ISO-Pascal

ISO-Pascal

for the BBC Microcomputer Model B

Contents

ISO-Pascal on two language ROMs

A disc (suitable for 40 or 80 track disc drives) containing an extended compiler for systems with a 6502 Second Processor and various extensions and demonstration programs, details of which are given overleaf

Instructions for inserting the ISO-Pascal language ROMs in the BBC Microcomputer (leaflet enclosed)

Pascal from BASIC - a tutorial course in Pascal

ISO-Pascal on the BBC Microcomputer and Acorn Electron - the reference manual for Acornsoft ISO-Pascal

A reference card

A function key card for use with the editor

Loading instructions

Instructions for entering ISO-Pascal are given at the beginning of *ISO-Pascal on the BBC Microcomputer and Acorn Electron.*

Command summary

The immediate mode commands available in Acornsoft ISO-Pascal are as follows:

CLOSE
Close all open files on the selected filing system

COMPILE
Compile to and from memory.

COMPILE source file
Compile from source file memory.

COMPILE > object-file
Compile from memory to object file.

COMPILE source-file object-file
Compile from source file to object file.

EDIT [source-file]
Call the editor, optionally loading a source file.

GO [arguments]
RUN the object file in memory, passing the optional arguments if the T option was used at compilation.

LOAD object-file
Load the specified object file.

MODE number
Change the display mode to the one specified

RUN object-file [arguments]
Load and run the specified code file.

SAVE object-file
Save the memory code file under the name given

TRACE [0, 1 or 2]
Set the current TRACE level

Editor pattern matching

Patterns used by the search and replace commands in the ISO-Pascal text editor consist of combinations of literal text with special characters. Literal text is case independent except when used with the special characters (to indicate ranges etc).

The special search characters are as follows:

.	matches any character
@	matches any alphanumeric (0-9, A-Z, a-z and _)
#	matches any digit (0-9)
[xyz]	matches any of x, y and z
a-z	matches any character between a and z (inclusive)
$	matches the carriage return character
¦c	matches CTRL. c
¦?	matches the DELETE character (ASCII127)
¦!c	matches character code c+128
~c	matches anything but c (c may be wildcard)
\c	matches c (where c would otherwise have a special meaning)
*c	matches zero or more of c (shortest match)

Examples:

$* $	matches all blank lines
# * # ~ #	matches all integer constants

The special characters available for replacements are:

$	carriage return
¦c	CTRL c
¦?	DELETE
¦!c	character code c+128
\c	c (where c would otherwise have a special meaning)
&	whatever was matched by the pattern
%n	field number n (0-9). where a field is a wildcard character, a multiple match (* c), an inverted match (~c), a range (a-z) or a choice ([13579]). Fields are numbered from the leftmost (which is 0).

Examples:

#/&&	duplicates all digits (eg 12 becomes 1122)
.. /%1%0	reverses alternate characters (eg r2d3 becomes 2r3d)

Editor command summary

The cursor movement and function key commands available in the editor are as follows:

BBC editor	Function	Electron editor
Up arrow	Move up to a line	Up arrow
Down arrow	Move down a line	Down arrow
Left arrow	Move left a character	Left arrow
Right arrow	Move right a character	Right arrow
SHIFT up	Move up a page	FUNC N
SHIFT down	Move down a page	FUNC M
SHIFT left	Move to start of line	FUNC <
SHIFT right	Move to end of line	FUNC >
CTRL up	Move to top of text	FUNC Z
CTRL down	Move to end of text	FUNC X
DELETE	Delete left of the cursor	DELETE
COPY	Delete at the cursor	COPY
SHIFT COPY	Initiate cursor-edit mode	FUNC :
TAB	Move cursor to non-space	FUNC A
f0	Find a line number	FUNC Q
f1	Issue MOS command	FUNC W
f2	Load the text in a file	FUNC E
f3	Save the text to a file	FUNC R
f4	Find and replace a string	FUNC T
f5	Global count/replace string	FUNC Y
f6	Set marker	FUNC U
f7	Copy a block of text	FUNC I
f8	Send text to printer	FUNC O
f9	Restore old text	FUNC P
SHIFT f0	Toggle <CR> display	FUNC 1
SHIFT f1	Toggle insert/overtype	FUNC 2
SHIFT f2	Insert text from a file	FUNC 3
SHIFT f3	*** NOT USED ***	FUNC 4
SHIFT f4	Quit from the editor	FUNC 5
SHIFT f5	*** NOT USED ***	FUNC 6
SHIFT f6	Clear marker(s)	FUNC 7
SHIFT f7	Move a block of text	FUNC 8
SHIFT f8	Delete a block of text	FUNC 9
SHIFT f9	Delete the text	FUNC 0

Error numbers/messages produced by the compiler

The table below lists all of the error numbers that the compiler produces, and the messages that are associated with them These messages are printed automatically when {$F+} compiler option is specified when using discs Additional information is printed by specifying the {$<CTRL@> +} option in the first lime of the source file.

1	Variable identifier expected
2	Comma expected / missing parameter.
3	. expected
4	: expected
5	; expected
6	Type mismatch
7	(expected
8	) expected
9	(expected
10	] expected
11	Can t assign a real to an integer.
12	RHS not compatible with LHS type mismatch
13	Bad statement start
14	Not LSO-Pascal (use compiler option X+ to allow extensions).
15	Equals expected
16	If INPUT or OUTPUT is used then it must be declared in program header.
17	Missing parameter(s).
18	Parameter can t be a packed var.
19	Missing semicolon
20	For loop control variable must be declared in the variable declaration part of this procedure / function
21	Assignment operator := expected
22	.. expected
23	Actual and formal parameters should both be either packed or unpacked
24	A label was declared in this block but was not defined
25	Hex number too large.
26	Variable too big for memory.
27	Too much code for code buffer, claim larger area using compiler option C .
28	Set base type must be max0 .. 255.
29	BEGIN expected
30	Too many procedures (max 127).
31	Missing body of FORWARD pro/func
32	DO expected
33	Label not declared
34	This label does not prefix a statement which is in the same statement sequence that contains the GOTO statement
35	END expected / missing semicolon
36	This label should prefix a statement at the outermost level of statement nesting in a block
37	Label not declared in this block
38	Label already defined

39	Label already declared
40	Label must be a sequence of digits 0 to 9999.
41	Array element selector is not the same type as the array s index type.
42	Unpacked array variable expected
43	Component types of both arrays must be the same.
44	OF expected
45	Packed array variable expected
46	Can t pass a conformant array as a value parameter.
47	PROGRAM expected
48	Can t pass a bound identifier as a var parameter.
49	Function result type mismatch
50	Formal parameter is a procedure and actual parameter is a function or vice versa
51	THEN expected
52	TO expected
53	Procedural/functional parameter expected
54	UNTIL expected
55	Can t altar the value of this variable because it is the control variable of an active FOR loop.
56	Control variable must be an entire variable ie not an array element or field of a record
57	Too many digits
58	Premature end of file.
59	Can only output integers in hex
60	Too many parameters
61	String parameter expected
62	Undeclared identifier expected
63	For loop initial & final values must be same type as control variable.
64	For loop control variable must be ordinal type.
65	Record s field identifier expected
66	Can only assign value to current function identifier.
67	Current function identifier is only allowed on LHS of assignment
68	Ordinal parameter expected
69	Parameter must be a file variable.
70	Parameter must be a textfile.
71	Constant already specifies a variant part in this record
72	Constant does not specify a variant
73	Variant constant/ tag-type mismatch
74	Too many variant constants.
75	Pointer s base type must be record in order to have variant constants
76	Formal parameters have the same conformant array type but the actual parameters are not of the same type.
77	Can only have variant constants if type pointed to is a record
78	Set base type and IN operand are not the same type.
79	Real parameter expected
80	Real / integer parameter expected

Error numbers/messages (continued)

81	Integer parameter expected
82	Text file variable expected
83	Filename string expected
84	Temp files do not have filenames
85	Can t have a file as a parameter to READ/WRITE.
86	File and parameter type mismatch
87	Can t read / write this type.
88	Only reals can have a decimal place.
89	File must be of type TEXT to do WRITELN/READLN.
90	Type mismatch between actual and formal parameter.
91	Procedure/function has no arguments.
92	File variable expected
93	Bad filename.
94	Control variable threatened by nested procedure / function
95	Procedural parameter list mismatch
96	Function id is unassigned
97	Structured types containing a file component cannot be assigned to each other.
98	File type must be TEXT to allow use of field widths
99	Can t assign value to function parameter identifier.
100	Set of all tag-constants does not equal the set of all values specified by the tag- type.
101	Can t pass tag-field as var param
102	A variable appeared in the program header but was not defined
103	Too many stmt sequences (max 255).
104	Can t redefine identifier because it has been used earlier in this block
105	No hex reals allowed
106	Can only pack conformant arrays.
107	Case value must be ordinal type.
108	Index limits out of range.
109	Standard file already declared
110	File variable expected
111	Constant expected
112	Can t sign non-numeric expressions.
113	Type mismatch between case constant and case expression
114	Bad pointer type.
115	Type identifier expected
116	Duplicate case constant
117	Subrange limits must be scalar.
118	Upper and lower limits must be same type.
119	Low bound exceeds high bound
120	Ordinal type expected
121	Too many dimensions for interpreter.
122	Set member must have ordinal type.
123	Can t have file of file(s).
124	Set member must have an ordinal value of 0 to 255.
125	Unresolved pointer type.
126	Function type expected
127	Digit expected.

128	Function type must be ordinal, real or pointer.
129	Illegal character detected
130	Unexpected EOF in a comment or a string constant
131	File already declared permanent
132	Unresolved pointer base type.
133	Pointer base type identifier is not a type identifier.
134	Structured type expected
135	Tag type expected
136	Ordinal constant expected
137	Field does not belong to this record
138	Procedure or function id expected
139	Sets are not of the same base type.
140	Procedure/function already declared
141	Variant selector type does not match variant constant type.
142	Pointer type expected
143	Permanent files must be declared in global variable section.
144	Packed conformant arrays must be single dimension
145	Can t change this compiler option once it is set
146	Component type mismatch
147	Set members must have the same type.
148	Variable is not a file or pntr type.
149	Missing index / spurious comma
150	Variable is not a record
151	Variable is not an array.
152	Numbers must be terminated by a non alphabetic character.
153	Permanent file not declared in global variable section
154	Decimal places field-width must be an integer expression
155	Field-width must be integer value.
156	Can t assign a value to a conformant array bound identifier.
157	Can t have EOLN in string constants
158	Can t have a file variable contained in a value parameter.
159	Illegal operation on these operands.
160	Index type mismatch
161	Boolean type expected
162	Can t use function id in this way.
163	Integer operands needed for this operation.
164	Procedure identifier has been used before its defining occurrence.

{ These are fatal errors and cause termination of the compilation }

165	Id table overflow (increase table size using compiler option I).
166	Too many nested records / procedures
167	To compile using disc Pascal, use DCOMP <source> <object>.
168	Code and source filenames the same.