCH_NUMBER       = -115,   // 'BatchNumber' not specified or out of range
  ERR_CL_UPD_MASK_INVALID   = -116,   // 'UpdateMask' not specified or invalid
  ERR_CL_UPD_MASK_MISMATCH  = -117,   // 'UpdateMask' contains bits incompatible with the required operation
  ERR_CL_COUNTER_VALUE      = -118,   // 'CounterValue' not specified or invalid
  ERR_CL_ENV_CNT_LIMIT      = -119,   // 'PortableCntLimitEnvN' is invalid
  ERR_CL_ENV_NUM_INVALID    = -120,   // Invalid environment number
  ERR_CL_INI_FILE_ERROR     = -121,   // INI-file error
};
```

10 History

- Version 4.1.2 (Mar-24)
  - added support for ICP2-ISO(G3) and ICP2-LAN(G3) programmers
  - Advanced Command Line: added sections [Update_byte_A], [Update_byte_B,…, [Update_byte_Z] – see [Update_byte_XXX]. Those sections are applied to the following operations: /m1, /a or /v

- Version 3.1.4 (Aug-22)
  - added switch /zr for ICP programmer reset

- Version 3.1.3 (Mar-21, no changes till Jan-22)
  - added UniversalHexLoader, GangNumberBoxesSet and GangChannelsBox1…GangChannelsBox16

- Version 3.1.2 (Jul-20)
  - 32-bit and 64-bit versions are available
  - added switch /zc for COM selection
  - don't use switch /? for help, use /h only
  - Advance Command Line: added memory preserve operation, see PreserveSpace

- Version 1.1.12/2.1.1 (Jan-2019):
  - Minor text corrections, no version change

- Version 1.1.12/2.1.1 (Jun-2019):
  - Added PortableCntLimitEnv1 and PortableEnvButLock

- Version 1.1.12/2.1.0 (Mar-2019):
  - Added “Advanced Command Line”

- Version 1.1.12 (Jul-2018):
  - “IcpWinComLine.exe” provides more diagnostic messages in case of incorrect or insufficient parameters

- Version 1.1.11 (Jun-2017):
  - Added switch /y to load security bit

- Version 1.1.10 (Mar-2017):
  - Done correction: “auto01.res” is saved for all errors if /g, /g1 or /g2 is specified (previous: it was saved for programming errors only)
  - Added memory spaced: 32 = BOOT memory and 64 = OTP

- Version 1.1.8 (Jan-2015):
  - Added switch /q to switch environment

- Version 1.1.7 (Nov-2013):
  - Added switch /o to enable/disable gap eliminator for PM or/and DM

- Version 1.1.6 (Jan-2012):
  - Added switch /u to transfer secure environment to programmer

- Version 1.1.5 (Nov-2008):
  - Switch /x can be used to speed-up the ICP operation

- Version 1.1.4 (Jan-2008):
  - Switch /g can be used with switch /n in order save individual results in file "ch_i.res"

- Version 1.1.3 (July-2007):
  - Added switches /? and /h to show command line and DLL versions in message box
  - Added switch /t for environment transfer
  - Expanded support for GANG operation (expanded switch /g: g2 will save global and individual results in “auto01.res” file, added switch /n to get individual results)
  - Expanded switch /d: in case of /d2 message box will appear for bad result only

- Version March-2007: initial version

11 Warranty

Softlog Systems (2006) Ltd. warrants this product against defects in materials and workmanship for a period of 1 (one) year. This warranty will not cover programmers that, in the opinion of Softlog Systems, have been damaged due to abuse, improper use, disassembly, replacement of parts or attempted repair by anyone other than an authorized Softlog Systems service technician.

This product must be returned to the supplier for warranty service within the stated period. The buyer shall pay all shipping costs and other charges or assessments for the product by the supplier.

Softlog Systems shall not be liable for any indirect, incidental, or consequential damages, regardless of whether liability is based upon breach of warranty, negligence, strict liability in tort, or any other theory, Softlog Systems will never be liable in an amount greater than the purchase price of the products described by this express warranty. No agent, distributor, salesperson, or wholesale or retail dealer has the authority to bind Softlog Systems to any other affirmation, representation, or warranty concerning these goods.

12 Contact

Softlog Systems (2006) Ltd.

6 Hayotzrim St. Or-Yehuda 6021820 Israel

Phone:	972-3-9515359
Fax:	972-3-9527520
Web:	www.softlog.com
E-mail:	sales@softlog.com, support@softlog.com

13 Copyright Notice

Windows is a registered trademark of Microsoft Corporation. Microchip, MPLAB, PIC and dsPIC are registered trademarks of Microchip Technology Incorporated.