

# 1. Introduction

## 1.1. General

The purpose of this document is to be used as a checklist at the Code Review process, to identify each rule by its Quality Factor, and to define the severity level of each rule

## 1.2. Quality Factors

Type	הערות
Maintainability	Rules which define how to write a program in such a way that it will be easy to maintain
Formatting	Rules which define the visual format of the program.
Documentation	Where to put documentation, what to write in documentation and what to document
Performance	Rules which define how to write efficient programs (I/O, CPU, Memory)
General	General rules as which commands to use and use of decision tables and state tables
Naming Convention	Methods of naming variables, sections, ...
Structured	Rules concerning the structured programming
Recommendations	Local special recommendations

## 1.3. Remarks

- If the code written has to violate a rule, add a comment before the code that will detail the reason of the violation

# 2. Maintainability (M)

ID	Rule Description	Sev.	ACRT
1.	The program has to be written as simple and readable as possible.	High	
2.	<a href="#">Avoid duplicate code.</a>	High	+

ID	Rule Description	Sev.	ACRT
3.	<a href="#">Do not write programs having more than 1500 lines in the PROCEDURE DIVISION.</a>	Med	+
4.	<a href="#">Do not use more than one nested cobol function.</a>	Med	
5.	Do not use hard-coded constants in PROCEDURE DIVISION. Define constant variables in COXX (Except 0 and 1 and in X-ERROR)	Med	+
6.	Do not use more than three levels of nesting statements.	Med	+
7.	<a href="#">The number of parameters must be equal to the number of arguments (the OMITTED option is allowed).</a>	Med	
8.	Write short sections (up to 100 lines of code not including comments).	Med	+
9.	<a href="#">Each SECTION should be limited to performing a single, well-defined task.</a>	Med	
10.	<a href="#">Do not use more than five arguments in the Linkage SECTION (01 Level).</a>	Med	+
11.	<a href="#">Do not use initial values in the WORKING STORAGE, except for constants or titles.</a>	Low	
12.	<a href="#">Always indicate the order of operation explicitly by using parentheses.</a>	Low	
13.	<a href="#">Divide complex expressions into a number of expressions. (including complex IFs)</a>	Low	
14.	Use ON EXCEPTION for dynamic subroutines (dynamic CALL )	Low	
15.	<a href="#">Variables must be alphabetically ordered by prefix</a>	Low	+
16.	<a href="#">Place the variable definition in the corresponding group according to its meaning(constants in COXX, counters in CNXX..)</a>	Low	
17.	<a href="#">Arrange the SECTIONS in PROCEDURE DIVISION by prefixes in ascending order.</a>	Low	+
18.	<a href="#">Do not use the ALSO clause in the EVALUATE statement</a>	Med	-
19.	Use PERFORM SECTION only and without THRU and not PERFORM paragraph.	Med	+
20.	<a href="#">Do not use recursive PERFORM and CALL</a>	Med	
21.	<a href="#">A section that performs a general task will have a "PXX" prefix</a>	Low	
22.	The following sections are mandatory: A-MAIN, W-INIT, X-ERROR, Z-FINISH	Low	+
23.	Add an END clause to the following statements: <a href="#">EVALUATE</a> , IF, <a href="#">READ</a> , inline <a href="#">PERFORM</a> , <a href="#">CALL</a>	Low	+
24.	<a href="#">The GOBACK, EXIT PROGRAM will be used only in the A-MAIN SECTION OR X-ERROR</a>	Low	+
25.	Do not use the following statements: ALTER, <a href="#">RENAME</a> , GO TO DEPENDING ON, NEXT SENTENCE, MOVE CORRESPONDING, MOVE CORR, ENTRY, ACCEPT DATE	Med	+
26.	<a href="#">Groups must be sorted by group names, and variables must be sorted within the group</a>	Low	+
27.	<a href="#">Never change the index inside a loop (PERFORM VARYING)</a>	Med	
28.	Do not use STOP RUN	Low	+
29.	<a href="#">Use LOW-VALUE instead of X'00' and HIGH-VALUE instead of X'FF'</a>	Low	+
30.	<a href="#">Use COPY for records in FD and SD</a>	Med	+
31.	Use only EVALUATE TRUE.	Med	+

### 3. Formatting (F)

ID	Rule Description	Sev.	ACRT
1.	<a href="#">Use separating comment lines before and after SECTIONS and DIVISIONs.</a>	Low	+
2.	<a href="#">Align the reserved words: PIC, VALUE, OCCURS, INDEXED BY, DEPENDING ON.</a>	Low	+
3.	<a href="#">Separate 01 levels by using empty lines.</a>	Low	+
4.	<a href="#">Each section must include the paragraph: X-EXIT or X-EXIT except in Declaratives (X is the section prefix)</a>	Low	+
5.	<a href="#">The exit paragraph name must have the structure: X-EX. EXIT Or X-EXIT. EXIT. (X is the section prefix)</a>	Low	
6.	<a href="#">The exit paragraph must include only the EXIT statement:</a>	Low	
7.	<a href="#">Do not overuse blank lines.</a>	Low	
8.	Each source line must contain a single command only	Low	
9.	<a href="#">Do not write spaces after the opening parenthesis, "(", and before the closing parenthesis, ")".</a>	Low	
10.	<a href="#">Add a space around assignment and binary/conditional operators.</a>	Low	
11.	<a href="#">Add a space after commas</a>	Low	
12.	<a href="#">There must be a clear indentation (4 spaces) between levels.</a>	Low	
13.	<a href="#">Place the else statement in its own line, except for else if.</a>	Low	
14.	<a href="#">If a condition is written on more than one line, place the "AND/OR" to the right of the conditions.</a>	Low	
15.	<a href="#">If the "THEN" clause is empty, add a CONTINUE statement</a>	Low	
16.	Removed	Low	
17.	<a href="#">Do not use continuation line (‘-‘ in column 7) in the Working Storage Section</a>	Low	+
18.	<a href="#">Use THEN for every IF</a>	Low	+
19.	<a href="#">All four DIVISIONs must be declared in every program</a>	Low	+
20.	<a href="#">Use only the paragraph X-EXIT except in Declaratives</a>	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.	<a href="#">Use the standard format for Program Header documentation: Purpose, Description, Calling statement, Arguments, Input, Output, Programmer, Date</a>	Low	
3.	<a href="#">Add before each group of variables an according documentation</a>	Low	
4.	<a href="#">Each group of source lines which performs an isolated action must be explained by a comment.</a>	Low	
5.	<a href="#">A section documentation must be written before each section in PROCEDURE DIVISION</a>	Low	
6.	<a href="#">The documentation of a section must include at least the purpose and the</a>	Low	

	Rule Description	Sev.	ACRT
	<a href="#">description of the flow</a>		

## 5. Performance (P)

ID	Rule Description	Sev.	ACRT
1.	<a href="#">There must not be any unused code.</a>	High	+
2.	<a href="#">Avoid using recursion in performance situations. Recursive calls consume time and additional memory.</a>	High	
3.	<a href="#">Do not write inside loop, expressions whose values do not change.</a>	High	
4.	<a href="#">Minimize I/O operations.</a>	Med	
5.	<a href="#">Do not use CPU-consuming statements if another solution exists.</a>	Med	
6.	<a href="#">There must not be any unused variables.</a>	Med	+
7.	<a href="#">If the logical expression is in format A AND B, make sure that A is more likely to receive the value <b>False</b> than B.</a>	Med	
8.	<a href="#">If the logical expression is in format A OR B, make sure that A is more likely to receive the value <b>True</b> than B.</a>	Med	
9.	<a href="#">Assign and use another variable in place of a direct array entry that is used repeatedly.</a>	Med	
10.	Do not use mixed variable types in computations	Med	
11.	<a href="#">For COMP-3/PACKED-DECIMAL variables use an odd number of digits</a>	Med	+
12.	<a href="#">For COMP/BINARY variables use 4, 9 or 18 digits</a>	Med	+
13.	<a href="#">Use SIGNED numeric fields instead of unsigned (Except for report variables (DXXX))</a>	Med	+
14.	<a href="#">Use COMPUTE instead of other complex statments (with GIVING...)</a>	Low	+
15.	For switches and indexes use PIC S9(4) COMP/BINARY	Med	
16.	<a href="#">Prefix SECTIONS according to the frequency of usage (to avoid paging)</a>	Med	
17.	<a href="#">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</a>	Low	
18.	<a href="#">Use ALL in SEARCH and FUNCTION.. (TB (..)) with updated DEPENDING ON variable</a>	Med	
19.	<a href="#">Always prefer the intrinsic functions.</a>	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.	<a href="#">Resource closing should always be done in the Z-FINISH or in X-ERROR</a>	High	
4.	<a href="#">Before every division calculation, check that the denominator is not zero.</a>	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.	<a href="#">Use decision or state machines to handle conditions wherever possible.</a>	Med	
10.	Program data structure levels should be only: 01, 03, 05 ... (Not 78, 77 or 66).	Med	+
11.	<a href="#">Initialize variables before first use.</a>	Med	
12.	<a href="#">Do not use the following clauses: LABEL RECORD IS, DATA RECORD IS, DATE-COMPILED, DATE-WRITTEN, AUTHOR</a>	Med	
13.	<a href="#">Use DE MORGAN rules for more than two "&amp;&amp; !" or two "   !".</a>	Low	
14.	<a href="#">RETURN-CODE valid values: 1-3 for information, 5-7 for remarks, 9-999 or greater than 1999 for errors</a>	Low	
15.	<a href="#">Reads and Writes to central files or tables will be performed only by Read/Write routines</a>	Low	
16.	<a href="#">Use INDEXED BY in each table definition</a>	Low	+
17.	<a href="#">USE DEPENDING ON in each table definition</a>	Low	+
18.	Use SEARCH ALL only for tables with more than 50 entries	Med	
19.	Use INITIALIZE only in the W-INIT Section.	Med	+
20.	<a href="#">To initialize a table more than once, use a temporary area initialized during the W-INIT Section</a>	Med	
21.	<a href="#">COPY is allowed only in the DATA DIVISION</a>	Med	
22.	<a href="#">Every program must return a status (RETURN-CODE)</a>	Low	
23.	Omitted	Low	-
24.	<a href="#">To avoid overflow in arrays, the program must prevent it by proper checks</a>	Med	
25.	<a href="#">Check the return code after calling a program if it returns a status</a>	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.	<a href="#">Use meaningful names, with standard abbreviations only for variable names, section names and paragraph name. (Use the Data Dictionary)</a>	Med	+
2.	Use standard naming conventions of name prefix for variables, section names and file and record names..	Low	+
3.	Use WS Prefix for FILE STATUS variable name.	Low	+
4.	Define file names in the format: <XX> [<P>] NN-<filename> (where XX is one of: I, O, IO, S, R, <P> is 1-2 letter prefix pf the file and NN is a 2 digits numerator)	Low	+
5.	The file record prefix must have the same prefix as the corresponding file	Low	+

ID	Rule Description	Sev.	ACRT
6.	The file record suffix must be "-R" Or "-R-FD"	Low	+
7.	Use a standard prefix structure (XXNN- where XX is one of CO, CN, IX, PR, LD, LH, LT, SW, TB, WS 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 english	Low	
10.	Add a prefix of 1-4 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-MAIN 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
<b>Amt</b>	amount
<b>Avg</b>	average
<b>C</b>	character
<b>cnt, n</b>	counter
<b>Cst</b>	customer
<b>Db</b>	database
<b>Dd</b>	day (in date)
<b>E</b>	exception
<b>Err</b>	error
<b>Hh</b>	hour (in time)
<b>i, j, k</b>	indices
<b>Id</b>	identification
<b>Ind</b>	indicator
<b>Ix</b>	index
<b>Len</b>	length
<b>Max</b>	maximum
<b>Min</b>	minimum
<b>Mm</b>	month (in date)
<b>Mmm</b>	month (in date)
<b>Mn</b>	minutes (in time)
<b>Msg</b>	message
<b>Num</b>	number
<b>Obj</b>	object
<b>Off</b>	offset
<b>Pk</b>	primary key
<b>Pos</b>	position
<b>Prod</b>	product
<b>Rc</b>	return code
<b>Rec</b>	record
<b>S</b>	string
<b>Ss</b>	seconds (in time)
<b>Sts</b>	status
<b>Tot</b>	total
<b>Yy</b>	year (in date)
<b>Yyyy</b>	year (in date)

## 11. Standard Prefixes

<b>Prefix</b>	<b>Meaning</b>
<b>CO</b>	Constants
<b>CN</b>	Counters
<b>IX</b>	Indexes
<b>LD</b>	Line Details - in Reports
<b>LH</b>	Line Header - in Reports
<b>LT</b>	Line Total - in Reports
<b>PR</b>	Parameters
<b>SW</b>	Switches
<b>TB</b>	Tables
<b>WS</b>	General Purpose