

1. מבוא

1.1. כללי

מטרת המסמך היא לזהות ולהגדיר תקן לכתיבת תוכניות בשפת COBOL

1.2. מדדי איכות

הערות	סוג
חוקים לכתיבת תוכנית בצורה כזו שהתחזוקה שלה תהיה קלה וקצרה.	Maintainability
חוקים הקשורים לצורה הויזואלית של התוכנית, כולל שימוש ברווחים, שיטות הקינון, וכו'...	Formatting
חוקים הקשורים לתעוד התוכנית. כולל מה יתועד, באיזו צורה, וכו'...	Documentation
חוקים הקשורים לכתיבת תוכניות יעילות יותר כולל שימוש נכון ב- I/O, CPU, Memory	Performance
חוקים כלליים כגון באלו פקודות לא להשתמש, שיש להשתמש בטבלאות החלטה במידת האפשר, וכו'...	General
שיטות לקביעת שמות משתנים, תוכניות, Sections, וכו'...	Naming Convention
חוקים לכתיבת תוכניות מבניות	Structured
המלצות שונות	Recommendations

1.3. הערות

- בכל מקרה בו הקוד הנכתב **אינו** עפ"י התקן, יש להוסיף – לפני קטע הקוד – הערת הסבר ממנה ניתן להבין מדוע הקוד נכתב כפי שנכתב ואת מספר החוק שהופר.

2. Maintainability (M)

ID	Rule Description	Sev.	ACRT
1.	The program has to be written as simple and readable as possible.	High	
2.	Avoid duplicate code.	High	+
3.	Do not write programs having more than 1500 lines in the PROCEDURE DIVISION.	Med	+

ID	Rule Description	Sev.	ACRT
4.	Do not use more than one nested cobol function.	Med	
5.	Do not use hard-coded constants in PROCEDURE DIVISION. Define constant variables in CXX (Except 0 and 1 and in ZT-TAKALA)	Med	+
6.	Do not use more than three levels of nesting statements.	Med	+
7.	The number of parameters must be equal to the number of arguments (the OMITTED option is allowed).	Med	
8.	Write short sections (up to 100 lines of code not including comments).	Med	+
9.	Each SECTION should be limited to performing a single, well-defined task.	Med	
10.	Do not use more than five arguments in the Linkage SECTION (01 Level).	Med	+
11.	Do not use initial values in the WORKING STORAGE, except for constants or titles.	Low	
12.	Always indicate the order of operation explicitly by using parentheses.	Low	
13.	Divide complex expressions into a number of expressions. (including complex IFs)	Low	
14.	Use ON EXCEPTION for dynamic subroutines (dynamic CALL)	Low	
15.	Variables must be alphabetically ordered by prefix	Low	+
16.	Place the variable definition in the corresponding group according to its meaning(constants in CXX, counters in MXX...)	Low	
17.	Arrange the SECTIONs in PROCEDURE DIVISION by prefixes in ascending order.	Low	+
18.	Do not use the ALSO clause in the EVALUATE statement	Med	-
19.	Use PERFORM SECTION only and without THRU and not PERFORM paragraph.	Med	+
20.	Do not use recursive PERFORM and CALL	Med	
21.	A section that performs a general task will have a "PXX" prefix	Low	
22.	The following sections are mandatory: A-TOCHNIT-RASHIT, ZA-ATCHALAT-TOCHNIT, ZT-TAKALA, ZZ-SIUM-TOCHNIT	Low	+
23.	Add an END clause to the following statements: EVALUATE , IF, READ , inline PERFORM , CALL	Low	+
24.	The GOBACK, EXIT PROGRAM will be used only in the A-TOCHNIT-RASHIT SECTION OR ZT-TAKALA SECTION	Low	+
25.	Do not use the following statements: ALTER, RENAME , GO TO DEPENDING ON, NEXT SENTENCE, MOVE CORRESPONDING, MOVE CORR, ENTRY, ACCEPT DATE	Med	+
26.	Groups must be sorted by group names, and variables must be sorted within the group	Low	+
27.	Never change the index inside a loop (PERFORM VARYING)	Med	
28.	Do not use STOP RUN	Low	+
29.	Use LOW-VALUE instead of X'00' and HIGH-VALUE instead of X'FF'	Low	+
30.	Use COPY for records in FD and SD	Med	+
31.	Use only EVALUATE TRUE.	Med	+

3. Formatting (F)

ID	Rule Description	Sev.	ACRT
1.	<u>Use separating comment lines before and after SECTIONS and DIVISIONs.</u>	Low	+
2.	<u>Align the reserved words: PIC, VALUE, OCCURS, INDEXED BY, DEPENDING ON.</u>	Low	+
3.	<u>Separate 01 levels by using empty lines.</u>	Low	+
4.	<u>Each section must include the paragraph: X-EXIT or X-EX except in Declaratives (X is the section prefix)</u>	Low	+
5.	<u>The exit paragraph name must have the structure: X-EX. EXIT Or X-EXIT. EXIT. (X is the section prefix)</u>	Low	
6.	<u>The exit paragraph must include only the EXIT statement:</u>	Low	
7.	<u>Do not overuse blank lines.</u>	Low	
8.	Each source line must contain a single command only	Low	
9.	<u>Do not write spaces after the opening parenthesis, "(", and before the closing parenthesis, ")".</u>	Low	+
10.	<u>Add a space around assignment and binary/conditional operators.</u>	Low	
11.	<u>Add a space after commas</u>	Low	
12.	<u>There must be a clear indentation (4 spaces) between levels.</u>	Low	
13.	<u>Place the else statement in its own line, except for else if.</u>	Low	+
14.	<u>If a condition is written on more than one line, place the "AND/OR" to the right of the conditions.</u>	Low	
15.	<u>If the "THEN" clause is empty, add a CONTINUE statement</u>	Low	
16.	Removed	Low	
17.	<u>Do not use continuation line ('- ' in column 7) in the Working Storage Section</u>	Low	+
18.	<u>Use THEN for every IF</u>	Low	+
19.	<u>All four DIVISIONs must be declared in every program</u>	Low	+
20.	<u>Use only the paragraph X-EXIT except in Declaratives</u>	Med	+
21.	Align IF, ELSE and END-IF	Med	+

4. Documentation (D)

	Rule Description	Sev.	ACRT
1.	Total comment lines, must not exceed 60% of the code.	Med	+
2.	<u>Use the standard format for Program Header documentation: Purpose, Description, Calling statement, Arguments, Input, Output, Programmer, Date</u>	Low	
3.	<u>Add before each group of variables an according documentation</u>	Low	
4.	<u>Each group of source lines which performs an isolated action must be explained by a comment.</u>	Low	
5.	<u>A section documentation must be written before each section in PROCEDURE DIVISION</u>	Low	
6.	<u>The documentation of a section must include at least the purpose and the</u>	Low	

	Rule Description	Sev.	ACRT
	description of the flow		

5. Performance (P)

ID	Rule Description	Sev.	ACRT
1.	There must not be any unused code.	High	+
2.	Avoid using recursion in performance situations. Recursive calls consume time and additional memory.	High	
3.	Do not write inside loop, expressions whose values do not change.	High	
4.	Minimize I/O operations.	Med	
5.	Do not use CPU-consuming statements if another solution exists.	Med	
6.	There must not be any unused variables.	Med	+
7.	If the logical expression is in format A AND B, make sure that A is more likely to receive the value False than B.	Med	
8.	If the logical expression is in format A OR B, make sure that A is more likely to receive the value True than B.	Med	
9.	Assign and use another variable in place of a direct array entry that is used repeatedly.	Med	
10.	Do not use mixed variable types in computations	Med	
11.	For COMP-3/PACKED-DECIMAL variables use an odd number of digits	Med	+
12.	For COMP/BINARY variables use 4, 9 or 18 digits	Med	+
13.	Use SIGNED numeric fields instead of unsigned (Except for report variables (DXXX))	Med	+
14.	Use COMPUTE instead of other complex statments (with GIVING...)	Low	+
15.	For switches and indexes use PIC S9(4) COMP/BINARY	Med	
16.	Prefix SECTIONS according to the frequency of usage (to avoid paging)	Med	
17.	Use ADD, SUBTRACT, MULTIPLY instead of COMPUTE and DIVIDE when the GIVING clause is not needed and there is no more than one sending variable	Low	
18.	Use ALL in SEARCH and FUNCTION..(TB(..)) with updated DEPENDING ON variable	Med	
19.	Always prefer the intrinsic functions.	Med	
20.	Use COMP/BINARY in non-DISPLAY numeric fields (IX, SW, CN).	Med	+
21.	Use READY TRACE only during the testing phases	Med	+

6. General (G)

ID	Rule Description	Sev.	ACRT
1.	A code is not considered complete if the compiler generates warnings.	High	
2.	All resources opened by the application must be explicitly closed	High	

ID	Rule Description	Sev.	ACRT
3.	Resource closing should always be done in the ZZ-SIUM-TOCHNIT or in ZZ-TAKALA section	High	
4.	Before every division calculation, check that the denominator is not zero.	High	
5.	Ensure that every file you open will be closed.	High	
6.	Do not close a resource you did not open.	High	
7.	Do not compare floating-point values for equality or inequality.	High	
8.	All possible errors must be handled.	Med	
9.	Use decision or state machines to handle conditions wherever possible.	Med	
10.	Program data structure levels should be only: 01, 03, 05 ... (Not 78, 77 or 66).	Med	+
11.	Initialize variables before first use.	Med	
12.	Do not use the following clauses: LABEL RECORD IS, DATA RECORD IS, DATE-COMPILED, DATE-WRITTEN, AUTHOR	Med	
13.	Use DE MORGAN rules for more than two "&& !" or two " !".	Low	
14.	RETURN-CODE valid values: 1-3 for information, 5-7 for remarks, 9-999 or greater than 1999 for errors	Low	
15.	Reads and Writes to central files or tables will be performed only by Read/Write routines	Low	
16.	Use INDEXED BY in each table definition	Low	+
17.	USE DEPENDING ON in each table definition	Low	+
18.	Use SEARCH ALL only for tables with more than 50 entries	Med	
19.	Use INITIALIZE only in the ZA-ATCHALAT-TOCHNIT.	Med	+
20.	To initialize a table more than once, use a temporary area initialized during the ZA-ATCHALAT-TOCHNIT	Med	
21.	COPY is allowed only in the DATA DIVISION	Med	
22.	Every program must return a status (RETURN-CODE)	Low	
23.	Do not use the EXIT SECTION statement. (EXIT PARAGRAPH statement is permitted)	Low	+
24.	To avoid overflow in arrays, the program must prevent it by proper checks	Med	
25.	Check the return code after calling a program if it returns a status	Med	
26.	PROGRAM-ID has to be identical to the file name (without the extension)	Low	+
27.	Use a FILE STATUS for every file in the program	Low	

7. Naming Convention (NC)

ID	Rule Description	Sev.	ACRT
1.	Use meaningful names, with standard abbreviations only for variable names, section names and paragraph name. (Use the Data Dictionary)	Med	+
2.	Use standard naming conventions of name prefix for variables, section names and file and record names..	Low	+
3.	Use MH Prefix for FILE STATUS variable name.	Low	+
4.	Define file names in the format : <XX> [<P>] NN-<filename> (where XX is one of: K, P, KP, S, D, <P> is 1-2 letter prefix pf the file and NN is a 2 digits	Low	+

ID	Rule Description	Sev.	ACRT
	numerator)		
5.	The file record prefix must have the same prefix as the corresponding file	Low	+
6.	The file record suffix must be "-R" Or "-R-FD"	Low	+
7.	Use a standard prefix structure (XXNN- where XX is one of C, DB, EZ, H, I, K, M, MH, PR, S, SH, SD, SF, SW, TV and NN is a 2 digit numerator) - See Appendix 10	Low	
8.	Redefined variable must include R, RX, RX1, RX2 . . . or R9, R91, R92 . . . as a suffix	Low	+
9.	The names must be written in hebrew with english characters (phonetic hebrew)	Low	
10.	Add a prefix of 2-4 letters to each section in the PROCEDURE DIVISION	Low	
11.	The Linkage Section 01 variable prefix must be LS and a numerator	Low	+
12.	Variable names must be declared up to 30 characters long	Low	+

8. Structured Programming (SP)

ID	Rule Description	Sev.	ACRT
1.	Use WHEN OTHER for every EVALUATE.	Med	+
2.	Removed	Med	
3.	GO TO is permitted only to the EXIT label of the same SECTION	High	+
4.	A-TOCHNIT-RASHIT contains only PERFORM, EVALUATE, IF, GOBACK, EXIT PROGRAM and SORT statements.	Med	+
5.	Encapsulate EXEC SQL in I/O programs	Med	+

9. Recommendations (R)

ID	Rule Description	Sev.	ACRT
1.	Limit usage of ON SIZE ERROR and ROUNDED (Performance)	Med	
2.	Prefer usage of MULTIPLY instead of DIVIDE. (Performance)	Med	

10. Standard Abbreviations

Abbreviation	Meaning
amt	amount
avg	average
c	character
cnt, n	counter
cst	customer
db	database
dd	day (in date)
e	exception
err	error
hh	hour (in time)
i, j, k	indices
id	identification
ind	indicator
ix	index
len	length
max	maximum
min	minimum
mm	month (in date)
mmm	month (in date)
mn	minutes (in time)
msg	message
num	number
obj	object
off	offset
pk	primary key
pos	position
prod	product
rc	return code
rec	record
s	string
ss	seconds (in time)
sts	status
tot	total
yy	year (in date)
yyyy	year (in date)

11. Standard Prefixes

Prefix	Meaning
C	Constants
DB	Data Base
EZ	Ezer
H	Hodaot
I	Indexes
K	Kelet
M	Monim
MH	Maftehot Hashavaa
PR	Parameters
S	Shurot Hadpasa
SH	Shurot Header
SD	Shurot Details
SF	Shurot Footer
SW	Switches
TV	Tavlaot