



ICS Triplex ISaGRAF Inc.
www.isagraf.com

ISaGRAF complies with the requirements set forth in IEC 61131-3, for the following language features:

Common Elements					
Table #	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
1	1	X	X	X	Required character set
	2	X	X	X	Lower case characters
	3a	X	X	X	Number sign (#)
	4a	X	X	X	Dollar sign (\$)
	5a	X	X	X	Vertical bar ()
	6a	X	X	X	Subscript delimiters []
2	1	X	X	X	Upper case and numbers
	2	X	X	X	Upper case, numbers, embedded underlines
	3	X	X	X	Upper case, numbers, leading or embedded underlines
3	1	X	X	X	Comments
4	1	X	X	X	Integer Literals
	2	X	X	X	Real Literals
	3	X	X	X	Real Literals with exponents
	4	X	X	X	Base 2 Literals
	5	X	X	X	Base 8 Literals
	6	X	X	X	Base 16 Literals
	7	X	X	X	Boolean zero or one (<i>note</i> : in variable initial values)
	8	X	X	X	Boolean FALSE and TRUE
5	1	X	X	X	Character strings
6	2	X	X	X	Dollar sign
	3	X	X	X	Single quote
	4	X	X	X	Line Feed
	5	X	X	X	New line
	6	X	X	X	Form Feed
	7	X	X	X	Carriage return
	8	X	X	X	Tab
	7	1a	X	X	X
	1b	X	X	X	Duration literal without underlines: long prefix
	2a	X	X	X	Duration literal with underlines: short prefix
	2b	X	X	X	Duration literal with underlines: long prefix
8	1			X	Date literals (long prefix)
	2			X	Date literals (short prefix)

Common Elements (continued)					
Table #	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
10	1	X	X	X	BOOL, Boolean (<i>note</i> : 8 bits for this type)
	2		X	X	SINT, Short Integer
	3			X	INT, Integer
	4	X	X	X	DINT, Double Integer
	5			X	LINT, Long Integer
	6			X	USINT, Unsigned Short Integer
	7			X	UINT, Unsigned Integer
	8			X	UDINT, Unsigned Double Integer
	9			X	ULINT, Unsigned Long Integer
	10	X	X	X	REAL, Real numbers
	11			X	LREAL, Long Reals
	12	X	X	X	TIME, Duration
	13			X	DATE, Date (only)
	16	X	X	X	STRING, Variable-length character string (<i>note</i> : 255 character maximum)
10	17		X	X	BYTE, (<i>note</i> : mapped on USINT)
	18			X	WORD, (<i>note</i> : mapped on UINT)
	19		X	X	DWORD, (<i>note</i> : mapped on UDINT)
	20			X	LWORD, (<i>note</i> : mapped on ULINT)
12	4		X	X	Array data types (<i>Note</i> : in the dictionary)
	5		X	X	Structured data types (<i>Note</i> : in the dictionary)
13	Supported	X	X	X	(<i>Note</i> : For the implemented data types)
15	1	X	X	X	Input Location
	2	X	X	X	Output Location
	4		X	X	Single bit size
	5		X	X	Single bit size
	6		X	X	Byte (8 bits) size
	7			X	Word (16 bits) size
	8	X	X	X	Double word (32 bits) size
	9			X	Long (quad) word (64 bits) size
16	Supported	X	X	X	(<i>Note</i> : variable declaration is made into the dictionary)
		X	X	X	Also initialization and retain supported
17	1	X	X	X	Declaration of directly represented, non-retentive variable (<i>Note</i> : to be done in the I/O wiring tool)
	3	X	X	X	Declaration of locations of symbolic variables (<i>Note</i> : to be done in the dictionary and I/O wiring tool)
	4	X	X	X	Array location assignment (<i>Note</i> : to be done in the dictionary and the I/O wiring tool)
	5	X	X	X	Automatic memory allocation of symbolic variables (<i>Note</i> : in the dictionary)
	6		X	X	Array declaration (<i>Note</i> : in the dictionary)
	7		X	X	Retentive array declaration (<i>Note</i> : in the dictionary)
	8		X	X	Declaration of structured variables (<i>Note</i> : in the dictionary)
18	1	X	X	X	Initialization of directly represented, non retentive variables (<i>Note</i> : only for outputs)
	3	X	X	X	Location and initial value assignment to symbolic variables (<i>Note</i> : only for outputs)

Common Elements (continued)					
Table #	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
18	5	X	X	X	Initialization of symbolic variables
	6		X	X	Array initialization
	8		X	X	Initialization of structured variables
	9	X	X	X	Initialization of constants (<i>Note</i> : constants are Read-Only variables)
19	1	X	X	X	Negated input
20	1	X	X	X	Use of EN and ENO Required for LD
	3	X	X	X	FBD without EN and ENO
20a	2	X	X	X	In-out variable declaration (graphical)
21	1	X	X	X	Overloaded functions (<i>see</i> : documentation for details)
	2	X	X	X	Typed functions
22	1	X	X	X	ANY_TO_** conversion functions (** can be BOOL, DINT, REAL, TIME, STRING, plus SINT for version 4.2x)
	2	X	X	X	TRUNC
23	1	X	X	X	ABS (<i>Note</i> : input and output type is REAL)
	2	X	X	X	SQRT (<i>Note</i> : input and output type is REAL)
	4	X	X	X	LOG (<i>Note</i> : input and output type is REAL)
	6	X	X	X	SIN (<i>Note</i> : input and output type is REAL)
	7	X	X	X	COS (<i>Note</i> : input and output type is REAL)
	8	X	X	X	TAN (<i>Note</i> : input and output type is REAL)
	9	X	X	X	ASIN (<i>Note</i> : input and output type is REAL)
	10	X	X	X	ACOS (<i>Note</i> : input and output type is REAL)
	11	X	X	X	ATAN (<i>Note</i> : input and output type is REAL)
24	12	X	X	X	ADD
	13	X	X	X	MUL
	14	X	X	X	SUB
	15	X	X	X	DIV
	16	X	X	X	MOD (<i>Note</i> : input and output type is DINT)
	17	X	X	X	EXPT
	18	X	X	X	MOVE (<i>Note</i> : name is "1 gain")
25	1	X	X	X	SHL
	2	X	X	X	SHR (<i>Note</i> : Most significant bit is duplicated on left)
	3	X	X	X	ROR
	4	X	X	X	ROL
26	5	X	X	X	AND
	6	X	X	X	OR
	7	X	X	X	XOR
	8	X	X	X	NOT
27	1	X	X	X	SEL (<i>Note</i> : only for base types)
	2a	X	X	X	MAX (<i>Note</i> : only for base types)
	2b	X	X	X	MIN (<i>Note</i> : only for base types)
	3	X	X	X	LIMIT (<i>Note</i> : only for base types)
	4	X	X	X	MUX (<i>Note</i> : doesn't exist, instead are MUX4 and MUX8)
28	5	X	X	X	GT
	6	X	X	X	GE
	7	X	X	X	EQ

Common Elements (continued)					
Table #	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
28	8	X	X	X	LE
	9	X	X	X	LT
	10	X	X	X	NE
29	1	X	X	X	LEN (<i>Note: name is MLEN</i>)
	2	X	X	X	LEFT
	3	X	X	X	RIGHT
	4	X	X	X	MID
	5	X	X	X	CONCAT (<i>Note: doesn't exist, instead use "+"</i>)
29	6	X	X	X	INSERT
	7	X	X	X	DELETE
29	8	X	X	X	REPLACE
	9	X	X	X	FIND
30	1	X	X	X	ADD for TIME
	4	X	X	X	SUB for TIME
32	Supported	X	X	X	Function Block I/O parameter usage
33	4	X	X	X	Input/Output declaration (<i>Note: to be done in the dictionary</i>)
	9a	X	X	X	R_TRIG
	9b	X	X	X	F_TRIG
34	1	X	X	X	SR
	2	X	X	X	RS
35	1	X	X	X	R_TRIG
	2	X	X	X	F_TRIG
36	1	X	X	X	CTU
	2	X	X	X	CTD
	3	X	X	X	CTUD
37	1	X	X	X	TP
	2a	X	X	X	TON
	3a	X	X	X	TOF
38	Supported	X	X	X	
39	11	X	X	X	Declaration of directly represented, non-retentive variable (<i>Note: to be done in the I/O wiring tool</i>)
	13	X	X	X	Declaration of locations of symbolic variables (<i>Note: to be done in the dictionary and I/O wiring tool</i>)
	14		X	X	Array location assignment (<i>Note: to be done in the dictionary and the I/O wiring tool</i>)
	15	X	X	X	Initialization of directly represented, non retentive variables (<i>Note: only for outputs</i>)
	17	X	X	X	Location and initial value assignment to symbolic variables (<i>Note: only for outputs</i>)
	19	X	X	X	Use of directly represented variables

SFC Language Elements					
Table #	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
40	1	X	X	X	Step and initial step – graphic
	2	X	X	X	Step and initial step – text (<i>Note:</i> supported in files)
	3a	X	X	X	Step flag
	3b	X	X	X	Step flag
	4	X	X	X	Step elapsed time
41	1	X	X	X	Transition in ST language
	2	X	X	X	Transition in LD language
	5	X	X	X	Textual equivalent of feature 1 (<i>Note:</i> supported in file)
	7	X	X	X	Use of transition name
	7a	X	X	X	Transition condition using LD (<i>Note:</i> in this case name is implicit)
	7d	X	X	X	Transition condition using ST
42	1	X	X	X	Boolean var
	2l	X	X	X	LD in action block
	2s	X	X	X	Inclusion of SFC elements in action
	3s	X	X	X	Textual declaration in ST language
	3i	X	X	X	Textual declaration in IL language
43	1	X	X	X	Action block
	2	X	X	X	Concatenated action blocks
	3	X	X	X	Textual step body (<i>Note:</i> supported in file)
	4	X	X	X	Action block “d” field
44	1	X	X	X	Qualifier (<i>Note:</i> restrictions apply, see Table # 45)
	2	X	X	X	Action name
	3	X	X	X	Boolean indicator (<i>Note:</i> is interpreted with the feature # 2)
	4	X	X	X	IL language
	5	X	X	X	ST language
	6	X	X	X	LD language
45	2	X	X	X	Non-Stored
	3	X	X	X	Overriding Reset
	4	X	X	X	Set
	11	X	X	X	Pulse (rising edge)
	12	X	X	X	Pulse (falling edge)
46	1	X	X	X	Single sequence
	2c	X	X	X	Divergence of sequence selection
	3	X	X	X	Convergence of sequence selection
	4	X	X	X	Simultaneous sequences
	5a, 5b, 5c	X	X	X	Sequence Skip
	6a, 6b, 6c	X	X	X	Sequence loop
	7	X	X	X	Directional arrows (<i>Note:</i> only for jumps)
47	40	X	X	X	Supported
	41	X	X	X	Textual

SFC Language Elements (continued)					
Table #	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
47	42	X	X	X	Textual
	43	X	X	X	Supported
48	40	X	X	X	Supported
	41	X	X	X	Supported
	42	X	X	X	Supported
	43	X	X	X	Supported
	45	X	X	X	Supported
	46	X	X	X	Supported
	57	X	X	X	Supported

CONFIGURATIONS, RESOURCES, PROGRAMS and TASK Elements					
Table #	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
49	1		X	X	CONFIGURATION construction (<i>Note: graphical declaration</i>)
	2		X	X	Binding mechanism
	3		X	X	RESOURCE construction (<i>Note: graphical declaration</i>)
	4	X	X	X	VAR_GLOBAL construction (<i>Note: in dictionary</i>)
	5a	X	X	X	Periodic TASK (<i>Note: only one TASK supported, implicit declaration</i>)
	5b		X	X	Cycle on event
	6a	X	X	X	PROGRAM declaration with TASK association (<i>Note: every PROGRAMs are associated to the only TASK</i>)
	7	X	X	X	Declaration of directly represented variables in VAR_GLOBAL (<i>Note: to be done in the I/O Wiring tool</i>)
	10a		X	X	VAR_ACCESS construction (<i>Note: to be done in the Binding Tool</i>)
	10d		X	X	Access path to GLOBAL variables in RESOURCES (<i>Note: to be done in the Binding Tool</i>)
	10e		X	X	Binding mechanism
50	1a	X	X	X	Textual declaration of periodic TASK (<i>Note: as there is only one TASK inside a RESOURCE, only INTERVAL property is supported</i>)
	1b		X	X	Cycle on event
	4a	X	X	X	Graphical association with PROGRAMS (within RESOURCES)

IL Language Elements

Table #	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
51a	Supported	X	X	X	Instruction Fields
51b	2	X	X	X	Parenthesized expression (short form)
52	1	X	X	X	LD N
	2	X	X	X	ST N
	3	X	X	X	S, R
	4	X	X	X	AND N,(
	5	X	X	X	& N,(
	6	X	X	X	OR N,(
	7	X	X	X	XOR N,(
	8	X	X	X	ADD (
	9	X	X	X	SUB (
	10	X	X	X	MUL (
	11	X	X	X	DIV (
	12	X	X	X	GT (
	13	X	X	X	GE (
	14	X	X	X	EQ (
	15	X	X	X	NE (
52	16	X	X	X	LE (
	17	X	X	X	LT (
	18	X	X	X	JMP C,N
	19	X	X	X	CAL C,N
	20	X	X	X	RET C,N
	21	X	X	X)

ST Language Elements

Table #	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
55	1	X	X	X	Parenthesization
	2	X	X	X	Function evaluation
	4	X	X	X	Negation
	5	X	X	X	Complement
	6	X	X	X	Multiply
	7	X	X	X	Divide
	8	X	X	X	Modulo
	9	X	X	X	Add
	10	X	X	X	Subtract
	11	X	X	X	Comparison
	12	X	X	X	Equality
	13	X	X	X	Inequality
	14	X	X	X	Bool AND

ST Language Elements (continued)					
Table #	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
	15	X	X	X	Bool &
	16	X	X	X	Bool XOR
	17	X	X	X	Bool OR
56	1	X	X	X	Assignment
	2	X	X	X	FB invocation and FB output usage
	3	X	X	X	RETURN
	4	X	X	X	IF
	5	X	X	X	CASE
	6	X	X	X	FOR
	7	X	X	X	WHILE
	8	X	X	X	REPEAT
	9	X	X	X	EXIT
	10	X	X	X	;

Common Graphical Elements					
Table #	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
57	2	X	X	X	Horizontal lines
	4	X	X	X	Vertical lines
	6	X	X	X	Horizontal/Vertical connection
	8	X	X	X	Line crossing without connection
	10	X	X	X	Corners
	12	X	X	X	Blocks
58	1	X	X	X	Unconditional JUMP FBD
	2	X	X	X	Unconditional JUMP LD
	3	X	X	X	Conditional JUMP FBD
	4	X	X	X	Conditional JUMP LD
	5	X	X	X	Conditional RETURN LD
	6	X	X	X	Conditional RETURN FBD
	7	X	X	X	Unconditional RETURN from FUNCTION or FUNCTION BLOCK (<i>Note</i> : implicit end of a function or function block)
	8	X	X	X	Unconditional RETURN in LD language

LD Language Elements

Table #	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
59	1	X	X	X	Left power rail
	2	X	X	X	Right power rail
60	1	X	X	X	Horizontal link
	2	X	X	X	Vertical link
61	1	X	X	X	Normally open contact
	3	X	X	X	Normally closed contact
	5	X	X	X	Positive transition-sensing contact
	7	X	X	X	Negative transition-sensing contact
62	1	X	X	X	Coil
	2	X	X	X	Negated coil
	3	X	X	X	SET (latch) coil
	4	X	X	X	RESET (latch) coil
	8	X	X	X	Positive transition-sensing coil
	9	X	X	X	Negative transition-sensing coil

FBD Language Elements

Table #	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
There are no compliance tables listed in this part of the specification.					

The ISaGRAF system complies with the requirements of IEC 61131-3, for the following software model features as listed:

Software Model					
Section	Feature #	version 3.x	version 4.x	version 5.x	Feature Description
1.4.1					Software Model:
		X	X	X	Multiple FBs per program
		X	X	X	Multiple Programs per Resource
		X	X	X	Multiple Tasks per Resource
			X	X	Multiple Resources per Configuration
			X	X	Multiple Configurations per project
		X	X	X	Global Variables
		X	X	X	Access Paths
		X	X	X	Instance specific initializations
1.4.2					Communication Model:
	Figure 2a	X	X	X	Data Flow connections within Programs
	Figure 2b	X	X	X	Communication via Global Variables
	Figure 2c	X	X	X	Communication via Function Blocks
	Figure 2d		X	X	Communication via Access paths