

Contents

[Preface](#)

[Chapter 1 Overview](#)

- [1.1 Outline of PowerFORM RTS Functions](#)
- [1.2 Development environment of PowerFORM RTS](#)
- [1.3 How to Use PowerFORM RTS](#)

[Chapter 2 Operating the PowerFORM RTS](#)

- [2.1 Developing Application Programs](#)
- [2.2 Executing Application Programs](#)

[Chapter 3 Basic Functions](#)

- [3.1 Using PowerFORM RTS](#)
- [3.2 Specifying Fields](#)
- [3.3 Printing Forms](#)
- [3.4 Processing Depending on Print Paper](#)

[Chapter 4 Extended Functions](#)

- [4.1 Form Overlay](#)
- [4.2 Built-in Media](#)
- [4.3 Bar code printing](#)
- [4.4 OCR-B font printing](#)

[Chapter 5 Environment Setting Files](#)

[Chapter 6 Program Interfaces](#)

[Appendix A Trouble Analysis and Recovery](#)

[Appendix B Special Processing](#)

Windows95 and WindowsNT is a trademark of Microsoft Corporation in the United States and other countries.

FUJITSU LIMITED

Preface

PowerFORM RTS V1.0 is a service program that supports input-output processing for a printer running under Microsoft(R) Windows(R) 95 or Microsoft(R) WindowsNT(R) Workstation operating system Version 3.51 or Microsoft(R) WindowsNT(R) Workstation operating system Version 4.0 or Microsoft(R) WindowsNT(R) Server Network operating system Version 3.51, or Microsoft(R) WindowsNT(R) Server Network operating system Version 4.0.

The PowerFORM RTS Users Guide describes the functions and usage of PowerFORM RTS. This Users Guide is intended for creators of output programs that use form descriptors.

Related manuals

While reading this handbook, you may also wish to use the following manuals:

- Various manuals for Windows(R)95 or WindowsNT(R)

To create form descriptors, use the following manual:

- PowerFORM Users Guide

To create application programs in COBOL, use the following manual:

- COBOL85 Users Guide

MS-DOS, Microsoft, Windows95, and WindowsNT are registered trademark of Microsoft Corporation.

© 1997 Fujitsu Limited. All Rights Reserved.

Chapter 1 Overview

This chapter explains the basics of PowerFORM RTS.

1.1 Outline of PowerFORM RTS Functions

1.1.1 What is PowerFORM RTS?

1.1.2 PowerFORM RTS Features

1.1.3 Using PowerFORM RTS

1.2 Development environment of PowerFORM RTS

1.3 How to Use PowerFORM RTS

1.3.1 What are records?

1.3.2 Form descriptor

1.3.3 Printer information files

1.1 Outline of PowerFORM RTS Functions

This section gives an outline of the PowerFORM RTS functions.

[1.1.1 What is PowerFORM RTS?](#)

[1.1.2 PowerFORM RTS Features](#)

[1.1.3 Using PowerFORM RTS](#)

1.1.1 What is PowerFORM RTS?

The message editing facile tool (PowerFORM RTS) is the service library called when an application program outputs data using a printer.

When PowerFORM RTS is used, the application program need not consider the complicated control codes of the printer. That is, the application program need only consider the output data itself.

The fixed data part of the output data need not be included in the application program. You need define the fixed data only in form descriptors. Then, PowerFORM RTS automatically prints the fixed data for you. In addition to the fixed data, form descriptors can also be used to define output information such as output data locations, length, color, and ruled lines. In this case, such output information need not be considered at all in the application program.

Through PowerFORM RTS, application programs using forms are easily created.

Figure 1.1.1 shows the PowerFORM RTS operating environment.

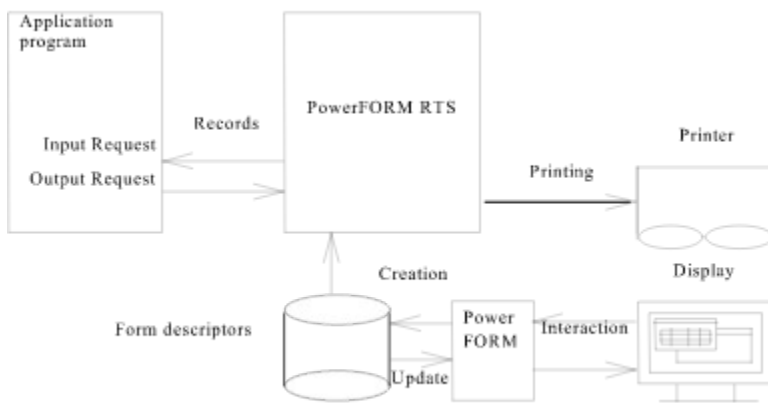


Figure 1.1.1 PowerFORM RTS Operation

1.1.2 PowerFORM RTS Features

- **PowerFORM RTS uses form descriptors with an application interface common to two or more systems.**
Form descriptors used by PowerFORM RTS have specifications common to two or more systems. Therefore, a system can use form descriptors created by other systems. The functions supported by PowerFORM RTS are based on industry standard specifications.
- **Arbitrary fields to be processed**
Output requests for arbitrary field groups can be issued to PowerFORM RTS.
- **Numeric data edit format**
PowerFORM RTS supports 50 or more numeric edit patterns.
- **Dynamic field attributes**
The character attributes of field such as underline, bold and italic in form descriptors can be changed from within the application programs.
- **Form overlay**
Overlay data common to various systems can be printed in the same way as form descriptors.
- **Medium data**
Bit map data and can be printed.

1.1.3 Using PowerFORM RTS

Complete the following steps before using the service library.:

- Design the form layouts.
- Create the printer information files.
- Create the application programs.
- Execute the application programs.

To use PowerFORM RTS, create a form descriptor using PowerFORM. PowerFORM is a tool for creating or updating a descriptor during interaction with the display screen.

In addition to the above descriptors, prepare a printer information file. Use an editor to define printer operation methods.

The application programs to be used with PowerFORM RTS must be created using COBOL program language.

After the above preparations have been completed, execute the application programs.

1.2 Development environment of PowerFORM RTS

To develop application programs in COBOL, use the following product:

- COBOL85 V3.0 or later

1.3 How to Use PowerFORM RTS

This section explains how to use PowerFORM RTS.

[1.3.1 What are records?](#)

[1.3.2 Form descriptor](#)

[1.3.3 Printer information files](#)

1.3.1 What are records?

PowerFORM RTS uses records to communicate (send and receive data) with application programs. Records contain the areas for all the fields defined in a form descriptor with the exception of static fields. The record parts assigned to these fields are record data fields. Records have an area where an application program can change the attributes of each field. PowerFORM RTS then posts the status of each changed field. This area is called a field control field.

All the record data fields and field control fields are generically referred to as record data areas.

1) Data storage format

The format for storing data in a record data field depends on the data type.

The following explains the types of data that can be processed in respective fields:

- **Numeric fields**

External decimal numbers are stored. For details, see 2), "Format of numeric fields."

- **Alphanumeric fields**

ASCII codes are stored.

2) Format of numeric fields

A numeric field is represented by an external decimal integer.

Table 1.3.1 lists the data format of external decimal integers.

		Table 1.3.1 Data format of external decimal integers									
		0	1	2	3	4	5	6	7	8	9
None	Character representation	0	1	2	3	4	5	6	7	8	9
	Hexadecimal representation	30	31	32	33	34	35	36	37	38	39
Positive	Character representation	@	A	B	C	D	E	F	G	H	I
	Hexadecimal representation	40	41	42	43	44	45	46	47	48	49
Negative	Character representation	P	Q	R	S	T	U	V	W	X	Y
	Hexadecimal representation	50	51	52	53	54	55	56	57	58	59

1.3.2 Form descriptor

Form descriptors must be created using PowerFORM.

To use PowerFORM RTS, the directory of the printer information file that contains the above descriptors must be specified. The extension of the descriptor file names and the number of registered descriptors must be specified.

1.3.3 Printer information files

A printer information file is prepared for each window to be opened. It is created using any text editor. The file name consists of up to no more than eight alphanumeric characters and is in text file format.

To open a window, the corresponding printer information file must be specified using its full path name. If only the file name is specified, the directory of the printer information file must be stated in the environment variable, MEFTDIR. If the environment variable MEFTDIR is not defined or a correct directory name is not specified, PowerFORM RTS searches the current directory.

If a printer information file is not found in the current directory, it is assumed that the file was omitted. PowerFORM RTS then uses the default values. For an explanation of the information that can be specified in a printer information file, see [Printer Information File](#).

Chapter 2 Operating the PowerFORM RTS

This chapter explains the preparations before using PowerFORM RTS and explains the development and execution of application programs. For an explanation of installing the PowerFORM RTS, refer to README.TXT stored in the master floppy disk of PowerFORM RTS.

[2.1 Developing Application Programs](#)

[2.2 Executing Application Programs](#)

2.1 Developing Application Programs

This section explains how to develop application programs in the COBOL85 language.

a) Coding the application programs

An application program must be coded as a output using print file with FORMAT clause or presentation file.

b) Compilation and linkage

The compile option FORMLIB must be specified, and code the form descriptor storage directory by path name.

2.2 Executing Application Programs

Setting the environment variables

To execute an application program that uses PowerFORM RTS, set the following environment variables:

1) **PATH**

Sets the PowerFORM RTS-installed directory in the environment variable PATH.

2) **MEFTDIR**

Specifies the storage directory for the printer information files. Two or more directories can be specified by delimiting them with a semicolon. The following shows an example of specifying C:\DIR1 and C:\DIR2:

```
SET MEFTDIR=C:\DIR1;C:\DIR2
```

Samples of a printer information file are stored, with file names MEFTPRC, in the PowerFORM RTS install directory. The user can copy these files into an arbitrary directory. By changing these copied files, the user can create a printer information file appropriate for the actual execution conditions.

Chapter 3 Basic Functions

The application program uses the form descriptor to output data to the printer. Three PowerFORM RTS functions are used for printing. Finally, the physical aspects (the form and the type of paper) must be defined. This chapter covers these basic printing operations:

3.1 Using PowerFORM RTS

3.1.1 Open

3.1.2 Output

3.1.3 Close

3.2 Specifying fields

3.3 Printing Forms

3.3.1 Partition output

3.3.2 Output field decorations

3.3.3 Printing ruled lines and shadings

3.4 Processing Depending on Print Paper

3.4.1 Printing on continuous forms

3.4.2 Printing on cut forms

3.1 Using PowerFORM RTS

This section uses the results of printing the sales slip shown in Figure 3.1 to illustrate the basic functions of PowerFORM RTS.

Sales slip				
Customer code 0201			Slip No. 0001	
Customer name A				
store Delivery date 96. 4. 1				
No.	Item code	Item	Quantity	Amount
1	0005	Rince	4	2,000
2	0006	Shampoo	4	2,000
3	0020	Conditioner	2	2,000
4				
5				
			Totals 20,000	

Figure 3.1.1 Printing a sales slip

Figure 3.1.2 is a program that prints a sales slip such as that shown in Figure 3.1.1.

~~ an omission of a middle part ~~

```
INPUT-OUTPUT SECTION.  
FILE-CONTROL.  
SELECT print file ASSIGN TO PRTFILE  
ORGANIZATION IS SEQUENTIAL  
ACCESS MODE IS SEQUENTIAL  
FORMAT IS PRT-FORMAT  
GROUP IS PRT-GROUP  
.
```

~~ an omission of a middle part ~~

```
DATA DIVISION.  
FILE SECTION.  
FD print file.  
COPY BILL OF XMDLIB.
```

~~ an omission of a middle part ~~

```
PROCEDURE DIVISION.  
OPEN OUTPUT print file  
.  
INITIALIZE BILL.  
MOVE BILL TO PRT-FORMAT.  
MOVE SCREEN TO PRT-GROUP.  
WRITE BILL.  
  
CLOSE print file.
```

Figure 3.1.2 Sample Program: Printing a Sales Slip

When the application program uses the form descriptor, the following PowerFORM RTS functions are used:

[3.1.1 Open](#)

[3.1.2 Output](#)

[3.1.3 Close](#)

The program shown in Figure 3.1.2 is used to explain how to output the form shown in Figure 3.2.1 below.

3.1.1 Open

The open function declares the start of output to the printer and specifies the printer information file. To output data to the printer, always execute the open function before using other functions. In Figure 3.1.2, the following statement opens the printer:

OPEN OUTPUT *print file*

File identifier specified in ASSIGN clause :Printer information file name

FILE STATUS clause :Return code storage area at abnormal termination of open process

After the operation is complete, be sure to close the file opened by the application program. If it is not closed, data is not printed and internal resources can remain allocated.

3.1.2 Output

Output sends fixed data in the form descriptor, data provided by the application program, and overlay data to the printer. Actual printing is done during the close process.

In Figure 3.1.2, the following statement outputs data to the printer:

WRITE BILL

.MOVE SCREEN TO PRT-GROUP.	: Name of field group containing all fields in the descriptor
MOVE BILL TO PRT-FORMAT.	:Specification of form descriptor BILL
FILE STATUS clause	: Return code storage area at abnormal termination

If close is not executed and the data output continues, the temporary space on the disk may become insufficient, causing a system error. Execute the printing process by using the close function or increase the temporary space if necessary. Create the temporary space by setting environment variable TEMP or TMP. Refer to Windows(R)95 or WindowsNT(R) manuals for details.

3.1.3 Close

Close declares the end of output to the printer and disables any PowerFORM RTS functions except Open. After the close function is executed, printing starts.

In Figure 3.1.2, the following statement closes the printer:

CLOSE *print file*

FILE STATUS clause :Return code storage area at abnormal termination of close process

3.2 Specifying Fields

Output is executed for an field.

When the form descriptor is created, add the partition names to the partition.

Specify the partition name in GROUP clause, and execute WRITE sentence.

3.3 Printing Forms

This section explains the output process when PowerFORM RTS is used to print forms.

[3.3.1 Partition output](#)

[3.3.2 Output field decorations](#)

[3.3.3 Printing ruled lines and shadings](#)

3.3.1 Partition output

In partition output, the position of the field to be printed is not specified when creating the descriptor. Instead, the position of the field to be printed is dynamically specified by the program that prints the field. The position of the immediately preceding printed field is used as a reference. The print position is dynamically specified by feeding any number of lines before or after an field group is printed.

Partition field groups include the floating partition field group and fixed partition field group. The print position of only the floating partition field group can be dynamically specified. The position of the fixed partition field group is specified when creating the descriptor. The field group is always printed at the position with the left top end of the form used as an origin.

When partition field group is printed, execute the form feed function or line feed function as required. If the descriptor is changed, the system executes form feed processing according to the vertical size of the form.

If a fixed partition before the position of the immediately preceding output partition is output, form is fed. The partition is then printed at the specified position.

When a fixed partition field group is printed, the specified number of lines to be fed becomes valid for printing before line feed only.

The following print methods are valid for partition output:

- Printing before line feed
- Printing after line feed

1) Printing before line feed

Printing before line feed prints a partition field group, then feeds any number of lines. The number of lines is specified during output. If 0 is specified, no lines are fed. If 0 is specified for the number of lines and a floating partition field group is continuously printed using partition output, the first line of the field group overlaps with the last line of the previously printed field group. To avoid this processing error, specify other than 0 for the number of lines or leave the last or first line of the partition field group blank when creating the descriptor.

Figure 3.3.2 shows an example of printing before line feed.



The above partition field group is used to execute printing after line feed twice consecutively. Four lines are fed before printing.

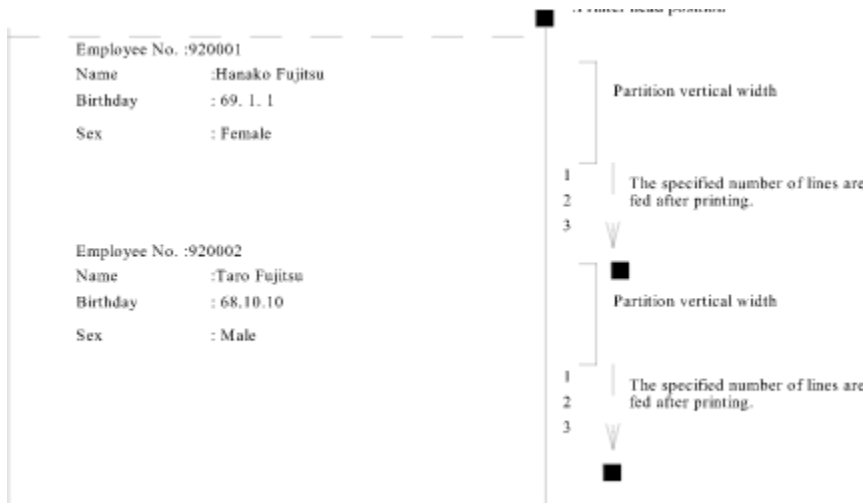


Figure 3.3.1 Example of printing before line feed

```
MOVE descriptor name TO PRT-FORMAT.
MOVE partition name TO PRT-GROUP.
```

~~ an omission of a middle part ~~

```
WRITE record BEFOR ADVANCING 4
```

~~ an omission of a middle part ~~

```
WRITE record BEFOR ADVANCING 4
```

Figure 3.3.2 Sample Program: Printing before line feed

2) Printing after line feed

Printing after line feed feeds any number of lines, then prints a partition field group.

The number of lines is specified during output. If 0 is specified, no lines are fed. If 0 is specified for the number of lines and a floating partition field group is continuously printed using partition output, the first line of the field group overlaps with the last line of the previously printed field group. To avoid this output error, specify a value other than 0 for the number of lines or leave the last or first line of the partition field group blank when creating the descriptor.

Printing after line feed is only valid for a floating partition field group. When a fixed partition field group is printed, the number of feed lines specified by the application program is ignored. The fixed partition field group is always printed at the position specified in the descriptor.

Figure 3.3.3 shows an example of printing after line feed.



The above partition field group is used to execute printing after line feed twice consecutively. Four lines are fed before printing.

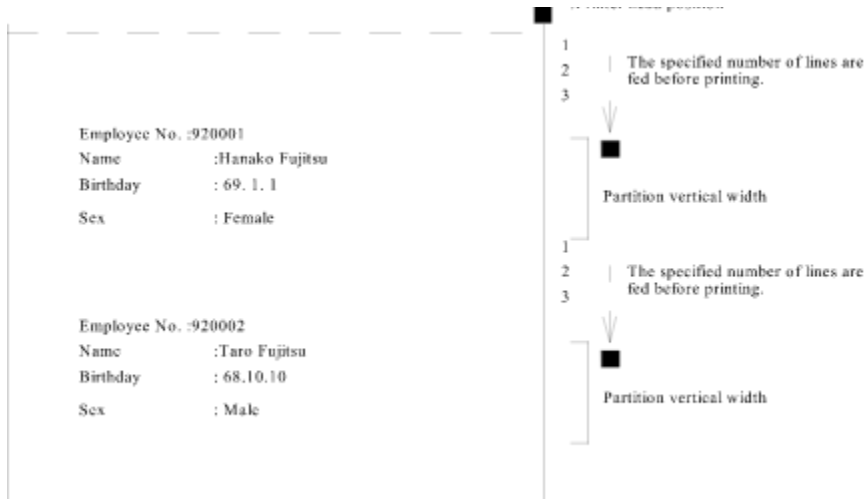


Figure 3.3.3 Example of printing after line feed

```
MOVE descriptor name TO PRT-FORMAT.
MOVE partition name TO PRT-GROUP.
```

~~ an omission of a middle part ~~

```
WRITE record AFTER ADVANCING 4
```

~~ an omission of a middle part ~~

```
WRITE record AFTER ADVANCING 4
```

Figure 3.3.4 Sample Program: Printing after line feed

3.3.2 Output field decorations

The application program can edit and print data set in the record data area for each field.

Field decorations are as follows:

These decorations are explained below.

1) Character color output

The color of the character to be printed is specified for each field when creating the descriptor. [Setting the field color attribute](#) enables the color specified when creating the descriptor to be changed dynamically and disables printing of data for the field to be output.

One of the following seven colors can be specified for the character color:

- Black
- Red
- Green
- Yellow
- Blue
- Pink
- Cyan

2) Character highlight

The application program can print decorating data by using [Setting the field highlight attribute](#).

One of the following seven can be specified for the decoration:

- Bold
- Italic
- Bold-Italic
- Underline (H)
- Bold with underline (J)
- Italic with underline (K)
- Bold-Italic with underline (L)

Underline is drawn for the character strings.

Moreover specify the parenthesized value for the field control to specify the underlined character.

3) Font specification

Specify the font on the entire form using [PRTFONT\(Output font name\)](#) and [FONTFACE \(Font face name\)](#) in the printer information file.

When the form descriptor is created, a font number from 0 to 255 can be specified for each field. Specify the font with the value from [FONT000 to FONT255 \(Font number specification\)](#) corresponding to the specified number.

4) Numeric editing

Numeric fields can be printed with data edited.

The edit format is specified when creating the descriptor.

See Table 3.3.1, "Edit Format and Example of Edit and Output" for an example of the print results when the edit format is specified.

Table 3.3.1 lists examples of editing formats and editing output.

	Data value	123456	001234	-00123.4	00000.0	-0.34
Number of all digits	6	6	6	6	6	2
Number of decimal digits	0	0	0	1	1	2
Editing format						
999,999,999,999,999,999	123,456	001,234	00,123.4	00,000.0	.34	
ZZZ,ZZZ,ZZZ,ZZZ,ZZZ,ZZ9	123,456	1,234	123.4	0.0	.34	
999999999999999999	123456	001234	00123.4	00000.0	.34	

ZZZZZZZZZZZZZZZZZZZZ	123456	1234	123.4	0.0	.34
,,***,***,***,***9	123,456	**1,234	***123.4	*****0.0	.34
\$\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$9	\$123,456	\$1,234	\$123.4	\$0.0	\$.34
999,999,999,999,999,999CR	123,456	001,234	00,123.4CR	00,000.0	.34CR
ZZZ,ZZZ,ZZZ,ZZZ,ZZZ,ZZ9CR	123,456	1,234	123.4CR	0.0	.34CR
9999999999999999999CR	123456	001234	00123.4CR	00000.0	.34CR
ZZZZZZZZZZZZZZZZZZZZ9CR	123456	1234	123.4CR	0.0	.34CR
,,***,***,***,***9CR	123,456	**1,234	***123.4CR	*****0.0	.34CR
\$\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$9CR	\$123,456	\$1,234	\$123.4CR	\$0.0	\$.34CR
999,999,999,999,999,999-	123,456	001,234	00,123.4-	00,000.0	.34-
ZZZ,ZZZ,ZZZ,ZZZ,ZZZ,ZZ9-	123,456	1,234	123.4-	0.0	.34-
9999999999999999999-	123456	001234	00123.4-	00000.0	.34-
ZZZZZZZZZZZZZZZZZZZZ9-	123456	1234	123.4-	0.0	.34-
,,***,***,***,***9-	123,456	**1,234	***123.4-	*****0.0	.34-
\$\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$9-	\$123,456	\$1,234	\$123.4-	\$0.0	\$.34-
-999,999,999,999,999,999	123,456	001,234	-00,123.4	00,000.0	-.34
-ZZZ,ZZZ,ZZZ,ZZZ,ZZZ,ZZ9	123,456	1,234	- 123.4	0.0	-.34
-9999999999999999999	123456	001234	-00123.4	00000.0	-.34
-ZZZZZZZZZZZZZZZZZZZZ9	123456	1234	- 123.4	0.0	-.34
-***,***,***,***,***,***9	123,456	**1,234	-***123.4	*****0.0	-.34
-\$\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$9	\$123,456	\$1,234	- \$123.4	\$0.0	-\$.34
-----,---,---,---,---,---9	123,456	1,234	-123.4	0.0	-.34
-----9	123456	1234	-123.4	0.0	-.34
ZZZZZZZZZZZZZZZZZZZZ	123456	1234	123.4	(*1)	.34
2-digit delimited editing Z9.Z9.Z9	12.34.56	0.12.34	(*2)	0.0.0	(*2)
2-digit delimited editing 99.99.99	12.34.56	00.12.34		00.00.00	
ZZZZZZZZZZZZZZZZZZZZ-	123456	1234	123.4-	(*1)	.34-
-ZZZZZZZZZZZZZZZZZZZZ	123456	1234	- 123.4	-.34	
999,999,999,999,999,999DB	123,456	001,234	00,123.4DB	00,000.0	.34DB
ZZZ,ZZZ,ZZZ,ZZZ,ZZZ,ZZ9DB	123,456	1,234	123.4DB	0.0	.34DB
9999999999999999999DB	123456	001234	00123.4DB	00000.0	.34DB
ZZZZZZZZZZZZZZZZZZZZ9DB	123456	1234	123.4DB	0.0	.34DB
,,***,***,***,***9DB	123,456	**1,234	***123.4DB	*****0.0	.34DB
\$\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$9DB	\$123,456	\$1,234	\$123.4DB	\$0.0	\$.34DB
999,999,999,999,999,999+	123,456+	001,234+	00,123.4-	00,000.0+	.34-
ZZZ,ZZZ,ZZZ,ZZZ,ZZZ,ZZ9+	123,456+	1,234+	123.4-	0.0+	.34-
9999999999999999999+	123456+	001234+	00123.4-	00000.0+	.34-
ZZZZZZZZZZZZZZZZZZZZ9+	123456+	1234+	123.4-	0.0+	.34-
,,***,***,***,***9+	123,456+	**1,234+	***123.4-	*****0.0+	.34-
\$\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$9+	\$123,456+	\$1,234+	\$123.4-	\$0.0+	\$.34-
+999,999,999,999,999,999	+123,456	+001,234	-00,123.4	+00,000.0	-.34
+ZZZ,ZZZ,ZZZ,ZZZ,ZZZ,ZZ9	+123,456	+ 1,234	- 123.4	+ 0.0	-.34
+9999999999999999999	+123456	+001234	-00123.4	+00000.0	-.34
+ZZZZZZZZZZZZZZZZZZZZ9	+123456	+ 1234	- 123.4	+ 0.0	-.34
+***,***,***,***,***,***9	+123,456	+**1,234	-***123.4	+*****0.0	-.34
+\$\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$,\$\$\$9	+\$123,456	+ \$1,234	- \$123.4	+ \$0.0	-\$.34
++++,+++,\$\$\$,\$\$\$,\$\$\$,\$\$\$9	+123,456	+1,234	-123.4	+0.0	-.34
+++++9	+123456	+1234	-123.4	+0.0	-.34
ZZZZZZZZZZZZZZZZZZZZ+	123456+	1234+	123.4-	(*1)	.34-
+ZZZZZZZZZZZZZZZZZZZZ	+ 123456	+ 1234	- 123.4	-.34	

"_": 1-byte blank

"_": 2-byte blanks

*1 Zero is suppressed, and no data is displayed.

*2 The editing result is undefined.

Table 3.3.1 Examples of editing formats and editing output

5) Character expansion editing and output

The character extended editing display function repeatedly displays the first character of a record data field up to a certain field length. Use Field display attribute setting at output to execute character extension editing. This editing

function is valid only when an attribute of an input-output or output field is numeric. Figure 3.3.5 shows an output example.

```
Setting the contents of the item output for a certain item
item, ? ? ? ? ? ? ? ? : Arbitrary data

Print result
*****
item field length
```

Figure 3.3.5 Character extension editing

6) Multipoint and multipitch
Specify an optional character size and pitch for output fields.

3.3.3 Printing ruled lines and shadings

The application program can overprint ruled lines and shading on printing data.

Ruled lines and shading are specified when creating the descriptor. The ruled lines and shadows are always printed at the fixed position.

Specify the following information fields when creating the descriptor:

1) Line information

Line information is as follows:

- Ruled line type

Ruled line type - a combination of line type and line width:

Line type: Solid line, dotted line, chain line, and dashed line

Line width: Fine line and bold line

- Ruled line color available are:

Black, red, green, yellow, blue, pink, or cyan

2) Shading information

Shading information is as follows:

- Shading pattern

Specify one of the 10 types of shading patterns.

- Shading color available are:

Black, red, green, yellow, blue, pink, or cyan

3.4 Processing Depending on Print Paper

This section explains how to establish the print environment.

The print environment must be set up before running the print program. The type of paper used for printing determines how the print environment is set up. The user selects the processing to suit the paper type.

[3.4.1 Printing on continuous forms](#)

[3.4.2 Printing on cut forms](#)

3.4.1 Printing on continuous forms

This section explains how to print on continuous forms supplied by the paper feed tractor.

- Specify "continuous forms" for [FORMKIND \(form type\)](#) in the printer information file.
- Prepare the serial printer for printing continuous forms.
- Specify "user-defined size" for the printer driver form size. If "user-defined size" is not supported, the paper fed by the paper size specified on the control panel.

* In case of WindowsNT system, "user-defined size" for the form size cannot be specified. Specify the custom form or continuous forms for the form size.

The size specified for the descriptor size is used for printing.

If "printing on continuous forms by setting on control panel" is specified in [PHYPAGE \(setting form size when continuous forms are printed\)](#) in the printer information file, the physical form size specified on control panel is used. Setting of this case is as follows:

- Specify "continuous forms" for [FORMKIND \(form type\)](#) in the printer information file.
- Prepare the serial printer for printing continuous forms.
- Specify "printing on continuous forms by setting on control panel" for [PHYPAGE \(setting form size when continuous forms are printed\)](#) in the printer information file.
- Specify "printing form size on control panel" for [MAPINFO FORMSIZE\(form size specification\)](#) in the printer information file.

3.4.2 Printing on cut forms

The printing of cut forms depends on how the paper is supplied in the following ways:

- Specify "cut forms" for [FORMKIND \(form type\)](#) and "hopper 1", "hopper 2", "hopper 3", "hopper 4", or "cut-sheet feeder" for [SUPLY \(paper supply orientation\)](#) in the printer information file.

- Prepare the serial printer with cut-sheet feeder ready for printing cut forms. If this specification is set for a serial printer without a cut-sheet feeder, cut forms are fed manually.

Paper may not be supplied from the paper supply orientation specified for [SUPLY \(paper supply orientation\)](#). In this case, correspond to paper supply orientation on control panel and paper supply orientation specified for [SUPLY \(paper supply orientation\)](#) in using [MAPINFO SUPLY \(paper supply orientation specification\)](#) in the printer information file. Moreover correspond to form size on control panel and form size for [FORMSIZE \(form size\)](#) in using [MAPINFO FORMSIZE \(paper size specification\)](#) in the printer information file.

Chapter 4 Extended Functions

PowerFORM RTS has several special printer functions. These functions allow the printing of standard forms (forms overlay), manipulation of graphic image, and bar codes. This chapter explains:

[4.1 Form Overlay](#)

[4.2 Built-in Media](#)

[4.3 Bar code printing](#)

[4.4 OCR-B font printing](#)

4.1 Form Overlay

An application program can overlap and print an overlay file and descriptor. The printed output is created using the form overlay function. Up to 128 kilobytes of data can be overlaid.

To use the form overlay function, complete the following steps before running the print program:

- Create an overlay file.
- Specify the overlay file name in the descriptor or printer information file.
- Specify the overlay file storage directory name for the [OVLDIR \(overlay file storage directory name\)](#) in the printer information file.

If the overlay file name is specified in both the descriptor and printer information file, the overlay file name specified for the descriptor is valid.

If the overlay file storage directory is not specified in the printer information file, the current directory is used as the overlay file storage directory.

Up to 10 overlay file names can be specified for the descriptor, but only the first overlay file name is valid.

4.2 Built-in Media

When a built-in media data fields is specified in the form descriptors, the bit map format data with the extension "BMP" can be output.

To output bit map data, define the format of the built-in media field to be visual and the type to be file name data when creating the form descriptors.

Clipping and centering

To print image data in the bit map format with the extension "BMP", write the application program as follows:

- Write to the field control field.
Set the field control field of the built-in media field to output in bit map format by [Setting the field display attribute](#) .
- Specify the bit map file.
Specify the bitmap file by writing the file name in the built-in media field field of the record to be used for printing.
- Print the bit map.
Enter the partition name containig the outputting built-in media field to the one name specified in GROUP clause.
Specify the bit map file for record data.

The follwing code shows an example of outputting an includes media field in COBOL.

"PICTURE.BMP" is printed in built-in media field S001 as the bit map.

```
MOVE "SAMP" TO PRT-FORMAT.  
MOVE "B" TO EDIT-MODE OF S001 OF SAMP.  
MOVE "PICTURE.BMP" TO S001 OF SAMP.  
MOVE "S001" TO PRT-GROUP.  
WRITE SAMP.
```

Clipping and centering

Built-in media fields can be clipped and centered.

If "yes" is set to clip a built-in media field, bit map data is output according to the size of the built-in media field. If "no" is set, bit map data is enlarged or reduced according to the size of the built-in media field without changing the aspect ratio of image data. Table 4.2.1 shows the printing result when "yes" is specified. Table 4.2.2 shows the printing result when "no" is specified.

Table 4.2.1 Printing result when "yes" is specified		
Collating sequence	Media data	Print
Size of media data > Size of built-in media field		
Size of media data ≤ size of built-in media field		



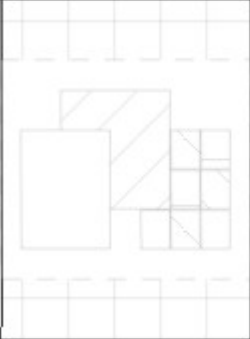
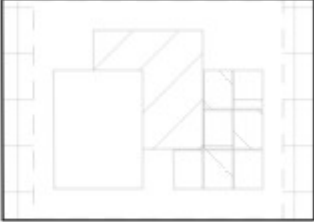
Table 4.2.2 Printing result when "no" is specified		
Similarity	Media data	Print
The shape of media data and that of built-in media field are similar.		
The shape of media data and that of built-in media field are not similar.		



When "yes" is selected to center a built-in media field, media data is printed at the center of the built-in media field. The specification is valid when "no" is specified in whether to clip a built-in media field.

Table 4.2.2 shows the print position depending on whether centering is set.

Table 4.2.2 Print position depending on whether centering is specified

Centering	Print position	
No		
Yes		



4.3 Bar code printing

The application program prints the range specification bar code and basic module width bar code.

- Range specification bar code

When the form descriptor is created, specify the maximum width of the printable bar code in the specified area. The printing start point is determined during printing so that the bar code is positioned at the center of the specified width.

- basic module width bar code

This function prints bar code in basic module width specified. The features of this case are as follows:

- In case the bar code is larger than the field length, the bar code is printed within the field area. Therefore the bar code may not be printed in basic module width specified in the descriptor.
- If the basic module width specified in the descriptor is changed for printer device density, the bar code may not be printed in basic module width specified in the descriptor. Because of the value less than 1 dot is omitted.
- If the basic module width specified in the form descriptor is $9(1/1440)$ inch unit), the bar code is printed in basic module width $14(1/1440)$ inch unit).

If alphanumeric characters are specified for bar code printing and are output by the application program for the generated bar code field, the bar code is printed in the specified area.

The bar code is printed under the following conditions:

- The vertical and horizontal size can be specified.
- A check character can be added, and printing orientation and character printing can be specified.

4.4 OCR-B font printing

Alphanumeric field specified OCR-B font in the form descriptor can be printed in OCR-B font.

The character printed in OCR-B font is as follows.

Numeral 0123456789
Alpha-capital character ABCDEFGHIJKLMN**OP**QRSTUVWXYZ
Particular character *+.=/.,:;<>#&\$%

To print in OCR-B font, set the OCR-B font attached PowerFORM RTS.

Setting procedure is as follows.

- Select Font setting on control panel.
- Click Addbutton.
- Specify the directory stored OCR-B font FJOCRB10.ttf attached PowerFORM RTS. Refer to README.TXT.
- Select OCR-B FJ 10dpi(True Type) in the list box of Font select , and click OK button. Then copy the font to system directory.

Alphanumeric field specified OCR-B font is output in 10CPI.

Chapter 5 Environment Setting Files

This chapter explains the environment setting files required by the application program in order to use PowerFORM RTS.

The environment setting file include the printer information file for printing. The user creates the environment setting files according to the PowerFORM RTS functions used by a program.

The environment setting file is a text file created using any text editor. Up to 512 characters can be specified on one line. Always add a line feed code to the end of each line. Figure 5.1 shows the format.

[Format]

```
keyword value  
*comment
```

[Parameters]

keyword:

Keyword of the environment starts in column 1.

value:

After a keyword place one or more blanks, then enter the value for that keyword.

comment:

From the beginning of a line, code an asterisk (*) in column 1 of the line, then enter the comment. To code a comment in the middle of a line containing a keyword, insert one or more blanks followed by an asterisk (*), then enter the comment.

[Example]

Specify the print origin position:

```
PRTPOSMX 1000  
PRTPOSMY 1000
```

Figure 5.1 Format of environment setting file

Set the following information in the environment setting files:

[Printer Information File](#)

Printer Information File

The printer information file is valid information from open processing to close processing for print file. Besides the printer information file, the information specified in the I-control record or form descriptor is valid. The order of priority is as follows:

1. If the information specified in I-control record the specification is valid.
2. If not specified in I-control record, specification of the descriptor is valid.
3. If not specified in I-control record and descriptor, specification of the printer information file is valid.

Table 5.1.1 lists the information that can be specified in the printer information file.

Information classification	Contents	Keyword
Print control information	Print name	<u>PRTID</u>
	Output printer device name	<u>PRTDRV</u>
	Output font name	<u>PRTFONT</u>
	Font face name	<u>FONTFACE</u>
	Font number specification	<u>FONT000 to FONT255</u>
	Print format	<u>PRTFORM</u>
	Number of copies	<u>COPYG</u>
	Form type	<u>FORMKIND</u>
	Paper supply orientation specification on continuous forms	<u>MAPINFO FORMKIND</u>
	Paper supply orientation	<u>SUPLY</u>
	Paper supply orientation specification	<u>MAPINFO SUPLY</u>
	Form size	<u>FORMSIZE</u>
	Form size specification	<u>MAPINFO FORMSIZE</u>
	Character output alignment	<u>JALIGN</u>
	Whether the print stop dialog box is displayed	<u>PRTDIALG</u>
	Setting form size when continuous	<u>PHYPAGE</u>
	Printing start column position	<u>PRTPOSMX</u>
Printing start line position	<u>PRTPOSMY</u>	
Both-side information	Both-side specification	<u>SIDE</u>
	Bonding orientation specification	<u>BSUP</u>
Descriptor information	Descriptor storage directory	<u>MEDDIR</u>
	Descriptor extension	<u>MEDSUF</u>
Overlay information	Overlay file storage directory name	<u>OVLDIR</u>
	Overlay file name	<u>OVLNAME</u>
	Overlay file extension	<u>OVLPSUF</u>
Built-in media	Built-in media storage directory name	<u>MEDIADIR</u>

This section explains the settings of the keywords.

If the keywords and values set in the printer information file are invalid, an error does not occur but the standard values (default) are used instead.

PRTID (Print name)

[Explanation]

Specify the print request name. The name specified in this keyword is displayed on the print manager display screen.

[Setting]

Specify a print name using up to 16 characters.

[Default]

The printer information file name is displayed.

PRTDRV (Output printer device name)

[Explanation]

Specify the using printer device.

[Setting]

To specify the printer name enclosed in double quotation marks, refer to the list displayed when the printer icon in the Control Panel window is selected.

[Default]

The default printer is used.

PRTFONT (Output font name)

[Explanation]

Specify the font type to be used for printing.

[Setting]

DEV: Default font

The font type to be used for printing is Courier New. If Courier New is not found, The font type is used system font.

FAC: Face name specification

The face name of the font to be used is specified for the font face name.

[Default]

DEV: Default

FONTFACE (Font face name)

[Explanation]

Specify the face name of the font if "Face name specification" is specified by [PRTFONT\(output font name\)](#)

[Setting]

Use up to 31 characters including the face name and delimit the font face name with double quotation marks.
See the font sample list displayed when the icon in the <control panel> window is selected to set the font face name.

[Default]

Assumed that DEV(Default) is specified for [PRTFONT](#)

FONT000 to FONT255 (Font number specification)

[Explanation]

Specify the font face name for the keyword corresponding to the font number specified in the form descriptor.

[Setting]

Use up to 31 characters. Includes the face name and delimit the font face with double quotation marks.

Refer to the font list displayed when the font icon in the Control Panel window is selected to set the font name.

[Default]

Font specified by FONTFACE

PRTFORM (Print format)

[Explanation]

Specify whether data is printed in portrait (vertical), landscape (horizontal), or LP mode. This specification is valid if "Default" is specified for the form size in the descriptor. LP print is used to print 136 column data as a line printer image on A4 paper. LP mode is valid only for a laser printer.

[Setting]

"PO": Portrait
"LA": Landscape
"LP": LP

[Default]

"PO": Portrait

COPYG (Number of copies)

[Explanation]

Specify the number of copies to print the same page repeatedly.
Only the driver that can specify the number of copies is valid.

[Setting]

Specify the number of copies.

[Default]

1

FORMKIND (Form type)

[Explanation]

Specify continuous form or cut paper. If a page printer are used, specify cut paper.

[Setting]

"F": Continuous paper

"C": Cut paper

[Default]

"F": Continuous paper

MAPINFO FORMKIND (Paper supply orientation specification on continuous forms)

[Explanation]

Correspond to paper supply orientation on control panel and paper supply orientation on continuous paper printing by using this specification.

If the paper is supplied incorrectly, specify this specification.

[Setting]

MAPINFO FORMKIND,F=*paper supply orientation name of the driver*

Example)

MAPINFO FORMKIND,F=tractorfeeder

[Default]

Paper supply orientation for continuous paper is set automatically.

SUPLY (Paper supply orientation)

[Explanation]

Specify the cut sheet feeder, hopper 1, hopper 2, hopper 3, hopper 4, manual, or default as the paper feed slot. This specification is valid when "default" is specified for the paper supply orientation in the descriptor.

If default is specified, settings on the control panel are valid.

[Setting]

CF: Cut sheet feeder

H1: Hopper 1

H2: Hopper 2

H3: Hopper 3

H4: Hopper 4

MA: Manual feed

NO: Default

[Default]

"CF": Cut sheet feeder

[Notes]

If the paper is not supplied from the specified paper orientation, specify [MAPINFO SUPPLY \(paper supply orientation specification\)](#).

To use automatic paper supply function, specify Default for this specification, [FORMSIZE \(Paper saize\)](#) and [MAPINFO FORMSIZE \(Form size specification\)](#) surely.

MAPINFO SUPPLY (Paper supply orientation specification)

[Explanation]

On printing cut paper, correspond to paper supply orientation on control panel and paper supply orientation for [SUPPLY \(paper supply orientation\)](#) by using this specification.

[Setting]

MAPINFO SUPPLY, *paper supply orientation specified for [SUPPLY \(paper supply orientation\)](#) =paper supply orientation name of driver*

Example) In case of XL5600(paper supply orientation : BIN1)

MAPINFO SUPPLY,H1=BIN1

[Default]

Paper may not be supplied from the paper supply orientation specified by [SUPPLY \(paper supply orientation\)](#).

FORMSIZE (Paper size)

[Explanation]

Specify the paper size as. A3, A4, A5, B4, B5, or postcard. This specification is valid if "Default" is specified for the paper size in the descriptor.

[Setting]

"A3": A3
"A4": A4
"A5": A5
"B4": B4
"B5": B5
"PS": Postcard
"LT": Letter
"LG": Legal
"NO": Default
"OT": Other

[Default]

"A4": A4

[Notes]

If OT is specified, specify [MAPINFO FORMSIZE \(paper size specification\)](#)

If this keyword is specified NO and paper size in descriptor is specified Default, the setting on control panel is valid.

This keyword may be invalid. In case of this, specify [MAPINFO FORMSIZE \(paper size specification\)](#).

MAPINFO FORMSIZE (Paper size specification)

[Explanation]

Correspond to paper supply orientation on control panel and paper supply orientation specified by [FORMSIZE \(paper size\)](#) by using this specification.

[Setting]

MAPINFO FORMSIZE, *paper size specified for [FORMSIZE \(paper size\)](#) =paper size name of driver*

Example) Specification of paper supply orientation and paper size on continuous printing

FORMKIND F

PHYPAGE Y

FORMSIZE OT

MAPINFO FORMKIND,F=tractorfeeder

MAPINFO FORMSIZE,OT=15 x 11 in

[Default]

Forms may not be printed on the paper specified by [FORMSIZE \(paper size\)](#).

If paper size is specified this keyword on continuous printing, specify "printing on continuous forms by setting on control panel" for [PHYPAGE \(setting form size when continuous forms are printed\)](#) surely.

JALIGN (Character output alignment)

[Explanation]

Specify whether to align the printing position in the line direction with the bottom end or top end of line pitch specified.

[Setting]

"DC": Alignment with bottom end

"UC": Alignment with top end

[Default]

"DC": Alignment with bottom end

[Notes]

- If alignment with bottom end is specified when the device font is used, characters which run above the top end of the paper cannot be printed.

PRTDIALG (Whether print stop dialog box is displayed)

[Explanation]

Specify whether to display the dialog box used to stop printing during form print processing.

[Setting]

"Y": Yes

"N": No

[Default]

"N": No

PHYPAGE (Setting paper size when continuous forms are printed)

[Explanation]

When continuous forms are printed, specify whether the user-defined size specified in the Control Panel printer driver is used.

[Setting]

Y: Continuous forms of the user-defined size are used.

N: Continuous forms of the size specified in the form descriptor are used.

[Default]

N: Continuous forms of the size specified in the form descriptor are used.

[Note]

If "continuous forms of the size specified in the form descriptor are used" is specified, specify a user-defined size for the printer driver paper size. If a user-defined size is not specified, forms are not correctly printed.

PRTPOSMX (Printing start column position)

[Explanation]

Specify the printing start column position.

[Setting]

Set the horizontal position using an integer representing 1/100 mm.

[Default]

The printer default is used.

If [PRTPOSMY \(printing start line position\)](#) is specified, 0 is set.

PRTPOSMY (Printing start line position)

[Explanation]

Specify the printing start line position.

[Setting]

Set the vertical position using an integer representing 1/100 mm.

[Default]

The printer default is used.

If [PRTPOSMX \(printing start column position\)](#) is specified, 0 is set.

SIDE (Both-side specification)

[Explanation]

Specify the both-side printing. Only the printer that have the both-side function is valid.
Specify both-side in the form descriptor, and set in the printer information file as follows.

[Setting]

Y: Both-side printing
N: Not both-side printing

[Default]

N: Not both-side printing

BSUP (Bonding orientation specification)

[Explanation]

On both-side printing, bonding orientation of printing matter can be specified. Upper or left specification is valid.

[Setting]

L: Left

U: Upper

[Default]

U: Upper

MEDDIR (Descriptor storage directory)

[Explanation]

Specify the descriptor storage directory. Multiple directories can be specified by delimiting them with a semicolon.

[Setting]

Directory name

[Default]

If the printer information file name is specified only in [Open processing](#) by file name, the current directory contains the descriptor.

If the printer information file name is specified using the full path, the directory containing the printer information file contains the descriptor.

MEDSUF (Descriptor extension)

[Explanation]

Specify the extension added to the descriptor file name. If four or more characters are specified for the extension, only the first three characters are valid.

Specify "NONE" if there is no extension.

[Setting]

Extension

[Default]

The extension SMD is used.

OVLDIR (Overlay file storage directory name)

[Explanation]

Specify the overlay file storage directory name. Multiple directories can be specified by delimiting them with a semicolon.

[Setting]

Directory name

[Default]

Current directory

OVLpname (Overlay file name)

[Explanation]

Specify the overlay file name. This specification is valid if the overlay file name is not specified in the descriptor.

[Setting]

Specify the overlay file name using up to eight characters.

[Default]

The overlay file is not printed.

OVLPSUF (Overlay file extension)

[Explanation]

Specify the extension added to the overlay file name. If four or more characters are specified for the extension, only the first three characters are valid.

Specify "NONE" if there is no extension.

[Setting]

Specify the extension added to the overlay file name.

[Default]

The extension OVD is used.

MEDIADIR (Built-in media storage directory name)

[Explanation]

Specify the built-in media storage directory.

Multiple directories can be specified by delimiting them with a semicolon. If the built-in media file name is specified using the full path by a program, the specification overrides MEDIADIR.

[Setting]

Specify built-in media storage directory.

[Default]

Current directory or directory specified by directory path

Chapter 6 Program Interfaces

This chapter explains the program interfaces provided by PowerFORM RTS. The program interfaces are as follows:

[Function Interfaces](#)

Function Interfaces

This section explains the function interfaces provided by PowerFORM RTS.

The names, formats, functions, parameters, examples, diagnosis, and notes about individual functions are explained below.

[Open processing](#)

[Output processing](#)

[Close processing](#)

[Setting the descriptor name](#)

[Setting the field display attribute](#)

[Setting the field highlight attribute](#)

[Setting the field color attribute](#)

[Obtaining the return code](#)

Open processing

Format

OPEN OUTPUT *file-name*.

Function

Open processing starts processing the print file.

Parameters

ASSIGN clause

Specify the printer information file name using a file-identifier. Refer to COBOL85 Users Guide. A file name can be specified by a full path. Only a file name can also be specified by omitting the path name. To specify only a file name, set the directory that contains the printer information file in the environment variable MEFTDIR.

Diagnosis

Return code is stored in the FILE STATUS clause.

Note

Assume that only a file name is specified and the environment variable MEFTDIR is undefined or that a correct directory name is not specified. The PowerFORM RTS retrieves the current directory. Assume that the current directory does not contain the printer information file nor that the full path is specified in ASSIGN clause and the path name is invalid. PowerFORM RTS assumes that the printer information file is omitted and the standard value (default) is specified and operates.

Output processing

Format

WRITE *record-name*

Function

Output processing edits and outputs data contained in the record data area to the printer according to the field attribute. The application program stores the output data in the record data field of the field to be output before calling this function.

To call this function immediately after open processing, always use the [Setting the descriptor name](#) function to set the descriptor name to be used.

Parameters

GROUP clause:

Specify the partition name to be output.

Example

Print the field S001 in the form descriptor "SAMP".

```
MOVE SAMP TO PRT-FORMAT.  
MOVE S001 TO PRT-GROUP.  
WRITE SAMP.
```

Diagnosis

Return code is stored in the FILE STATUS clause.

Close processing

Format

CLOSE *file-name*.

Function

Close processing terminates processing of the print file.

Parameters

none

Diagnosis

Return code is stored in the FILE STATUS clause.

Setting the descriptor name

Format

Specify in the FORMAT clause.

Function

Specify the name of the descriptor to be used for output processing for one name specified in FORMAT clause. To call an output function immediately after open processing and switch a descriptor, use this function to set the descriptor name, then call the output function.

Parameters

FORMAT clause:

Specify the name of the descriptor to be used for output. Specify a character string of up to eight characters for the descriptor file name, excluding the file extension.

Setting the field display attribute

Format

Specify in EDIT-MODE of the field.

Function

Dynamically sets the field display attribute.

Parameters

EDIT-MODE:

Specify the display attribute to be set. The values and their meanings are as follows:

(blank): The field is to be output.

X: The field is excluded from output processing.

*: Characters are expanded and edited.

B: The bit map is output.

-: The deletion line is output.

=: The double deletion line is output.

Example

The field name R001 in the field group name S001 is expanded and edited.

```
MOVE SAMP TO PRT-FORMAT.  
MOVE * TO EDIT-MODE OF R001.  
MOVE S001 TO PRT-GROUP.  
WRITE SAMP.
```

Notes

- This function has no return value.

Setting the field highlight attribute

Format

Specify in the EDIT-OPTION of the field.

Function

Dynamically sets the field highlight attribute.

Parameters

EDIT-OPTION:

Specify the highlight attribute to be set. The values and their meanings are as follows:

(blank): The previously set attribute is not changed.

B: Print in Italic.

U: Print with underline.

I: Print in Italic

C: Print Bold-Italic.

E: Print in Bold with underline.

F: Print in Italic with underline.

G: Print in Bold-Italic with underline.

Example

The field name R001 in the field group name S001 is Printed with underline.

```
MOVE SAMP TO PRT-FORMAT.  
MOVE U TO EDIT-OPTION OF R001.  
MOVE S001 TO PRT-GROUP.  
WRITE SAMP.
```

Notes

- This function has no return value.

Setting the field color attribute

Format

Specify in EDIT-COLOR of the field.

Function

Dynamically sets the field color attribute.

Parameters

EDIT-COLOR:

Specify the color attribute to be set. The values and their meanings are as follows:

(blank): The previously set attribute is not changed.

M: An field with the color attribute in the descriptor is output.

N: A nondisplay field is output.

B: An field is output in blue.

R: An field is output in red.

P: An field is output in pink.

G: An field is output in green.

T: An field is output in cyan.

Y: An field is output in yellow.

W: An field is output in white.

A: An field is output in black.

Example

Change the color of the character of the field name "R001" in the field group S001 to blue.

```
MOVE SAMP TO PRT-FORMAT.  
MOVE B TO EDIT-COLOR OF R001.  
MOVE S001 TO PRT-GROUP.  
WRITE SAMP.
```

Notes

- This function has no return value.

Obtaining the return code

Format

Check the FILE STATUS clause after execution.

Function

The return code is stored in the FILE STATUS clause. See [Trouble Analysis and Recovery](#), "Trouble Analysis and Action" for details on the return codes.

Parameter

none

Example

Check whether the system error is occurred on output.

```
MOVE SAMP TO PRT-FORMAT.  
MOVE S001 TO PRT-GROUP.  
WRITE record-name.  
IF PRT-STATUS = CX THEN.  
~ Processing if the outputting error is occurred in the field group S001 ~  
END-IF.  
~ Program error processing ~
```

Appendix A Trouble Analysis and Recovery

Appendix A explains PowerFORM RTS errors and operator responses for error recovery. An error code without a return code label in this table indicates an error that occurs only in COBOL programs.

<u>Return codes</u>	<u>Codes</u>
<u>MEFD_RC_END</u>	00
<u>MEFD_RC LENG</u>	08
<u>MEFD_RC IN</u>	09
<u>MEFD_RC_OPEN</u>	10
<u>MEFP_RC BUSY</u>	19
<u>MEFD_RC WRK</u>	21
<u>MEFD_RC MED</u>	22
<u>MEFD_RC MDMEM</u>	23
<u>MEFD_RC FLD</u>	24
<u>MEFD_RC INE</u>	32
<u>MEFD_RC MALOAD</u>	42
<u>MEFD_RC OPMDA</u>	44
<u>MEFP_RC MALINE</u>	62
<u>MEFP_RC PMITEM</u>	84
<u>MEFP_RC SYSTOP</u>	86
<u>MEFP_RC NOOVLF</u>	91
<u>MEFP_RC OVLLOAD</u>	9A
<u>MEFP_RC MANYOPEN</u>	C7
<u>MEFP_RC WNO</u>	CF
<u>MEFD_RC SYS</u>	CX
<u>MEFD_RC NOUSEC</u>	F1
<u>MEFD_RC NOSETD</u>	F2
<u>MEFD_RC NOPOST</u>	F3

MEFD_RC_END

[Alphanumeric code]

00

[Explanation]

A normal end occurred.

MEFD_RC LENG

[Alphanumeric code]

08

[Explanation]

The record in the specified descriptor could not be used because the record was longer than the record area of the application program.

[Operator response]

- Enlarge the record area of the application program.
- To enlarge the record area, copy the record of the descriptor.

MEFD_RC_IN

[Alphanumeric code]

09

[Explanation]

An internal error occurred. The internal area of PowerFORM RTS may have been destroyed.

[Operator response]

- Call your systems engineer because a system error may have occurred.

MEFD_RC_OPEN

[Alphanumeric code]

10

[Explanation]

Open processing failed. One of the following errors occurred:

- The printer information file name was not specified in the [Open processing](#) function or 0 was specified.
- The printer that was specified as the output printer in the printer information file is inactive.

[Operator response]

- Activate the printer that was specified as the output printer by using the control panel.

MEFP_RC_BUSY

[Alphanumeric code]

19

[Explanation]

Open processing failed. One of the following errors occurred:

- The printer is used by other application.
- The printer has been already opened. Moreover open processing was executed for the printer.

[Operator response]

- Check that the printer is not used.
- Open the printer after close processing.

MEFD_RC_WRK

[Alphanumeric code]

21

[Explanation]

A memory shortage occurred, the work area could not be allocated, and processing stopped.

[Operator response]

- Increase the available memory by terminating other unnecessary programs.

MEFD_RC_MED

[Alphanumeric code]

22

[Explanation]

A descriptor read error occurred. One of the following errors occurred:

- The specified descriptor file was not found.
- The reference authority for the descriptor file was not found.

[Operator response]

- Before setting a descriptor name by using the [Setting the descriptor name](#) function, check whether an attempt was made to execute the [Output processing](#) function, etc.
- Check that the descriptor name specified in the [Setting the descriptor name](#) function is valid.
- Check that the name of the directory that contains the descriptor specified for [MEDDIR](#) in the printer information file is valid.

MEFD_RC_MDMEM

[Alphanumeric code]

23

[Explanation]

A descriptor could not be read because of a memory shortage.

[Operator response]

- Increase the available memory by terminating other unnecessary programs.

MEFD_RC_FLD

[Alphanumeric code]

24

[Explanation]

The specified field name or field group name is invalid. One of the following errors occurred:

- The specified partition field group was not defined in the descriptor.

[Operator response]

- Specify the name that was defined in the descriptor.

MEFD_RC_INE

[Alphanumeric code]

32

[Explanation]

An error was detected while reading the descriptor. One of the following errors occurred:

- The descriptor is incomplete.
- The function defined in the descriptor is not supported.

[Operator response]

- Correct the error when creating the descriptor.
- Check that the function defined in the descriptor is supported by PowerFORM RTS.

MEFD_RC_MALOAD

[Alphanumeric code]

42

[Explanation]

The specified media data could not be loaded.

[Operator response]

- Store the media data in the specified directory.

MEFD_RC_OPMDA

[Alphanumeric code]

44

[Explanation]

The memory space for the specified media data could not be allocated, or creating of the specified bar code was failed.

[Operator response]

- Release an unused memory space.
- Check the bar code field specified in the descriptor.

MEFP_RC_MALINE

[Alphanumeric code]

62

[Explanation]

When the field was moved according to the specified feed lines, the field position exceeded the vertical size of the descriptor.

[Operator response]

- Do not specify an field movement position that exceeds the vertical size of the descriptor.

MEFP_RC_PMITEM

[Alphanumeric code]

84

[Explanation]

The following error was detected in print data output processing:

- An field name was specified.

[Operator response]

- Specify an field group name.

MEFP_RC_SYSTOP

[Alphanumeric code]

86

[Explanation]

A printer error was detected. The following error occurred:

- Some print files were opened, and output processing was executed for each of print files mutually.
- Output processing of forms were executed continuously before close processing. Therefore temporal area of the disk was insufficient
- A system error occurred.

[Operator response]

- Not output mutually, when some print files were opened.
- Enlarge temporal area.
- Call your systems engineer because a system error may have occurred.

MEFP_RC_NOOVL

[Alphanumeric code]

91

[Explanation]

An overlay file was read unsuccessfully. The following error occurred:

- The specified overlay file was not found.

[Operator response]

- Check that the overlay file exists in the directory specified with keyword [OVLDIR](#) of the printer information file.

MEFP_RC_OVLLOAD

[Alphanumeric code]

9A

[Explanation]

The overlay DLL could not be loaded.

[Operator response]

- Check that PowerFORM RTS has been correctly installed.

MEFP_RC_MANYOPEN

[Alphanumeric code]

C7

[Explanation]

An error occurred because too many files were opened.

[Operator response]

- Close unnecessary files.

MEFP_RC_WNO

[Alphanumeric code]

CF

[Explanation]

An internal error was detected. The internal area of PowerFORM RTS may have been destroyed.

[Operator response]

Call your systems engineer because a system error may have occurred.

MEFD_RC_SYS

[Alphanumeric code]

CX

[Explanation]

If this error occurs, subsequent operations are not guaranteed.

[Operator response]

- Call your systems engineer because a system error may have occurred.

MEFD_RC_NOUSEC

[Alphanumeric code]

F1

[Explanation]

Open processing failed. The following error occurred:

- A computer was used without this system installed.

If the maximum number of users who can simultaneously use the PowerFORM RTS LANPACK product is specified, the following error occurred:

- The number of computers that were used simultaneously exceeded the specified maximum number of users.

[Operator response]

- Install the system correctly.
- Increase the maximum number of LANPACK users.

MEFD_RC_NOSETD

[Alphanumeric code]

F2

[Explanation]

An internal error occurred. The internal area of PowerFORM RTS may have been destroyed. The information required to execute the specified processing was not set. For example, the answer area address was not set.

[Operator response]

- Call your systems engineer because a system error may have occurred.

MEFD_RC_NOPOST

[Alphanumeric code]

F3

[Explanation]

An internal error occurred. The internal area of PowerFORM RTS may have been destroyed.

[Operator response]

- Call your systems engineer because a system error may have occurred.

Appendix B Special Processing

PowerFORM RTS supports special processing as follows.

Stopping Printing

An application program can stop the current printing operation as follows:

- Specify [PRTDIALOG](#) in the printer information file.

[Printing stop dialog box](#)

[Printing stop procedure](#)

Printing stop dialog box

When "Yes" is specified for the [PRTDIALG](#) in the printer information file, the printing stop dialog box is displayed on the screen. When the "Cancel" button displayed in the printing stop dialog box is selected, the current printing operation is stopped.

Printing stop procedure

All the data contained on the page that was being processed when the "Cancel" button was selected is invalidated. However, pages for which processing has already completed are printed.

The printing stop operation is validated at the second or later output operation. At the first output operation, it is not possible to determine whether the "Cancel" button was selected. If the second or later output operation terminates abnormally, the application program checks that printing stops by using the [Obtaining the return code](#) function. When printing has stopped, the application program must always close the printer.

If printing has stopped, execute the close operation.

Figure B.1 shows the printing stop procedure.

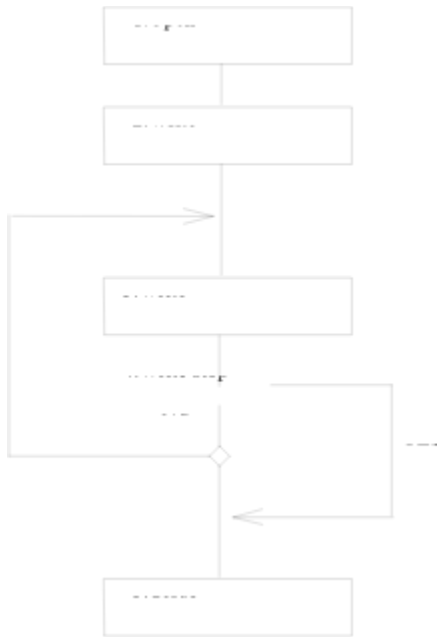


Figure B.1 Printing stop procedure

The flow shown in Figure B.1 is as follows:

- 1 Open
The printer is opened.
- 2,3 Write
Fields or field groups are printed in the partition output mode.
- 4 Write stop
Whether printing has stopped is checked using the [Obtaining the return code](#) function.
If a descriptor is switched during output processing, another print job may interrupt.
- 5 Close
The printer is closed.

