SOFTWARE INFORMATION

Issued by: S & I Systems Department Software Support and Development Group Building TQV-3, Eindhoven The Netherlands.

Number : 4022 152 20141

Date : December 1980

Subject : Software Release Bulletin RTL/2 rel. 2.0

SOFTWARE RELEASE BULLETIN

RTL/2 release 2.0

0. TABLE OF CONTENTS.

1. GENERAL

1.1 Introduction.

This bulletin describes the product 'RTL/2 P800'.

Information is given about the contents of the product, about the way to implement it in your system and about the behaviour of the product.

1.2 Main items of this release.

This release contains a RTL/2 compiler which is able to run under control of the P800 DOS system release 8.1 and the P800 MAS system release 6.
The only restriction is the available amount of memory which is for DOS about 20K words and for MAS the maximum of 32K words.

Also a LVED and LVER program is available on both systems.

The runtime package contains an adapted base-program (RRBPGA, RRBPGB) to be able to pass parameters at program activation.
A macro (RRBPG) is available to generate suitable programs either for memory resident programs or disc resident programs.

The standard stream I/O support is split up into a non-reentrant part (RRSIOA) and a reentrant part (RRSIOB).
The consequence of the change in the stream I/O support is that when a program uses I/O, a call to the procedure RRINIO must be made before opening any stream.
E.g.

```
        .
        .
        .
EXT PROC () RRINIO;
        .
        .
        .
ENT PROC RRJOB ();
        .
        .
    RRINIO ();
        .
    RROPO (x);
    RROPI (y);
```

The FORTRAN interface has bee adapted to the Extended FORTRAN compiler for parameter passing via register A13 (OPTM=YES).
where RTL/2 release 1.4 supported the parameter passing via A4 (OPTM=NO, equivalent for the "old" FORTRAN compilers).

2. RELEASE DESCRIPTION

2.1 Ordering.

The RTL/2 package has to be ordered separately by using the 12-NC number:

8701 352 82000

2.2 Products.

The RTL/2 release contains the following parts.

PV 380-190 RTL/2 Basic Package

- RTL/2 compiler
- Control routines
- Base programms
- Stream I/O formatting
- Stream I/O support
- Supervisory control procedures
- Mathematical library.

PV 380-191 RTL/2 Linkage Verifier.

2.3 System generation

- The RTL/2 compiler is delivered as load module with unique identification per user.

- For delivery on different media a generation procedure will be present to fill a user entry on disk with the available and other necessary member(s).

- In case of delivery on cassette or tape the program FILEXC will be used to transport the overlay load module RTL/2. This program FILEXC must be present in the system. For DOS the program should be present in the current userid.

- Filecode /10 should be assigned to the cassette(s) or tape before running the procedure to read the RTL/2 package.
  To create a user entry with the RTL/2 package the following actions are to be taken for DOS:

```
USERID: <userid>                    userid of package to be put on.
   ASG    /E1,TKnn                  assign file code /E1 to cassette
      or ASG    /E1,MTmm            or tape with concerning procedure.
   RDA    /A
   KPF    /A,READ
   ASG    /10,TKnn                  file code necessary to read
      or ASG    /10,MTmm            cassette or tape.
   ASG    /E0,DK,READ
```

or for MAS:

```
:JOB USID=<userid>,DAD=/Fx      userid of package to be put on.
ASG    FCOD=/E1,DVCE=TKnn       assign file code /E1 to cassette
 or ASG    FCOD=/E1,DVCE=MTmm   or tape with concerning procedure.
LIB
CSF    /E1,ONAM=/A,ODAD=/Fx
KPF    /A,UF,READ
LEN
ASG    FCOD=/10,DVCE=TKnn       file code necessary to read cassette
 or ASG    FCOD=/10,DVCE=MTmm   or tape with concerning procedure.
ASG    FCOD=/E0,FNAM=READ
```

2.4 Contents.

Contents of RTL/2 release 2.00

Object modules:

MLRATN	RRBPGA	SLRFR
MLREXP	RRBPGB	SLRFW
MLRFRT	RRBUFF	SLRFWF
MLRLOG	RRSIO	SLRIR
MLRSIN	RTLR00	SLRIW
MLRSRT	RTLR01	SLRIWF
	RTLR03	SLRRR
	RTLR08	SLRRW
	RTLR09	SLRRWF
	RTLR11	SLRTR
	RTLR12	SLRTW
	RTLR14	
	RTLR16	
	RTLR17	
	RTLR18	
	RTLR23	
	RTLR26	
	RTLR28	

Load modules: RTL (overlay load module).
LVED (verifier editor).
LVER (verifier).

Procedures: M:PROC with 2 procedures in it:
$RTL and $LVE for DOS-systems
or
B:PROC with 2 procedures in it:
%%RTL and %%LVE for MAS-systems.

Source: RRBPGA and RRBPGB.

UF-file: RRBPG (macro).

The procedures are necessary to run the RTL compiler as well as the linkage verifier.

Others:

A member R:XREF(UF) is created and contains the cross reference information of the runtime modules.

2.5 Environment

- The RTL/2 compiler uses the instruction set of the P856 and generates source modules with this instructionset. Additionally the floating point instruction can be generated (option FP) but by default these arithmetic functions are done by control routines.

- The P800 DOS systems does not support the floating point processor.

- The compiler needs 20 K words to run (minimum).

3. RELEASE COMPONENT DESCRIPTION

Appendix 1 contains a survey of all solved problems.

3.1 RTL/2 P800 compiler.

- Running the compiler with the parameter TEST=YES is used for debugging the compiler and generates NO assembler source module.

- The RTL load module occupies 53 granules on X1215 - disc.
  The module is an overlay load module which is created by the overlay linkage editor.

- Code statements in combination with assembler-comment uses compiler table space. To avoid using compiler table space use RTL-comment (%.....%). This comment is not placed in the assembler source module.

- As the output of the compiler is assembler code, there are restrictions on the following names:

  P, A1, A2,.....A10, A11,.....A15.

- The step condition codes (MAS BCP) issued by the RTL/2 compiler are:

/20	warnings
/40	errors
/60	failures

- If the parameter FORT=YES is specified, the compiler performs FORTRAN interfacing for non-reentrant FORTRAN subroutines. As the reentrant interface is different from the non-reentrant, the user should perform these interfaces himself.

3.2 RTL/2 linkage verifier editor.

- This program is a tool to create one member of cross reference data. It uses one standard member R:XREF, which contains all the runtime system data, and asks for the names of the cross reference data files of the user.

- The limitation on a cross reference data member is 1000 lines.

- The load module LVED occupies 2 granules on disc.

3.3 RTL/2 linkage verifier.

The linkage verifier processes the cross reference module as generated by the RTL/2 Linkage Verifier Editor.
For the relevant error message see P800 RTL/2 user manual.

- The linkage verifier handles 300 entries and 600 externals.

- The load module LVER occupies 8 granules on disc.

3.4 RTL/2 procedures.

The procedures as delivered with the RTL/2 package might need some local adaptions e.g. to names of system userid.

3.5 RTL/2 runtime routines.

3.5.1 Base program.

- The object version is RRBPGA (non-reentrant part) and RRBPGB (reentrant part).
- The runtime stack is 512 bytes.
- The macro (UF) RRBPG allows to generate to a disc-resident or memeory-resident base program.
  The next modules can be generated:

RRBPC	memory resident
RRBPD	disc resident
RRGEL	non-reentrant RTL/2 error handler
UERR	RTL/2 application error handler
RRDATA	RTL/2 SVC DATA declarations.

The RRBPGA and RRBPGB base programs.

There are two base programs RRBPGA and RRBPGB to start the RTL/2 program. RRBPGA is the not reentrant part and RRBPGB the reentrant.

They can be used with the operating systems DOS, DRTM and MAS.

The functions of these programs are:

- Initialization of stack and registers.
- Initialization of SVC DATA items.
- Provide the standard error procedure (RRGEL).

From the base program the user's main procedure RRJOB is called and after the return the exit to the monitor is performed.

Having the source of these programs (included in the RTL/2 package) the user can modify them for his own purposes. E.g. he can increase stack space, add SVC procedures (monitor requests), modify the standard error procedure etc.

Some general modifications possible in RRBPGB.

Addition of SVC DATA.

The user can decide to add some SVC DATA to the base program. E.g. if the user wants to add the SVC DATA brick

```
SVC DATA JOHN;
    INT A,B;
ENDDATA;
```

he should add:

```
        ENTRY   JOHN
SVCLEN  EQU     168+4       previous lenght + length
                            of the data brick in
                            bytes (even boundary)
JOHN    EQU     RRIODT+2
STACK   EQU     JOHN+4
```

If they have to be initialized to the value X, the initalization will look like:

```
        LDKL    A1,X
        ST      A1,JOHN,A13
        ST      A1,JOHN+2,A13
```

Addition of SVC procedures.

SVC procedures (LKM requests) can be added too. E.g. the user can add the LKM number Y with name RRLKMY. Additions to the base program would be:

```
        ENTRY   RRLKMY
RRLKMY  EQU     Y
```

The error procedure RRGEL.

The errorprocedure RRGEL can be generated in two versions, the long version (with PROGSZ EQU 1) and the short version (with PROGSZ EQU 0). The short version prints error number and line number. The long version also prints the contents of the used part of the stack. Both versions print the errormessage (error number and line number) on the operator's typewriter too.

The user can change this module for his own purposes. For example he can print the text of the errormessages instead of the errornumber. However, it is not necessary to change the errormodule RRGEL to do this. It is also possible to do this in the RTL/2 program using the errorlabel ERL and the errornumber ERN in the SVC DATA brick RRERR. But in this case every RTL/2 program, that wants to print the errormessages with their text should have this routine included.

Increase the stack.

It might be necessary to increase the RTL/2 program stack. When the user wants to increase this stack with X words, he should change

```
STLEN   EQU     512
```

into

```
STLEN   EQU     512+2*X
```

The stack, that RTL/2 uses for its control routines (A14-stack) is also used by FORTRAN, when FORTRAN subroutines are included. So it is possible that this stack should be increased too. If this stack has to be increased by X words, SILEN should be changed in the same way and the instruction

```
ADKL    A13,24
```

(immediately before initialization of SVC DATA) has to be changed into

```
ADKL    A13,24+2*X
```

When the required changes are made, the base program can be reassembled and included in the RTL/2 object library.

References:

RTL/2 Run time environment on P800
Publ. nr. 5122 991 28131

P800 DOM RTL/2 User Manual
Publ. nr. 5122 991 28112

3.5.2 Stream I/O.

3.5.2.1 Standard library procedures.

- The following object modules are available
  (for the function description see the relevant manual(s)):

  SLRFR
  SLRFW
  SLRFWF
  SLRIR
  SLRIW
  SLRIWF
  SLRNLS
  SLRRR
  SLRRW
  SLRRWF
  SLRTR
  SLRTW
  RRSIOA
  RRSIOB
  RRBUFF

3.5.2.2 Stream I/O support procedures

- When opening or closing a disc file (DL) the disc files are rewound except in case of file codes /EO and /EE (CCI-command files).

- In case of closing a printer stream (output).
  :EOF is written.

- Default I/O orders are as specified in the user manual:
  standard read (2) and standard write (6)

- Only standard P800 equipment is supported.

- Max. buffer size in RRSIOA/RRSIOB is 256 bytes.

- Max. number of streams is 8.

3.5.3 Control routines

- The following object modules are available.
  For function description see the relevant manual(s):

  RTLR00
  RTLR01
  RTLT03
  RTLR08
  RTLR09
  RTLR11
  RTLR12
  RTLR14
  RTLR16
  RTLR17
  RTLR18
  RTLR23
  RTLR26
  RTLR28

3.5.4 Mathematical routines

The following modules are present as object:

MLRATN - tangent in radians of a real
MLREXP - exponential (e)
MLRLOG - log base e/log base 10
MLRSIN - sin/cos of real
MLRSRT - real square root
MLRFRT - fraction square root.

The function and calling sequence are described in the user manual.

4. MANUALS

Relevant documentation:

RTL/2 language specification	5122 011 28950
RTL/2 system standard	5122 011 28960
RTL/2 standard stream I/O	5122 011 28970
RTL/2 training manual	5122 011 28980
RTL/2 language reference card	5122 011 28990
P800 RTL/2 runtime environment	5122 991 28130
P800 RTL/2 user manual	5122 991 28110
P800 RTL/2 fortran interface capability.	5122 991 28120

Adaptions of these manuals are described in appendix 2.

PC-NR	SUBNR	STATUS	PROGRAM	REL. IN	PROBLEM DESCRIPTION	REL. OUT	CHANGED MODULE(S) OR REMARKS
01379	SART0084	UR-SO	RTL/2	01.02	BASE PROGRAM DOES NOT DETECT ERROR CODE 65 IN CASE OF TOO LESS CORE, LIKE THE OLD VERSION DID.	S02	RRBPGB.
01401		RO-NA	RTL/2	01.02	IF IN RRBPQ, THE LKM GET BUFFER RETURNS THE INFO 'NO BUFFER', ERROR NUMBER 65 IS PRINTED AND THE PROGRAM BRANCHES TO LABEL JOBEXT, HERE THE PROGRAM TRIES TO RELEASE A BUFFER THAT IS NOT YET AVAILABLE/RESERVED.	S02	SOLVED BY PC01379
01406		RO-SO	RTL/2	01.02	BASE PROGRAM RRBPQ: AFTER OBTAINING STACK SPACE FROM THE DYNAMIC AREA, THE STACK END ADDRESS INDICATES A LENGTH OF TWO BYTES LONGER THAN SUGGESTED BY THE EQUIVALENCE STLEN.	S02	SOLVED.
01418	SART0090	UR-SO	RTL/2	01.02	FOR GENERAL RELEASE BUFFER, A CHECK ON A7 SHOULD BE DONE.	S02	RRRBF.
01426		RO-SO	RTL/2	02	DECIMAL FILE CODES NOT ACCEPTED IN OPT-STATEMENT.	S02	OPROC.
01629		RO-SO	RTL/2	01.03	THE STREAM I/O PROCEDURE IREAD DOES NOT GIVE AN ERROR IF AN INTEGER IS TOO LARGE. INSTEAD THE INTEGER IS REDUCED TO A NUMBER MODULO 32K.	S02	IREAD.
01859	TPG00019	RO-NA	RTL/2	01.03	RTL/2 COMPILER GIVES NO INFORMATION UNDER "MODE INFORMATION". "USED" IS ALWAYS 0, EVEN WHEN OVERFLOW RESULTS IN CATASTROPHIC ERROR.	S01.04 S02	SOLVED BY PC01912
01912	NLSE0009	RO-SO	RTL/2	01.03	IN THE TABLE OF "USE OF RESOURCES" AT COMPILATION TIME, ALWAYS A ZERO IS PRINTED IN THE COLUMN "USED" FOR "MODE INFORMATION".	S01.04 S02	STREE.
02633		RO-SO	RTL/2	01.03	COMPILER DOES NOT PRINT VERSION NUMBER WHEN SOURCE LISTING IS SUPPRESSED.	S02	CRCP11.
02749	SART0102	UR-SO	RTL/2	01.03	CLOSING AN OUTPUT FILE (RROPOS) WITH SINGLE BUFFER, RESULTS IN I/O ERROR (ILLEGAL LENGTH C008) LENGTH NOT FILLED IN ECB. OPEN WITH RROPD (DOUBLE BUFFER OUTPUT) WORKS CORRECT.	S02	RRS108.
02778	OVRK0008	RO-SO	RTL/2	01.03	RTL/2 STREAM I/O-TREAD: TREAD TAKES THE SYSTEM DEFAULT TIMEOUT BECAUSE THE ECB IS ONE WORD TOO SHORT.	S02	NG10 RRCIPA RRCIPB.

PC-NR	SUBNR	STATUS	PROGRAM	REL. IN	PROBLEM DESCRIPTION	REL. OUT	CHANGED MODULE(S) OR REMARKS
03392	NLSE0075	RO-NA	RTL/2	01.03	COMPILER GIVES ALWAYS ZERO FOR MODE INFORMATION RESOURCES USED.	S01.04 S02	SOLVED BY PC01912
03696	SART0104	RO-SO	RTL/2	01.04	DELETE OUTPUT FILE CODES (ASM OUTPUT + XREF OUTPUT) WHEN RTL/2 FAILS.	S02	CRCP11.
03697	SART0105	RO-SO	RTL/2	01.04	THE DEFAULT FOR PARAMETER VERS SHOULD BE IN ANY CASE ZERO (VERS=0), ALTHOUGH E.O. DAD SPECIFIED.	S02	UPROC.
03828		RO-NA	RTL/2	01.03	NO CHECK ON OVERFLOW IN "IREAD()".	S02	SOLVED BY PC01629
04092	NLMS0009	RO-SO	RTL/2	01.03	STREAM I/O IS VERY INEFFICIENT, DUE TO THE FACT THAT EVERY CALL FOR IN OR OUT LEADS TO A LKM WAIT REQUEST.	S02	RRSIO.
04122	NLMS0016	RO-SO	RTL/2	01.03	WHEN RRCLOSO IS CALLED, IT SHOULD: A. NOT GIVE A SENDBUFF CALL IF THE CURRENT BUFFER IS EMPTY. B. WAIT FOR COMPLETION OF OUTSTANDING I/O BEFORE GIVING A WRITE EOF LKM.	S02	RRSIO.
04295	NLMS0020	UR-NA	RTL/2	01.04	COMPILER STILL GENERATES ERRONEOUS "ML 2,/FFF2,A12" AT RETURN STATEMENT.	S02	SOLVED BY PC04303
04303	NLSE0129	UR-SO	RTL/2	01.04	BEFORE LEAVING A PROCEDURE, THE CORRECT VALUE TO BE RETURNED (REG. A1), IS DESTROYED BY A SUPERFLUOUS INSTRUCTION "ML 1,/FFF0,A12".	S02	BP10.
04415		UR-NA	RTL/2	01.04	COMPILER REPORTS "PROGRAM TOO LONG" WHEN TRANSLATING.	S02	SOLVED BY PC04303
04562	SASW0045	RO-SO	RTL/2	01.04	THE COMPILER GENERATES AN OUTPUT LINE OVER 80 CHARACTERS. THE PIECE OF CODING IS IN A CODE SECTION.	S02	BP70.

END OF REPORT

APPENDIX 2

Manual adaptions

General

Next manuals concerning the use of RTL/2 on a P800 are submitted to changes. Changes caused by this release but also by updates of release 1. All changes are summarized in this release bulletin.
It concerns the manuals:

- Runtime environment on P800
  (publ.nr. 5122 991 28131)
- DOM RTL/2 user manual
  (publ.nr. 5122 991 28111)
- FORTRAN interface capability for P800 DOS
  (publ.nr. 5122 991 28122)

As RTL/2 release 2 covers both the use of RTL/2 under DOM and MAS control, all references to DOM also count for MAS.

2.1.3. (page 2/1)
Appendix II must be appendix III.

DOM RTL/2 user manual

Title must be read as
RTL/2 user manual

2.1.4. (page 2/6)

Add the next new front-end failure messages (catastrophic errors):

25	Table space exceeded
26	Invalid table switch
27	D specified greater than I
28	Too many generated labels

The maximum number of front-end error messages, which can be issued in one compiler run is 50. When this amount is exceeded, the message
** TOO MANY FAILURES **
will be given.

2.2.5. (page 2/16)

Add the next back-end failure message (warning message):

204	ext/ent name more than 6 characters - truncated

The maximum number of back-end error messages, which can be issued during one compiler run is 25. See also remark for front-end error messages.

2.3.2. (page 2/17)

Replace the whole paragraph by next information.

Source Module Size Constraints

The resource limits are not absolute anymore, but are flexible dependant on the user requirements (Flexi-Interface).

The space available for compiler interface tables is by default divided into proportions for the various resources as below:

ratio	resource	unit costs(bytes)	
1	Array bound pool(A)	2	
8	Identifiers/Variables(I)	16	*
5/10	Gen. labels/Mode info(G)	2/1	*
30	Name characters(N)	1	
20	Constant pool(C)	1	
60	String pool(S)	1	
5	At brick lvl/Id.names(D)	2	*

* Space used by the Front-end is re-used by the Back-end.

When a module under compilation would overflow the limits of one or more of the resources (but not all of them), the user can reserve specific amounts of these resources, after which the remaining space is divided in the standard ratio for the other resources. This can be done in the OPT-card when calling the

compiler, using the table parameters TA1 to TA7. The particular resources are identified by the letter between the parentheses in the table above.

2.4.1. (page 2/18)

Replace the whole paragraph by next information.

The command format to run the compiler under DOS(release 8.1) is:

RTL /S|<file name>[,NL]

followed by an option card:

OPT␣[LIST= YES | NO] [,PARLIST=YES | NO] [,MODE=SYST | APP] [,FORT=YES | NO] [,SYMB=YES | NO] [,RES= YES | NO][,TEST=YES | NO] [,CARDFMT=(n:m) | (1:72)] [,XREF=<filecode>|<filename> | /D8][,OP1=option][,OP2=option].....[,OP5=option] [,TA1=table spec] [,TA2=table spec].....[TA7=table spec] [,STACK= 512] [,OBJLIST=YES | NO]

Under MAS (release 6.3) the compiler has to be run by the command RTL, without parameters. The option card is extended with:

PROG=<filecode>|<filename> | /E1 [,DAD=/Fx | <current dad>] [,USID=<userid> | <current user>] [,VERS=<n> | 0]

Continuation of the option card is done by a semicolon (;) after a parameter-specification.

2.4.2. (page 2/18)

Replace the whole paragraph by the next information.

Parameters

Running the compiler under DOS, the specification of /S or filename tells the compiler, that he can find the RTL source on the temporary file /S or on a source file of the current user.

If NL is specified, the assembler listing is suppressed. The assembler processor is called automatically under DOS if no errors occurred during the RTL compilation. Under MAS the user must call the assembler processor himself.

If LIST=YES is specified, the RTL source listing will be output on filecode /02. LIST=NO means no listing.

The parameter list may be output with also the default values for the parameters not specified, by specifying PARLIST=YES. If NO, this list will not be output.

MODE is set to SYST, if the source is to be compiled in systems mode. If the source is to be compiled in applications mode, MODE is set to APP.

If FORT=YES is specified, FORTRAN interfacing is performed, to make it possible to use FORTRAN subroutines.

SYMB=YES is specified, when additional information has to be output. This information consists of a concordance table showing the line numbers on which each variable occurs.

If RES=YES is specified, a table showing the use of resources and their limits and a table with the brick sizes will be output.

Additional debug information may be printed by specifying TEST=YES. In this case no assembler source is produced.

The user can change the card format by specifying CARDFMT=(n:m); n is the first relevant column and m is the last one (0<n<m<=80).

XREF specifies the filecode or the name of the datafile, where the cross reference data generated by the compiler will be stored. Is fixed to /D8 for DOS.

By means of OP1 to OP5, a source statement of the form:

```
OPTION (n) opitem,opitem,....
```

may be modified by specifying the OPTION number in brackets followed by opitems separated by semicolons. The number in the OPn parameter has no significance.

e.g. a source statement

```
OPTION (2) NS;
```

is modified by specifying

```
OP1 = (2) NW;TR
```

to `OPTION (2) NW,TR;`

The source listing, however, is not modified.

By means of TA1 to TA7 resource table sizes may be modified. The letter identifying the selected resource must be followed by a decimal number specifying the number of units for that resource. the number in the TAn parameter has no significance.
e.g. the parameters

```
TA1=I300,TA2=N4000
```

allow using 300 identifiers and 4000 name characters in the module to be compiled.

NOTE: When using "table parameters", the user should not attempt to set the value for D greater than that for I and take care that this does not happen when the space remaining after some specification is divided among the resources not specified. Neither the user should attempt to specify more space then available.

STACK=512 is not used.

OBJLIST=YES and OBJLIST=NO are not used.

The additional parameters for MAS:

The source to be compiled can be found on the file with specified filecode or filename using the keyword PROG.

DAD specifies the dad, where the source file is catalogued.

USID specifies the user in which directory the source file can be found.

VERS specifies the version number of the catalogued source file.

2.4.3. (page 2/19)

Replace the whole paragraph by the next information.

Filecodes used by the compiler

The compiler uses the filecodes /D4 (=/S), /D7 (compiler scratch file), /D8 and /D9. After the compilation /D4 (DOS) or /D9 (MAS) contains the generated assembler source. The cross reference information is default put on /D8 and /D7 is scratched. If the compilation was not succesful, /D4 still contains the RTL/2 source and /D9 is scratched.

2.4.4. (page 2/20)

The statement on line 0:

␣IDENT␣<name of up to 6 characters>␣<comment>

is copied to the assembler output file in the form:

\IDENT\<name>\<comment>

2.4.5. (page 2/20)

Replace the whole paragraph by the next information.

Examples

To compile PROGRM with a listing, in applications mode, use:

```
DOS:RTL PROGRM,NL
   RTL: OPT LIST=NO,....


MAS: RTL
   RTL: OPT PROG=PROGRM,LIST=NO,....
     ASM
   ASM: OPT PROG=/D9,LIST=NO
```

2.4.6. (page 2/21)

Filecode /BE must be filecode /D7.

Report format

The compiler, during the compilation of programs, outputs a report to filecode /02. This report in the full form contains:

(a) Compiler identity.

(b) The source program with line numbers, if LIST=YES.

(c) FORTRAN subprogram report, if FORT=YES.

(d) A warning, if compiled as systems module (MODE=SYST).

(e) Compiler error diagnostics and warning messages in the format:

ON LINE AT SYMBOL LAST IDENTIFIER NO. ERROR MESSAGE

Front-end and Back-end errors are separated.

(f) Resource tabel, if RES=YES.

(g) Concordance tabel, if SYMB=YES.

(h) List of brick names and sizes, if RES=YES.

(i) The message "COMPILATION FAILS" or "COMPILATION OK".

2.2. (page 2/1)

Replace FOR=Y[ES] by
FORT=Y ES | N O

3.1. (page 3/1)

The actual FORTRAN situation (Extended FORTRAN) is described in the next manuals:

P800 FORTRAN Reference Manual, 5122 991 11405
Supplement 5122 991 31071

The supplement describes the Extended FORTRAN Compiler for MAS, but the information contained in the supplement applies to the DOS compiler, with the following exceptions:

- the processor call under DOS is
  HSF /S
  or
  HSF <program name>
  in which /S is the current temporary source file, and <program name> the name of a catalogued program
- in the OPT control statement, the parameters PROG=,USID=,DAD=,VERS= does not apply to the DOS Extended FORTRAN compiler
- under DOS, the default value for the FPP= parameter in the OPT statement is FPP=S.

The extended FORTRAN has to be used with the specification OPTM=YES.

3.3.3. (page 3/3)

The address of the parameter list is contained in register A13.

SECTION 7. Example (page 7/3)

This example must be completed with the call to RRINIO and the procedure body of RRJOB must be indicated as a block. The example is changed to:

```
ENT PROC RRJOB();
RRINIO();
BLOCK
   <body of RRJOB>
ENDBLOCK;
ENDPROC;
```