

source outline

ctl-opt

control specifications

default *LIBL/RPGLEHSPEC
QRPGLE/DFTLEHSPEC

dcl-pr

prototypes

dcl-pi

procedure interface

dcl-f

global definitions

dcl-s

(files, variables, data structures,

dcl-ds

arrays, constants, enumera-

dcl-c

tions)

dcl-enum

exec sql

SQL precompiler options

SQL cursor declarations

main source

begsr

global subroutines

dcl-proc

procedure specification

dcl-pi

procedure interface

dcl-f

local definitions

dcl-s

(files, variables, data structures,

dcl-ds

arrays, constants, enumera-

dcl-c

tions)

dcl-enum

procedure body

begsr

local subroutines

main section

subprocedure section

free-form statements



syntax rules

comment: // continuation character: +
 end instruction: ; variable naming: don't start with number
 built-in functions: %function-name(parm1:parm2:...) i

data types keywords

	Tipo	Keyword	Length
alphanumeric	character fixed length	char	1-16773104
	character varying length	varchar	1-16773100
	character UCS fixed length	ucs2	1-8386552
	character UCS varying length	varucs2	1-8386550
	graphic fixed length	graph	1-8386552
	graphic varying length	vargraph	1-8386550
numeric	binary	bindec	1-9
	float	float	4, 8
	integer	int	3, 5, 10, 20
	unsigned	uns	3, 5, 10, 20
	zoned decimal	zoned	63,63
date/time	packed decimal	packed	63,63
	date	date	
	time	time	
	timestamp	timestamp	26-32
other	boolean	ind	1
	pointer	pointer	16 bytes
	pointer to procedure	pointer(*proc)	16 bytes
	object	object	

indicators

0 - *off / 1 - *on
 *inxx (xx 1 to 99)
 *inlr last record
 *inkx key function
 op.code: eval

F1	F2	F3	F4
KA	KB	KC	KD
F5	F6	F7	F8
KE	KF	KG	KH
F9	F10	F11	F12
KI	KJ	KK	KL
F13	F14	F15	F16
KM	KN	KP	KQ
F17	F18	F19	F20
KR	KS	KT	KU
F21	F22	F23	F24
KV	KW	KX	KY

logical operators

= <> > >= < <= i
 AND OR NOT IN i

operation codes

code extender: (a), (d),
 (e), (h), (m), (n),
 (p), (r), (t), (z)

assignment operators

= += -= *= /= **=
 op.code: clear, eval,
 evalr, reset

character variables

concatenation: +
 op.code: clear, eval,
 evalr, reset
 BIF: %char,
 %charcount, %check,
 %checker, %concat,
 %left, %lower, %max,
 %min, %replace,
 %right, %scan, %scanr,
 %scanrpl, %split,
 %str, %subst, %trim,
 %triml, %trimr,
 %ucs2, %upper, %xlate

numeric variables

operators: + - * / **
 op.code: clear, eval,
 reset
 BIF: %abs, %dec, %dech,
 %div, %editc, %editw,
 %float, %hival, %int,
 %inth, %loval, %max,
 %min, %rem, %sqrt,
 %uns, %unsh

arrays

definition: ... dim(max-entries |
 *var:maxentries | *auto:maxentries |
 *ctdata)

time → run-time
 → compile-time (CTDATA)
 → prerun-time

syntax: array-name(index | *| *next)

operator: in

operations code: for-each, sorta

BIF: %concatarr, %elem, %fields, %hival,
 %list, %lookupxx, %loval, %maxarr, %minarr,
 %split, %subarr, %tlookupxx, %xfoot

data structures

keyword definition: alias, based, const, ext,
 extfld, extname, inz, likeds, likerec, over-
 lay, pos, prefix, qualified, samepos, template
 op.code: clear, eval, eval-corr, reset
 reference to subfield qualified ds:

dsName.subfieldName

special data structures

file information (INFDS)

program status (PSDS)

indicator (INDDS)

caption

i reference

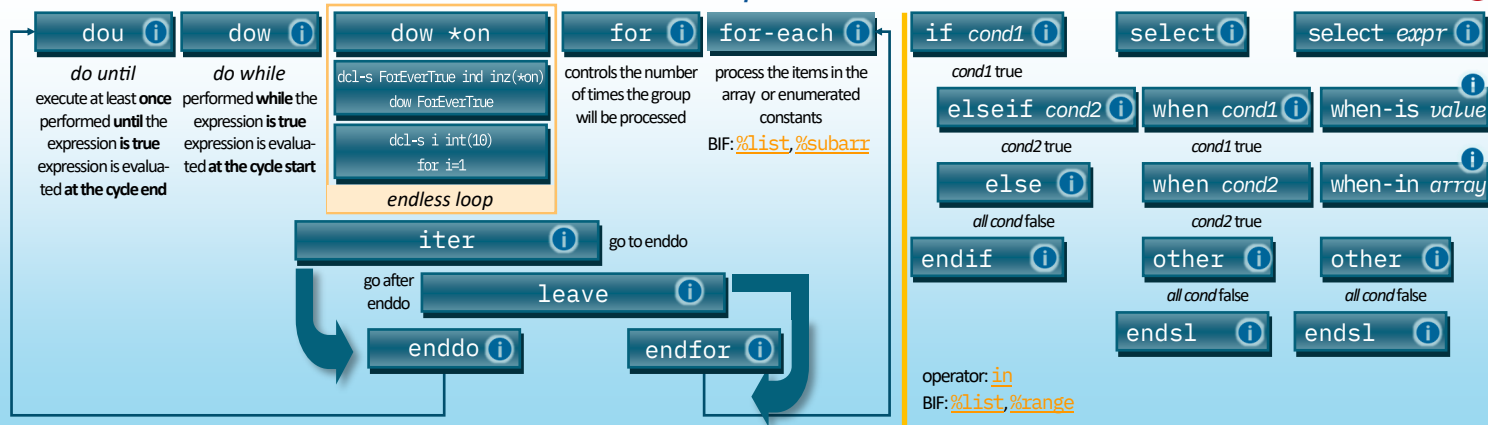
i examples

i articles

SQL data types (SQLTYPE())

	Tipo	Keyword	Length
binary	binary fixed length	binary	1-32766
	binary varying length	varbinary	1-32740
LOB	character Large Object	clob	1-16773100
	double byte large object	dbclob	1-8386550
	binary large object	blob	1-16773100
	LOB file reference	clob_file dbclob_file blob_file	
	LOB locator	clob_locator dbclob_locator blob_locator	
	LOB XML	xml_clob xml_dbclob xml_blob	1-16773100 1-8386550 1-16773100
other	XML file reference	xml_clob_file xml_dbclob_file xml_blob_file	
	XML locator	xml_locator	
	ROW ID	rowid	
	resultset locator	result_set_locator	

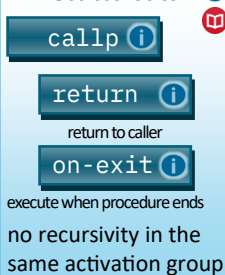
structured operations



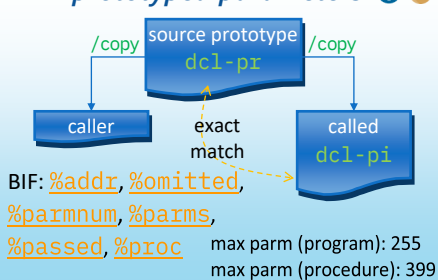
subroutines



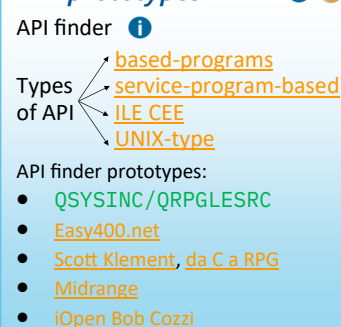
static call



prototyped parameters



API prototypes



data areas

keyword definition: [dtaara](#)
op.code: [in](#), [out](#), [unlock](#)
BIF: [%addr](#), [%alloc](#)
[*LDA](#)

variable info

BIF: [%decpos](#), [%len](#), [%nullind](#), [%size](#)

XML, JSON

op.code: [data-gen](#), [data-into](#), [xml-into](#), [xml-sax](#)
BIF: [%data](#), [%gen](#), [%handler](#), [%parser](#), [%xml](#)

date and time

keyword definition: [datfmt](#)
operator: [+](#), [-](#)
op.code: [eval](#), [test](#)
BIF: [%char](#), [%date](#), [%days](#), [%dec](#), [%diff](#), [%hours](#), [%minutes](#), [%months](#), [%mseconds](#), [%seconds](#), [%subdt](#), [%time](#), [%timestamp](#), [%years](#)
[constant prefix](#): [d](#), [t](#), [z](#)
[user date special word](#): [UDATE](#), [UMONTH](#), [UYEAR](#), [UDAY](#), [C*DATE](#), [*DATE](#), [*MONTH](#), [*YEAR](#), [*DAY](#)
[duration code](#): [*years/*y](#), [*months/*m](#), [*days/*d](#), [*hours/*h](#), [*minutes/*mn](#), [*seconds/*s](#), [*mseconds/*ms](#)

formats

embedded SQL

syntax: **exec sql . . . ;**
host variables: [:VarName](#)
used in clause: where, select, into, order by, set, values (insert), call
indicator variables: [int\(5\)](#)
support extended indicators: compilation [*extind](#)
SQL communication area: [SQLCA](#)
SQL descriptor area: [SQLDA](#)

Error handlers

Priority:
1. error extender ([e](#))
2. monitor group
3. subroutine [*pssr](#) (file)
keyword f-spec: [infsr](#)
4. [ILE condition handlers](#)
5. subroutine [*pssr](#) (programma)
op.code: [monitor](#)
BIF: [%error](#), [%status](#)

files

keyword: [alias](#), [commit](#), [extdesc](#), [extfile](#), [extmbr](#), [infds](#), [prefix](#), [qualified](#), [rename](#), [template](#), [usage](#), [usropt](#)
keyword for display/printer file: [indds](#), [oflind](#), [printer](#), [sfile](#), [workstn](#)
operations code
[keyed input](#): [chain](#), [reade](#), [readpe](#), [setll](#), [setgt](#)
[sequential input](#): [read](#), [readp](#)
[data manipulation](#): [delete](#), [update](#), [write](#), [unlock](#)
[commitment control](#): [commit](#), [rolbk](#)
[file](#): [close](#), [open](#)
[display file only](#): [exfmt](#), [readc](#)
BIF: [%eof](#), [%equal](#), [%error](#), [%fields](#), [%found](#), [%kds](#), [%open](#), [%status](#)

figurative constants

[*BLANK](#), [*BLANKS](#) = ''
[*ZERO](#), [*ZEROS](#) = 0
[*ON](#) = '1' / [*OFF](#) = '0'
[*NULL](#) = valore nullo
[*LOVAL](#) = valore minimo / [*HIVAL](#) = valore massimo
[*ALL 'x'](#) = ripete 'x' per la lunghezza massima della variabile

message

op.code: [snd-msg](#)
BIF: [%msg](#), [%target](#)

pointers

op.code:
[dealloc](#)
BIF: [%addr](#), [%alloc](#), [%paddr](#), [%realloc](#)

binder language

[STRPGMEXP](#) [PGMLVL](#) ([*CURRENT](#)) [SIGNATURE](#) ('V1R1')
[EXPORT](#) [SYMBOL](#) (p1)
[EXPORT](#) [SYMBOL](#) (p2)
...
[ENDPGMEXP](#)
[STRPGMEXP](#) [PGMLVL](#) ([*PRV](#)) [SIGNATURE](#) ('V1R0')
[EXPORT](#) [SYMBOL](#) (p1)
[ENDPGMEXP](#)

Compiler directives

start at column >= 7 or >=1 (for all free). End semicolon not required
[/COPY](#) or [/INCLUDE](#) [[library/file/member](#) | "[ifs_path](#)"]:
copy the external source member
[defining condition](#):
[/DEFINE](#) or [/UNDEFINE](#):
[/IF](#) [[NOT](#)] [DEFINED](#) (nome-direttiva)
...;
[/ELSEIF](#)
...;
[/ELSE](#)
...;
[/ENDIF](#)
[/EOF](#): ignore the rest of source