

ソース・アウトライン

- ctl-opt 制御仕様  
default \*LIBL/RPGLEHSPEC  
QRPGLE/DFTLEHSPEC
- dcl-pr フォントタイプ
- dcl-pi フォンション・インターフェース
- dcl-f グローバル定義  
dcl-s (ファイル, 変数  
dcl-ds , データ構造, 配列  
dcl-c , 定数)
- exec sql SQL プリコンパイルオプション  
SQL カール宣言
- メイン・ソース
- begsr グローバル・サブルーチン

実行時・コンパイル時

- dcl-proc フォンション・仕様
- dcl-pi フォンション・インターフェース
- dcl-f ローカル定義  
dcl-s (ファイル, 変数  
dcl-ds , データ構造, 配列, 定数)
- プログラムの本体
- begsr ローカル・サブルーチン

プログラムの本体

配列 指標 1~16, 773, 104  
定義: ... dim(要素数 | \*var:要素数 | \*auto:要素数 | \*ctdata)

いつ 実行時  
コンパイル時 (CTDATA)  
実行前時  
構文: 配列名(指標[\*] \*next)  
演算子: in  
命令コード: for-each, sorta  
関数: %concatarr, %elem, %fields, %list, %lookupxx, %maxarr, %minarr, %split, %subarr, %tlookupxx, %xfoot

データ構造 キーワード定義:  
alias, based, ext, extfld, extname, inz, likeds, likerec, overlay, pos, prefix, qualified, samepos, template  
命令コード: clear, eval, eval-corr, reset  
DSサブフィールドの参照:  
DS名, サブフィールド名  
特殊データ構造

- 凡例
- リファレンス
  - コード例
  - 参考記事

自由形式構文



構文ルール  
コメント: //  
終了: ;  
組込関数: %関数名(引数1:引数2)  
大文字と小文字区別なし  
ソース継続文字: '+'  
変数名の先頭に数値は使用不可

データ型 キーワード

	型	キーワード	長さ
英数字 EBCDIC 照合順序	文字列・固定長	char	1-16773104
	文字列・可変長	varchar	1-16773100
	UCS・固定長	ucs2	1-8386552
	UCS・可変長	varucs2	1-8386550
	グラフィック・固定長	graph	1-8386552
	グラフィック・可変長	vargraph	1-8386550
数値	ハナリ	bindec	1-9
	浮動小数	float	4, 8
	整数	int	3, 5, 10, 20
	符号なし整数	uns	3, 5, 10, 20
	ゾーン10進数	zoned	63, 63
日付/時刻	ハナリ	packed	63, 63
	日付	date	
	時刻	time	
その他	タイムスタンプ	timestamp	26-32
	真偽	ind	1
	ポインタ	pointer	16 bytes
	プログラムのポインタ	pointer(*proc)	16 bytes
	オブジェクト	object	

SQL データ型 【SQLTYPE()】

	型	キーワード	長さ	
ハナリ	ハナリ・固定長	binary	1-32766	
	ハナリ・可変長	varbinary	1-32740	
LOB	文字・LOB	clob	1-16773100	
	ダブルハナリ文字・LOB	dbclob	1-8386550	
	ハナリ・LOB	blob	1-16773100	
	LOBファイル参照	clob_file		
		dbclob_file		
		blob_file		
	LOB・ローケータ	clob_locator		
dbclob_locator				
blob_locator				
LOB XML	XML	xml_clob	1-16773100	
		xml_dbclob	1-8386550	
		xml_blob	1-16773100	
XMLファイル参照	xml_clob_file			
	xml_dbclob_file			
	xml_blob_file			
XML・ローケータ		xml_locator		
その他	行ID	rowid		
	結果セットローケータ	result_set_locator		

指標  
0 - \*OFF / 1 - \*ON  
\*INxx (xx 1~99)  
\*INLR : 最終コメント  
\*INKx : PF機能  
命令コード 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

論理演算子  
= <> > = < <=  
AND OR NOT IN

命令オプション  
命令拡張: (A), (D), (E), (H), (M), (N), (P), (R), (T), (Z)

割り当て演算子  
= += -= \*= /= \*\*=  
命令コード: clear, eval, evalr, reset

文字変数  
連結子: +  
命令コード: clear, eval, evalr, reset

BIF: %char, %charcount, %check, %checkr, %concat, %lower, %max, %min, %replace, %scan, %scanr, %scanrpl, %split, %str, %subst, %trim, %triml, %trimr, %ucs2, %upper, %xlate

数値変数  
演算子: + - \* / \*\*  
命令コード: clear, eval, reset  
BIF: %abs, %dec, %dech, %div, %editc, %editw, %float, %int, %inlh, %max, %min, %rem, %sqrt, %uns, %unsh

構造化命令

**dou** **dow** **dow \*on** **for** **for-each** **if** **select** **select expr**

*do until*  
必ず1回は実行する式が真になるまで実行される  
ループの終端 (ENDDO) で式は評価される

*do while*  
式が真の間実行される  
ループの開始 (DOW) で式は評価される

永久ループ  
decl-s ForEverTrue ind inz(\*on)  
dow ForEverTrue  
decl-s i int(10)  
for i=1

「FOR~ENDFOR」スタート中の繰り返し処理回数を制御する  
BIF: %list, %subarr

配列内の要素を繰り返し処理する  
BIF: %list, %subarr

条件1が真  
条件2が真  
全条件が偽  
endif

when 条件1  
when 条件2  
other  
endsl

when-is value  
when-in array  
other  
endsl

iter ENDDOへ  
leave ENDDOの後へ  
enddo  
endfor

演算子: in  
BIF: %list, %range

**サブルーチン**

**begsr** **exsr** **leavesr** **endsr**

exit  
\*inzsr  
初期化サブルーチン  
\*pssr  
エラー・サブルーチン

**静的呼び出し**

**callp** **return** **on-exit**

呼び出し元に戻る  
プログラム終了時に実行  
同一活動化グループに再帰性がない

**プロトタイプ引数**

ソース プロトタイプ  
呼出元 呼出先  
完全一致  
decl-p1

BIF: %addr, %omitted, %parnum, %parms, %passed, %proc

最大引数(プログラム): 255  
最大引数(プログラム): 399

**API プロトタイプ**

API finder  
APIのタイプ  
プログラムベース  
サービスプログラムベース  
ILE CEE  
UNIXタイプ

API finder プロトタイプ:  
- QSYSINC/QRPGLESRC  
- Easy400.net  
- Scott Klement, C to RPG  
- Midrange  
- IOpen Bob Cozzi

**データエリア**

キーワード定義: dtaara  
命令コード: in, out, unlock  
BIF: %addr, %alloc  
%LDA

**変数情報**

BIF: %decpos, %len, %nullind, %size

**XML, JSON**

命令コード: data-gen, data-into, xml-into, xml-sax  
BIF: %data, %gen, %handler, %parser, %xml

**オブジェクト**

BIF: %this

**日付と時刻**

キーワード定義: datfmt  
演算子: +, -  
命令コード: eval, test  
BIF: %char, %date, %days, %dec, %diff, %hours, %minutes, %months, %mseconds, %seconds, %subdt, %time, %timestamp, %years

定数リテラル: d, t, z 例) T'xx:xx:xx'  
ユーザ日付の特殊語: UDATE, UMONTH, UYEAR, UDAY, C\*DATE, \*DATE, \*MONTH, \*YEAR, \*DAY  
日付期間コード: \*years/\*y, \*months/\*m, \*days/\*d, \*hours/\*h, \*minutes/\*mn, \*seconds/\*s, \*mseconds/\*ms

書式

**埋め込みSQL**

構文: exec sql ... ;  
ホスト変数: :変数名  
句で使用: where, select, into, order by, set, values (insert), call  
標識変数: int(5)  
拡張キーワード標識: コンパイル時 \*extind

SQL連絡域: SQLCA  
SQL記述子域: SQLDA

**エラー・ハンドラー**

優先順:  
1. エラー拡張 (e)  
2. モニター・グループ  
3. 例外サブルーチン \*pssr (ファイル)  
F仕様書キーワード: infsr  
4. ILE条件処理 \*pssr (プログラム)  
5. 例外サブルーチン \*pssr (プログラム)  
6. RPG省略エラー・ハンドラー  
命令コード: monitor  
BIF: %error, %status

**ファイル** **データベース** **ディスプレイ** **プリンター**

キーワード: alias, commit, extdesc, extfile, extmbr, infds, prefix, qualified, rename, template, usage, usropr  
画面/帳票ファイルのキーワード: indds, oflind, printer, sfile, workstn

命令コード  
キー読み取り: chain, reade, readpe, settl, setgt  
順次読み取り: read, readp  
データ操作: delete, update, write, unlock  
DBコミット制御: commit, rolbk  
ファイル: close, open  
画面ファイル専用: exfmt, readc  
BIF: %eof, %equal, %error, %fields, %found, %kds, %open, %status

**ポインター**

命令コード: dealloc  
BIF: %addr, %alloc, %paddr, %realloc

**バインド言語**

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

**表意定数**

\*BLANK, \*BLANKS = ''  
\*ZERO, \*ZEROS = 0  
\*ON = '1' / \*OFF = '0'  
\*NULL = Null値  
\*LOVAL = 最小値 / \*HIVAL = 最大値

**メッセージ**

命令コード: snd-msg  
BIF: %msg, %target

**コンパイル指示ディレクティブ**

開始桁が7桁 (又は完全フリーは1桁) 以降。文末のセミコロンは不要  
/COPY 又は /INCLUDE [ライブラリ/ファイル,メンバー-|"IFSパス"] : 外部ソースメンバーをコピーする  
条件付きコンパイル:  
/DEFINE ディレクティブ名 又は /UNDEFINE:  
/IF [NOT] DEFINED (ディレクティブ名)  
...;  
/ELSEIF  
...;  
/ELSE  
...;  
/ENDIF  
/EOF: 残りのソースを無視する



## 謝辞

「RPG IV free - 対話式・チートシート【日本語版】」は、Marco Riva 氏の「RPG IV free - interactive cheatsheet」を元に、株式会社中部システム（CSC）で翻訳しています。

まずは何をにおいても、我々日本のIBM i開発者の為に、この素晴らしいRPGチートシートを提供してくれた、Marco Riva 氏 に、多大なる感謝の意を表明致します。

また、彼は母国イタリアのコミュニティ FAQ400 にも参加しており、そこで” 何故このチートシートを制作するに至ったか？” を述べております。

[FAQ400 - RPG IV free cheatsheet]

(<https://blog.faq400.com/en/programming/rpg-en/rpg-iv-free-cheatsheet-2/>)

何故我々CSCが、「RPG IV free - interactive cheatsheet」を日本語翻訳をしようと考えたかについて少しお話させて頂きます。

「オープン系の技術者も取り込める」の様なキャッチーなフレーズが、IBMi技術者の界隈で叫ばれる中、果たしてFF-RPGの資料やサンプルは、どれだけあるのでしょうか？

そんな時、我々はRiva氏の「RPG IV free - interactive cheatsheet」に出会いました。その出会いのきっかけを作ってくれたのは、我々の良き友のAlan Seiden氏とSeiden Groupでした。

それは、英語の資料でした。しかし、多くの日本人技術者は、資料が英語であることで、それがどんなに素晴らしいモノだとしても目にする機会を失いがちです。

我々は、それを見た時「これを日本の技術者の為に是非日本語化したい！」と即座に思い、友人のSeiden氏を通じて、Riva氏に連絡をとりました。

彼も母国がイタリアのため、我々日本人の悩み、” 英語の資料が日本語になっている事が、どんなに素晴らしい事か” との理解を共有し、快くこの日本語化に賛同してくれたのです。

我々のこの日本語翻訳のタスクが、これからRPGに出会うであろう技術者の少しでも助けになれば幸いです。

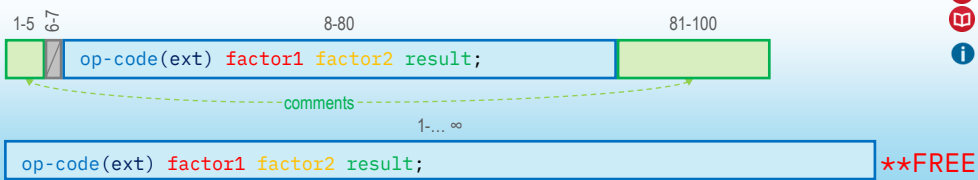
## 免責事項

我々は、この文書内で公開された情報、記事、関連リンク等について、絶対的な正確性を保証しません。文書内には、外部サイトへのリンクが含まれている場合がありますが、その目的は、読者がテーマに関連する詳細情報を得やすくするためです。リンクされた外部サイトは、我々の管理下になく、そのリンクの正確性や内容については責任を負いません。文書を利用したことによって生じたいかなる損害についても、一切の責任を負いかねます。ご利用の際には、必ず自己責任においてご利用下さい。我々は、予告なしにいつでもこの条項を変更する権利を保有します。この文書のオリジナルの著作権は、Marco Riva氏が有します。また、日本語翻訳版の著作権は株式会社中部システムが有します。この文書の無断転載・再配布は固く禁止されています。個人利用以外の使用を希望される場合は、事前に許可を得る必要があります。

source outline



free-form statements



syntax rules

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

indicators

0 - \*off / 1 - \*on  
 \*inx (xx da 1 a 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

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
	packed decimal	packed	63,63
date/time	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	

logical operators

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

operation codes

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

assignment operators

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

arrays

index from 1 to 16.773.104  
 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, %list, %lookupxx, %maxarr, %minarr, %split, %subarr, %tlookupxx, %xfoot

data structures

keyword definition: alias, based, ext, extfld, extname, inz, likeds, likerec, overlay, 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**

- reference
- examples
- 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	xml_clob, xml_dbclob, xml_blob
LOB XML	XML file reference	xml_clob_file, xml_dbclob_file, xml_blob_file	
	XML locator	xml_locator	
	ROW ID	rowid	
other	resultset locator	result_set_locator	

character variables

concatenation: +  
 op.code: clear, eval, evalr, reset  
 BIF: %char, %charcount, %check, %checkr, %concat, %lower, %max, %min, %replace, %scan, %scanr, %scanrp, %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, %int, %inth, %max, %min, %rem, %sqrt, %uns, %unsh



### structured operations

**do until**  
execute at least once performed until the expression is true expression is evaluated at the cycle end

**do while**  
performed while the expression is true expression is evaluated at the cycle start

**do \*on**  
dcl-s ForEverTrue ind inz(\*on)  
do ForEverTrue  
dcl-s i int(10)  
for i=1  
**endless loop**

**for**  
controls the number of times the group will be processed  
BIF: [%list](#), [%subarr](#)

**for-each**  
process the items in the array  
BIF: [%list](#), [%subarr](#)

**if cond1**  
cond1 true  
**elseif cond2**  
cond2 true  
**else**  
all cond false  
**endif**

**select**  
cond1 true  
**when cond1**  
cond2 true  
**when cond2**  
all cond false  
**other**  
all cond false  
**endsl**

**select expr**  
cond1 true  
**when-is value**  
cond2 true  
**when-in array**  
all cond false  
**other**  
all cond false  
**endsl**

**iter** go to enddo  
go after enddo  
**leave**  
**enddo**  
**endfor**

operator: [in](#)  
BIF: [%list](#), [%range](#)

### subroutines

**begsr** → **exsr**  
exit **leavesr**  
**endsr**

\*inzsr  
initial subroutine  
\*pssr  
error subroutine

### static call

**callp**  
**return**  
return to caller  
**on-exit**  
execute when procedure ends  
no recursivity in the same activation group

### prototyped parameters

source prototype **dcl-pr** /copy  
called **dcl-pi** /copy  
exact match

BIF: [%addr](#), [%omitted](#), [%parmnum](#), [%parms](#), [%passed](#), [%proc](#)  
max parm (program): 255  
max parm (procedure): 399

### API prototypes

API finder

Types of API

- based-programs
- service-program-based
- ILE CEE
- UNIX-type

API finder prototypes:

- QSYSINC/QRPGLESRC
- Easy400.net
- Scott Klement, da C a RPG
- Midrange
- iOpen Bob Cozzi

### 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](#)

### object

BIF: [%this](#)

### 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:

- error extender ([e](#))
- monitor group
- subroutine [\\*pssr](#) (file)
- [ILE condition handlers](#)
- subroutine [\\*pssr](#) (program)
- RPG default error handler

op.code: [monitor](#)  
BIF: [%error](#), [%status](#)

### files

keyword: [alias](#), [commit](#), [extdesc](#), [extfile](#), [extmbr](#), [infds](#), [prefix](#), [qualified](#), [rename](#), [template](#), [usage](#), [usropr](#)  
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](#)

### 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](#)

### 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](#)

### 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



## With great appreciation

"RPG IV free - interactive cheatsheet (Japanese version)" is based on "RPG IV free - interactive cheatsheet" by Marco Riva and translated by Chubu System Corporation (CSC).

First and foremost, we would like to express our great appreciation to Mr. Marco Riva for providing this wonderful RPG cheat sheet for us Japanese IBM i developers.

He also participates in the community famous FAQ400 in his home country of Italy, where he explains "Why did I create this cheat sheet?"

[FAQ400 - RPG IV free cheatsheet]

(<https://blog.faq400.com/en/programming/rpg-en/rpg-iv-free-cheatsheet-2/>)

Let me tell you a little about why we at CSC decided to translate "RPG IV free - interactive cheatsheet" into Japanese.

With catchy phrases like "open-source engineers can be included," how many FF-RPG documents and samples are really available in the IBM i engineering community?

Then we came across Riva's "RPG IV free - interactive cheatsheet". It was our good friend Alan Seiden and the Seiden Group who helped us make that connection.

It was a document in English. However, many Japanese engineers tend to lose the opportunity to see the materials, no matter how wonderful they are, because they are in English.

When we saw it, we immediately thought, "We really want to translate this into Japanese for Japanese engineers! We contacted Mr. Riva through our friend Seiden.

He is also from Italy, so he shared our concern, "How wonderful it is to have English documents in Japanese," and he gladly agreed to help us translate them into Japanese.

We hope that our Japanese translation task will be of some help to engineers who will encounter RPGs in the future.

### Disclaimer

We do not guarantee the absolute accuracy of the information, articles, related links, etc. published within this document. The document may contain links to external sites, the purpose of which is to facilitate the reader in obtaining further information related to the subject matter. The linked external sites are not under our control and we are not responsible for the accuracy or content of such links. We cannot be held liable for any damages whatsoever resulting from the use of the documents. Use of the site is at your own risk. We reserve the right to change these terms at any time without notice. The original copyright of this document is owned by Marco Riva. The Japanese translation of this document is copyrighted by Chubu System Co., Ltd. Unauthorized reproduction or redistribution of this document is strictly prohibited. If you wish to use this document for purposes other than personal use, you must obtain prior permission.