



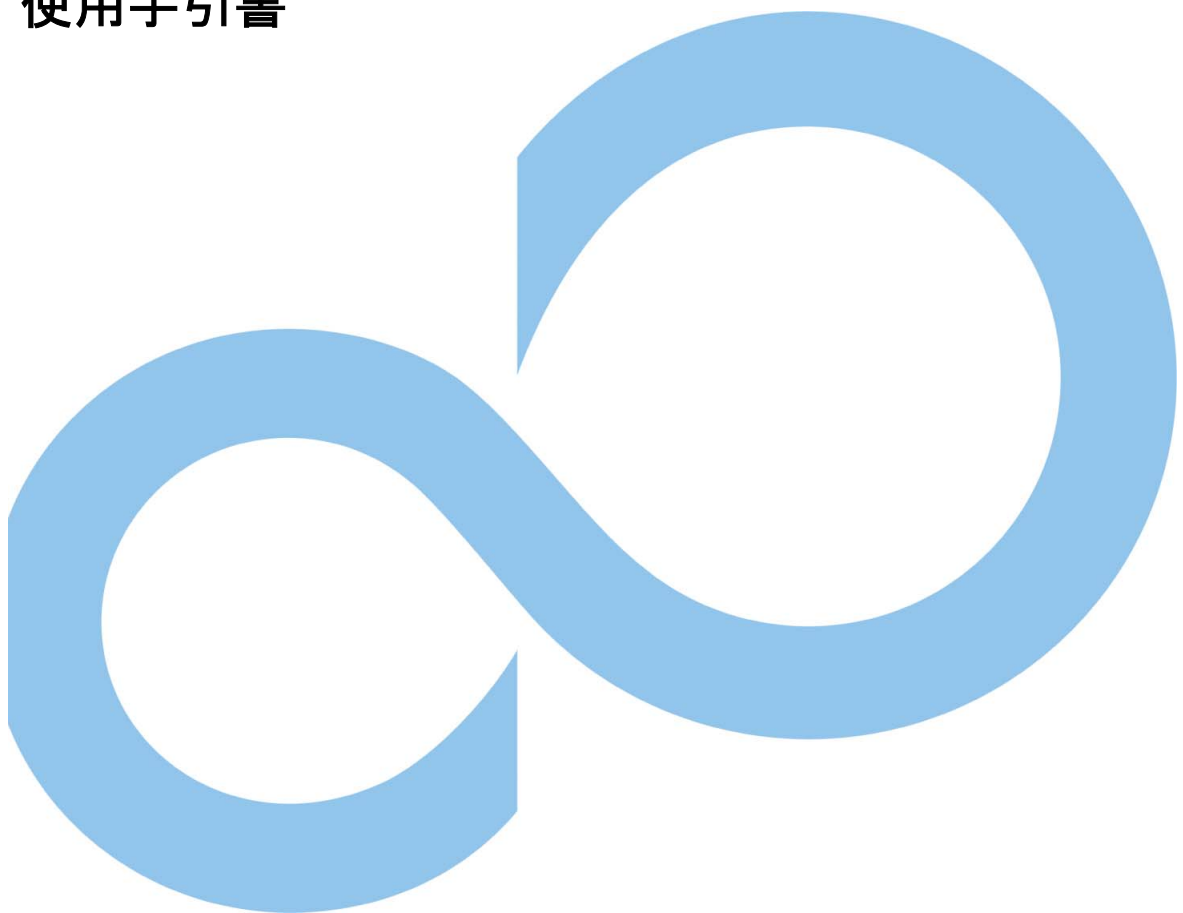
Microsoft® Windows® 95  
Microsoft® Windows® 98  
Microsoft® Windows® Me  
Microsoft® Windows NT®

Microsoft® Windows® 2000  
Microsoft® Windows® XP  
Microsoft® Windows Server™ 2003

B1JW-6291-01Z2

# 帳票印刷クラスライブラリ for COBOL V7.0

## 使用手引書



FMVシリーズ, FMV-DESKPOWER, FMV-BIBLO

Net  **OBOL**

 **FUJITSU**



---

# まえがき

“帳票印刷クラスライブラリ for COBOL V7.0”は、“NetCOBOL”を使って帳票の印刷を行うソフトウェアを開発/運用するためのクラス群と実行のための環境です。

当クラスライブラリで使用する“帳票定義体”の作成/編集には“PowerFORM”が必要です。

“PowerFORM”は“FORM”に同梱されているコンポーネントです。“オーバーレイ定義体”の作成/編集には、“FORMオーバーレイオプション”が必要です。帳票の印刷には“MeFt”が必要です。電子帳票の出力には、“SystemWalker/ListWORKS”または“SystemWalker/e-DocGenerator”が必要です。必要な製品については“[準備するもの](#)”を参照してください。

## 製品の呼び名について

本書では、各製品を次のように略記しています。あらかじめご了承ください。

- 「Microsoft(R) Windows(R) 95 operating system」  
→ 「Windows(R) 95」
- 「Microsoft(R) Windows(R) 98 operating system」  
→ 「Windows(R) 98」
- 「Microsoft(R) Windows(R) Millennium Edition」  
→ 「Windows(R) Me」
- 「Microsoft(R) Windows NT(R) Workstation operating system Version 4.0」  
→ 「Windows NT(R)」
- 「Microsoft(R) Windows NT(R) Server Network operating system Version 4.0」  
→ 「Windows NT(R)」
- 「Microsoft(R) Windows NT(R) Server Network operating system Version 4.0, Terminal Server Edition」  
→ 「Windows NT(R)」
- 「Microsoft(R) Windows NT(R) Server Network operating system, Enterprise Edition Version 4.0」  
→ 「Windows NT(R)」
- 「Microsoft(R) Windows(R) 2000 Professional operating system」  
→ 「Windows(R) 2000」
- 「Microsoft(R) Windows(R) 2000 Server operating system」  
→ 「Windows(R) 2000」
- 「Microsoft(R) Windows(R) 2000 Advanced Server operating system」  
→ 「Windows(R) 2000」
- 「Microsoft(R) Windows(R) XP Professional operating system」  
→ 「Windows(R) XP」または、「Windows(R) XP Professional」
- 「Microsoft(R) Windows(R) XP Home Edition operating system」  
→ 「Windows(R) XP」または、「Windows(R) XP Home Edition」
- 「Microsoft(R) Windows Server™ 2003, Standard Edition」  
→ 「Windows Server™ 2003」または、「Windows Server™ 2003 Standard Edition」
- 「Microsoft(R) Windows Server™ 2003 Enterprise Edition」  
→ 「Windows Server™ 2003」または、「Windows Server™ 2003 Enterprise Edition」
- 「Windows(R) 95」、「Windows(R) 98」、「Windows(R) Me」、「Windows NT(R)」、「Windows(R) 2000」および「Windows Server™ 2003」  
→ 「Windows(R)」

## 本書の目的

本書は、帳票印刷クラスライブラリの使い方とクラスの構成などについて説明します。

NetCOBOLの言語仕様については“COBOL 文法書”を参照してください。NetCOBOLを使ったソフトウェアの開発方法やデバッグのしかたについては“NetCOBOL 使用手引書”を参照してください。

---

帳票定義体とオーバレイ定義体の作成方法については、“PowerFORMヘルプ”を参照してください。  
帳票を印刷するために必要な MeFtについては“MeFt 説明書” (MeFtオンラインマニュアル)を参照してください。

## 本書の対象読者

本書は、COBOLのオブジェクト指向プログラミングを利用した帳票印刷プログラムを開発される方を対象としています。

## 前提知識

本書を読むにあたって、以下の知識が必要です。

- COBOLの文法に関する基本的な知識
- COBOLのオブジェクト指向プログラミングに関する基本的な知識
- “PowerFORM”を使った帳票定義体の設計/作成に関する基本的な知識

## 本書の位置付け

NetCOBOLシリーズにおける本書の位置付け、および関連マニュアルについては、“NetCOBOL 解説書”を参照してください。

## 登録商標について

本書に記載している登録商標を、以下に示します。

Microsoft, Windows, Windows NTは、米国 Microsoft Corporationの米国およびその他の国における登録商標です。

Adobe, Acrobat, Acrobat Readerは Adobe Systems Incorporated (アドビシステムズ社) の商標です。

2002年 7月

All Rights Reserved, Copyright (C) 富士通株式会社 1998-2002

---

# 目次

第1章	帳票印刷クラスライブラリの概要	1
1.1	帳票印刷クラスライブラリとは	2
1.2	準備するもの	4
1.3	エンハンス機能	5
第2章	開発方法	7
2.1	帳票定義体とオーバーレイ定義体の作成	8
2.2	プログラムの作成	9
2.3	プログラムの翻訳	10
2.4	オブジェクトファイルのリンク	11
2.5	アプリケーションの実行	12
第3章	印刷処理	13
3.1	アプリケーションの設計	14
3.1.1	帳票の設計	14
3.1.2	業務処理と印刷処理	15
3.1.3	印刷処理の記述の分割	16
3.2	処理の流れ	18
3.2.1	帳票印刷の呼出し	18
3.2.2	呼び出すイベント処理メソッドの選択	19
3.3	処理の記述	20
3.3.1	帳票印刷イベント処理クラスを使った印刷処理の記述	20
3.3.2	帳票印刷制御クラスを使った印刷処理の記述	23
3.3.3	業務処理での帳票機能の呼出しの記述	26
第4章	イベント	31
4.1	帳票印刷クラスライブラリのイベント	32
4.2	印刷の開始と終了に関するイベントの発生	35
4.2.1	印刷の開始と終了に関するイベント発生の流れ	35
4.2.2	発生するイベントの種類	35
4.3	基本的なイベントの発生	37
4.3.1	基本的なイベント発生の流れ	37
4.3.2	発生するイベントの種類	38
4.3.3	改ページに関するイベント発生の流れ	43
4.4	複数の帳票定義体を利用する場合のイベントの発生	44
4.4.1	最初のレコード読み込みイベントでの帳票定義体の切替え	44
4.4.2	2回目以降のレコード読み込みイベントでの帳票定義体の切替え	46
4.5	イベントの発生に影響する事項	50
4.5.1	帳票定義体名プロパティの設定	50
4.5.2	印刷種別プロパティの設定	50
4.5.3	帳票印刷メソッドの印刷モード引数	51
第5章	印刷する帳票項目の操作	53
5.1	帳票項目の文字の設定/参照	54
5.1.1	COPY文で取り込んだ帳票定義体の集団項目を使つての設定/参照	54
5.2	帳票項目の属性(プロパティ)の設定/参照	59
5.2.1	帳票項目名によるプロパティの設定/参照	59
第6章	エラー処理	65
6.1	例外エラー	66
6.2	印刷エラー	72
第7章	クラスの構成と説明	73
7.1	FJPRT-CONTROLクラス(帳票印刷制御クラス)	74
7.1.1	説明	74
7.1.2	解説	74

7.1.3	メソッド/プロパティ	76
7.2	FJPRT-EVENT-HANDLERクラス(帳票印刷イベント処理クラス)	80
7.2.1	説明	80
7.2.2	解説	80
7.2.3	メソッド	82
7.3	FJPRT-EXCEPTIONクラス(帳票印刷例外クラス)	83
7.3.1	説明	83
7.3.2	解説	83
7.3.3	メソッド	83
第8章	帳票印刷制御クラスのメソッド/プロパティ	85
8.1	印刷に関するメソッド	86
8.1.1	PRINT-CONTROLメソッド(帳票印刷メソッド)	86
8.1.2	PAGE-FEEDメソッド(改ページメソッド)	88
8.1.3	NEXT-DE-PRINT-LINEメソッド(次明細印刷行参照メソッド)	89
8.1.4	NEXT-PRINT-LINEメソッド(次印刷行参照メソッド)	92
8.1.5	SKIP-PARTITIONメソッド(パーティション抑止メソッド)	92
8.1.6	EXPORT-PRINTER-INFORMATIONメソッド(プリンタ情報生成メソッド)	93
8.1.7	PRINTER-DEVICE-NUMBERメソッド(プリンタデバイス数参照メソッド)	95
8.1.8	PRINTER-DEVICE-NAMESメソッド(プリンタデバイス名参照メソッド)	96
8.2	イベント処理メソッド	98
8.2.1	OPEN-REPORTメソッド(帳票開設イベント処理メソッド)	98
8.2.2	CLOSE-REPORTメソッド(帳票閉鎖イベント処理メソッド)	100
8.2.3	ERROR-MESSAGEメソッド(エラーメッセージイベント処理メソッド)	101
8.2.4	CANCEL-PRINTメソッド(印刷キャンセルイベント処理メソッド)	103
8.2.5	READ-RECORDメソッド(レコード読み込みイベント処理メソッド)	104
8.2.6	CALCULATE-DEメソッド(明細計算イベント処理メソッド)	106
8.2.7	PRINT-DEメソッド(明細出力イベント処理メソッド)	107
8.2.8	OPEN-FORMメソッド(帳票定義体開設イベント処理メソッド)	110
8.2.9	CLOSE-FORMメソッド(帳票定義体閉鎖イベント処理メソッド)	111
8.2.10	CB-HEADメソッド(制御頭書きイベント処理メソッド)	111
8.2.11	CB-FOOTメソッド(制御脚書きイベント処理メソッド)	114
8.2.12	PAGE-HEADメソッド(ページ頭書きイベント処理メソッド)	116
8.2.13	PAGE-FOOTメソッド(ページ脚書きイベント処理メソッド)	117
8.2.14	REPORT-HEADメソッド(報告書頭書きイベント処理メソッド)	119
8.2.15	REPORT-FOOTメソッド(報告書脚書きイベント処理メソッド)	120
8.3	帳票項目のプロパティに関するメソッド	123
8.3.1	SET-RECORD-CONTENTメソッド(レコード内容の設定メソッド)	123
8.3.2	GET-RECORD-CONTENTメソッド(レコード内容の参照メソッド)	123
8.3.3	GET-PRE-RECORD-CONTENTメソッド(前レコード内容の参照メソッド)	124
8.3.4	SET-TEXT-COLORメソッド(文字色プロパティ設定メソッド)	125
8.3.5	GET-TEXT-COLORメソッド(文字色プロパティ参照メソッド)	125
8.3.6	SET-DISABLE-FIELDメソッド(項目抑止プロパティ設定メソッド)	127
8.3.7	GET-DISABLE-FIELDメソッド(項目抑止プロパティ参照メソッド)	127
8.3.8	SET-EXTEND-CHARACTERメソッド(文字引伸しプロパティ設定メソッド)	128
8.3.9	GET-EXTEND-CHARACTERメソッド(文字引伸しプロパティ参照メソッド)	128
8.3.10	SET-FONT-STRIKE-OUTメソッド(抹消線プロパティ設定メソッド)	129
8.3.11	GET-FONT-STRIKE-OUTメソッド(抹消線プロパティ参照メソッド)	129
8.3.12	SET-FONT-ITALICメソッド(イタリックプロパティ設定メソッド)	130
8.3.13	GET-FONT-ITALICメソッド(イタリックプロパティ参照メソッド)	130
8.3.14	SET-FONT-BOLDメソッド(ボールドプロパティ設定メソッド)	131
8.3.15	GET-FONT-BOLDメソッド(ボールドプロパティ参照メソッド)	131
8.3.16	SET-FONT-UNDERLINEメソッド(下線プロパティ設定メソッド)	132

8.3.17	GET-FONT-UNDERLINEメソッド(下線プロパティ参照メソッド)	132
8.3.18	SET-FIELD-UNDERLINEメソッド(項目下線プロパティ設定メソッド)	133
8.3.19	GET-FIELD-UNDERLINEメソッド(項目下線プロパティ参照メソッド)	133
8.4	帳票定義体に関するプロパティ	135
8.4.1	FORM-NAMEプロパティ(帳票定義体名プロパティ)	135
8.4.2	DE-NAMEプロパティ(明細名プロパティ)	137
8.4.3	CONTROL-PARTITION-NAMEプロパティ(制御パーティション名プロパティ)	139
8.4.4	OVERLAY-NAMEプロパティ(オーバーレイ名プロパティ)	140
8.4.5	FORM-DIRECTORYプロパティ(帳票定義体格納ディレクトリプロパティ)	141
8.4.6	OVERLAY-DIRECTORYプロパティ(オーバーレイ格納ディレクトリプロパティ)	142
8.4.7	MEDIA-DIRECTORYプロパティ(メディア格納ディレクトリプロパティ)	143
8.4.8	FORM-SUFFIXプロパティ(帳票定義体拡張子プロパティ)	143
8.4.9	OVERLAY-SUFFIXプロパティ(オーバーレイ拡張子プロパティ)	144
8.5	印刷エラーに関するプロパティ	145
8.5.1	PRINT-JOB-ERROR-CODEプロパティ(印刷ジョブエラーコードプロパティ)	145
8.6	印刷行/ページ/タックに関するプロパティ	146
8.6.1	PRINTING-PAGE-NUMBERプロパティ(印刷中ページ番号プロパティ)	146
8.6.2	PRINTING-LINE-NUMBERプロパティ(印刷中行番号プロパティ)	146
8.6.3	INIT-PAGE-NUMBERプロパティ(最初のページ番号プロパティ)	147
8.6.4	PAGE-RANGEプロパティ(ページ範囲プロパティ)	148
8.6.5	START-PAGE-NUMBERプロパティ(開始ページ番号プロパティ)	148
8.6.6	END-PAGE-NUMBERプロパティ(終了ページ番号プロパティ)	149
8.6.7	START-TACK-NUMBERプロパティ(タック開始番号プロパティ)	150
8.7	入力データに関するプロパティ	151
8.7.1	AUTOMATIC-RECORD-CHANGEプロパティ(レコード自動切替えプロパティ)	151
8.7.2	RECORD-EOFプロパティ(レコードEOFプロパティ)	152
8.7.3	RECORD-COUNTプロパティ(レコード回数プロパティ)	153
8.8	印刷制御に関するプロパティ	154
8.8.1	PRINT-TYPEプロパティ(印刷種別プロパティ)	154
8.8.2	PRINT-COUNTプロパティ(印刷回数プロパティ)	155
8.8.3	REQUEST-CANCELプロパティ(キャンセル要求プロパティ)	155
8.8.4	PRINTER-NAMEプロパティ(プリンタ名プロパティ)	156
8.8.5	PRINT-JOB-NAMEプロパティ(印刷ジョブ名プロパティ)	157
8.8.6	PRINT-JOB-CONDITIONプロパティ(印刷ジョブ状態プロパティ)	158
8.8.7	PAPER-TYPEプロパティ(用紙種別プロパティ)	158
8.8.8	PRINT-COPIESプロパティ(印刷部数プロパティ)	159
8.8.9	PAPER-SIZEプロパティ(用紙サイズプロパティ)	159
8.8.10	PAPER-SIZE-HEIGHTプロパティ(用紙縦サイズプロパティ)	161
8.8.11	PAPER-SIZE-WIDTHプロパティ(用紙横サイズプロパティ)	161
8.8.12	PRINT-POSITIONプロパティ(印刷原点プロパティ)	162
8.8.13	PRINT-POSITION-TOPプロパティ(印刷原点縦位置プロパティ)	163
8.8.14	PRINT-POSITION-LEFTプロパティ(印刷原点横位置プロパティ)	164
8.8.15	REDUCTION-TYPEプロパティ(縮小種別プロパティ)	164
8.8.16	COMPACT-PRINT-TYPEプロパティ(コンパクト印刷種別プロパティ)	165
8.8.17	OVERLAY-NAME-TO-COMPACT-PRINTプロパティ(コンパクト印刷用オーバーレイ名プロパティ)	167
8.8.18	PRINTING-DIALOGプロパティ(印刷中ダイアログプロパティ)	167
8.8.19	PRINTING-DIALOG-CANCEL-BUTTONプロパティ(印刷中ダイアログのキャンセルボタンプロパティ)	168
8.8.20	PRINTING-DIALOG-MINIMIZE-BOXプロパティ(印刷中ダイアログ最小化ボックスプロパティ)	169
8.8.21	PRINTER-INFORMATION-FILENAMEプロパティ(プリンタ情報ファイル名プロパティ)	170
8.9	イベント処理に関するプロパティ	172
8.9.1	EVENT-HANDLERプロパティ(イベント処理オブジェクトプロパティ)	172

8.10	電子帳票制御のプロパティに関するメソッド	174
8.10.1	_SET-E-REPORT-TYPEメソッド(電子帳票種別プロパティ設定メソッド)	174
8.10.2	_GET-E-REPORT-TYPEメソッド(電子帳票種別プロパティ参照メソッド)	174
8.10.3	_SET-E-REPORT-CONTROL-FILENAMEメソッド(電子帳票制御ファイル名プロパティ設定メソッド)	175
8.10.4	_GET-E-REPORT-CONTROL-FILENAMEメソッド(電子帳票制御ファイル名プロパティ参照メソッド)	175
8.10.5	_SET-E-REPORT-DESTINATIONメソッド(電子帳票宛先名プロパティ設定メソッド)	176
8.10.6	_GET-E-REPORT-DESTINATIONメソッド(電子帳票宛先名プロパティ参照メソッド)	176
8.10.7	_SET-E-REPORT-FILENAMEメソッド(電子帳票ファイル名プロパティ設定メソッド)	179
8.10.8	_GET-E-REPORT-FILENAMEメソッド(電子帳票ファイル名プロパティ参照メソッド)	179
8.11	印刷制御のプロパティに関するメソッド	181
8.11.1	_SET-PRINT-PREVIEWメソッド(印刷プレビュープロパティ設定メソッド)	181
8.11.2	_GET-PRINT-PREVIEWメソッド(印刷プレビュープロパティ参照メソッド)	181
第9章	帳票印刷イベント処理クラスのメソッド	185
9.1	イベント処理メソッド	186
9.1.1	OPEN-REPORTメソッド(帳票開設イベント処理メソッド)	186
9.1.2	CLOSE-REPORTメソッド(帳票閉鎖イベント処理メソッド)	188
9.1.3	ERROR-MESSAGEメソッド(エラーメッセージイベント処理メソッド)	189
9.1.4	CANCEL-PRINTメソッド(印刷キャンセルイベント処理メソッド)	191
9.1.5	READ-RECORDメソッド(レコード読み込みイベント処理メソッド)	192
9.1.6	CALCULATE-DEメソッド(明細計算イベント処理メソッド)	195
9.1.7	PRINT-DEメソッド(明細出力イベント処理メソッド)	196
9.1.8	OPEN-FORMメソッド(帳票定義体開設イベント処理メソッド)	199
9.1.9	CLOSE-FORMメソッド(帳票定義体閉鎖イベント処理メソッド)	200
9.1.10	CB-HEADメソッド(制御頭書きイベント処理メソッド)	201
9.1.11	CB-FOOTメソッド(制御脚書きイベント処理メソッド)	203
9.1.12	PAGE-HEADメソッド(ページ頭書きイベント処理メソッド)	205
9.1.13	PAGE-FOOTメソッド(ページ脚書きイベント処理メソッド)	206
9.1.14	REPORT-HEADメソッド(報告書頭書きイベント処理メソッド)	208
9.1.15	REPORT-FOOTメソッド(報告書脚書きイベント処理メソッド)	210
第10章	帳票印刷例外クラスのメソッド	213
10.1	例外エラー情報参照メソッド	214
10.1.1	GET-CLASS-NAMEメソッド(クラス名の参照メソッド)	214
10.1.2	GET-METHOD-NAMEメソッド(メソッド名の参照メソッド)	214
10.1.3	GET-CODEメソッド(コードの参照メソッド)	215
10.1.4	GET-MESSAGEメソッド(メッセージの参照メソッド)	215
第11章	マルチスレッド	217
11.1	マルチスレッド環境下での動作	218
11.2	マルチスレッドプログラムの作成	219
11.3	マルチスレッド固有の留意事項	220
第12章	Unicode	221
12.1	Unicode環境下での動作	222
12.2	Unicodeプログラムの作成	223
12.3	Unicode固有の留意事項	224
第13章	電子帳票	225
13.1	電子帳票の概要	226
13.2	SystemWalker/ListWORKS連携	227
13.3	SystemWalker/e-DocGenerator連携	228
付録A	エラーコード一覧	229
A.1	例外エラーコード	229
A.2	印刷ジョブエラーコード	230



---

## 第1章 帳票印刷クラスライブラリの概要

---

ここでは、帳票印刷クラスライブラリ(以降で、当クラスライブラリと呼ぶことがあります)の概要と、準備するものについて説明します。

[帳票印刷クラスライブラリとは](#)

[準備するもの](#)

[エンハンス機能](#)

---

## 1.1 帳票印刷クラスライブラリとは

“帳票印刷クラスライブラリ”は、帳票のレイアウトなどを定義した“帳票定義体”（および“オーバーレイ定義体”）を使用して帳票の印刷を行うソフトウェアの開発/運用のためのクラスと実行環境の総称です。

当クラスライブラリを利用することによる利点には次のようなことが挙げられます。

“集計表”の印刷処理のように、従来の手続き型のプログラミングでは処理の記述内容が複雑となるような帳票印刷のアプリケーションの開発を容易にし、開発時間を短縮できます。

当クラスライブラリを利用して顧客に帳票部品(クラス)を提供するような用途に使えます。この場合、顧客は帳票部品のオブジェクトに自らが作成した部品(クラス)のオブジェクトを登録するだけで、加工元の情報を転記する程度の処理の記述で、複雑な帳票印刷の処理を行うようにすることも可能です。

当クラスライブラリは、電子帳票にも対応しています。詳細については“[電子帳票](#)”を参照してください。

なお、帳票印刷クラスライブラリを使ったプログラミングは、ファイル節/OPEN文/WRITE文などを使った帳票印刷のプログラミングとは異なります。オブジェクト指向を用いたイベント駆動型のプログラミングを行うことになります。詳細については“[印刷処理](#)”を参照してください。

下表に帳票印刷クラスライブラリで使用可能な帳票定義体の種類を挙げます。なお、FORMで作成した帳票定義体(拡張子:smd)やPowerFORMで作成した“ファイルインタフェース”の帳票定義体(拡張子:pmd)は当クラスライブラリでは利用できません。PowerFORM V4.0L10以前で作成/編集した帳票定義体(拡張子:pmd)はファイルインタフェースの帳票定義体であり、同様に利用できません。“クラスインタフェース”の帳票定義体を新規に作成してください。

表：利用可能な帳票定義体の種類

帳票定義体作成ソフトウェア	帳票定義体の分類	帳票定義体の種別	利用可能/不可能
PowerFORM	クラスインタフェース 1 (拡張子: pxd)	ページ形式	
		段組み伝票形式	
		集計表形式	
		段組み集計表形式 2	
		タックシール形式	
	ファイルインタフェース 1 (拡張子: pmd)	自由パーティション形式	×
		ページ形式	×
		段組み伝票形式	×
		集計表形式	×
		タックシール形式	×
FORM	(拡張子: smd)	自由形式	×
		伝票形式	×
		自由パーティション形式	×
		集計表パーティション形式	×

[記号の意味]

：利用可能、×：利用不可能。

1: PowerFORMで“ウィザード”を使って帳票定義体を新規に作成する際に、どの分類(どのインタフェース)の帳票定義体を作成するか確認されます。

2: [参照] “[段組み集計表形式](#)”

**[段組み集計表形式]**

本書では、帳票定義体の種別が集計表形式で、かつ帳票定義体のプロパティの[段組み]タブで“ 段組みを行う ”にチェックをしたものを、特に“ 段組み集計表形式 ”と呼んで区別することがあります。段組み集計表形式の帳票定義体と、段組みを行わない集計表形式の帳票定義体では、印刷時の動作が一部異なることがあり、その差異を説明するためにこの用語を使っています。

## 1.2 準備するもの

ここでは、開発/運用のそれぞれの場合に何が必要かを明記します。必要となるソフトウェアのバージョン/レベルについては、当クラスライブラリのソフトウェア説明書を参照してください。開発のための環境には以下が必要です。

COBOLコンパイラ  
PowerFORM (FORMに同梱)  
FORMオーバレイオプション (必要に応じて)  
MeFt  
SystemWalker/ListWORKS (必要に応じて)  
SystemWalker/e-DocGenerator (必要に応じて)

運用のための環境には以下が必要です。

MeFt  
COBOLランタイムシステム  
SystemWalker/ListWORKS (必要に応じて)  
SystemWalker/e-DocGenerator (必要に応じて)

COBOLコンパイラは、COBOLソースプログラムの翻訳、オブジェクトファイルのリンク、実行、デバッグのために必要です。

PowerFORMは、当クラスライブラリで利用できる帳票定義体を作成/編集するために必要です。

PowerFORMは、FORMに同梱されています。

FORMオーバレイオプションは、PowerFORMやFORMを使ってオーバレイ定義体を作成/編集するために必要です。

MeFtは、帳票の印刷のために必要なランタイムを含んでいます。

COBOLランタイムシステムは、COBOLアプリケーションの実行のために必要です。

SystemWalker/ListWORKSおよびSystemWalker/e-DocGeneratorは、電子帳票を出力するために必要です。詳細については“[電子帳票](#)”を参照してください。

以上の必要なソフトウェアを表にまとめると次のようになります。

表: 必要なソフトウェア

ソフトウェア	開発	運用
COBOLコンパイラ		-
PowerFORM (FORMに同梱)		-
FORMオーバレイオプション		-
MeFt		
COBOLランタイムシステム	-	
SystemWalker/ListWORKS		
SystemWalker/e-DocGenerator		

[記号の意味]

： 必須、   ： 任意、 -： 不要。

## 1.3 エンハンス機能

V1.0L40と比べて、V7.0L10では以下の機能がエンハンスされました。

“印刷プレビュープロパティ設定メソッド”を使って、印刷プレビュー機能をサポートしました。印刷プレビュー機能の詳細については“MeFt 説明書”を参照してください。

V1.0L30と比べて、V1.0L40では以下の機能がエンハンスされました。

“[電子帳票](#)”に対応し、帳票の電子化が可能になりました。電子帳票の詳細については“MeFt 説明書”の“帳票の電子化（SystemWalker連携）”を参照してください。



---

## 第2章 開発方法

---

ここでは帳票印刷クラスライブラリを使った開発方法について説明します。

[帳票定義体とオーバーレイ定義体の作成](#)

[プログラムの作成](#)

[プログラムの翻訳](#)

[オブジェクトファイルのリンク](#)

[アプリケーションの実行](#)

---

## 2.1 帳票定義体とオーバレイ定義体の作成

PowerFORMを使って帳票定義体(必須)とオーバレイ定義体(任意)を作成します。



注意

帳票印刷クラスライブラリで使用できない種類の帳票定義体があるので注意してください。詳細については“[表: 利用可能な帳票定義体の種類](#)”を参照してください。  
帳票定義体の作成/編集に関するその他のことがらについては、“PowerFORMヘルプ”を参照してください。



## 2.2 プログラムの作成

帳票印刷クラスライブラリが提供するクラスを使ったプログラムを作成します。

プログラム作成の際には、リポジトリ段落に使用するクラス名を指定してください。リポジトリ段落については、“COBOL文法書”の“リポジトリ段落”を参照してください。

Unicodeプログラムの作成については、“[Unicode](#)”を参照してください。

電子帳票を出力するプログラムの作成については、“[電子帳票](#)”を参照してください。

## 2.3 プログラムの翻訳

プログラムを翻訳します。

翻訳の際には、帳票印刷クラスライブラリのリポジトリファイルが必要です。翻訳オプションで、リポジトリファイルが格納されているフォルダを“リポジトリファイルの入力先フォルダ”として指定してください。リポジトリファイルについては、“NetCOBOL 使用手引書”の“リポジトリ”を参照してください。

[例]

帳票印刷クラスライブラリを C:\Program Files\NetCOBOL にインストールし、翻訳オプション REPIN でリポジトリファイルの入力先フォルダ名を指定する場合：

```
REPIN("C:\Program Files\NetCOBOL\Rep")
```

また、帳票印刷クラスライブラリでは、プロパティの設定値や、例外エラーコードの判定のために使用する記号定数の宣言を持った登録集ファイルを提供しています。

この登録集ファイルを使用するためには、翻訳オプションで、登録集ファイルが格納されているフォルダを“登録集ファイルのフォルダ”として指定する必要があります。

[例]

帳票印刷クラスライブラリを C:\Program Files\NetCOBOL にインストールし、翻訳オプション LIB で登録集ファイルのフォルダを指定する場合：

```
LIB("C:\Program Files\NetCOBOL")
```

マルチスレッドプログラムの翻訳については、“[マルチスレッドプログラムの作成](#)”を参照してください。

Unicodeプログラムの翻訳については、“[Unicodeプログラムの作成](#)”を参照してください。

## 2.4 オブジェクトファイルのリンク

オブジェクトファイルをリンクして実行形式のファイルを生成します。

オブジェクトファイルをリンクする際は、インポートライブラリ `F3BICFPR.lib` を結合する必要があります。ただし、プログラム構造を動的プログラム構造とする主プログラムのリンクには、このインポートライブラリは不要です。プログラム構造については、“NetCOBOL 使用手引書”の“プログラム構造”を参照してください。

インポートライブラリは、帳票印刷クラスライブラリを `C:\Program Files\NetCOBOL` にインストールした場合、次のフォルダに格納されています。

[例]

`C:\Program Files\NetCOBOL`

## 2.5 アプリケーションの実行

作成したアプリケーションプログラム(実行形式のファイル)を実行します。

プログラム構造が動的プログラム構造である場合、アプリケーションを実行する際に、エントリ情報ファイルが必要です。

帳票印刷クラスライブラリで必要となるエントリ情報を、FJPR TENT.infファイルにまとめました。エントリ情報ファイルを作成し、FJPR TENT.infファイルの情報を記述してください。エントリ情報については、“NetCOBOL 使用手引書”の“クラスとメソッドのエントリ情報”を参照してください。

エントリ情報を持つ FJPR TENT.infは、帳票印刷クラスライブラリを C:\Program Files\NetCOBOL にインストールした場合、次のフォルダに格納されています。

[例]

C:\Program Files\NetCOBOL

---

## 第3章 印刷処理

---

ここでは、帳票印刷クラスライブラリを使ったアプリケーションの設計と、処理の流れ、処理の記述について説明します。

[アプリケーションの設計](#)

[処理の流れ](#)

[処理の記述](#)

---

## 3.1 アプリケーションの設計

ここでは、アプリケーション全体の設計と、印刷処理の設計に関する以下のことについて説明をします。

[帳票の設計](#)

[業務処理と印刷処理](#)

[印刷処理の記述の分割](#)

### 3.1.1 帳票の設計

ここでは、帳票の設計について説明します。帳票の設計とは、帳票定義体と印刷処理の両方を使って、どのような帳票を印刷するのかを設計することを指します。

以下に帳票定義体の設計と、印刷処理の設計のための観点をあげます。

#### a. どのような外観の帳票を印刷したいのか

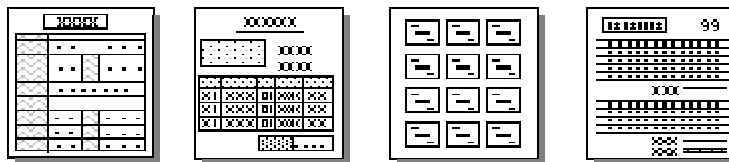
以下の設計は、PowerFORMを使って帳票定義体/オーバーレイ定義体を作成/編集することにより行います。

用紙サイズ/用紙方向

印刷する用紙の大きさと、印刷する方向を縦/横のどちらにするかを決定します。

帳票定義体の種別

どのタイプ(形式)の帳票定義体を使って、目的の帳票を印刷するかを決定します。目的の帳票が、どの形式の帳票定義体に適しているかは、“PowerFORMヘルプ”(オンラインヘルプ)を参照して判断してください。



帳票定義体の種別と形式の対応については、“[表：利用可能な帳票定義体の種類](#)”を参照してください。

レイアウト

どんなデータをどの位置に印刷するのか設計し、パーティション(区画)や項目として帳票定義体上に配置します。

集計表形式の帳票定義体では、制御頭書きパーティションや制御脚書きパーティションの印刷に影響する、ブレイク項目の定義も行います。[参照] “[ブレイク項目](#)”

飾りつけ

必要に応じて、罫線、枠、固定リテラル文字、イメージなどを配置します。

#### b. どのような印刷処理を行うか

以下の設計は、帳票印刷クラスライブラリを使った印刷処理の記述で行います。

印刷するデータはどこから得るのか(データの入力ソース)

COBOLファイル/データベース/オペレータによるキー入力など、データの入力ソースを決定します。

データの入力としては、どのような処理を行うのか

例えば、索引ファイルをデータの入力ソースとする場合、以下のようなことを決定します。

- レコードは、順読込みで入力するのか
- レコードキーとして、どれを使うのか

- レコードの読み込みの繰返しは、回数により制限を設けるのか、ファイル終了条件まで繰返し行うのか
  - ファイル終了条件を検出した場合、印刷を終了するのか
- データの入力以外に、計算など、どのような処理を行うのか
- 例えば、以下のようなことが考えられます。
- 入力したデータをもとに、和や積を求めて印刷する
  - 入力したデータをもとに求めた和や積の合計を、ヘッダやフッタのパーティション(区画)に印刷する
  - ある項目(ブレイク項目)が持つ値が変わったときに、ヘッダやフッタのパーティションを前後に囲んで印刷する
  - ページ番号/ページ数を印刷する
  - 数ページ目から、印刷する帳票(印刷に利用する帳票定義体やオーバーレイ定義体)を、別のものに切り替える
  - ある条件のときに、文字の太さ、または印刷される色などを変更する
  - 印刷エラーが発生した場合に、画面への表示を行いオペレータに通知する
  - 例外エラーが発生した場合に、画面に警告を表示して、印刷の続行/中断を行う
  - オペレータにより印刷がキャンセルされた場合に、その旨を画面に表示する

### 3.1.2 業務処理と印刷処理

ここではアプリケーションの設計に関係する、業務処理と印刷処理について説明します。  
帳票印刷を行うアプリケーションは、以下の 2つの処理に分けて設計します。

#### 業務処理

業務処理とは、印刷に関係のない業務を行ったり、帳票印刷の機能呼び出ししたりする手続きのことです。業務処理は、プログラム定義またはクラスのメソッド定義に記述し、アプリケーションのメインとなる処理を行います。

#### 印刷処理

印刷処理とは、帳票の印刷に関連する処理を行う手続きのことです。具体的には、印刷するデータを持つデータベースや COBOLファイルのレコードを読んだり、そのデータをもとに計算を行ったりする処理などを行います。印刷処理は、帳票印刷クラスライブラリが提供するクラスを、オブジェクト指向の機能を使って再利用することによって記述します。

既存のアプリケーションに、帳票印刷の機能の呼出しを部品のように組み込むことによって、帳票印刷の機能を追加することができます。この場合、既存のアプリケーションが業務処理となります。

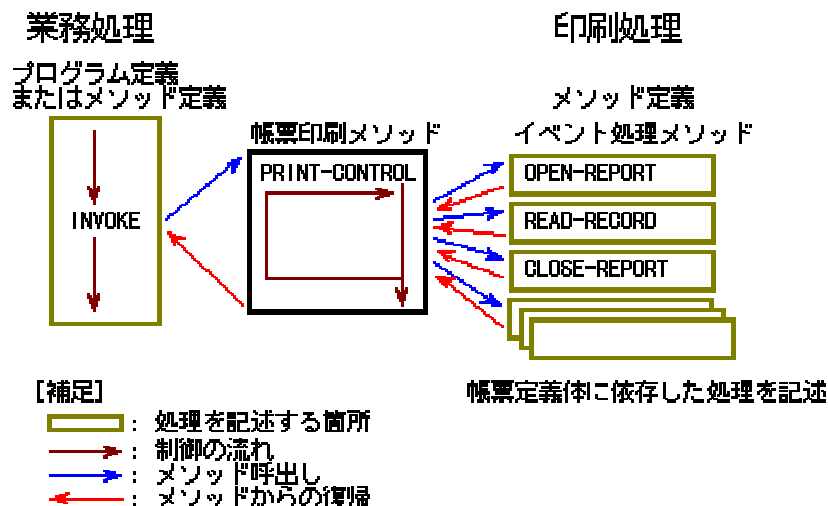
業務処理が帳票印刷の機能呼び出すと、印刷処理が呼び出され、印刷が行われます。帳票印刷の機能の呼出しとは、帳票印刷制御クラス(FJPRT-CONTROL)または、その子クラスのオブジェクトが持つ、帳票印刷メソッド(PRINT-CONTROL)を呼び出すことを言います。

帳票の印刷中(帳票印刷メソッド(PRINT-CONTROL)の実行中)は、さまざまな種類の“イベント”が発生します。イベントの発生は、イベントに対する処理が記述される“イベント処理メソッド”を呼び出すことでアプリケーションに通知します。印刷処理の記述は、イベント処理メソッドに処理を記述していくことで行います。

イベントについては、以下を参照してください。

#### [イベント](#)

下図に業務処理と印刷処理の呼出し関係を示します。



図：業務処理と印刷処理の呼出し関係

### 3.1.3 印刷処理の記述の分割

ここでは、印刷処理の記述の分割について説明します。

印刷処理の記述の方法は、大きく分けて次の2つがあります。これらの特徴をもとにして、印刷処理の記述の方法を決定します。

a. 印刷処理を1つのクラスに記述する方法

以下のクラスの、イベント処理メソッドを上書き(オーバーライド)して印刷処理を記述します。

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)

以下に、この方法の特徴を挙げます。

- 1) 印刷処理を記述するクラスとして、1つだけを利用すれば済む。
  - 後述の、印刷処理を2つのクラスに分けて記述する方法では、処理を記述するクラスによって、処理の記述方法を変える必要があります。また、2つのクラスで別々に、同じイベント処理メソッドを上書きした場合に、どのイベント処理メソッドが有効になるかなどを考慮しなくて済みます。
- 2) 印刷処理を行うための COBOLソースが1本である。
  - 作成/保守するクラスが、COBOLソースとして1つで済みます。

b. 印刷処理を2つのクラスに分けて記述する方法

印刷処理を、イベントごとに以下のいずれかのクラスに分けて記述します。記述は、イベント処理メソッドを上書きすることによって行います。

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)

帳票印刷制御クラス(FJPRT-CONTROL)

以下に、この方法の特徴を挙げます。

- 1) 印刷処理を分けることにより、何をしているのが明確になる。
  - データの入力を行うクラスと、計算を行うクラスに分けて設計することによって、互いの印刷処理の目的がはっきりとします。
- 2) 印刷処理を、部品のように目的別に分離できる。
  - 例えば、計算を行う印刷処理を持つクラスを作ります。そして、データの入力を行うクラスをいくつか作ります。このような構成にすると、データの入力を行うクラスを置き換えるだけで、別の入力ソースを使った帳票印刷が可能になります。



3) 印刷処理の開発を分担できる。

- データの入力を行うクラスを開発し、他の部署などが開発した計算を行うクラスと組み合わせて、印刷処理を行うことができます。

なお、印刷処理の記述に使うこれらのクラスは、ともに同じ種類のイベント処理メソッドを持っています。

これらのクラスについては、以下を参照してください。

“ [帳票印刷イベント処理クラス\(FJPRT-EVENT-HANDLER\)](#) ”

“ [帳票印刷制御クラス\(FJPRT-CONTROL\)](#) ”

下表に、推奨する印刷処理の分け方を示します。

表：推奨する印刷処理の分け方

イベント/イベント処理メソッド名	帳票印刷イベント処理クラス (FJPRT-EVENT-HANDLER)	帳票印刷制御クラス (FJPRT-CONTROL)	記述する印刷処理
帳票開設イベント/ OPEN-REPORT		-	データ入力ソースの オープン
帳票閉鎖イベント/ CLOSE-REPORT		-	データ入力ソースの クローズ
レコード読み込みイベント/ READ-RECORD		-	データ入力ソースからの読み込み
その他のイベント	-		イベントによる

[補足]

- : このクラスを継承し、このイベント処理メソッドに、印刷処理を記述します。
- : 印刷処理を記述しません。

## 3.2 処理の流れ

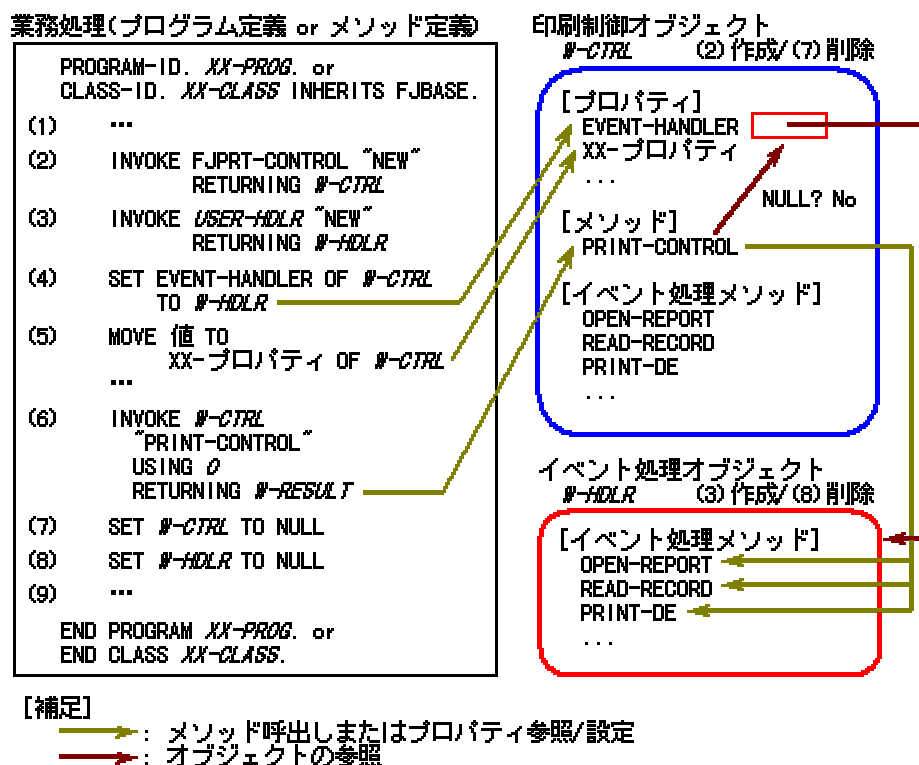
ここでは、業務処理と印刷処理で記述する処理の流れに関連することがらについて説明します。

[帳票印刷の呼出し](#)

[呼び出すイベント処理メソッドの選択](#)

### 3.2.1 帳票印刷の呼出し

ここでは、業務処理が、どのような処理で帳票印刷の機能呼び出すかを、簡単に説明します。  
下図に、帳票印刷の機能の呼出しを示します。



図：帳票印刷の機能の呼出し

[処理の流れの説明]

- (1) 必要に応じて、印刷に関係のない処理などを行います。
- (2) 帳票印刷制御クラス(FJPRT-CONTROL)のオブジェクト(印刷制御オブジェクト)を作成します。[参照] “[オブジェクトの作成](#)”
- (3) 帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)クラスを継承して、印刷処理を記述した子クラス(例では USER-HDLRクラス)の、オブジェクト(イベント処理オブジェクト)を作成します。[参照] “[オブジェクトの作成](#)”
- (4) イベント処理オブジェクトプロパティ(EVENT-HANDLER)に、(3)で作成したイベント処理オブジェクトを登録します。[参照] “[イベント処理オブジェクトの登録](#)”
- (5) 必要に応じて、印刷制御オブジェクトのプロパティに、値を設定します。

- (6)  
帳票印刷メソッド(PRINT-CONTROL)を呼びます。これにより、間接的に印刷処理が呼び出されて、帳票の印刷が行われます。[参照] “[帳票の印刷](#)”  
呼び出される印刷処理は、(4)で登録したイベント処理オブジェクトが持つ、印刷処理を記述したイベント処理メソッドになります。  
印刷時にどのようにイベントが発生するかについては、“[イベント](#)”を参照してください。
- (7)  
印刷制御オブジェクトを削除します。[参照] “[オブジェクトの削除](#)”
- (8)  
イベント処理オブジェクトを削除します。[参照] “[オブジェクトの削除](#)”
- (9)  
必要に応じて、印刷に関係のない処理などを行います。

### 3.2.2 呼び出すイベント処理メソッドの選択

イベント処理オブジェクトと印刷制御オブジェクトは、ともに同じ名前のイベント処理メソッドを持っています。帳票の印刷時にイベントが発生すると、どちらかのオブジェクトが持つイベント処理メソッドが呼ばれます。

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)のイベント処理メソッドを上書きして印刷処理を記述した場合は、イベント処理オブジェクトのイベント処理メソッドが呼ばれます。

帳票印刷制御クラス(FJPRT-CONTROL)のイベント処理メソッドを上書きして印刷処理を記述した場合は、印刷制御オブジェクトのイベント処理メソッドが呼ばれます。

帳票印刷イベント処理クラスと、帳票印刷制御クラスの両方のメソッドを上書きして印刷処理を記述した場合は、イベント処理オブジェクトのイベント処理メソッドが呼ばれます。

これらを下表にまとめます。

表：呼び出すイベント処理メソッド

	イベント処理メソッドの上書き	帳票印刷イベント処理クラス (FJPRT-EVENT-HANDLER)	
		する	しない
帳票印刷制御クラス (FJPRT-CONTROL)	する	イベント処理オブジェクトのイベント処理メソッドを呼ぶ	印刷制御オブジェクトのイベント処理メソッドを呼ぶ
	しない	同上	イベント処理を行わない



印刷制御オブジェクトにイベント処理オブジェクトを登録しない場合は、印刷制御オブジェクトのイベント処理メソッドが呼び出されます。

## 3.3 処理の記述

ここでは具体的に、どのように業務処理と印刷処理の記述をするのかについて説明します。

[帳票印刷イベント処理クラスを使った印刷処理の記述](#)

[帳票印刷制御クラスを使った印刷処理の記述](#)

[業務処理での帳票機能の呼出しの記述](#)

### 3.3.1 帳票印刷イベント処理クラスを使った印刷処理の記述

以下のいずれかの場合、帳票印刷イベント処理クラスを継承して、イベント処理メソッドを上書きすることにより、印刷処理を記述します。

印刷処理を1つのクラスに記述する

印刷処理を2つのクラスに分けて記述する

ここでは具体例を挙げて、以下の書き方について説明します。

[帳票印刷イベント処理クラスの子クラスの定義](#)

[帳票印刷イベント処理クラスの子クラスのオブジェクト定義](#)

#### 3.3.1.1 帳票印刷イベント処理クラスの子クラスの定義

以下に、帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の子クラスの書き方について、具体例を挙げて説明します。

```
*>.1.*.:...2.:...3.:...4.:...5.:...6.:...7.*.:...8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER-CLASS INHERITS  FJPRT-EVENT-HANDLER.  *>...[1]
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.  *>.....[2]
SYMBOLIC CONSTANT
COPY  FJPRTSC.
.
REPOSITORY.  *>.....[3]
CLASS  FJPRT-EVENT-HANDLER
CLASS  FJPRT-CONTROL
.
IDENTIFICATION DIVISION.  *>.....[4]
FACTORY.
:
END FACTORY.
IDENTIFICATION DIVISION.  *>.....[5]
OBJECT.
:
END OBJECT.
END CLASS  USER-HANDLER-CLASS.  *>.....[6]
*>.1.*.:...2.:...3.:...4.:...5.:...6.:...7.*.:...8
```

[例の説明]

[1]

見出し部のクラス名段落(CLASS-ID)に、作成する子クラス名を記述し、INHERITS句に継承

の対象となるクラスの FJPRT-EVENT-HANDLERを記述します。

[2]

[参照] “ [COPY文による記号定数の取込み](#) ”

#### [COPY文による記号定数の取込み]

環境部の構成節の特殊名段落(SPECIAL-NAMES)に、SYMBOLIC CONSTANT句を記述します。続いて COPY文に原文名 FJPRTSC を記述して、帳票印刷クラスライブラリが提供する記号定数を持った登録集ファイルである FJPRTSC.cblを取り込みます。この COPY文が取り込む記号定数は、例外エラーのエラーコードの判定やプロパティの設定/参照で用います。

[3]

環境部の構成節のリポジトリ段落(REPOSITORY)を記述し、継承するクラスで利用するクラス名をクラス指定子を指定して列挙します。利用するクラスは次のとおりです。

FJPRT-EVENT-HANDLER  
FJPRT-CONTROL



- FJPRT-CONTROLクラスを継承した子クラスに、(上書きではなく)追加した機能(メソッド/プロパティ)を利用したい場合、さらにその子クラス名をクラス指定子として記述してください。

[4]

必要に応じて、ファクトリ定義を記述します。ファクトリオブジェクトの機能を追加/変更しないかぎり、記述を省略できます。

[5]

オブジェクト定義を記述します。オブジェクト定義に記述する内容については、次節で説明します。

[6]

クラス終わり見出し(END CLASS)に、[1]で名付けた子クラス名を記述します。

### 3.3.1.2 帳票印刷イベント処理クラスの子クラスのオブジェクト定義

以下に、帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の子クラスのオブジェクト定義の書き方について、具体例を挙げて説明します。

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
:
IDENTIFICATION DIVISION.
OBJECT.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION. *>.....[1]
FILE-CONTROL.
    SELECT IDX-FILE ASSIGN TO SYS006
    ORGANIZATION IS INDEXED
    RECORD KEY IS F-KEY1
    ACCESS MODE IS SEQUENTIAL.
DATA DIVISION.
FILE SECTION. *>.....[2]
    FD IDX-FILE.
    01 F-INDEXED.
        02 F-KEY1 PIC X(10).
        02 F-DATA2 PIC 9(4).
WORKING-STORAGE SECTION.
    COPY BILL OF XMDLIB. *>.....[3]
    01 W-FIELD1-NAME PIC X(6) VALUE "FIELD1". *>...[4]
    01 W-OBJECT-DATA-NAME PIC 9(5). *>.....[5]
:
PROCEDURE DIVISION.
*>
IDENTIFICATION DIVISION. *>.....[6]
METHOD-ID. OPEN-REPORT OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-CTRLOBJ OBJECT REFERENCE FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    OPEN INPUT IDX-FILE
    MOVE "BILL" TO FORM-NAME OF L-CTRLOBJ
    EXIT METHOD.
END METHOD OPEN-REPORT.
*>
IDENTIFICATION DIVISION. *>.....[7]
METHOD-ID. READ-RECORD OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-CTRLOBJ OBJECT REFERENCE FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    READ IDX-FILE
    AT END
        MOVE B"1" TO RECORD-EOF OF L-CTRLOBJ
    NOT AT END
        INVOKE L-CTRLOBJ "GET-RECORD-CONTENT" USING BILL
        MOVE "TEXT-DATA" TO FIELD1 OF BILL
        MOVE F-KEY1 TO KEY1 OF BILL
        MOVE F-DATA2 TO DATA2 OF BILL

```

```

        INVOKE L-CTRLOBJ "SET-RECORD-CONTENT" USING BILL
        INVOKE L-CTRLOBJ "SET-TEXT-COLOR"
                USING W-FIELD1-NAME FJPRT-SC-COLOR-BLUE
    END-READ
    EXIT METHOD.
END METHOD READ-RECORD.
*>
:
END OBJECT.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

## [1],[2]

印刷処理で COBOL ファイルを扱う場合、ファイルを利用するための定義を行います。

## [3]

以下の、帳票項目を参照/設定するためのメソッドを使用するには、オブジェクト定義のデータ部の作業場所節に COPY 文を記述します。この COPY 文によって、取り込まれた宣言を、帳票定義体で定義されている項目として、間接的に設定/参照可能になります。

[レコード内容の参照メソッド\(GET-RECORD-CONTENT\)](#)

[レコード内容の設定メソッド\(SET-RECORD-CONTENT\)](#)

[前レコード内容の参照メソッド\(GET-PRE-RECORD-CONTENT\)](#)

この COPY 文については、“[COPY 文の記述](#)”を参照してください。上記のメソッドの利用方法については“[帳票印刷クラスライブラリとの集団項目の受け渡し](#)”を参照してください。

## [4]

[帳票項目のプロパティに関するメソッド](#)を使用するために、オブジェクト定義のデータ部の作業場所節に、そのメソッドの引数用の“帳票項目名”を持たせる変数を初期値指定で定義します。詳細については“[帳票項目名を格納する領域の宣言](#)”を参照してください。帳票項目のプロパティを利用するには“[メソッドによる帳票項目プロパティの設定/参照](#)”を参照してください。

## [5]

必要に応じて、オブジェクトデータとなるデータ項目を定義します。これは、オブジェクト内で共有できるデータ項目となります。

## [6],[7]

印刷処理を記述したいイベントに対応する[イベント処理メソッド](#)を、上書き(オーバーライド)して処理を記述します。

イベント処理メソッドの中で、印刷制御オブジェクトが持つ[メソッド/プロパティ](#)を利用する場合は、印刷制御オブジェクトを指す一意名として、“イベント処理メソッドの第一引数に渡される印刷制御オブジェクト参照データ項目”を記述してください。なお、メソッド/プロパティにアクセスするときに、例外エラーが発生することがあります。例外エラーについては“[例外エラー](#)”と、各メソッド/プロパティの説明を参照してください。

### 3.3.2 帳票印刷制御クラスを使った印刷処理の記述

以下の場合、帳票印刷制御クラスを継承して、イベント処理メソッドを上書きすることにより、印刷処理を記述します。

印刷処理を2つのクラスに分けて記述する

ここでは具体例を挙げて、以下の書き方について説明します。

[帳票印刷制御クラスの子クラスの定義](#)

[帳票印刷制御クラスの子クラスのオブジェクト定義](#)

### 3.3.2.1 帳票印刷制御クラスの子クラスの定義

以下に、帳票印刷制御クラス(FJPRT-CONTROL)の子クラスの書き方の具体例を挙げます。

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-CONTROL-CLASS INHERITS  FJPRT-CONTROL.  *>.....[1]
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.  *>.....[2]
    SYMBOLIC CONSTANT
    COPY  FJPRTSC.
.
REPOSITORY.  *>.....[3]
    CLASS  FJPRT-CONTROL
.
IDENTIFICATION DIVISION.  *>.....[4]
FACTORY.
:
END FACTORY.
IDENTIFICATION DIVISION.  *>.....[5]
OBJECT.
:
END OBJECT.
END CLASS  USER-CONTROL-CLASS.  *>.....[6]
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

[例の説明]

[1]

見出し部のクラス名段落(CLASS-ID)に、作成する子クラス名を記述し、INHERITS句に継承の対象となるクラスの FJPRT-CONTROLを記述します。

[2]

環境部の構成節の特殊名段落(SPECIAL-NAMES)に、SYMBOLIC CONSTANT句を記述します。続いて COPY文に原文名 FJPRTSC を記述して、帳票印刷クラスライブラリが提供する記号定数を持った登録集ファイルである FJPRTSC.cblを取り込みます。

この COPY文が取り込む記号定数は、[例外エラー](#)のエラーコードの判定やプロパティの設定/参照で用います。

[3]

環境部の構成節のリポジトリ段落(REPOSITORY)を記述し、継承するクラスで利用するクラス名をクラス指定子を指定して列挙します。利用するクラスは次のとおりです。

FJPRT-CONTROL

[4]

必要に応じて、ファクトリ定義を記述します。ファクトリオブジェクトの機能を追加/変更しないかぎり、記述を省略できます。

[5]

オブジェクト定義を記述します。オブジェクト定義に記述する内容については、次節で説明します。

[6]

クラス終わり見出し(END CLASS)に、[1]で名付けた子クラス名を記述します。



### 3.3.2.2 帳票印刷制御クラスの子クラスのオブジェクト定義

以下に、帳票印刷制御クラス(FJPRT-CONTROL)の子クラスのオブジェクト定義の書き方について、具体例を挙げて説明します。

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY BILL OF XMDLIB. *>.....[1]
    01 W-DATA2-NAME PIC X(6) VALUE "FIELD1". *>...[2]
    01 W-OBJECT-DATA-NAME PIC 9(5). *>.....[3]
:
PROCEDURE DIVISION.
*>
IDENTIFICATION DIVISION. *>.....[4]
METHOD-ID. PRINT-DE OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.
    INVOKE SELF "GET-RECORD-CONTENT" USING BILL
    IF DATA2 OF BILL < 20 THEN
        MOVE FJPRT-SC-COLOR-RED TO W-COLOR
    ELSE
        MOVE FJPRT-SC-COLOR-BLACK TO W-COLOR
    END-IF
    INVOKE SELF "SET-TEXT-COLOR"
        USING W-DATA2-NAME W-COLOR
    EXIT METHOD.
END METHOD PRINT-DE.
*>
:
END OBJECT.
:
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

[例の説明]

[1]

以下の、帳票項目を参照/設定するためのメソッドを使用するには、オブジェクト定義のデータ部の作業場所節に COPY文を記述します。この COPY文によって、取り込まれた宣言を、帳票定義体で定義されている項目として、間接的に設定/参照可能になります。

[レコード内容の参照メソッド\(GET-RECORD-CONTENT\)](#)

[レコード内容の設定メソッド\(SET-RECORD-CONTENT\)](#)

[前レコード内容の参照メソッド\(GET-PRE-RECORD-CONTENT\)](#)

このCOPY文については、“[COPY文の記述](#)”を参照してください。上記のメソッドの利用方法については“[帳票印刷クラスライブラリとの集団項目の受け渡し](#)”を参照してください。

[2]

[帳票項目のプロパティに関するメソッド](#)を使用するために、オブジェクト定義のデータ部の作業場所節に、そのメソッドの引数用の“帳票項目名”を持たせる変数を初期値指定で定義します。詳細については“[帳票項目名を格納する領域の宣言](#)”を参照してください。

帳票項目のプロパティを利用するには“ [メソッドによる帳票項目プロパティの設定/参照](#) ”を参照してください。

[3] 必要に応じて、オブジェクトデータとなるデータ項目を定義します。これは、オブジェクト内で共有できるデータ項目となります。

[4] 印刷処理を記述したいイベントに対応する [イベント処理メソッド](#) を、上書き(オーバーライド)して処理を記述します。  
イベント処理メソッドの中で、印刷制御オブジェクトが持つ [メソッド/プロパティ](#) を利用する場合は、印刷制御オブジェクトを指す一意名として SELF を記述してください。なお、メソッド/プロパティにアクセスするときに、例外エラーが発生することがあります。例外エラーについては“ [例外エラー](#) ”と、各メソッド/プロパティの説明を参照してください。

### 3.3.3 業務処理での帳票機能の呼出しの記述

ここでは、業務処理から、帳票印刷の機能呼び出すために行う記述について説明します。  
以下のそれぞれの場合について説明します。

[印刷処理を1つのクラスに記述する場合の記述](#)

[印刷処理を2つのクラスに分けて記述する場合の記述](#)

#### 3.3.3.1 印刷処理を1つのクラスに記述する場合の記述

---

以下に具体例を挙げます。なお、ここではプログラム定義の業務処理の例を挙げています。

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
PROGRAM-ID.  PRT-PROG.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.  *>.....[1]
    SYMBOLIC CONSTANT
    COPY  FJPRTSC.
.
REPOSITORY.  *>.....[2]
    CLASS  FJPRT-CONTROL
    CLASS  USER-HANDLER-CLASS
.
DATA DIVISION.
WORKING-STORAGE SECTION.
:
01  W-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.      *>.....[3]
01  W-HDLROBJ  OBJECT REFERENCE  USER-HANDLER-CLASS.  *>.....[4]
01  W-RESULT   PIC 1(1)  DISPLAY.
:
PROCEDURE DIVISION.
:
    INVOKE  FJPRT-CONTROL      "NEW"  RETURNING  W-CTRLOBJ  *>.....[5]
    INVOKE  USER-HANDLER-CLASS  "NEW"  RETURNING  W-HDLROBJ  *>.....[6]
    SET  EVENT-HANDLER  OF  W-CTRLOBJ  TO  W-HDLROBJ  *>.....[7]
    MOVE  値  TO  プロパティ名  OF  W-CTRLOBJ  *>.....[8]
:
    INVOKE  W-CTRLOBJ  "PRINT-CONTROL"
        USING  0  RETURNING  W-RESULT  >.....[9]
    SET  W-CTRLOBJ  TO  NULL  *>.....[10]
    SET  W-HDLROBJ  TO  NULL  *>.....[11]
:
END PROGRAM  PRT-PROG.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

## [例の説明]

[1]

環境部の構成節の特殊名段落(SPECIAL-NAMES)に、SYMBOLIC CONSTANT句を記述します。続いて COPY文に原文名 FJPRTSC を記述して、帳票印刷クラスライブラリが提供する記号定数を持った登録集ファイルである FJPRTSC.cblを取り込みます。  
この COPY文が取り込む記号定数は、[例外エラー](#)のエラーコードの判定やプロパティの設定/参照で用います。

[2]

環境部の構成節のリポジトリ段落(REPOSITORY)を記述し、継承するクラスで利用するクラス名をクラス指定子を指定して列挙します。利用するクラスは次のとおりです。

FJPRT-CONTROL

FJPRT-EVENT-HANDLERを継承した子クラス(例では USER-HANDLER-CLASS)

[3]

FJPRT-CONTROLクラスのオブジェクト参照一意名を宣言します。

[4]

FJPRT-EVENT-HANDLERを継承した子クラスの、オブジェクト参照一意名を宣言します。

- [5]  
印刷制御オブジェクトを作成します。
- [6]  
イベント処理オブジェクトを作成します。
- [7]  
印刷制御オブジェクトに、イベント処理オブジェクトを登録します。
- [8]  
必要に応じて、印刷制御オブジェクトのプロパティに値を設定します。
- [9]  
帳票印刷メソッド (PRINT-CONTROL) を呼び出します。これによって帳票の印刷とイベントの発生が始まり、このメソッドから復帰するまでに印刷が終了します。
- [10]  
印刷制御オブジェクトを削除します。
- [11]  
イベント処理オブジェクトを削除します。

### 3.3.3.2 印刷処理を2つのクラスに分けて記述する場合の記述

---

以下に具体例を挙げます。なお、ここではプログラム定義の業務処理の例を挙げています。

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
PROGRAM-ID.  PRT-PROG.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
    SPECIAL-NAMES.  *>.....[1]
        SYMBOLIC CONSTANT
        COPY  FJPRTSC.
        .
REPOSITORY.  *>.....[2]
    CLASS  USER-HANDLER-CLASS
    CLASS  USER-CONTROL-CLASS
    .
DATA DIVISION.
WORKING-STORAGE SECTION.
    :
    01  W-HDLROBJ  OBJECT REFERENCE  USER-HANDLER-CLASS.  *>.....[3]
    01  W-CTRLOBJ  OBJECT REFERENCE  USER-CONTROL-CLASS.  *>.....[4]
    01  W-RESULT  PIC 1(1)  DISPLAY.
    :
PROCEDURE DIVISION.
    :
    INVOKE  USER-CONTROL-CLASS  "NEW"  RETURNING  W-CTRLOBJ  *>.....[5]
    INVOKE  USER-HANDLER-CLASS  "NEW"  RETURNING  W-HDLROBJ  *>.....[6]
    SET  EVENT-HANDLER  OF  W-CTRLOBJ  TO  W-HDLROBJ  *>.....[7]
    MOVE  値  TO  プロパティ名  OF  W-CTRLOBJ  *>.....[8]
    :
    INVOKE  W-CTRLOBJ  "PRINT-CONTROL"
        USING  0  RETURNING  W-RESULT  >.....[9]
    SET  W-CTRLOBJ  TO  NULL  *>.....[10]
    SET  W-HDLROBJ  TO  NULL  *>.....[11]
    :
END PROGRAM  PRT-PROG.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

## [例の説明]

## [1]

環境部の構成節の特殊名段落(SPECIAL-NAMES)に、SYMBOLIC CONSTANT句を記述します。続いて COPY文に原文名 FJPRTSC を記述して、帳票印刷クラスライブラリが提供する記号定数を持った登録集ファイルである FJPRTSC.cblを取り込みます。  
この COPY文が取り込む記号定数は、[例外エラー](#)のエラーコードの判定やプロパティの設定/参照で用います。

## [2]

環境部の構成節のリポジトリ段落(REPOSITORY)を記述し、継承するクラスで利用するクラス名をクラス指定子を指定して列挙します。利用するクラスは次のとおりです。  
FJPRT-EVENT-HANDLERを継承した子クラス(例では USER-HANDLER-CLASS)  
FJPRT-CONTROLを継承した子クラス(例では USER-CONTROL-CLASS)

## [3]

FJPRT-EVENT-HANDLERを継承した子クラスの、オブジェクト参照一意名を宣言します。

## [4]

FJPRT-CONTROLを継承した子クラスの、オブジェクト参照一意名を宣言します。

- [5]  
印刷制御オブジェクトを作成します。
- [6]  
イベント処理オブジェクトを作成します。
- [7]  
印刷制御オブジェクトに、イベント処理オブジェクトを登録します。
- [8]  
必要に応じて、印刷制御オブジェクトのプロパティに値を設定します。
- [9]  
帳票印刷メソッド (PRINT-CONTROL) を呼び出します。これによって帳票の印刷とイベントの発生が始まり、このメソッドから復帰するまでに印刷が終了します。
- [10]  
印刷制御オブジェクトを削除します。
- [11]  
イベント処理オブジェクトを削除します。

---

## 第4章 イベント

---

ここでは、帳票印刷クラスライブラリを利用する上で理解が不可欠である“イベント”について説明します。

[帳票印刷クラスライブラリのイベント](#)

[印刷の開始と終了に関するイベントの発生](#)

[基本的なイベントの発生](#)

[複数の帳票定義体を利用する場合のイベントの発生](#)

[イベントの発生に影響する事項](#)

## 4.1 帳票印刷クラスライブラリのイベント

帳票印刷クラスライブラリにおけるイベントについて説明します。

帳票印刷クラスライブラリのイベントとは、帳票の印刷時にクラスライブラリが、利用者が記述した処理を実行すべきと判断したさまざまな場面/条件で発生するものです。

イベントにはいろいろな種類があります。ある特定のイベントが発生するとその特定のイベントのために利用者が記述した処理が帳票印刷クラスライブラリから呼び出されて実行されます。この呼出しがプログラムへのイベントが発生したことの通知となります。

利用者は、イベントが発生した際に行う処理の内容を、“イベント処理メソッド”に記述します。イベント処理メソッドはイベントの発生によってクラスライブラリより呼び出されます。イベント処理メソッドは、以下のクラスが持っています。これらのクラスはすべての種類のイベントに対応するイベント処理メソッドを持ちます。利用者はこれらのクラスを継承して、イベント処理メソッドを上書き(オーバーライド)してイベント発生時に行う処理を記述します。

[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)

[帳票印刷イベント処理クラス\(FJPRT-EVENT-HANDLER\)](#)

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)を利用したイベント処理を行うには、イベント処理オブジェクトをイベント処理オブジェクトプロパティ(EVENT-HANDLER)に設定(登録)する必要があります。詳細については“[イベント処理オブジェクトの登録](#)”を参照してください。イベントにはそれが発生する流れ(順番/条件)があります。イベントが発生する流れと各イベントで利用者が記述する処理については、以下を参照してください。全体的なイベント発生の流れは、“[図: 印刷の開始と終了に関するイベント発生の流れ](#)”と“[図: 基本的なイベント発生の流れ](#)”を合わせて参照してください。

[印刷の開始と終了に関するイベントの発生](#)

[印刷の開始と終了に関するイベント発生の流れ](#)

[発生するイベントの種類](#)

[基本的なイベントの発生](#)

[基本的なイベント発生の流れ](#)

[発生するイベントの種類](#)

[改ページに関するイベント発生の流れ](#)

[複数の帳票定義体を利用する場合のイベントの発生](#)

[最初のレコード読み込みイベントでの帳票定義体の切替え](#)

[2回目以降のレコード読み込みイベントでの帳票定義体の切替え](#)

また、イベント発生の流れに影響を与える事項があります。以下の節を参照してください。

[イベントの発生に影響する事項](#)

[帳票定義体名プロパティの設定](#)

[印刷種別プロパティの設定](#)

[帳票印刷メソッドの印刷モード引数](#)

イベントの種類については“[表: イベントの一覧](#)”を参照してください。

帳票定義体の種別によっては特定のイベントが発生しないことがあります。また帳票定義体の種別によってはイベントの名前と処理対象となるパーティションの名前が一致していないものがあります。詳細については“[表: 帳票定義体の種別と発生するイベントの一覧](#)”を参照してください。



表: イベントの一覧

イベントの種類 (イベント処理 メソッド名)	イベントが発生するタイミング	そのイベントで記述する 主な処理
<a href="#">帳票開設</a> (OPEN-REPORT)	<a href="#">帳票印刷メソッド (PRINT-CONTROL)</a> を呼び出した直後	帳票印刷に関する初期化処理、入力ファイルのオープン。
<a href="#">レコード読み込み</a> (READ-RECORD)	明細/ボディ部/ページの印刷の繰り返しごと	印刷する文字の設定。[参照] “ <a href="#">帳票項目の文字の設定/参照</a> ”。入力ファイルのレコードの読み込み。
<a href="#">エラーメッセージ</a> (ERROR-MESSAGE)	印刷時のエラー ( <a href="#">印刷エラー</a> ) の発生	<a href="#">印刷エラー</a> に対する処理
<a href="#">印刷キャンセル</a> (CANCEL-PRINT)	オペレータの操作または <a href="#">キャンセル要求プロパティ (REQUEST-CANCEL)</a> による帳票印刷のキャンセル	印刷をキャンセルした旨のメッセージを表示
<a href="#">帳票閉鎖</a> (CLOSE-REPORT)	<a href="#">帳票印刷メソッド (PRINT-CONTROL)</a> の終了の直前	帳票開設に対応する終了処理、入力ファイルのクローズ。
<a href="#">帳票定義体開設</a> (OPEN-FORM)	帳票開設イベントの後。 または帳票印刷途中で帳票定義体切替え後 (ページ脚書きイベントとページ頭書きイベントの間) の帳票定義体開設時。[参照] “ <a href="#">複数の帳票定義体を利用する場合のイベントの発生</a> ”	帳票定義体に関する初期化処理
<a href="#">帳票定義体閉鎖</a> (CLOSE-FORM)	帳票閉鎖イベントの前。 または帳票印刷途中で帳票定義体切替え前 (ページ脚書きイベントとページ頭書きイベントの間) の帳票定義体閉鎖時。[参照] “ <a href="#">複数の帳票定義体を利用する場合のイベントの発生</a> ”	帳票定義体開設に対応する終了処理
<a href="#">明細計算</a> (CALCULATE-DE)	レコード読み込みイベントの正常終了	明細/ボディ部/ページ内の派生処理
<a href="#">報告書頭書き</a> (REPORT-HEAD)	報告書頭書きの印刷の直前	報告書頭書きに印刷する文字の設定。[参照] “ <a href="#">帳票項目の文字の設定/参照</a> ”
<a href="#">ページ頭書き</a> (PAGE-HEAD)	ページ頭書きの印刷の直前	ページ頭書きに印刷する文字の設定。[参照] “ <a href="#">帳票項目の文字の設定/参照</a> ”
<a href="#">制御頭書き</a> (CB-HEAD)	制御頭書きの印刷の直前	制御頭書きに印刷する文字の設定。[参照] “ <a href="#">帳票項目の文字の設定/参照</a> ”
<a href="#">明細出力</a> (PRINT-DE)	明細の印刷の直前	明細/ボディ部の合計の計算、明細/ボディ部/ページ内の項目のプロパティ変更。[参照] “ <a href="#">帳票項目のプロパティに関するメソッド</a> ”
<a href="#">制御脚書き</a> (CB-FOOT)	制御脚書きの印刷の直前	制御脚書きに印刷する文字の設定。[参照] “ <a href="#">帳票項目の文字の設定/参照</a> ”
<a href="#">ページ脚書き</a> (PAGE-FOOT)	ページ脚書きの印刷の直前	ページ脚書きに印刷する文字の設定。[参照] “ <a href="#">帳票項目の文字の設定/参照</a> ”

イベントの種類 (イベント処理 メソッド名)	イベントが発生するタイミング	そのイベントで記述する 主な処理
<a href="#">報告書脚書き</a> (REPORT-FOOT)	報告書脚書きの印刷の直前	報告書脚書きに印刷する文字 の設定。[参照] “ <a href="#">帳票項目の 文字の設定/参照</a> ”

表：帳票定義体の種別と発生するイベントの一覧

	帳票定義体の種別	ページ 形式	段組み 伝票形式	集計表 形式	タックシー ル形式
イベント	帳票開設				
	レコード読み込み				
	エラーメッセージ				
	印刷キャンセル				
	帳票閉鎖				
	帳票定義体開設				
	帳票定義体閉鎖				
	明細計算	1	1	1	1
	報告書頭書き	-	-		-
	ページ頭書き	-	2		-
	制御頭書き	-	-		-
	明細出力	1	1	1	1
	制御脚書き	-	-		-
	ページ脚書き	-	3		-
	報告書脚書き	-	-		-

## [記号の説明]

： イベント発生あり、 - ： イベント発生なし。

1: 以下のように帳票定義体の種別によってイベントに対応するパーティションが異なります。

帳票定義体の種別	イベントに対応するパーティション
ページ形式	ページ全体のパーティション
集計表形式	明細パーティション
段組み伝票形式/タックシール形式	ボディ部のパーティション

2: イベントに対応するパーティションはヘッダ部のパーティションです。

3: イベントに対応するパーティションはテイル部のパーティションです。

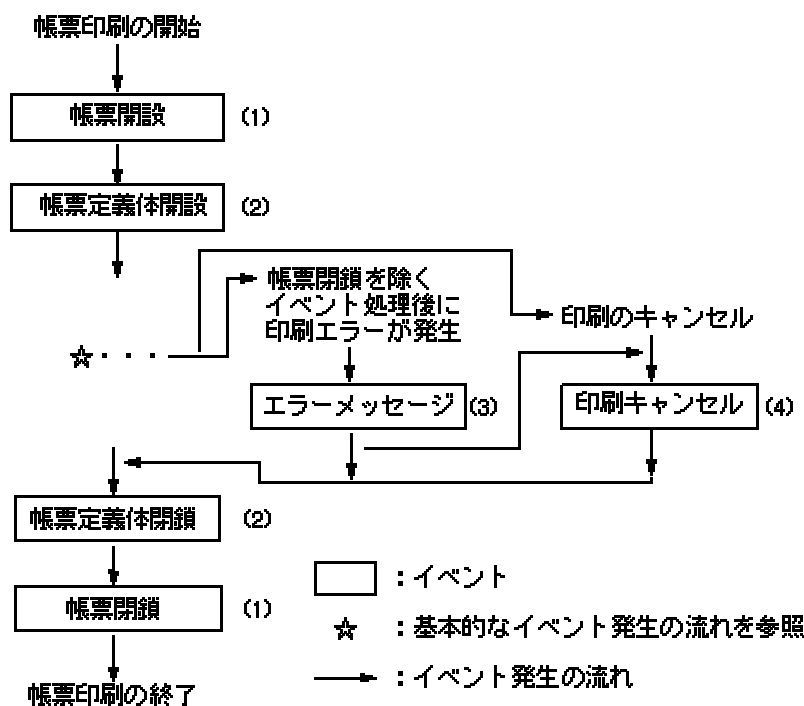
## 4.2 印刷の開始と終了に関するイベントの発生

ここでは、以下について説明します。

[印刷の開始と終了に関するイベント発生の流れ](#)  
[発生するイベントの種類](#)

### 4.2.1 印刷の開始と終了に関するイベント発生の流れ

印刷の開始と終了に関するイベント発生の流れを、下図に示します。この図に“[図：基本的なイベント発生の流れ](#)”を合わせると、全体のイベント発生の流れになります。



図：印刷の開始と終了に関するイベント発生の流れ

### 4.2.2 発生するイベントの種類

印刷の開始と終了に関連して発生するイベントには、次のものがあります。

[帳票開設イベント](#) (OPEN-REPORTメソッド呼出し)  
[帳票閉鎖イベント](#) (CLOSE-REPORTメソッド呼出し)  
[帳票定義体開設イベント](#) (OPEN-FORMメソッド呼出し)  
[帳票定義体閉鎖イベント](#) (CLOSE-FORMメソッド呼出し)  
[エラーメッセージイベント](#) (ERROR-MESSAGEメソッド呼出し)  
[印刷キャンセルイベント](#) (CANCEL-PRINTメソッド呼出し)

以下に“[図：印刷の開始と終了に関するイベント発生の流れ](#)”に記述されている番号に対応した説明をします。

### (1) 帳票開設イベントと帳票閉鎖イベント

帳票開設イベントおよび帳票閉鎖イベントでは、ほかのイベントの処理を記述する上で必要になる、初期設定や終了に関する処理を記述します。

イベント処理メソッドについては以下を参照してください。

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[帳票開設メソッド\(OPEN-REPORT\)](#)

[帳票閉鎖メソッド\(CLOSE-REPORT\)](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[帳票開設メソッド\(OPEN-REPORT\)](#)

[帳票閉鎖メソッド\(CLOSE-REPORT\)](#)

### (2) 帳票定義体開設イベントと帳票定義体閉鎖イベント

帳票定義体開設イベントおよび帳票定義体閉鎖イベントでは、ほかのイベントの処理を記述する上で必要となる、帳票定義体に対する初期設定や終了に関する処理を記述します。

イベント処理メソッドについては以下を参照してください。

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[帳票定義体開設メソッド\(OPEN-FORM\)](#)

[帳票定義体閉鎖メソッド\(CLOSE-FORM\)](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[帳票定義体開設メソッド\(OPEN-FORM\)](#)

[帳票定義体閉鎖メソッド\(CLOSE-FORM\)](#)

### (3) エラーメッセージイベント

エラーメッセージイベントでは、ほかの任意のイベント処理後に[印刷エラー](#)が発生した場合に行う処理を記述します。

イベント処理メソッドについては以下を参照してください。

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[エラーメッセージメソッド\(ERROR-MESSAGE\)](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[エラーメッセージメソッド\(ERROR-MESSAGE\)](#)

### (4) 印刷キャンセルイベント

印刷キャンセルイベントでは、オペレータの指示などにより印刷が中止された場合に行う処理を記述します。

印刷中ダイアログでオペレータが印刷中止を指示した場合、エラーメッセージイベントの後に印刷キャンセルイベントが発生します。

イベント処理メソッドについては以下を参照してください。

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[印刷キャンセルメソッド\(CANCEL-PRINT\)](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[印刷キャンセルメソッド\(CANCEL-PRINT\)](#)

## 4.3 基本的なイベントの発生

ここでは、以下について説明します。

[基本的なイベント発生の流れ](#)

[発生するイベントの種類](#)

[改ページに関するイベント発生の流れ](#)

### 4.3.1 基本的なイベント発生の流れ

基本的なイベント発生の流れを、以下の図に示します。

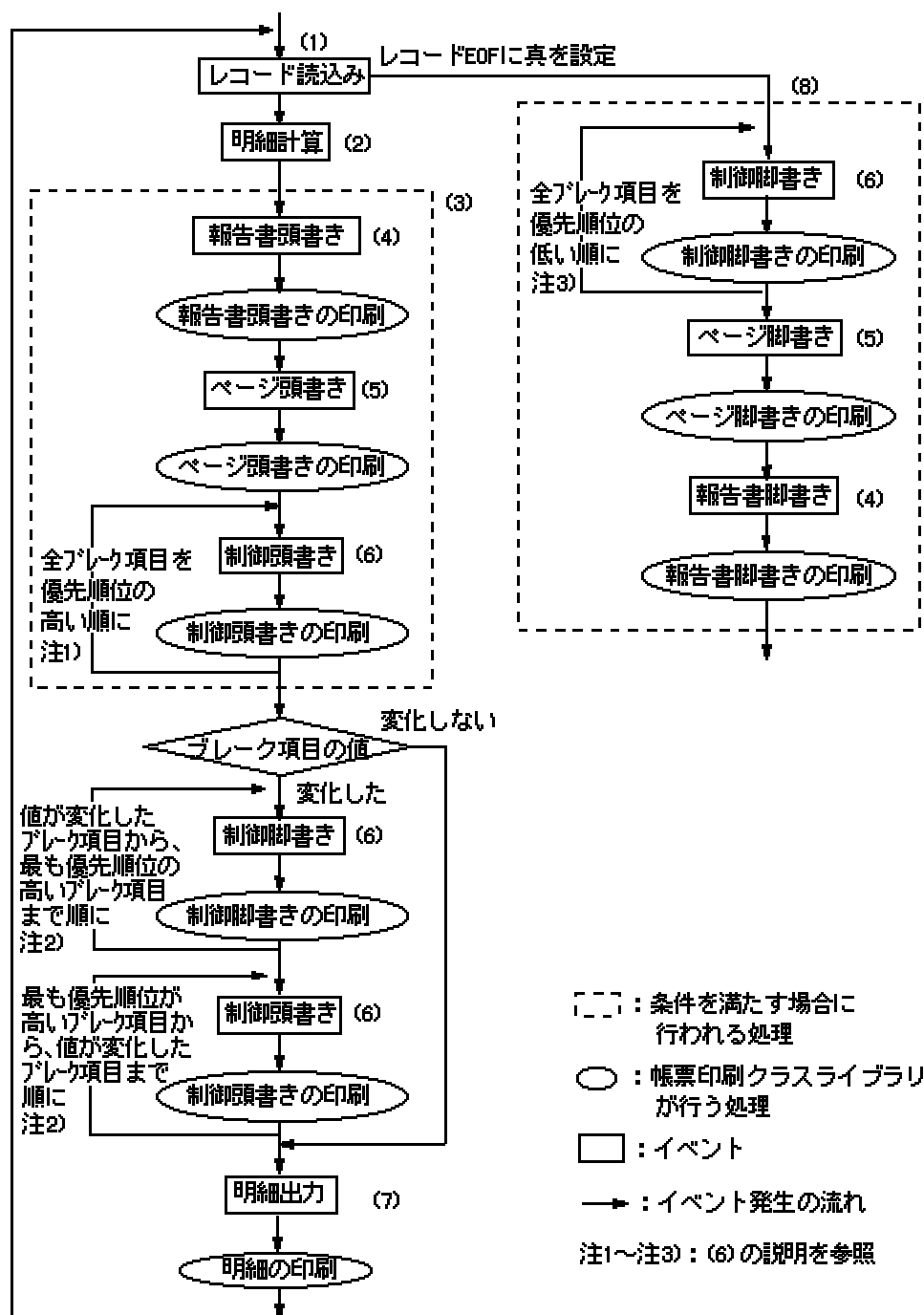


図: 基本的なイベント発生の流れ

### 4.3.2 発生するイベントの種類

基本的な流れで発生するイベントには、次のものがあります。

- [明細計算イベント](#) (CALCULATE-DEメソッド呼出し)
- [明細出力イベント](#) (PRINT-DEメソッド呼出し)
- [報告書頭書きイベント](#) (REPORT-HEADメソッド呼出し)
- [報告書脚書きイベント](#) (REPORT-FOOTメソッド呼出し)
- [ページ頭書きイベント](#) (PAGE-HEADメソッド呼出し)
- [ページ脚書きイベント](#) (PAGE-FOOTメソッド呼出し)

[制御頭書きイベント](#) (CB-HEADメソッド呼出し)

[制御脚書きイベント](#) (CB-FOOTメソッド呼出し)

[レコード読み込みイベント](#) (READ-RECORDメソッド呼出し)

以下に“[図：基本的なイベント発生の流れ](#)”に記述されている番号に対応した説明をします。

### (1) レコード読み込みイベント

レコード読み込みイベントでは、COBOLファイルやデータベースなどから読んできた1レコードの内容を、印刷するために帳票項目に設定する処理などを記述します。COBOLファイルやデータベースの読み込みでレコードやファイルの終端 (EOF) を検出したら、[レコードEOFプロパティ \(RECORD-EOF\)](#) に真を設定してください。

また、集計表形式の帳票定義体で複数の明細パーティションを定義している場合、このイベントで明細名を指定します。印刷で利用する明細は[明細名プロパティ \(DE-NAME\)](#) に値を設定し、指定します。

複数の帳票定義体を利用するために帳票定義体を切り替える場合は、[帳票定義体名プロパティ \(FORM-NAME\)](#) で帳票定義体を切り替えます。この場合のイベント発生の流れについては、“[複数の帳票定義体を利用する場合のイベントの発生](#)”を参照してください。

[帳票印刷メソッド \(PRINT-CONTROL\)](#) の“印刷モード引数”に 1 を指定した場合は、当イベントに対応するイベント処理メソッド (READ-RECORD) の上書き (オーバーライド) を省略することができます。

イベント処理メソッドについては以下を参照してください。

帳票印刷イベント処理クラス (FJPRT-EVENT-HANDLER) の

[レコード読み込みメソッド \(READ-RECORD\)](#)

帳票印刷制御クラス (FJPRT-CONTROL) の

[レコード読み込みメソッド \(READ-RECORD\)](#)

#### [前レコード]

レコード読み込みイベント発生前のレコードの内容が、“前レコード”として保存されます。保存された前レコード内容は、“ブレイク項目”の値が変化したかどうかの判断のために帳票印刷クラスライブラリ内部で使われます。[参照] “[ブレイク項目](#)”，“[前レコード内容の参照メソッド \(GET-PRE-RECORD-CONTENT\)](#)”

### (2) 明細計算イベント

明細計算イベントは、1つの明細/ボディ部/ページ内の、ある項目からほかの項目への計算を行うためなどに利用します。レコード読み込みイベントで値を設定していない明細/ボディ部/ページの項目は、このイベントで値を設定するよう処理を記述してください。

イベント処理メソッドについては以下を参照してください。

帳票印刷イベント処理クラス (FJPRT-EVENT-HANDLER) の

[明細計算メソッド \(CALCULATE-DE\)](#)

帳票印刷制御クラス (FJPRT-CONTROL) の

[明細計算メソッド \(CALCULATE-DE\)](#)

### (3) 最初の1レコードに関する処理

レコード読み込みイベントで設定された最初の1レコードに対して、明細計算イベントのあとに、報告書頭書きなど、先頭に印刷するパーティションの印刷が行われます。つまり、報告書頭書きイベントなどの処理を記述するときに、明細計算イベントで設定された項目の値を参照して利用することができます。

### (4) 報告書頭書きイベントと報告書脚書きイベント

報告書頭書きイベントと報告書脚書きイベントでは、それぞれ報告書頭書きパーティションと報

告書脚書きパーティションの項目に値を設定する処理を記述します。

当イベントは集計表形式の帳票定義体でだけ発生します。報告書頭書きパーティション、報告書脚書きパーティションの有無にかかわらずこれらのイベントは発生します。

イベント処理メソッドについては以下を参照してください。

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[報告書頭書きメソッド\(REPORT-HEAD\)](#)

[報告書脚書きメソッド\(REPORT-FOOT\)](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[報告書頭書きメソッド\(REPORT-HEAD\)](#)

[報告書脚書きメソッド\(REPORT-FOOT\)](#)

## (5) ページ頭書きイベントとページ脚書きイベント

ページ頭書きイベントとページ脚書きイベントは、それぞれページの先頭と末尾に印刷されるページ頭書きパーティションとページ脚書きパーティションの項目に値を設定する処理を記述します。

集計表形式の帳票定義体では、報告書頭書きの印刷のあとに、最初のページのページ頭書きが印刷されます。また、集計表形式の帳票定義体では、同様に最終ページのページ脚書きの印刷のあとに、報告書脚書きが印刷されます。

当イベントは段組み伝票形式または集計表形式の帳票定義体を使った帳票の印刷で発生します。ページ頭書きパーティション、ページ脚書きパーティションの有無にかかわらずイベントは発生します。

イベント処理メソッドについては以下を参照してください。

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[ページ頭書きメソッド\(PAGE-HEAD\)](#)

[ページ脚書きメソッド\(PAGE-FOOT\)](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[ページ頭書きメソッド\(PAGE-HEAD\)](#)

[ページ脚書きメソッド\(PAGE-FOOT\)](#)

## (6) 制御頭書きイベントと制御脚書きイベント

制御頭書きイベントと制御脚書きイベントでは、それぞれ制御頭書きパーティションと制御脚書きパーティションの項目に値を設定する処理を、それぞれのパーティションに対応づけられている“ブレイク項目”ごとに記述します。

イベント発生の対象となったブレイク項目の名前が、当イベントに対応するイベント処理メソッドの引数として渡されます。ブレイク項目ごとの処理の切りわけはこの引数の値を参照して行ってください。

当イベントは集計表形式の帳票定義体でだけ発生します。制御頭書きパーティション、制御脚書きパーティションの有無にかかわらずイベントは発生します。

イベント処理メソッドについては以下を参照してください。

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[制御頭書きメソッド\(CB-HEAD\)](#)

[制御脚書きメソッド\(CB-FOOT\)](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[制御頭書きメソッド\(CB-HEAD\)](#)

[制御脚書きメソッド\(CB-FOOT\)](#)

### [ブレイク項目]

“ブレイク項目”とは、その項目の値が変化したかどうか、明細計算イベントのあとで帳票印刷クラスライブラリによって常にチェックされる項目です。項目の値が変化したか否かの判定は“前レコード”が持つ項目との値の比較によってなされます。ブレイク項目として定義する項目、およびそのブレイク項目に対応する制御頭書きパーティションや制



御脚書きのパーティションは、PowerFORMで帳票定義体を作成/編集するときに定義します。

[参照] “[前レコード](#)”

たとえば、あるブレーク項目の値が変化するときそれまでの小計を印刷したい場合は、次のようにするとよいでしょう。

- 4) あるブレーク項目に対応する制御頭書きイベントで、変数の値を 0にする処理を記述します。
- 5) 明細出力イベントで、変数に値を加算していく処理を記述します。
- 6) あるブレーク項目に対応する制御脚書きイベントで、変数の値を制御脚書きパーティションの項目に設定する処理を記述します。

#### [コントロールブレークの制御方法]

あるブレーク項目の値が変化するときなどに、帳票印刷クラスライブラリは制御頭書きイベントや制御脚書きイベントを発生させます。この動作/制御のことを、“コントロールブレーク”といいます。

コントロールブレークの制御は、次のように処理されます。

最初のレコードを読み込んだとき:

制御頭書きイベントが、すべてのブレーク項目の優先順位が高い順に発生します (“[図: 基本的なイベント発生の流れ](#)”の図中に記述の注1を参照)。

ブレーク項目の値が変化するとき:

まず、一連の制御脚書きイベントが発生してから続いて一連の制御頭書きイベントが発生します。制御脚書きイベントは、値が変化したブレーク項目を始めとし、もっとも優先順位が高いブレーク項目まで優先順位が低い順に発生します。制御頭書きイベントは、最も優先度が高いブレーク項目を始めとし、値が変化したブレーク項目まで優先順位が高い順に発生します (“[図: 基本的なイベント発生の流れ](#)”の図中に記述の注2を参照)。

レコードの読み込みが終了したとき:

制御脚書きイベントが、すべてのブレーク項目の優先順位が低い順に発生します (“[図: 基本的なイベント発生の流れ](#)”の図中に記述の注3を参照)。

なお、ブレーク項目の優先順位は、そのブレーク項目に対応づけられている制御頭書きパーティションと制御脚書きパーティションによって決定します。

制御頭書きイベントの発生時のブレーク項目の優先順位はブレーク項目に対応づけられている制御頭書きパーティションに影響して次のとおりになります。

優先順位	最も高い	...	最も低い
印刷される順番	先	...	後
制御頭書きパーティション名	CH1, CH2, CH3	...	CH14, CH15, CH16

制御脚書きイベントの発生時のブレーク項目の優先順位はブレーク項目に対応づけられている制御脚書きパーティションに影響して次のとおりになります。

優先順位	最も高い	...	最も低い
印刷される順番	後	...	先
制御脚書きパーティション名	CF1, CF2, CF3	...	CF14, CF15, CF16



**注意**

複数のブレーク項目から同じ制御頭書きパーティションに対応づけられている場合と、複数のブレーク項目から同じ制御脚書きパーティションに対応づけられている場合の各ブレーク項目の優先順位は不定となります。帳票定義体を作成するときは、複数のブレーク項目に同じ制御頭書きパーティションや制御脚書きパーティションを対応づけないようにしてください。

下図に、帳票定義体に複数のブレイク項目が定義された場合の例を示します。ここでは、部名と課名をブレイク項目に指定し、それぞれの制御頭書きパーティションと制御脚書きパーティションを定義しています。

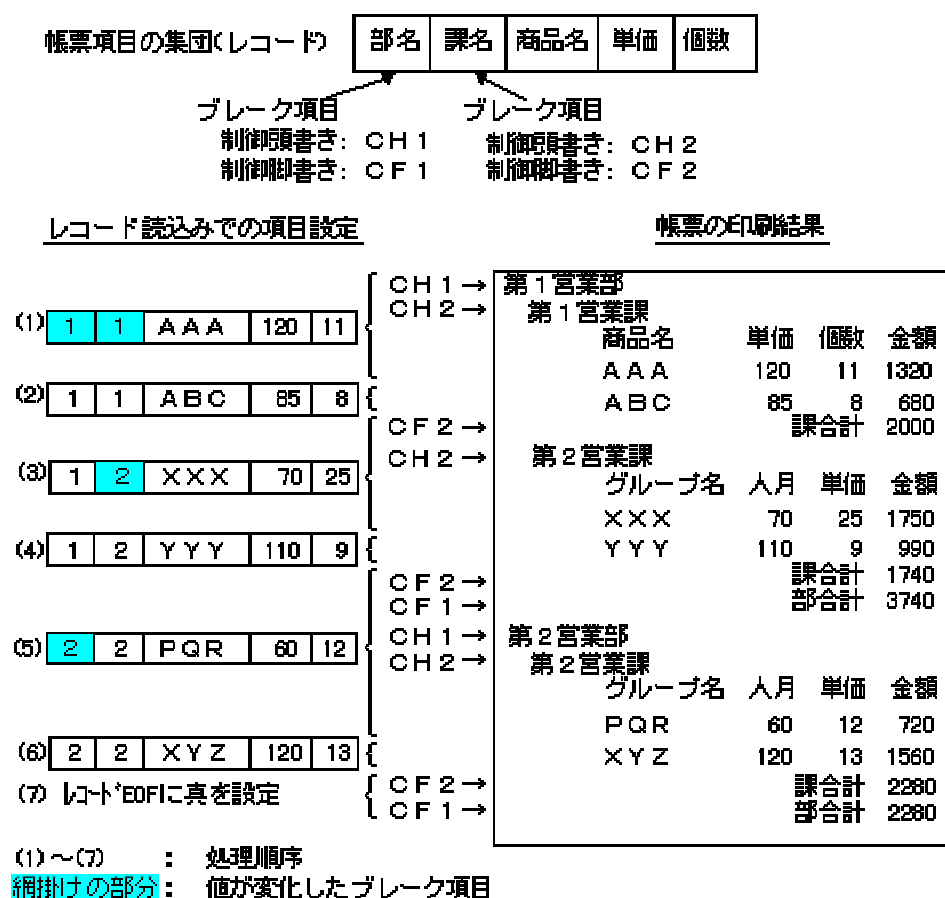


図: コントロールブレイクの制御の例

## (7) 明細出力イベント

明細出力イベントでは、主に各種の脚書きパーティションに印刷する合計を加算する処理を記述します。

また、明細/ボディ部/ページ内の帳票項目の属性を変更したい場合は、このイベントで変更する処理を書きます。

なお、ページの先頭以外で、前の明細/ボディ部のある項目が今回と同じ値なら印刷される文字を変える(たとえば、空白や“ ”などにする)という“集団印字”については、PowerFORMで帳票定義体を編集するときに指定します。このような処理を行いたい場合に集団印字を使えば、このイベントやその他のイベントにて、このような処理を記述する必要はなくなります。

イベント処理メソッドについては以下を参照してください。

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[明細出力メソッド\(PRINT-DE\)](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[明細出力メソッド\(PRINT-DE\)](#)

## (8) 最後のレコードに関する処理

レコード読み込みイベントで[レコードEOFプロパティ\(RECORD-EOF\)](#)に真を設定すると、報告書脚書きなどの印刷が行われます。ただし、レコードが1件もなかった場合、なにも印刷されません。



段組み伝票形式の帳票定義体を使用した場合以下のイベントは発生しません。

報告書頭書きイベント (REPORT-HEADメソッド呼出し)

報告書脚書きイベント (REPORT-FOOTメソッド呼出し)

制御頭書きイベント (CB-HEADメソッド呼出し)

制御脚書きイベント (CB-FOOTメソッド呼出し)

ページ形式とタックシール形式の帳票定義体を使用した場合、以下のイベントは発生しません。

報告書頭書きイベント (REPORT-HEADメソッド呼出し)

報告書脚書きイベント (REPORT-FOOTメソッド呼出し)

制御頭書きイベント (CB-HEADメソッド呼出し)

制御脚書きイベント (CB-FOOTメソッド呼出し)

ページ頭書きイベント (PAGE-HEADメソッド呼出し)

ページ脚書きイベント (PAGE-FOOTメソッド呼出し)

“表: 帳票定義体の種別により発生するイベントの一覧” に一覧表があります。

### 4.3.3 改ページに関するイベント発生の流れ

各パーティションを印刷するときにページ内に入らない場合、改ページしてから印刷されます。

改ページに伴って、ページ脚書きやページ頭書きが印刷されます。

下図に、改ページに関するイベント発生の流れを示します。

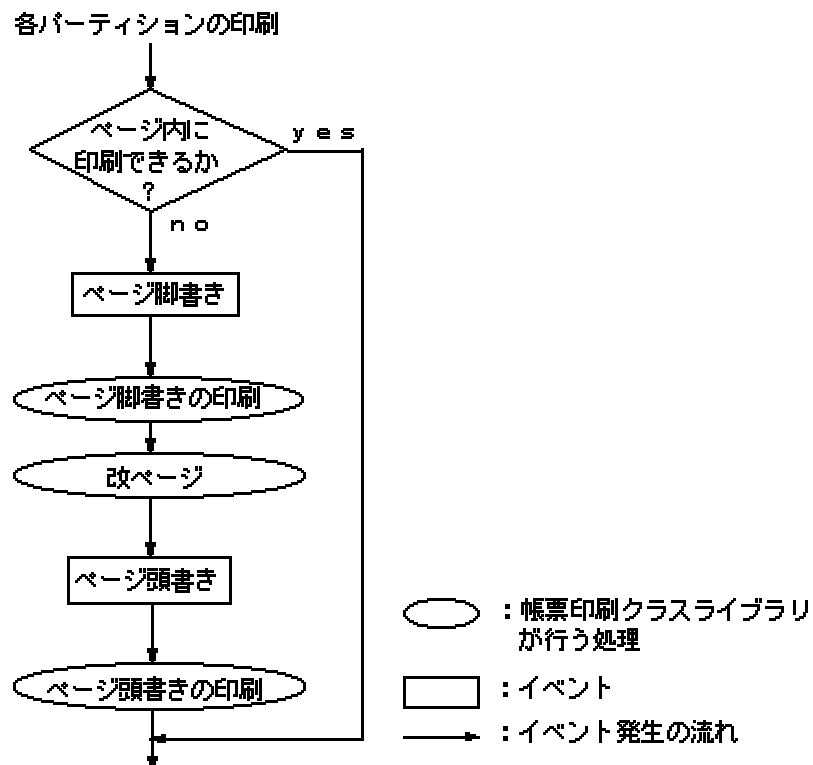


図: 改ページに関するイベント発生の流れ

## 4.4 複数の帳票定義体を利用する場合のイベントの発生

ここでは、複数の帳票定義体を利用する場合のイベント発生の流れについて説明します。  
帳票の印刷を行う1つのアプリケーションプログラムで、複数の帳票定義体を利用することができます。これは、帳票定義体を切り替えることによって実現できます。  
利用する帳票定義体を切り替えるには、以下のイベント内でこれから利用する帳票定義体名を[帳票定義体名プロパティ \(FORM-NAME\)](#)に設定します。

[レコード読み込みイベント](#) (READ-RECORDメソッド内)

なお、現在印刷の対象となっている帳票定義体名を参照する場合は、[帳票定義体名プロパティ \(FORM-NAME\)](#)を参照します。

帳票定義体の切替えを行う場合、レコードを読み込んだ回数によって、処理動作やイベント発生の流れが異なります。最初のレコード読み込みイベントでの切替えの場合と2回目以降のレコード読み込みイベントでの切替えの場合について以下に説明します。

[最初のレコード読み込みイベントでの帳票定義体の切替え](#)

[2回目以降のレコード読み込みイベントでの帳票定義体の切替え](#)



注意

タックシール形式の帳票定義体は、切替え前および切替え後の帳票定義体としては、使用することができません。

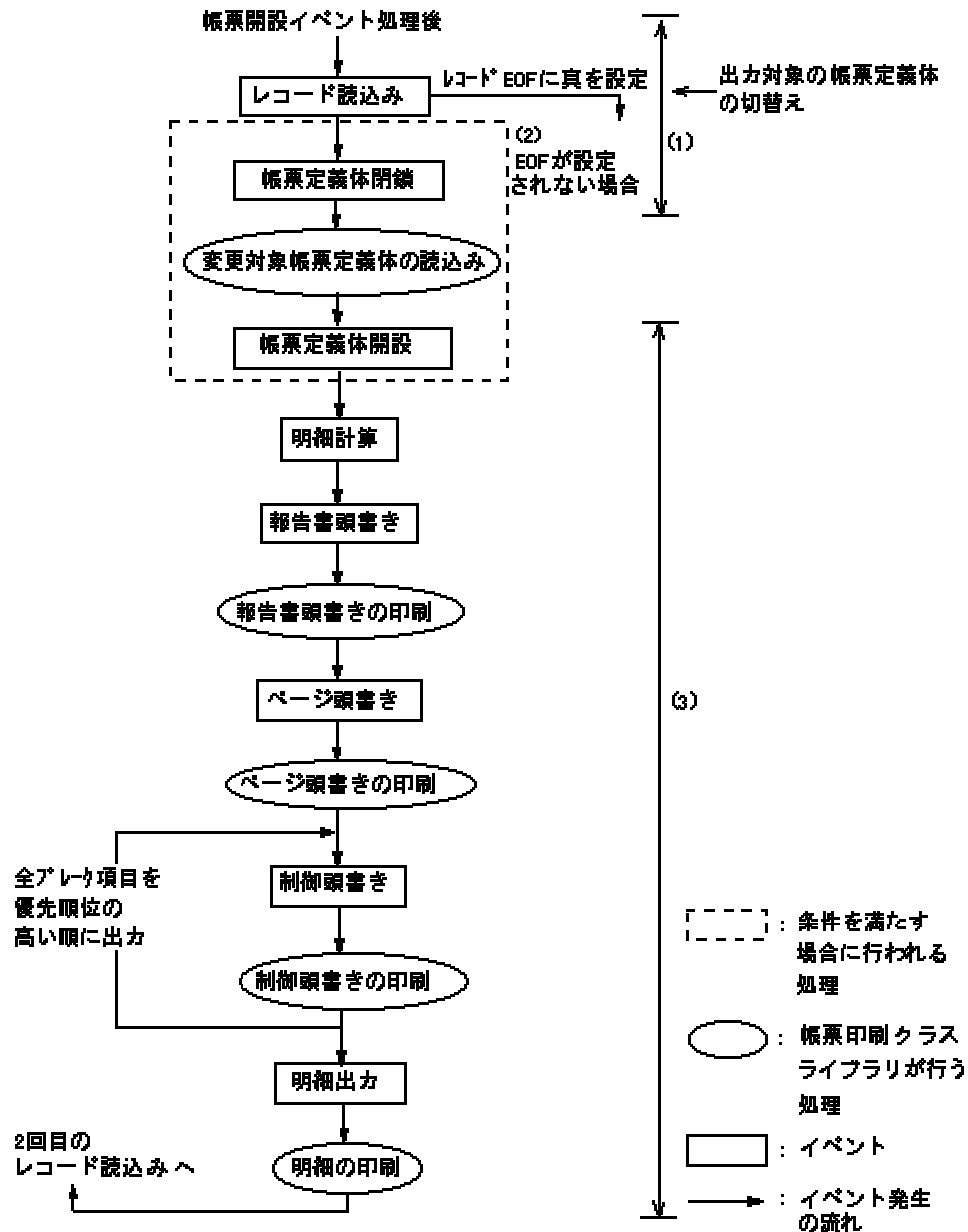
集計表形式の帳票定義体を切り替える場合、切替え前と切替え後の帳票定義体は、以下の点に留意して作成してください。

項目の構成、配置および長さを同じにする。

ブレイク項目に関する定義を同じにする。

### 4.4.1 最初のレコード読み込みイベントでの帳票定義体の切替え

帳票の印刷を開始した後、最初のレコード読み込みイベントで帳票定義体の切替えを行った場合のイベント発生の流れを以下に示します。



図：最初のレコード読み込みイベントで帳票定義体を切り替えた場合のイベント発生の流れ

以下に上図の説明をします。

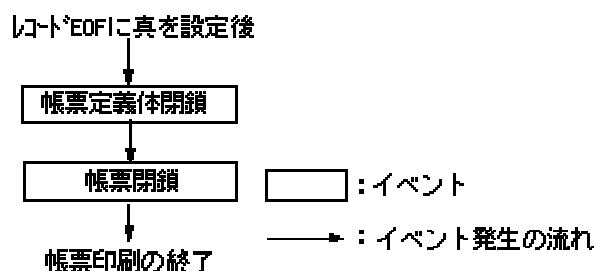
レコード読み込みイベントの処理が終わるまでの(1)の間では、帳票印刷クラスライブラリが認識している帳票定義体は、最初に**帳票定義体名プロパティ (FORM-NAME)**に指定してあった帳票定義体となります。レコード読み込みイベントで新たに設定した帳票定義体はまだ有効になっていません。

初回のレコード読み込みイベントで**レコードEOFプロパティ (RECORD-EOF)**に真(B"1")が設定されなかった場合は、(2)の囲み中の処理を行います。まず、最初に指定された帳票定義体に対して、帳票定義体閉鎖イベントが発生します。その後、帳票印刷クラスライブラリ内部で切替え対象の帳票定義体の読み込みが完了した時点で、切り替える帳票定義体に対する帳票定義体開設イベントが発生します。

帳票定義体の切替えが完了した(3)の間では、レコード読み込みイベントで設定した切替え後の帳票定義体に対してのイベントが発生し、パーティションの印刷などが行われます。

また、初回のレコード読み込みイベントで**レコードEOFプロパティ (RECORD-EOF)**に真(B"1")を設定した場合は、帳票定義体の切替えを行いません。この場合は、帳票定義体を切り替えない場合と

同じイベント発生の流れになります。  
初回のレコード読みイベントでレコードEOFプロパティに真を設定した場合のイベント発生の流れを以下に示します。

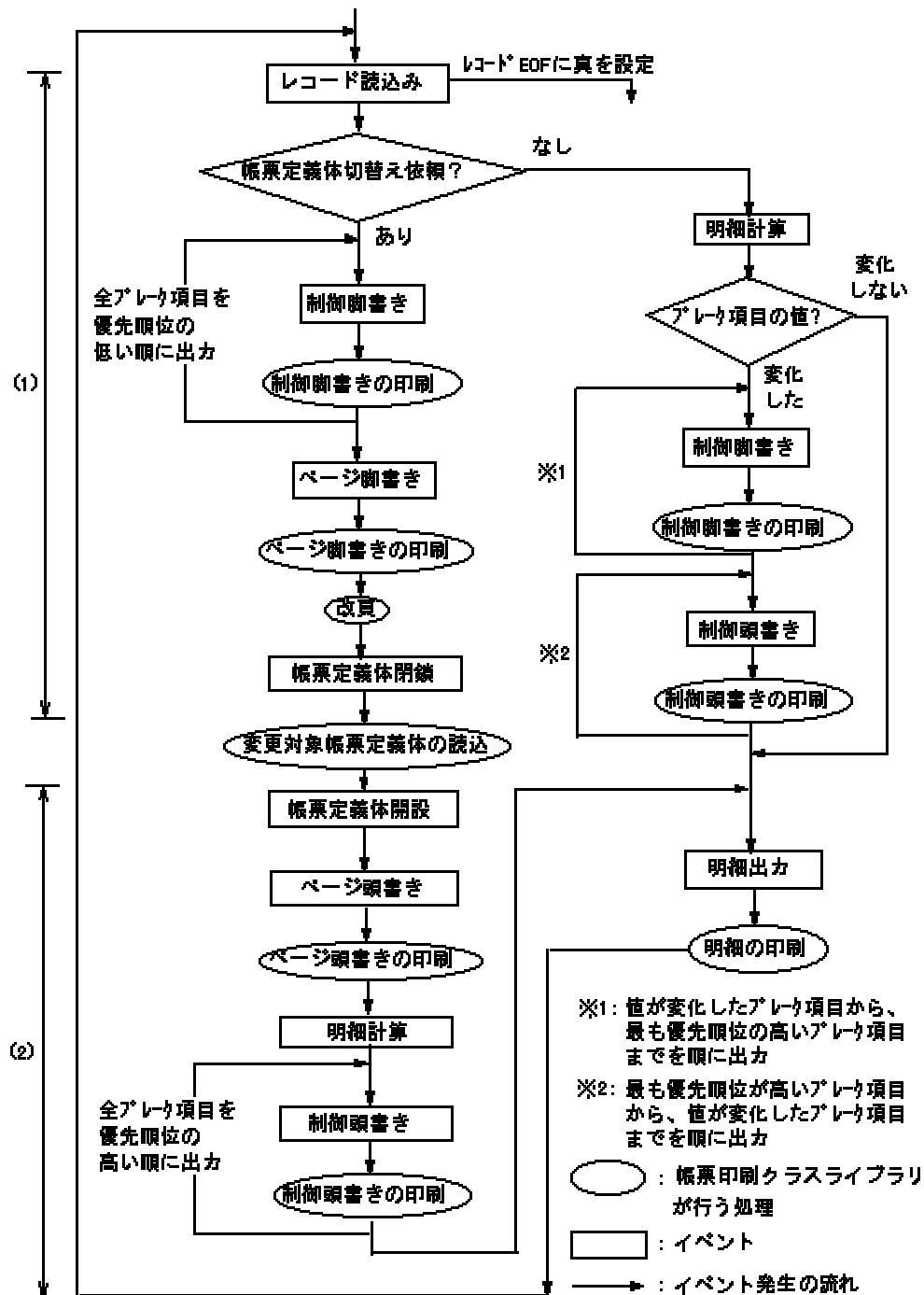


図：初回のレコード読みイベントでEOFが通知された場合のイベント発生の流れ

#### 4.4.2 2回目以降のレコード読みイベントでの帳票定義体の切替え

2回目以降のレコード読みイベントで帳票定義体の切替えを行う場合、切替え前の帳票定義体に対する印刷ページの行の制御を行うために、切替え前の帳票定義体での改ページを行ったあとに切替えが行われます。

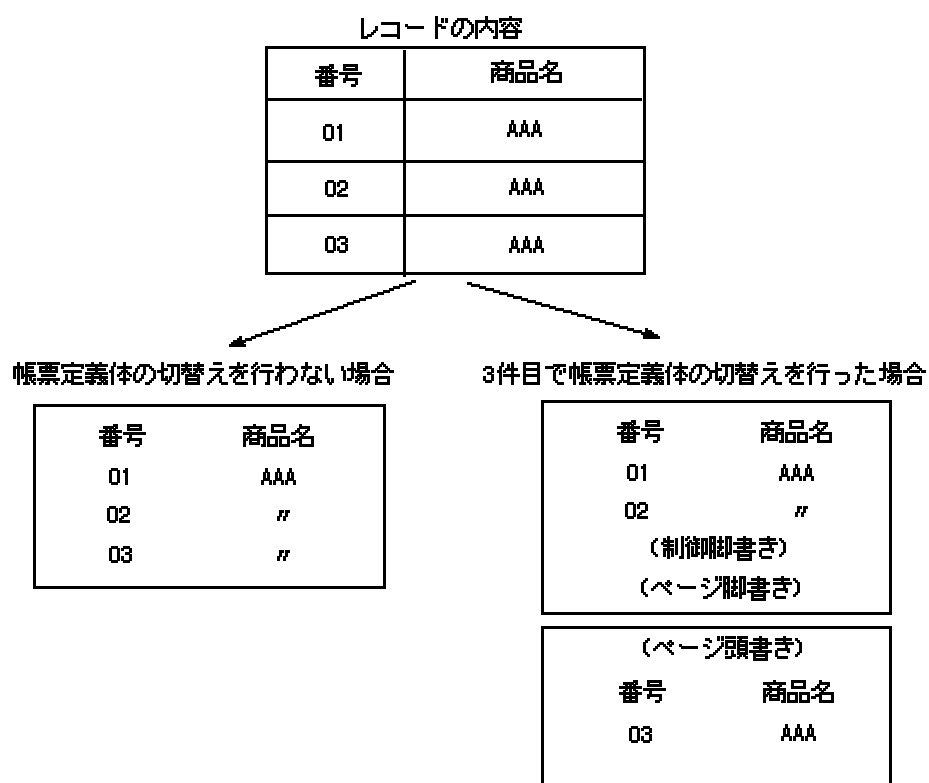
2回目以降のレコード読みイベントで帳票定義体の切替えを行う場合のイベント発生の流れを以下に示します。



図：2回目以降のレコード読み込みイベントで切替えを行う場合のイベント発生の流れ

2回目以降のレコード読み込みイベントで帳票定義体の切替え依頼を行い、改ページ後に帳票定義体の切替えを行う(1)の間では、切替え前の帳票定義体を使用してイベントが発生し、帳票出力を行います。集計表形式の印刷でコントロールブレイクを処理中の場合、全ブレイク項目に対する制御脚書きイベントが最も優先順位が低い方から順に発生して切替え前の帳票定義体に対するブレイク処理を終了します。

帳票定義体の切替えが完了した(2)の間では、切替え後の帳票定義体を使用します。このとき、“集団印刷”の状態は解除されます。たとえば、集計表形式の印刷で明細に集団印刷を指定した帳票項目である“商品名”について、帳票定義体の切替えを行った場合と行わない場合の印刷結果を以下に示します。



図：集団印刷指定した項目の印刷結果

また、レコード読みイベントで[レコードEOFプロパティ \(RECORD-EOF\)](#)に真("1")を設定した場合は、帳票定義体の切替えを行いません。この場合は、帳票定義体を切り替えない場合と同じイベント発生の流れになります。

2回目以降のレコード読みイベントでレコードEOFプロパティに真を設定した場合のイベント発生の流れを以下に示します。



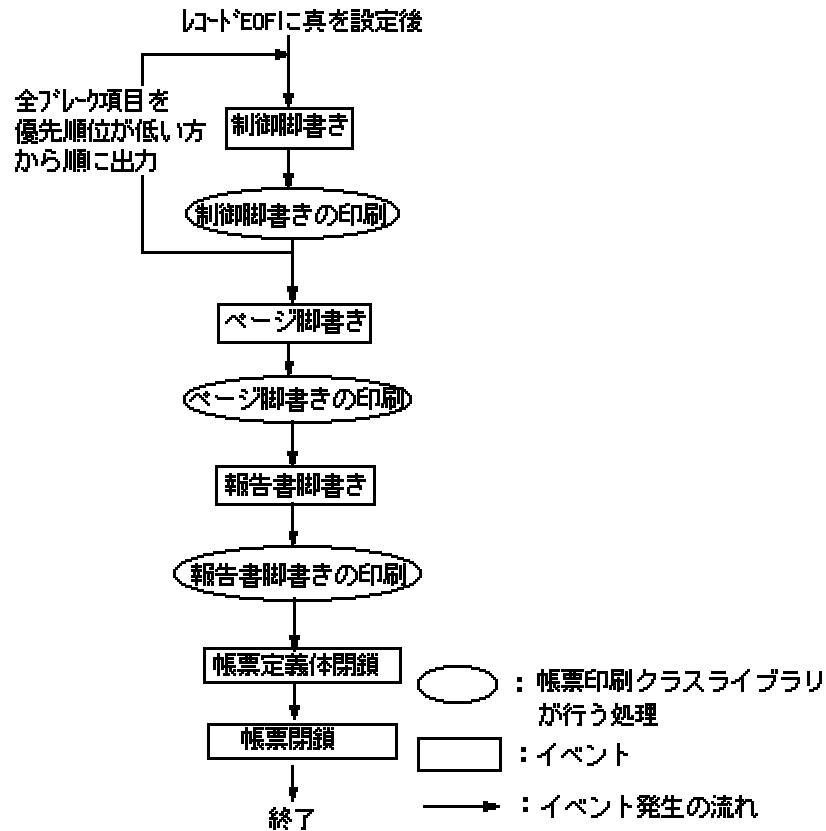


図: 2回目以降のレコード読み込みイベントでEOFを通知した場合のイベント発生の流れ

## 4.5 イベントの発生に影響する事項

ここでは、イベント発生の流れに影響する以下の事項について説明します。

[帳票定義体名プロパティの設定](#)

[印刷種別プロパティの設定](#)

[帳票印刷メソッドの印刷モード引数](#)

### 4.5.1 帳票定義体名プロパティの設定

イベント処理中に[帳票定義体名プロパティ \(FORM-NAME\)](#)に値を設定し直すことにより、イベントの発生に影響があります。これについては“[複数の帳票定義体を利用する場合のイベントの発生](#)”を参照してください。

### 4.5.2 印刷種別プロパティの設定

[印刷種別プロパティ \(PRINT-TYPE\)](#)に設定する種別により、イベントの発生や実際にどのように印刷されるかを次の表に示します。

表：印刷種別

種別	説明
通常印刷 1回読み	基本的なイベントの発生の流れでデータを処理しつつ、帳票定義体の情報に基づいて印刷します。
通常印刷 2回読み 1	1回目に基本的なイベントの発生の流れでデータを処理し、明細の合計などを算出したあと、2回目に再び基本的なイベント発生の流れでデータを処理しつつ、帳票定義体の情報に基づいて印刷します。各種頭書きに明細の合計などを印刷したい場合に使用します。
サマリ印刷 1回読み 2	基本的なイベント発生の流れでデータを処理しつつ、各種の頭書き/脚書きだけを印刷します(明細は印刷されません)。
サマリ印刷 2回読み 1 2	1回目に基本的なイベント発生の流れでデータを処理し、明細の合計などを算出したあと、2回目に再び基本的なイベント発生の流れでデータを処理しつつ、各種の頭書き/脚書きだけを印刷します(明細は印刷されません)。各種の頭書きに明細の合計などを印刷したい場合に使用します。

[補足]

1:

印刷種別の種類が、“通常印刷 2回読み”または“サマリ印刷 2回読み”の場合、帳票開設イベントから帳票閉鎖イベントまでの一連のイベントが 2回発生します。  
(最初に帳票開設イベントから帳票閉鎖イベントまでの一連のイベントが発生します。この間は印刷が行われません。そして再び帳票開設イベントから帳票閉鎖イベントまでの一連のイベントが発生します。このときは印刷が行われます。)

2:

印刷種別の種類の“サマリ印刷 1回読み”または“サマリ印刷 2回読み”は、集計表形式の帳票定義体を使用した場合に有効です。

集計表形式の帳票定義体以外の場合で“サマリ印刷 1回”を指定しても“通常印刷 1回”が指定されたものと見なします。またこの場合、同様に“サマリ印刷 2回”を指定しても“通常印刷 2回”が指定されたものと見なします。

### 4.5.3 帳票印刷メソッドの印刷モード引数

[帳票印刷メソッド\(PRINT-CONTROL\)](#)の第一引数である“印刷モード”には通常、0 を設定します。この場合、レコード読み込みイベントが繰り返し発生するようになります。基本的なイベントの発生については“[基本的なイベント発生の流れ](#)”を参照してください。

印刷処理を終了するためにこのイベントの繰り返しの発生を中断するには、[レコードEOFプロパティ\(RECORD-EOF\)](#)に真(B"1")を設定します。

印刷モード引数に 1を設定した場合、レコード読み込みイベントが一度だけ発生し、プログラムでレコード読み込みイベントの中で[レコードEOFプロパティ\(RECORD-EOF\)](#)に真(B"1")を設定してイベントの繰り返しの発生を抑止することが必要がなくなります。ただし、明細/ボディ部/ページが1件しか処理されずに印刷処理が終了することになります。



---

## 第5章 印刷する帳票項目の操作

---

ここでは、帳票定義体の作成/編集時に定義した、印刷時に任意に値を設定するための項目を操作する方法を説明します。

COBOLの用語の“項目”と区別するため、帳票印刷クラスライブラリでは、この項目のことを特に“帳票項目”と呼んでいます。

帳票項目に印刷したい文字をプログラムから設定/参照する方法や、帳票項目の属性(プロパティ)の設定/参照する方法について、以降で説明します。

[帳票項目の文字の設定/参照](#)

[帳票項目の属性\(プロパティ\)の設定/参照](#)



注意

これらの操作は、必ず特定のイベント処理メソッド内で行います。イベント処理メソッド以外でこれらの操作を行うと[例外エラー](#) (コード: [FJPRT-SCE-ILLEGAL-STATE](#))が発生しますので注意してください。

これらの操作が行えるイベントは以下です。

[帳票定義体開設イベント](#) (OPEN-FORMメソッド)

[帳票定義体閉鎖イベント](#) (CLOSE-FORMメソッド)

[レコード読み込みイベント](#) (READ-RECORDメソッド)

[明細計算イベント](#) (CALCULATE-DEメソッド)

[明細出力イベント](#) (PRINT-DEメソッド)

[報告書頭書きイベント](#) (REPORT-HEADメソッド)

[報告書脚書きイベント](#) (REPORT-FOOTメソッド)

[ページ頭書きイベント](#) (PAGE-HEADメソッド)

[ページ脚書きイベント](#) (PAGE-FOOTメソッド)

[制御頭書きイベント](#) (CB-HEADメソッド)

[制御脚書きイベント](#) (CB-FOOTメソッド)

---

## 5.1 帳票項目の文字の設定/参照

帳票印刷クラスライブラリは、MeFtとの情報の受け渡しのために“レコード”を管理しています。レコードとは、入出力ランタイム製品である MeFtと帳票項目の文字をやりとりするための領域です。

当クラスライブラリを使って帳票項目の値を設定/参照するには間接的にこのレコードを介して行われることになります。レコードについては“MeFt 説明書”の“レコードとは”を参照してください。

帳票項目の文字を設定/参照するには次の方法があります。

[COPY文で取り込んだ帳票定義体の集団項目を使っている設定/参照](#)

### 5.1.1 COPY文で取り込んだ帳票定義体の集団項目を使っている設定/参照

ここでは COPY文の記述のしかたと、帳票項目の文字の設定/参照および、設定/参照に必要な帳票印刷クラスライブラリとのレコード(帳票の集団項目)の受け渡しの方法について説明します。

[COPY文の記述](#)

[帳票印刷クラスライブラリとの集団項目の受け渡し](#)

[集団項目が持つ項目の文字の設定/参照](#)

#### 5.1.1.1 COPY文の記述

イベント処理メソッドが含まれるクラスのオブジェクトデータ項目として、帳票定義体を COPY文で取り込む記述をします。

この場合の、COPY文の構文は次のとおりです。COPY文の構文については“COBOL 文法書”を参照してください。

```
COPY FORM-NAME OF XMDLIB [REPLACING FORM-NAME BY NEW-NAME].
```

【補足】

**FORM-NAME :**

帳票定義体の名前を記述します。帳票定義体の取り込みにより宣言される集団項目の名前と帳票定義体の名前は同一になります。

**NEW-NAME :**

[前レコード](#)の内容を取得する場合のための集団項目の名前を記述します。

以下に COBOLソースの例を挙げます。[帳票印刷イベント処理クラス\(FJPRT-EVENT-HANDLER\)](#)を継承したクラスの記述例です。

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER-CLASS INHERITS FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS FJPRT-EVENT-HANDLER
    CLASS FJPRT-CONTROL
    CLASS FJPRT-EXCEPTION
    .
SPECIAL-NAMES.
    SYMBOLIC CONSTANT
    COPY  FJPRTSC.
    .
IDENTIFICATION DIVISION.
OBJECT.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  USERFORM OF XMDLIB.
    :
END CLASS  USER-HANDLER-CLASS.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```



#### 注意

複数の帳票定義体を切り替えて利用するクラスの場合、COPY文による帳票定義体の取り込みを複数記述する必要があります。

以下の例のように COPY文を列挙してください。

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
:
COPY  FORM-A OF XMDLIB.
COPY  FORM-B OF XMDLIB.
COPY  FORM-C OF XMDLIB.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

[前レコード](#)の内容を参照する処理を書く場合、REPLACING 指定を行って集団項目の名前を決めてください。

COPY文 の記述の例を以下に挙げます。

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
:
*> GET-RECORD-CONTENT, SET-RECORD-CONTENTメソッドで利用する集団項目
*> 集団項目名 FORM-A
   COPY FORM-A OF XMDLIB.

*> GET-PRE-RECORD-CONTENTメソッドで利用する集団項目
*> 集団項目名 FORM-A-PRE
   COPY FORM-A OF XMDLIB REPLACING FORM-A BY FORM-A-PRE.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

### 5.1.1.2 帳票印刷クラスライブラリとの集団項目の受け渡し

イベント処理メソッドで帳票項目の文字を設定する場合は、帳票項目への設定の前と設定の後にそれぞれ、以下のメソッド呼出しを記述します。

イベント処理メソッドで帳票項目の文字を参照だけする場合には、帳票項目の参照の前に以下のメソッド呼出しを記述してください。

これらの操作は、帳票印刷クラスライブラリが管理するレコード(帳票項目の集団)の内容を参照するためと、設定した文字を集団として帳票印刷クラに通知する(レコードに集団項目の内容を設定する)ために必要です。

設定/参照の前 (帳票印刷クラスライブラリが管理する帳票項目の集団を参照する)

```

INVOKE CONTROL-OBJECT-2 "GET-RECORD-CONTENT"
      USING [BY REFERENCE] FORM-NAME

```

設定の後 (設定した文字を集団として帳票印刷クラスライブラリに通知する)

```

INVOKE CONTROL-OBJECT-2 "SET-RECORD-CONTENT"
      USING [BY REFERENCE] FORM-NAME

```

【補足】

**FORM-NAME :**

帳票定義体の取込みで定義した集団項目の名前を記述します。

また、[レコード読み込みイベント](#)発生前に保存された帳票項目の文字(これを[前レコード](#)と呼びます)を参照することができます。

[前レコード](#)の帳票項目の文字を参照する場合には、イベント処理メソッドの参照の前に以下のメソッド呼出しを記述してください。

この操作は、帳票印刷クラスライブラリが持つ前レコードを COBOL側に通知するために必要に応じて使います。

参照の前 (帳票印刷クラスライブラリ が持つ帳票項目の集団を COBOL側に通知する)

```

INVOKE CONTROL-OBJECT-2 "GET-PRE-RECORD-CONTENT"
      USING [BY REFERENCE] GROUP-ITEM

```

【補足】

**GROUP-ITEM :**

帳票定義体の取込みで定義した集団項目の名前を記述します。



注意

複数の帳票定義体を切り替えて利用する場合、イベント処理メソッドの中で**帳票定義体名プロパティ(FORM-NAME)**を参照して扱う帳票定義体を意図的に切りわけるように処理を記述してください。

以下に COBOLソースの例を挙げます。 [帳票印刷制御クラス\(FJPRT-CONTROL\)](#) スライブラリ



を継承したクラスのイベント処理メソッド内の記述例です。

```
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
:
EVALUATE FORM-NAME OF SELF
WHEN      "FORM-A"
    INVOKE SELF "GET-RECORD-CONTENT" USING FORM-A
*>    FORM-Aの帳票項目への文字の設定
    MOVE "HAMAMATSU" TO CITY-NAME OF FORM-A
:
    INVOKE SELF "SET-RECORD-CONTENT" USING FORM-A
WHEN      "FORM-B"
    INVOKE SELF "GET-RECORD-CONTENT" USING FORM-B
*>    FORM-Bの帳票項目への文字の設定
    MOVE "SHIZUOKA" TO PREF-NAME OF FORM-B
:
    INVOKE SELF "SET-RECORD-CONTENT" USING FORM-B
END-EVALUATE
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
```

### 5.1.1.3 集団項目が持つ項目の文字の設定/参照

集団項目が持つ項目の文字を参照、設定するためには MOVE文を利用します。

以下のように記述してください。

#### 【設定】

```
MOVE ANY-VALUE TO FIELD-NAME OF FORM-NAME [(INDEX1 [INDEX2 [INDEX3]])]
```

#### 【参照】

```
MOVE FIELD-NAME OF FORM-NAME [(INDEX1 [INDEX2 [INDEX3]])] TO ANY-ITEM
```

#### 【補足】

##### FORM-NAME :

帳票定義体の取り込みで宣言された集団項目名を記述します。

##### FIELD-NAME :

帳票定義体の作成/編集時に定義した帳票項目の名前を記述します。

##### ANY-VALUE :

MOVE文の送り出し側に記述できる語を記述します。ただし、FIELD-NAME OF FORM-NAME の属性と一致している必要があります。

##### ANY-ITEM :

MOVE文の受け取り側に記述できる語(主にデータ項目)を記述します。ただし、FIELD-NAME OF FORM-NAME の属性と一致している必要があります。

##### INDEX1 :

FIELD-NAME が、一次元以上の繰返し構造を持つ場合、一次元目の添え字を与えます。

##### INDEX2 :

FIELD-NAME が、二次元以上の繰返し構造を持つ場合、二次元目の添え字を与えます。

##### INDEX3 :

FIELD-NAME が、三次元の繰返し構造を持つ場合、三次元目の添え字を与えます。



注意

この帳票項目の集団項目への設定/参照の前後に、帳票印刷クラスライブラリから集団項目への帳票項目集団の通知や、COBOL側から帳票印刷クラスライブラリへの集団項目の通知が必要となります。詳細については“[帳票印刷クラスライブラリとの集団項目の受け渡し](#)”を参照してください。

## 5.2 帳票項目の属性(プロパティ)の設定/参照

帳票項目が持つ文字以外の情報として、以下があります。これらは帳票印刷クラスライブラリが帳票項目のプロパティとして保持しています。

文字色、項目抑止、文字引伸し、抹消線、イタリック、ボールド、下線、項目下線。



これらのプロパティは、特殊レジスタ(EDIT-XXXX)を使って設定/参照することはできませんので注意してください。

ファイルインタフェースでは MeFt とのインタフェースであるレコードが項目制御部を持っています。帳票印刷クラスライブラリが使うクラスインタフェースでは、項目制御部はレコードから切り離されています。切り離された項目制御部は帳票印刷クラスライブラリがプロパティとして管理しています。

これらの帳票項目のプロパティを設定/参照するには次の方法があります。

[帳票項目名によるプロパティの設定/参照](#)

### 5.2.1 帳票項目名によるプロパティの設定/参照

ここでは帳票項目名を保持するデータ項目の宣言のしかたと、帳票項目のプロパティの設定/参照の方法について説明します。

[帳票項目名を格納する領域の宣言](#)

[メソッドによる帳票項目プロパティの設定/参照](#)

#### 5.2.1.1 帳票項目名を格納する領域の宣言

帳票項目名を使ってプロパティの設定/参照を行う場合、帳票項目の名前をデータ項目に持たせる必要があります。

帳票項目は“[繰返し回数を持たない場合](#)”と、“[繰返し回数を持つ場合](#)”があります。これによって記述方法が異なります。

##### c. 帳票項目が繰返し回数を持たない場合:

帳票項目の名前を格納するデータ項目は、オブジェクト定義の作業場所節に宣言します  
(1)。このときデータ項目は英数字の属性で、帳票項目名が入りきる長さで宣言し、初期値として帳票項目の名前を持たせます(2)。

以下に例を示します。

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
:
IDENTIFICATION DIVISION.
OBJECT.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.  *> (1)
    COPY  USERFORM OF XMDLIB.
    01  TANKA-NAME  PIC  X(5) VALUE  "TANKA".  *> (2)
    01  GOUKEI-NAME PIC  X(6) VALUE  "GOUKEI". *> (2)
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```



#### 注意

帳票項目が英字名るとき、帳票定義体の作成/編集時に定義したとおりに大文字/小文字をあわせて記述してください。

帳票項目名が日本語であるときは、次のように記述してください。

動作モードがシフトJISのときに帳票項目名の長さの2倍の長さを持つ英数字項目を宣言します。動作モードがUnicodeのときに帳票項目名の長さの3倍の長さを持つ英数字項目を宣言します。どちらの場合も、その初期値として日本語の文字を持つ英数字の属性の定数を記述します。

なお、英数字項目の長さを帳票項目名の3倍の長さで宣言すると、どちらの動作モードにも対応できるようになります。以下に例を示します。

```

*..1.*.....2.....3.....4.....5.....6.....7..*
:
01  単価 - 名    PIC  X(6) VALUE  "単価".
01  製品名 - 名  PIC  X(9) VALUE  "製品名".
:
*..1.*.....2.....3.....4.....5.....6.....7..*

```

#### d. 帳票項目が繰返し回数を持つ場合:

帳票項目の名前を格納するデータ項目は、オブジェクト定義の作業場所節に宣言します **(1)**。このときデータ項目は集団項目で宣言します **(2)**。

集団項目の先頭には、帳票項目名が入りきる長さの英数字の属性の FILLER項目を宣言します。初期値として帳票項目の名前を持たせます **(3)**。

2番目には、初期値に左括弧("(")を持つ英数字の属性の FILLER項目を記述します **(4)**。

3番目には、任意の桁数を持つ外部10進データ項目を記述します **(5)**。最後に、初期値に右括弧(")")を持つ英数字の属性の FILLER項目を記述します **(6)**。

以下に例を示します。

```

*..1.*.....2.....3.....4.....5.....6.....7..*
:
IDENTIFICATION DIVISION.
OBJECT.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.  *> (1)
COPY USERFORM OF XMDLIB.
01 TANKA-NAME.  *> (2)
02 FILLER PIC X(5) VALUE "TANKA".  *> (3)
02 FILLER PIC X(1) VALUE "(".  *> ....(4)
02 TANKA-INDEX PIC 9(2).  *> .....(5)
02 FILLER PIC X(1) VALUE ")".  *> ....(6)
:
*..1.*.....2.....3.....4.....5.....6.....7..*

```



注意

帳票項目が 2重に繰返し回数を持つ場合、あるいは 3重に繰返し回数を持つ場合は、それぞれ次のように記述してください。

－ 帳票項目が **2重に繰返し回数を持つ場合**：

集団項目の 3番目と 5番目の項目は、任意の桁数を持つ外部10進データ項目を記述します(1)。

集団項目の 4番目の項目は、初期値にカンマ(",")を持つ英数字の属性の FILLER項目を記述します(2)。

以下に例を挙げます。

```

*..1.*.....2.....3.....4.....5.....6.....7..*
:
01 TABLE-2D-NAME.
02 FILLER PIC X(8) VALUE "TABLE-2D".
02 FILLER PIC X(1) VALUE "(".
02 TABLE-2D-INDEX-1 PIC 9(2).  *> ..(1)
02 FILLER PIC X(1) VALUE ", ".  *> (2)
02 TANKA-2D-INDEX-2 PIC 9(2).  *> ..(1)
02 FILLER PIC X(1) VALUE ")".
:
*..1.*.....2.....3.....4.....5.....6.....7..*

```

－ 帳票項目が **3重に繰返し回数を持つ場合**：

集団項目の 3番目と 5番目と 7番目の項目は、任意の桁数を持つ外部10進データ項目を記述します(1)。

集団項目の 4番目と 6番目の項目は、初期値にカンマ(",")を持つ英数字の属性の FILLER項目を記述します(2)。

以下に例を挙げます。

```
*..1.*.....2.....3.....4.....5.....6.....7..*
:
01 三次元 - 名.
02  FILLER  PIC  X(9)  VALUE  "三次元".
02  FILLER  PIC  X(1)  VALUE  "(".
02  三次元 - 指標 1  PIC  9(2).  *> ....(1)
02  FILLER  PIC  X(1)  VALUE  ", ". *> (2)
02  三次元 - 指標 2  PIC  9(2).  *> ....(1)
02  FILLER  PIC  X(1)  VALUE  ", ". *> (2)
02  三次元 - 指標 3  PIC  9(2).  *> ....(1)
02  FILLER  PIC  X(1)  VALUE  ")".
:
*..1.*.....2.....3.....4.....5.....6.....7..*
```

### 5.2.1.2 メソッドによる帳票項目プロパティの設定/参照

帳票項目のプロパティの設定/参照を行うためには、[帳票項目に関するプロパティ設定/参照メソッド](#)を使います。

プロパティ設定/参照メソッドの第一引数に “ [帳票項目名を格納する領域の宣言](#) ” で定義したデータ項目を渡してください。

以下に例を挙げます。

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID. USER-HANDLER-CLASS INHERITS FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY FJPRTSC.
.
REPOSITORY.
CLASS FJPRT-EVENT-HANDLER
CLASS FJPRT-CONTROL
CLASS FJPRT-EXCEPTION
.
IDENTIFICATION DIVISION.
OBJECT.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
COPY USERFORM OF XMDLIB.
01 TANKA-NAME PIC X(5) VALUE "TANKA".
01 合計 - 名 PIC X(6) VALUE "合計".
:
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID. PRINT-DE OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
01 L-CTRLOBJ OBJECT REFERENCE FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
:
INVOKE L-CTRLOBJ "SET-TEXT-COLOR"
USING TANKA-NAME FJPRT-SC-COLOR-RED
INVOKE L-CTRLOBJ "SET-TEXT-COLOR"
USING 合計 - 名 FJPRT-SC-COLOR-BLUE
:
EXIT METHOD.
END METHOD PRINT-DE.
:
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

帳票項目は繰り返し項目を持つことがあります。この場合は帳票項目名となる集団項目が持つ外部10進項目に繰り返しによる指標値を設定して(1)からメソッドを呼び出してください(2)。以下に例を示します。

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
:
IDENTIFICATION DIVISION.
OBJECT.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY USERFORM OF XMDLIB.
    01 配列 - 名.
        02 FILLER PIC X(6) VALUE "配列".
        02 FILLER PIC X(1) VALUE "(" .
        02 配列 - 指標 PIC 9(2). *> (1)
        02 FILLER PIC X(1) VALUE ")".
:
    MOVE 1 TO 配列 - 指標 *> (1)
    INVOKE L-CTRLJOB "SET-TEXT-COLOR"
        USING 配列 - 名 FJPRT-SC-COLOR-RED *> (2)
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```



注意

繰り返し項目のための指標値の基数は 1 です。0 ではありませんので注意してください。



---

## 第6章 エラー処理

---

ここでは、帳票印刷クラスライブラリを利用する上で発生するエラーに対する処理について説明します。

印刷ジョブエラーコードと、例外エラーコードについては“ [エラーコード一覧](#) ”を参照してください。

帳票印刷クラスライブラリを使ったプログラミングでは必要に応じて以下のエラー処理を記述します。

[例外エラー](#)

[印刷エラー](#)

---

## 6.1 例外エラー

例外エラーとは、[帳票印刷例外クラス \(FJPRT-EXCEPTION\)](#) のオブジェクトを使ってエラーを通知することを言います。

例外エラーを知らせる方法は以下があります。

例外オブジェクト

[印刷制御オブジェクト](#)のプロパティやメソッドにアクセスすると、例外エラーが発生することがあります。

具体的な例外エラーの種類については“[例外エラーコード](#)”を参照してください。

プロパティやメソッドへのアクセスで発生した例外エラーを“例外オブジェクト”を使って検出するには、そのアクセスを行っているプログラム定義やイベント処理メソッドなどの手続き部に USE 文を使った“例外処理”を記述する必要があります。USE 文を使った例外処理の記述については“[NetCOBOL 使用手引書](#)”の“例外オブジェクト”を参照してください。



注意

例外オブジェクトが発生するプロパティやメソッドアクセスを行うプログラム定義やイベント処理メソッド内で USE 文を使った例外処理を記述しなかった場合は、例外オブジェクトの発生でプログラムが実行時エラーが発生しますので注意してください。

例外オブジェクトを使った例外エラーに対する処理の例として、“[a. プログラム定義で例外処理を記述する場合の例](#)”と“[b. イベント処理メソッドで例外処理を記述する場合の例](#)”を挙げます。

### e. プログラム定義で例外処理を記述する場合の例

```

*>.1.*.....2.....3.....4.....5.....6.....7...*.....8
*> プログラム定義で例外処理を記述する場合の例
IDENTIFICATION DIVISION.
PROGRAM-ID. EXCEPTION-SAMPLE.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS FJPRT-CONTROL
    CLASS USER-HANDLER-CLASS *> FJPRT-EVENT-HANDLERを継承した子クラス
    CLASS FJPRT-EXCEPTION *>.....[1]
.
SPECIAL-NAMES.
    SYMBOLIC CONSTANT
    COPY FJPRTSC. *> 例外エラーコード用記号定数などの取込み ....[2]
.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01 W-CONTROL-OBJ OBJECT REFERENCE FJPRT-CONTROL.
    01 W-HANDLER-OBJ OBJECT REFERENCE USER-HANDLER-CLASS.
    01 W-CODE          PIC S9(9) COMP-5.
    01 W-MESSAGE      PIC X(256).
    01 W-MSG-LENG     PIC S9(9) COMP-5.
    01 W-CLASS-NAME   PIC X(30).
    01 W-CLASS-LENG   PIC S9(9) COMP-5.
    01 W-METHOD-NAME PIC X(30).
    01 W-METHOD-LENG PIC S9(9) COMP-5.
PROCEDURE DIVISION.
DECLARATIVES. *> .....[3]
ANY-SECTION-NAME SECTION. *> 任意に節名を付ける .....[4]
    USE AFTER EXCEPTION FJPRT-EXCEPTION. *> .....[5]
    DISPLAY NC"例外エラーが発生しました。"
*> クラス名の参照
    INVOKE EXCEPTION-OBJECT "GET-CLASS-NAME"
        USING W-CLASS-NAME RETURNING W-CLASS-LENG *> ....[6]
    DISPLAY " CLASS-NAME: " W-CLASS-NAME (1 : W-CLASS-LENG)
*> メソッド名の参照
    INVOKE EXCEPTION-OBJECT "GET-METHOD-NAME"
        USING W-METHOD-NAME RETURNING W-METHOD-LENG *> ...[7]
    DISPLAY " METHOD-NAME: " W-METHOD-NAME (1 : W-METHOD-LENG)
*> メッセージの参照
    INVOKE EXCEPTION-OBJECT "GET-MESSAGE"
        USING W-MESSAGE RETURNING W-MSG-LENG *> .....[8]
    DISPLAY " MESSAGE: " W-MESSAGE (1 : W-MSG-LENG)
*> コードの参照
    INVOKE EXCEPTION-OBJECT "GET-CODE"
        RETURNING W-CODE *> .....[9]
    EVALUATE W-CODE
    WHEN FJPRT-SCE-SET-VALUE
        DISPLAY " CODE: FJPRT-SCE-SET-VALUE"
    WHEN FJPRT-SCE-FIELD-NAME
        DISPLAY " CODE: FJPRT-SCE-FIELD-NAME"

```

```

WHEN      FJPRT-SCE-ILLEGAL-STATE
  DISPLAY " CODE: FJPRT-SCE-ILLEGAL-STATE"
WHEN      FJPRT-SCE-ARRAY-INDEX
  DISPLAY " CODE: FJPRT-SCE-ARRAY-INDEX"
WHEN      FJPRT-SCE-INTERNAL-ERROR
  DISPLAY " CODE: FJPRT-SCE-INTERNAL-ERROR"
WHEN      FJPRT-SCE-INVALID-OBJECT
  DISPLAY " CODE: FJPRT-SCE-INVALID-OBJECT"
END-EVALUATE
.
END DECLARATIVES.  *> .....[10]
  INVOKE FJPRT-CONTROL      "NEW" RETURNING W-CONTROL-OBJ
  INVOKE USER-HANDLER-CLASS "NEW" RETURNING W-HANDLER-OBJ
  SET EVENT-HANDLER OF W-CONTROL-OBJ TO W-HANDLER-OBJ
  INVOKE W-CONTROL-OBJ "PRINT-CONTROL"
    USING 0 RETURNING W-RESULT
  SET W-CONTROL-OBJ TO NULL
  SET W-HANDLER-OBJ TO NULL
END PROGRAM EXCEPTION-SAMPLE.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

## [例の説明]

[1]

利用するクラス名として REPOSITORY段落に FJPRT-EXCEPTION を記述します。

[2]

環境部構成節特殊名段落の SYMBOLIC CONSTANT句に、FJPRTSC.cblを取り込むように COPY文を記述します。詳細については“[COPY文による記号定数の取込み](#)”を参照してください。

[3]

手続き部の先頭に DECLARATIVES を記述します。

[4]

DECLARATIVES に続き、節名を記述します。

[5]

節名に続き、USE文 を記述します。

[6]

例として、例外オブジェクトを使ってクラス名の参照をしています。

[7]

例として、例外オブジェクトを使ってメソッド名の参照をしています。

[8]

例として、例外オブジェクトを使ってメッセージの参照をしています。

[9]

例として、例外オブジェクトを使ってコードの参照をしています。

[10]

手続きの始めの直前に、END DECLARATIVES を記述します。

## f. イベント処理メソッドで例外処理を記述する場合の例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
*> イベント処理メソッドで例外処理を記述する場合の例
IDENTIFICATION DIVISION.
CLASS-ID USER-HANDLER-CLASS INHERITS FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS FJPRT-EVENT-HANDLER
    CLASS FJPRT-CONTROL
    CLASS FJPRT-EXCEPTION *>.....[1]
.
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY FJPRTSC. *> 例外エラーコード用記号定数などの取込み ...[2]
.
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
*> 帳票定義体を登録集として取り込み
COPY FORM1 OF XMDLIB.
*> 帳票項目名
01 W-FIELD1-NAME PIC X(6) VALUE "FIELD1".
01 W-FIELD2-NAME.
    02 FILLER PIC X(6) VALUE "FIELD2".
    02 FILLER PIC X(1) VALUE "(".
    02 W-FIELD2-INDEX PIC 9(2).
    02 FILLER PIC X(1) VALUE ")".
*>
PROCEDURE DIVISION.
*>
*> 例外エラーの情報を表示するメソッド
IDENTIFICATION DIVISION.
METHOD-ID. EXCEPTION-DISPLAY.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01 W-MESSAGE PIC X(256).
    01 W-METHOD-NAME PIC X(30).
    01 W-LENG PIC S9(9) COMP-5.
LINKAGE SECTION.
    01 L-EXCEPTION-OBJ OBJECT REFERENCE FJPRT-EXCEPTION.
PROCEDURE DIVISION USING L-EXCEPTION-OBJ.
*> メソッド名の参照
    INVOKE L-EXCEPTION-OBJ "GET-METHOD-NAME"
        USING W-METHOD-NAME RETURNING W-LENGTH
    DISPLAY " METHOD-NAME: " W-METHOD-NAME
*> メッセージの参照
    INVOKE L-EXCEPTION-OBJ "GET-MESSAGE"
        USING W-METHOD-NAME RETURNING W-LENGTH
    DISPLAY " MESSAGE: " W-METHOD-NAME
END METHOD EXCEPTION-DISPLAY.

```

```

*>
*> 帳票開設イベント処理メソッド(オーバーライド)
IDENTIFICATION DIVISION.
METHOD-ID. OPEN-REPORT OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-CONTROL-OBJ OBJECT REFERENCE FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CONTROL-OBJ.
    MOVE "FORM1" TO FORM-NAME OF L-CONTROL-OBJ
END METHOD OPEN-REPORT.
*>
*> レコード読み込みイベント処理メソッド(オーバーライド)
IDENTIFICATION DIVISION.
METHOD-ID. READ-RECORD OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01 W-EXCPOBJ OBJECT REFERENCE FJPRT-EXCEPTION.
LINKAGE SECTION.
    01 L-CONTROL-OBJ OBJECT REFERENCE FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CONTROL-OBJ.
DECLARATIVES. *>.....[3]
EXCEPTION-HANDLING SECTION. *>.....[4]
    USE AFTER EXCEPTION FJPRT-EXCEPTION. *>.....[5]
    DISPLAY "EXCEPTION-ERROR OCCURED IN READ-RECORD"
*> 例外エラーの情報を表示するメソッドを呼び出す。
    SET W-EXCPOBJ TO EXCEPTION-OBJECT AS FJPRT-EXCEPTION
    INVOKE SELF "EXCEPTION-DISPLAY"
        USING W-EXCPOBJ. *>.....[6]
*> 例外エラーが発生したら、無条件で印刷をキャンセルする。
    MOVE B"1" TO REQUEST-CANCEL OF L-CONTROL-OBJ *>.....[7]
.
END DECLARATIVES. *>.....[8]
    IF RECORD-COUNT OF L-CONTROL-OBJ = 2 THEN
        MOVE B"1" TO RECORD-EOF OF SELF
    EXIT METHOD
END-IF
*> レコード内容の参照
    INVOKE L-CONTROL-OBJ "GET-RECORD-CONTENT"
        USING FORM1
*> 各項目に値を設定
    MOVE "TEST1" TO FIELD1 OF FORM1
    MOVE "TEST2" TO FIELD2 OF FORM1 (1)
    MOVE "TEST3" TO FIELD2 OF FORM1 (2)
*> レコード内容の設定
    INVOKE L-CONTROL-OBJ "SET-RECORD-CONTENT"
        USING FORM1
*> 文字色の設定
    INVOKE L-CONTROL-OBJ "SET-TEXT-COLOR"
        USING W-FIELD1-NAME FJPRT-SC-COLOR-BLUE
    MOVE 1 TO W-FIELD2-INDEX
    INVOKE L-CONTROL-OBJ "SET-TEXT-COLOR"

```

```

        USING W-FIELD2-NAME FJPRT-SC-COLOR-RED
MOVE 2 TO W-FIELD2-INDEX
INVOKE L-CONTROL-OBJ "SET-TEXT-COLOR"
        USING W-FIELD2-NAME FJPRT-SC-COLOR-GREEN
EXIT METHOD.
END METHOD READ-RECORD.
END OBJECT.
END CLASS USER-HANDLER-CLASS.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

[1]

利用するクラス名として REPOSITORY段落に FJPRT-EXCEPTION を記述します。

[2]

環境部構成節特殊名段落の SYMBOLIC CONSTANT句に、FJPRTSC.cb1を取り込むように COPY文を記述します。詳細については“[COPY文による記号定数の取込み](#)”を参照してください。

[3]

例外処理を行うメソッドの手続き部の先頭に DECLARATIVES を記述します。

[4]

DECLARATIVES に続き、節名を記述します。

[5]

節名に続き、USE文を記述します。

[6]

例として、例外オブジェクトを使ってエラーメッセージの表示を行うメソッドを呼び出しています。

[7]

例として、印刷のキャンセルを行っています。例外オブジェクトを使ってメソッド名の参照をしています。

[8]

手続きの始めの直前に、END DECLARATIVES を記述します。

## 6.2 印刷エラー

[印刷制御オブジェクト](#)の[帳票印刷メソッド \(PRINT-CONTROL\)](#)の実行中に、印刷のための環境が誤っていた場合などに[エラーメッセージイベント](#)が発生します (ERROR-MESSAGEメソッドが呼び出されます)。

これは“印刷エラー”と呼ばれ、主にプリンタの異常や帳票定義体の不備などが原因で発生します。

印刷エラーを検出してそれに対応した処理を行うためには、以下のクラスのエラーメッセージメソッド (ERROR-MESSAGE) を上書き (オーバーライド) する必要があります。

[帳票印刷制御クラス \(FJPRT-CONTROL\)](#)

[帳票印刷イベント処理クラス \(FJPRT-EVENT-HANDLER\)](#)



注意

上書き (オーバーライド) するエラーメッセージメソッド (ERROR-MESSAGE) の中では、[印刷ジョブエラーコードプロパティ \(PRINT-JOB-ERROR-CODE\)](#) の値を表示させてオペレータに異常を知らせるなどの処置を行ってください。

エラーの種類によっては印刷の継続が不可能なものもあります。詳細については“[印刷ジョブエラーコード](#)”の一覧表を参照してください。



---

## 第7章 クラスの構成と説明

---

ここでは、帳票印刷クラスライブラリが提供するクラスの構成とそれについての説明をします。  
下図は帳票印刷クラスライブラリが提供するクラスの階層図です。



図：帳票印刷クラスライブラリのクラス階層

帳票印刷クラスが提供するクラスについては、以下を参照してください。

[FJPRT-CONTROLクラス\(帳票印刷制御クラス\)](#)

[FJPRT-EVENT-HANDLERクラス\(帳票印刷イベント処理クラス\)](#)

[FJPRT-EXCEPTIONクラス\(帳票印刷例外クラス\)](#)

## 7.1 FJPRT-CONTROLクラス(帳票印刷制御クラス)

ここでは帳票印刷制御クラス(FJPRT-CONTROL)について説明します。

[説明](#)

[解説](#)

[メソッド/プロパティ](#)

### 7.1.1 説明

帳票印刷制御クラス(FJPRT-CONTROL)は以下のプロパティ、メソッドを持っているクラスです。

印刷に影響するさまざまな属性(プロパティ)群

印刷時に発生するさまざまなイベントで処理する内容を記述するための中身が空のメソッド群 (この空のメソッドのことをまとめて“イベント処理メソッド”と呼びます)

印刷に関するメソッドなど

帳票印刷制御クラス(FJPRT-CONTROL)のメソッド/プロパティについては、以下を参照してください。

“[帳票印刷制御クラスのメソッド/プロパティ](#)”

### 7.1.2 解説

ここでは帳票印刷制御クラス(FJPRT-CONTROL)を使ったプログラミングに関する事項について解説します。

[オブジェクトの作成](#)

[オブジェクトの削除](#)

[帳票の印刷](#)

[イベント処理メソッドの上書き](#)

[イベント処理オブジェクトの登録](#)

[当クラスの継承](#)

[当クラスの親クラス](#)

#### 7.1.2.1 オブジェクトの作成

帳票印刷制御クラス(FJPRT-CONTROL)またはこれを継承した子クラスのオブジェクトは以下の方法で作成します。なお、この操作で作成したオブジェクトのことを“印刷制御オブジェクト”と呼びます。

帳票印刷制御クラス(FJPRT-CONTROL):

INVOKE FJPRT-CONTROL "NEW"

RETURNING [CONTROL-OBJECT](#)

または、

帳票印刷制御クラス(FJPRT-CONTROL)を継承した子クラス

(例: *USER-CONTROL-CLASS*):

INVOKE *USER-CONTROL-CLASS* "NEW"

RETURNING [CONTROL-OBJECT](#)

【補足】

***CONTROL-OBJECT*** :

作成された印刷制御オブジェクトが返ります。

帳票印刷制御クラス(FJPRT-CONTROL)またはそれを継承したクラス(例:  
*USER-CONTROL-CLASS*)のオブジェクト参照一意名を記述します。このデータ項目は  
 次の属性で宣言してください。またデータ項目の名前は任意に付けてください。

- [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL  
 または、
- [USAGE IS] OBJECT REFERENCE *USER-CONTROL-CLASS*

### 7.1.2.2 オブジェクトの削除

[印刷制御オブジェクト](#)が不要になったとき、次の方法で削除することによって、システムリソースを開放します。

```
SET CONTROL-OBJECT-1 TO NULL
```



COBOLランタイムシステムはオブジェクトの削除の直前に、暗にオブジェクトが持つ  
 \_FINALIZEメソッドを呼び出します。なお、印刷制御オブジェクトが持つ \_FINALIZEメソ  
 ッドをプログラムから明に呼び出さないでください。

### 7.1.2.3 帳票の印刷

帳票の印刷を行うには、[帳票印刷メソッド\(PRINT-CONTROL\)](#)を呼び出します。

```
INVOKE CONTROL-OBJECT-1 "PRINT-CONTROL"  

  USING PRT-MODE [RETURNING BOOL-VAL]
```

このメソッドを呼び出すと、帳票の印刷が開始されます。そしてこのメソッド呼出しから復帰す  
 るまでに帳票の印刷が終了しています。帳票の印刷中にはイベントが発生し、実際に印刷する文  
 字を設定する処理などを利用者がイベント処理として記述する必要があります。詳細については  
 “[イベント](#)”を参照してください。



帳票の印刷を行うためには“[イベント処理メソッドの上書き](#)”(オーバーライド)か“[イベン  
 ト処理オブジェクトの登録](#)”のいずれか、または両方をあらかじめ行っておく必要があり  
 ます。

### 7.1.2.4 イベント処理メソッドの上書き

帳票印刷制御クラス(FJPRT-CONTROL)が持つ[イベント処理メソッド](#)を上書き(オーバーライド)した  
 子クラスを記述することができます。

この方法は、主に[イベント処理オブジェクト](#)で処理しないイベントに対する処理を行うために用  
 います。これによってイベント処理オブジェクトと印刷制御オブジェクトで処理すべきイベント  
 を切り分けることが可能となります。また、イベント処理オブジェクトを登録せずに当クラスを  
 継承したクラスのオブジェクトだけで印刷処理を行う場合にも利用できます。

上書きして記述したイベント処理メソッドは、帳票の印刷時にイベントが発生するときにそのイ  
 ベントに対応するメソッドが呼び出されます。詳細については“[呼び出すイベント処理メソッド  
 の選択](#)”を参照してください。

当クラスが持つイベント処理メソッドの中身は空で、上書きしないかぎり何も処理を行いません。

### 7.1.2.5 イベント処理オブジェクトの登録

帳票印刷イベント処理クラス (FJPRT-EVENT-HANDLER) の子クラスのオブジェクトが持つ [イベント処理メソッド](#) を、帳票の印刷時にイベントが発生した際に呼び出すことができます。

これは、イベント処理オブジェクトを印刷制御オブジェクトの [イベント処理オブジェクトプロパティ \(EVENT-HANDLER\)](#) に登録することにより可能となります。

以下の方法で、帳票印刷イベント処理クラス (FJPRT-EVENT-HANDLER) を継承した子クラス (例: *USER-HANDLER-CLASS*) のイベント処理オブジェクトを登録します。

```
INVOKE USER-HANDLER-CLASS "NEW" RETURNING HANDLER-OBJECT
SET EVENT-HANDLER OF CONTROL-OBJECT-1 TO HANDLER-OBJECT-1
```

【補足】

ここでの *HANDLER-OBJECT* と *HANDLER-OBJECT-1* は同一のデータ項目を記述してください。

### 7.1.2.6 当クラスの継承

帳票印刷制御クラス (FJPRT-CONTROL) を継承した子クラスを作ることが可能です。この子クラスの作成で行えることは以下の 3点です。これら以外を行った子クラスのオブジェクトの動作は保証しませんので注意してください。

新たなファクトリ/オブジェクトデータ、ファクトリ/オブジェクトメソッド、プロパティの追加

イベント処理メソッドの上書き (オーバーライド)

メソッド: NEW, INIT, \_FINALIZE の上書き (オーバーライド)



注意

メソッド: NEW, INIT, \_FINALIZE を上書きするときは、そのメソッドの先頭で、親クラスが持つ同メソッドを呼び出す記述を必ず入れてください。

### 7.1.2.7 当クラスの親クラス

帳票印刷制御クラス (FJPRT-CONTROL) は FJBASE クラスを継承しています。FJBASE クラスについては “COBOL 文法書” および “NetCOBOL 使用手引書” を参照してください。

## 7.1.3 メソッド/プロパティ

帳票印刷制御クラス (FJPRT-CONTROL) には、以下のメソッド/プロパティがあります。

オブジェクトメソッド

印刷に関するメソッド	名前
<a href="#">PRINT-CONTROL</a>	帳票印刷メソッド
<a href="#">PAGE-FEED</a>	改ページメソッド
<a href="#">NEXT-DE-PRINT-LINE</a>	次明細印刷行参照メソッド
<a href="#">NEXT-PRINT-LINE</a>	次印刷行参照メソッド
<a href="#">SKIP-PARTITION</a>	パーティション抑止メソッド
<a href="#">EXPORT-PRINTER-INFORMATION</a>	プリンタ情報生成メソッド
<a href="#">PRINTER-DEVICE-NUMBER</a>	プリンタデバイス数参照メソッド

<u>印刷に関するメソッド</u>	名前
<a href="#"><u>PRINTER-DEVICE-NAMES</u></a>	プリンタデバイス名参照メソッド

<u>イベント処理メソッド</u>	名前
<a href="#"><u>OPEN-REPORT</u></a>	帳票開設イベント処理メソッド
<a href="#"><u>OPEN-FORM</u></a>	帳票定義体開設イベント処理メソッド
<a href="#"><u>READ-RECORD</u></a>	レコード読み込みイベント処理メソッド
<a href="#"><u>CALCULATE-DE</u></a>	明細計算イベント処理メソッド
<a href="#"><u>REPORT-FOOT</u></a>	報告書脚書きイベント処理メソッド
<a href="#"><u>REPORT-HEAD</u></a>	報告書頭書きイベント処理メソッド
<a href="#"><u>PRINT-DE</u></a>	明細出力イベント処理メソッド
<a href="#"><u>CB-FOOT</u></a>	制御脚書きイベント処理メソッド
<a href="#"><u>CB-HEAD</u></a>	制御頭書きイベント処理メソッド
<a href="#"><u>PAGE-FOOT</u></a>	ページ脚書きイベント処理メソッド
<a href="#"><u>PAGE-HEAD</u></a>	ページ頭書きイベント処理メソッド
<a href="#"><u>ERROR-MESSAGE</u></a>	エラーメッセージイベント処理メソッド
<a href="#"><u>CANCEL-PRINT</u></a>	印刷キャンセルイベント処理メソッド
<a href="#"><u>CLOSE-FORM</u></a>	帳票定義体閉鎖イベント処理メソッド
<a href="#"><u>CLOSE-REPORT</u></a>	帳票閉鎖イベント処理メソッド

<u>帳票項目のプロパティに関するメソッド</u>	名前
<a href="#"><u>SET-RECORD-CONTENT</u></a>	レコード内容の設定メソッド
<a href="#"><u>GET-RECORD-CONTENT</u></a>	レコード内容の参照メソッド
<a href="#"><u>GET-PRE-RECORD-CONTENT</u></a>	前レコード内容の参照メソッド
<a href="#"><u>SET-TEXT-COLOR</u></a>	文字色プロパティ設定メソッド
<a href="#"><u>GET-TEXT-COLOR</u></a>	文字色プロパティ参照メソッド
<a href="#"><u>SET-DISABLE-FIELD</u></a>	項目抑止プロパティ設定メソッド
<a href="#"><u>GET-DISABLE-FIELD</u></a>	項目抑止プロパティ参照メソッド
<a href="#"><u>SET-EXTEND-CHARACTER</u></a>	文字引伸しプロパティ設定メソッド
<a href="#"><u>GET-EXTEND-CHARACTER</u></a>	文字引伸しプロパティ参照メソッド
<a href="#"><u>SET-FONT-STRIKE-OUT</u></a>	抹消線プロパティ設定メソッド
<a href="#"><u>GET-FONT-STRIKE-OUT</u></a>	抹消線プロパティ参照メソッド
<a href="#"><u>SET-FONT-ITALIC</u></a>	イタリックプロパティ設定メソッド
<a href="#"><u>GET-FONT-ITALIC</u></a>	イタリックプロパティ参照メソッド
<a href="#"><u>SET-FONT-BOLD</u></a>	ボールドプロパティ設定メソッド
<a href="#"><u>GET-FONT-BOLD</u></a>	ボールドプロパティ参照メソッド
<a href="#"><u>SET-FONT-UNDERLINE</u></a>	下線プロパティ設定メソッド
<a href="#"><u>GET-FONT-UNDERLINE</u></a>	下線プロパティ参照メソッド
<a href="#"><u>SET-FIELD-UNDERLINE</u></a>	項目下線プロパティ設定メソッド
<a href="#"><u>GET-FIELD-UNDERLINE</u></a>	項目下線プロパティ参照メソッド

<u>電子帳票制御のプロパティに関するメソッド</u>	名前
<a href="#"><u>SET-E-REPORT-TYPE</u></a>	電子帳票種別プロパティ設定メソッド
<a href="#"><u>GET-E-REPORT-TYPE</u></a>	電子帳票種別プロパティ参照メソッド
<a href="#"><u>SET-E-REPORT-CONTROL-FILENAME</u></a>	電子帳票制御ファイル名プロパティ設定メソッド

<u>電子帳票制御のプロパティに関するメソッド</u>	名前
<u>GET-E-REPORT-CONTROL-FILENAME</u>	電子帳票制御ファイル名プロパティ参照メソッド
<u>SET-E-REPORT-DESTINATION</u>	電子帳票宛先名プロパティ設定メソッド
<u>GET-E-REPORT-DESTINATION</u>	電子帳票宛先名プロパティ参照メソッド
<u>SET-E-REPORT-FILENAME</u>	電子帳票ファイル名プロパティ設定メソッド
<u>GET-E-REPORT-FILENAME</u>	電子帳票ファイル名プロパティ参照メソッド

<u>印刷制御のプロパティに関するメソッド</u>	名前
<u>SET-PRINT-PREVIEW</u>	印刷プレビュープロパティ設定メソッド
<u>GET-PRINT-PREVIEW</u>	印刷プレビュープロパティ参照メソッド

## オブジェクトプロパティ

<u>帳票定義体に関するプロパティ</u>	名前
<u>FORM-NAME</u>	帳票定義体名プロパティ
<u>DE-NAME</u>	明細名プロパティ
<u>CONTROL-PARTITION-NAME</u>	制御パーティション名プロパティ
<u>OVERLAY-NAME</u>	オーバーレイ名プロパティ
<u>FORM-DIRECTORY</u>	帳票定義体格納ディレクトリプロパティ
<u>OVERLAY-DIRECTORY</u>	オーバーレイ格納ディレクトリプロパティ
<u>MEDIA-DIRECTORY</u>	メディア格納ディレクトリプロパティ
<u>FORM-SUFFIX</u>	帳票定義体拡張子プロパティ
<u>OVERLAY-SUFFIX</u>	オーバーレイ拡張子プロパティ

<u>印刷エラーに関するプロパティ</u>	名前
<u>PRINT-JOB-ERROR-CODE</u>	印刷ジョブエラーコードプロパティ

<u>印刷行/ページ/タックに関するプロパティ</u>	名前
<u>PRINTING-PAGE-NUMBER</u>	印刷中ページ番号プロパティ
<u>PRINTING-LINE-NUMBER</u>	印刷中行番号プロパティ
<u>INIT-PAGE-NUMBER</u>	最初のページ番号プロパティ
<u>PAGE-RANGE</u>	ページ範囲プロパティ
<u>START-PAGE-NUMBER</u>	開始ページ番号プロパティ
<u>END-PAGE-NUMBER</u>	終了ページ番号プロパティ
<u>START-TACK-NUMBER</u>	タック開始番号プロパティ

<u>入力データに関するプロパティ</u>	名前
<u>AUTOMATIC-RECORD-CHANGE</u>	レコード自動切替えプロパティ
<u>RECORD-EOF</u>	レコードEOFプロパティ
<u>RECORD-COUNT</u>	レコード回数プロパティ

<u>印刷制御に関するプロパティ</u>	名前
<u>PRINT-TYPE</u>	印刷種別プロパティ
<u>PRINT-COUNT</u>	印刷回数プロパティ
<u>REQUEST-CANCEL</u>	キャンセル要求プロパティ

<u>印刷制御に関するプロパティ</u>	名前
<u>PRINTER-NAME</u>	プリンタ名プロパティ
<u>PRINT-JOB-NAME</u>	印刷ジョブ名プロパティ
<u>PRINT-JOB-CONDITION</u>	印刷ジョブ状態プロパティ
<u>PAPER-TYPE</u>	用紙種別プロパティ
<u>PRINT-COPIES</u>	印刷部数プロパティ
<u>PAPER-SIZE</u>	用紙サイズプロパティ
<u>PAPER-SIZE-HEIGHT</u>	用紙縦サイズプロパティ
<u>PAPER-SIZE-WIDTH</u>	用紙横サイズプロパティ
<u>PRINT-POSITION</u>	印刷原点プロパティ
<u>PRINT-POSITION-TOP</u>	印刷原点縦位置プロパティ
<u>PRINT-POSITION-LEFT</u>	印刷原点横位置プロパティ
<u>REDUCTION-TYPE</u>	縮小種別プロパティ
<u>COMPACT-PRINT-TYPE</u>	コンパクト印刷種別プロパティ
<u>OVERLAY-NAME-TO-COMPACT-PRINT</u>	コンパクト印刷用オーバーレイ名プロパティ
<u>PRINTING-DIALOG</u>	印刷中ダイアログプロパティ
<u>PRINTING-DIALOG-CANCEL-BUTTON</u>	印刷中ダイアログのキャンセルボタンプロパティ
<u>PRINTING-DIALOG-MINIMIZE-BOX</u>	印刷中ダイアログ最小化ボックスプロパティ
<u>PRINTER-INFORMATION-FILENAME</u>	プリンタ情報ファイル名プロパティ

<u>イベント処理に関するプロパティ</u>	名前
<u>EVENT-HANDLER</u>	イベント処理オブジェクトプロパティ

## 7.2 FJPRT-EVENT-HANDLERクラス(帳票印刷イベント処理クラス)

ここでは帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)について説明します。

[説明](#)

[解説](#)

[メソッド](#)

### 7.2.1 説明

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)は、帳票印刷制御クラス(FJPRT-CONTROL)またはそれを継承したクラスのオブジェクト([印刷制御オブジェクト](#))を利用して帳票の印刷を行うときに発生するイベントを処理する“[イベント処理メソッド](#)”の実装を行うために利用します。当クラスは以下のメソッドを持っています。

印刷時に発生するさまざまなイベントで処理する内容を記述するためのメソッド群（このメソッドのことをまとめて“イベント処理メソッド”と呼びます）

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)のメソッドについては、以下を参照してください。

“[帳票印刷イベント処理クラスのメソッド](#)”

### 7.2.2 解説

ここでは帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)を使ったプログラミングに関する事項について解説します。

[当クラスの継承](#)

[オブジェクトの作成](#)

[オブジェクトの削除](#)

[イベント処理メソッドの上書き](#)

[イベント処理オブジェクトの登録](#)

[当クラスの親クラス](#)

#### 7.2.2.1 当クラスの継承

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)は、そのクラスを継承して、処理する必要があるイベントに対応するイベント処理メソッドを上書き(オーバーライド)した子クラスを記述するために利用します。

当クラスはイベント処理メソッドを上書きする以外に、以下のことを許しています。

新たなファクトリ/オブジェクトデータ、ファクトリ/オブジェクトメソッド、プロパティの追加

メソッド: NEW, INIT, \_FINALIZE の上書き(オーバーライド)



**注意**

メソッド: NEW, INIT, \_FINALIZE を上書きするときは、そのメソッドの先頭で、親クラスにある同メソッドを呼び出す記述を必ず入れてください。



### 7.2.2.2 オブジェクトの作成

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)は継承して子クラスを作ってから、その子クラスを利用します。帳票印刷イベント処理クラスのオブジェクトを直接作成する利用方法はありません。

当クラスを継承した子クラス(例: *USER-HANDLER-CLASS*)のオブジェクトは以下の方法で作成します。なお、この操作で作成したオブジェクトのことを“イベント処理オブジェクト”と呼びます。

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)を継承した子クラス

(例: *USER-HANDLER-CLASS*):

```
INVOKE  USER-HANDLER-CLASS  "NEW"
RETURNING HANDLER-OBJECT
```

【補足】

***HANDLER-OBJECT* :**

作成されたイベント処理オブジェクトが返ります。

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)を継承したクラス(例: *USER-HANDLER-CLASS*)のオブジェクト参照一意名を記述します。このデータ項目は次の属性で宣言してください。なお、データ項目には任意に名前を付けてください。

– [USAGE IS] OBJECT REFERENCE *USER-CONTROL-CLASS*

### 7.2.2.3 オブジェクトの削除

[イベント処理オブジェクト](#)が不要になったとき、次の方法で削除することによって、システムリソースを開放します。

```
SET HANDLER-OBJECT-1 TO NULL
```



注意

COBOLランタイムシステムはオブジェクトの削除時に暗にオブジェクトが持つ `_FINALIZE` メソッドを呼び出します。

### 7.2.2.4 イベント処理メソッドの上書き

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)は直接の利用(当クラスのオブジェクトを作成して利用すること)ができません。必ず、[イベント処理メソッド](#)を上書き(オーバーライド)した子クラスを記述してください。

当クラスのイベント処理メソッドを上書きした子クラスのオブジェクト(イベント処理オブジェクト)を作成し、印刷制御オブジェクトの[イベント処理オブジェクトプロパティ\(EVENT-HANDLER\)](#)に登録すると、帳票の印刷時にイベントが発生するときにそのイベントに対応するイベント処理メソッドが呼び出されるようになります。詳細については“[イベント処理オブジェクトの登録](#)”を参照してください。

当クラスが持つイベント処理メソッドは、上書きしない場合、登録元である帳票印刷制御オブジェクトのイベント処理メソッドを呼び出すようになっています。

### 7.2.2.5 イベント処理オブジェクトの登録

イベント処理オブジェクトの登録については、帳票印刷制御クラス(FJPRT-CONTROL)の解説“[イ](#)

[イベント処理オブジェクトの登録](#)”を参照してください。

### 7.2.2.6 当クラスの親クラス

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)は FJBASEクラスを継承しています。FJBASEクラスについては“COBOL文法書”および“NetCOBOL使用手引書”を参照してください。

## 7.2.3 メソッド

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)には、以下のメソッドがあります。

オブジェクトメソッド

<a href="#">イベント処理メソッド</a>	名前
<a href="#">OPEN-REPORT</a>	帳票開設イベント処理メソッド
<a href="#">OPEN-FORM</a>	帳票定義体開設イベント処理メソッド
<a href="#">READ-RECORD</a>	レコード読み込みイベント処理メソッド
<a href="#">CALCULATE-DE</a>	明細計算イベント処理メソッド
<a href="#">REPORT-FOOT</a>	報告書脚書きイベント処理メソッド
<a href="#">REPORT-HEAD</a>	報告書頭書きイベント処理メソッド
<a href="#">PRINT-DE</a>	明細出力イベント処理メソッド
<a href="#">CB-FOOT</a>	制御脚書きイベント処理メソッド
<a href="#">CB-HEAD</a>	制御頭書きイベント処理メソッド
<a href="#">PAGE-FOOT</a>	ページ脚書きイベント処理メソッド
<a href="#">PAGE-HEAD</a>	ページ頭書きイベント処理メソッド
<a href="#">ERROR-MESSAGE</a>	エラーメッセージイベント処理メソッド
<a href="#">CANCEL-PRINT</a>	印刷キャンセルイベント処理メソッド
<a href="#">CLOSE-FORM</a>	帳票定義体閉鎖イベント処理メソッド
<a href="#">CLOSE-REPORT</a>	帳票閉鎖イベント処理メソッド

## 7.3 FJPRT-EXCEPTIONクラス(帳票印刷例外クラス)

ここでは帳票印刷例外クラス(FJPRT-EXCEPTION)について説明します。

[説明](#)

[解説](#)

[メソッド](#)

### 7.3.1 説明

帳票印刷例外クラス(FJPRT-EXCEPTION)は、[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)またはその子クラスが持つメソッドやプロパティへのアクセスが失敗した([例外エラー](#)が発生した)場合のエラー処理を行うために利用します。

当クラスのオブジェクトは例外エラーの発生時に帳票印刷クラスライブラリによって作成されます。当クラスのオブジェクトを利用者が NEWメソッドで作成する利用方法はありません。

当クラスのメソッドについては、以下を参照してください。

“ [帳票印刷例外クラスのメソッド](#) ”

### 7.3.2 解説

ここでは帳票印刷例外クラス(FJPRT-EXCEPTION)を使ったプログラミングに関する事項について解説します。

[当クラスの親クラス](#)

#### 7.3.2.1 当クラスの親クラス

帳票印刷例外クラス(FJPRT-EXCEPTION)は FJBASEクラスを継承しています。FJBASEクラスについては“ COBOL文法書 ” および“ NetCOBOL使用手引書 ” を参照してください。

### 7.3.3 メソッド

帳票印刷例外クラス(FJPRT-EXCEPTION)には、以下のメソッドがあります。

オブジェクトメソッド

<a href="#">例外エラー情報参照メソッド</a>	名前
<a href="#">GET-CODE</a>	コードの参照メソッド
<a href="#">GET-MESSAGE</a>	メッセージの参照メソッド
<a href="#">GET-METHOD-NAME</a>	メソッド名の参照メソッド
<a href="#">GET-CLASS-NAME</a>	クラス名の参照メソッド



---

## 第8章 帳票印刷制御クラスのメソッド/プロパティ

---

ここでは、[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)が提供するメソッドとプロパティについての説明をします。

メソッドは以下に示すとおり、5つのカテゴリに分かれています。

[印刷に関するメソッド](#)

[イベント処理メソッド](#)

[帳票項目のプロパティに関するメソッド](#)

[電子帳票制御のプロパティに関するメソッド](#)

[印刷制御のプロパティに関するメソッド](#)

プロパティは以下に示すとおり、6つのカテゴリに分かれています。

[帳票定義体に関するプロパティ](#)

[印刷エラーに関するプロパティ](#)

[印刷行/ページ/タックに関するプロパティ](#)

[入力データに関するプロパティ](#)

[印刷制御に関するプロパティ](#)

[イベント処理に関するプロパティ](#)

プロパティは、帳票印刷制御クラス(FJPRT-CONTROL)の“オブジェクトプロパティ設定メソッド”または“オブジェクトプロパティ取出しメソッド”として実装されています。

オブジェクトプロパティについては“COBOL文法書”の“オブジェクトプロパティ”を参照してください。

---

## 8.1 印刷に関するメソッド

ここでは、[帳票印刷制御\(FJPRT-CONTROL\)](#)が提供する以下のカテゴリのメソッドについての説明をします。

印刷に関するメソッド

[PRINT-CONTROLメソッド\(帳票印刷メソッド\)](#)  
[PAGE-FEEDメソッド\(改ページメソッド\)](#)  
[NEXT-DE-PRINT-LINEメソッド\(次明細印刷行参照メソッド\)](#)  
[NEXT-PRINT-LINEメソッド\(次印刷行参照メソッド\)](#)  
[SKIP-PARTITIONメソッド\(パーティション抑止メソッド\)](#)  
[EXPORT-PRINTER-INFORMATIONメソッド\(プリンタ情報生成メソッド\)](#)  
[PRINTER-DEVICE-NUMBERメソッド\(プリンタデバイス数参照メソッド\)](#)  
[PRINTER-DEVICE-NAMESメソッド\(プリンタデバイス名参照メソッド\)](#)



注意

以下に、本文の引数/復帰値の項目を読む上での注意事項を挙げます。

引数/復帰値の項目に書かれている“属性”は、メソッドの定義上のインタフェースを表すものです。

PIC X ANY LENGTH の属性を持つ引数には、任意の長さを持つ英数字のデータ項目を記述してください。なお、この引数には値が返却されることがあります。これについては各メソッドの仕様を確認してください。

### 8.1.1 PRINT-CONTROLメソッド(帳票印刷メソッド)

#### 説明

[帳票の印刷](#)を開始します。

#### 記述形式

```

INVOKE  CONTROL-OBJECT-1  "PRINT-CONTROL"
        USING      PRT-MODE
        [RETURNING  BOOL-VAL]

```

【補足】

**CONTROL-OBJECT-1** :

操作の対象となる、[印刷制御オブジェクト](#)を持つ、オブジェクト参照一意名を記述します。[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)または、それを継承したクラス(例：[USER-CONTROL-CLASS](#))のオブジェクト参照一意名は、それぞれ次の属性で宣言してください。なお、オブジェクト参照一意名の名前は、任意に名付けてください。

[USAGE IS] OBJECT REFERENCE FJPRT-CONTROL

または、

[USAGE IS] OBJECT REFERENCE USER-CONTROL-CLASS

#### 引数

**PRT-MODE** 【引数の属性: PIC S9(4) [USAGE IS] COMP-5】:

印刷モードを渡します。引数の属性と同じ属性を持つデータ項目または定数を記述してください。詳細は“[イベントの発生に影響する事項](#)”を参照してください。

印刷モード	意味
-------	----

印刷モード	意味
0	<a href="#">レコードEOFプロパティ (RECORD-EOF)</a> に真 (B"1") が設定されるまで <a href="#">レコード読み込みイベント</a> が繰り返し発生して印刷処理が継続します。一般的にはこちらを使用します。
1	1件だけ <a href="#">レコード読み込みイベント</a> が発生して印刷がなされます。印刷処理が繰り返されることはありません。

## 復帰値

**BOOL-VAL** 【復帰値の属性: PIC 1(1) [USAGE IS] DISPLAY】:

以下の値を返します。復帰値の属性と同じ属性を持つデータ項目を記述してください。

値	意味
B"1" (真)	メソッドの正常終了
B"0" (偽)	メソッドの異常終了

## 例外エラー

例外エラーの発生はありません。

## 使用例

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
PROGRAM-ID. MAIN.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
CLASS FJPRT-CONTROL
CLASS USER-HANDLER-CLASS.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 W-CTRLOBJ OBJECT REFERENCE FJPRT-CONTROL.
01 W-HDLROBJ OBJECT REFERENCE USER-HANDLER-CLASS.
01 W-RESULT PIC 1(1) DISPLAY.
PROCEDURE DIVISION.
INVOKE FJPRT-CONTROL "NEW" RETURNING W-CTRLOBJ *>....[1]
INVOKE USER-HANDLER-CLASS "NEW" RETURNING W-HDLROBJ *>....[2]
SET EVENT-HANDLER OF W-CTRLOBJ TO W-HDLROBJ *>.....[3]
INVOKE W-CTRLOBJ "PRINT-CONTROL"
USING 0 RETURNING W-RESULT *>.....[4]
SET W-CTRLOBJ TO NULL *>.....[5]
SET W-HDLROBJ TO NULL *>.....[6]
END PROGRAM MAIN.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

[例の説明]

[1]

[FJPRT-CONTROL](#)クラスのオブジェクトを作成します。

[2]

[FJPRT-EVENT-HANDLER](#)クラスを継承して[イベント処理メソッドを上書き](#)して記述した [USER-HANDLER-CLASS](#)クラスのオブジェクトを作成します。

[3]

[2]で作成した[イベント処理オブジェクト](#)を[イベント処理オブジェクトとして登録](#)します。

- [4]  
帳票印刷メソッド (PRINT-CONTROL) を呼び出します。このメソッドの実行中に [イベント](#) が発生して、目的の帳票の印刷が行われます。
- [5]  
作成した [印刷制御オブジェクト](#) を削除します。
- [6]  
作成した [イベント処理オブジェクト](#) を削除します。

## 8.1.2 PAGE-FEEDメソッド(改ページメソッド)

### 説明

印刷対象となるパーティションを印刷した後で改ページを行います。  
改ページの後のイベントの発生については、“[改ページに関するイベント発生の流れ](#)” を参照してください。

### 記述形式

INVOKE [CONTROL-OBJECT-2](#) "PAGE-FEED"  
[RETURNING *BOOL-VAL*]

#### 【補足】

#### **CONTROL-OBJECT-2 :**

イベント処理メソッドで操作する対象となる、[印刷制御オブジェクト](#) を持つ、オブジェクト参照一意名を記述します。

具体的な記述は、当メソッドを呼び出す記述箇所によって、次のように異なります。

[帳票印刷制御クラス\(FJPRT-CONTROL\)](#) を継承した子クラスのイベント処理メソッド内では、以下を記述します。

SELF

[帳票印刷イベント処理クラス\(FJPRT-EVENT-HANDLER\)](#) を継承した子クラスのイベント処理メソッド内では、以下を記述します。

[イベント処理メソッド](#) の第一引数として渡ってくる、印刷制御オブジェクト引数のオブジェクト参照一意名

### 引数

なし。

### 復帰値

*BOOL-VAL* 【復帰値の属性: PIC 1(1) [USAGE IS] DISPLAY】:

以下の値を返します。復帰値の属性と同じ属性を持つデータ項目を記述してください。

値	意味
B"1" (真)	メソッドの正常終了
B"0" (偽)	メソッドの異常終了

### 例外エラー

例外エラーの発生はありません。



## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER-CLASS  INHERITS  FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS  FJPRT-EVENT-HANDLER
    CLASS  FJPRT-CONTROL
:
IDENTIFICATION DIVISION.
METHOD-ID.  PRINT-DE  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01 W-RESULT  PIC 1(1)  DISPLAY.
LINKAGE SECTION.
    01 L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.
    IF  FUNCTION REM (RECORD-COUNT OF L-CTRLOBJ, 10) = 0  THEN  *>...[1]
        INVOKE  L-CTRLOBJ  "PAGE-FEED"  RETURNING  W-RESULT  *>.....[2]
    END-IF
END METHOD  PRINT-DE.
:
END CLASS  USER-HANDLER-CLASS.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

[例の説明]

[1]

[レコード回数プロパティ \(RECORD-COUNT\)](#) の参照値を 10 で割った余りが 0 か否か判定します。

[2]

IF 文の条件が真なら、改ページメソッド (PAGE-FEED) を呼び出します。

[1] と [2] によって、明細またはボディ部の印刷を 10 回行った後に改ページをするように動作します。

### 8.1.3 NEXT-DE-PRINT-LINE メソッド (次明細印刷行参照メソッド)

#### 説明

次に印刷する明細 / ボディ部の先頭印刷行位置を返却します。

#### 記述形式

```

INVOKE  CONTROL-OBJECT-2  "NEXT-DE-PRINT-LINE"
RETURNING  SHORT-VAL

```

#### 引数

なし。

#### 復帰値

**SHORT-VAL** 【復帰値の属性: PIC S9(4) [USAGE IS] COMP-5】:

次に印刷する明細 / ボディ部の先頭印刷行位置を返します。復帰値の属性と同じ属性を持つデータ項目を記述してください。

先頭印刷行位置は、次に印刷する明細/ボディ部の位置を論理的に計算し、該当明細パーティション/ボディ部パーティションの先頭行位置を返します。コントロールブレイク処理中であれば、制御頭書きや制御脚書きパーティションの行数および印刷前・印刷後制御情報をブレイク階層分加味し、明細/ボディ部の位置を求めます。

なお、明細/ボディ部印刷位置が改ページによって次ページになる場合は、復帰値に 0 を返します。また、ページ最終行に達して次ページに印刷するデータがない場合は、-1 が返されます。

### 例外エラー

例外エラーの発生はありません。

### **注意事項**

当メソッドは、[制御頭書きイベント](#)、[制御脚書きイベント](#)、[明細出力イベント](#)のイベント処理中でだけ呼び出すことが有効です。

これらのイベントの処理では、各パーティションの印刷処理はまだ行われていません。そのため、通知される先頭印刷行位置は、各イベント処理直後に処理する明細/ボディ部（明細出力イベントで参照した場合、これに対応する明細/ボディ部）の出力行位置になります。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER-CLASS  INHERITS  FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
SPECIAL-NAMES.
    SYMBOLIC CONSTANT
    COPY  FJPRTSC.
.
    CLASS  FJPRT-EVENT-HANDLER
    CLASS  FJPRT-CONTROL
:
IDENTIFICATION DIVISION.
METHOD-ID.  PRINT-DE  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01  W-FIELD1-NAME  PIC  X(6).
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.
    MOVE  "FIELD1"  TO  W-FIELD1-NAME  *>.....[1]
    EVALUATE  L-CTRLOBJ::"NEXT-DE-PRINT-LINE"  *>.....[2]
    WHEN      20  THRU  39  *>.....[3]
        INVOKE  L-CTRLOBJ  "SET-TEXT-COLOR"
                USING  W-FIELD1-NAME  FJPRT-SC-COLOR-RED
    WHEN      OTHER  *>.....[4]
        INVOKE  L-CTRLOBJ  "SET-TEXT-COLOR"
                USING  W-FIELD1-NAME  FJPRT-SC-COLOR-BLACK
    END-EVALUATE
    EXIT METHOD.
END METHOD  PRINT-DE.
:
END CLASS  USER-HANDLER-CLASS.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

[1]

[文字色プロパティ設定メソッド\(SET-TEXT-COLOR\)](#)に渡す帳票項目名を作業場所節のデータ項目に設定します。

[2]

次に印刷される明細行の行位置(今回、印刷対象となる明細パーティションの行位置)を、次明細印刷行参照メソッド(NEXT-DE-PRINT-LINE)の行内呼出しで取り出し、判定条件とします。

[3]

次に印刷される明細行の行位置が 20から 39の範囲なら、明細パーティションにある "FIELD1" の帳票項目が印刷される色を[文字色プロパティ設定メソッド\(SET-TEXT-COLOR\)](#)で“赤”に変更します。

[4]

[3]の条件以外なら、明細パーティションにある "FIELD1" の帳票項目が印刷される色を[文字色プロパティ設定メソッド\(SET-TEXT-COLOR\)](#)で“黒”に変更します。

[1] ~ [4]によって、20行目から 39行目に印刷される明細パーティションを持つ "FIELD1" の帳票項目が印刷される色が“赤”となり、それ以外では“黒”で印刷されます。

### 8.1.4 NEXT-PRINT-LINEメソッド(次印刷行参照メソッド)

#### 説明

次に印刷するパーティションの先頭行位置を返却します。

#### 記述形式

INVOKE [CONTROL-OBJECT-2](#) "NEXT-PRINT-LINE"  
RETURNING *SHORT-VAL*

#### 引数

なし。

#### 復帰値

**SHORT-VAL** 【復帰値の属性: PIC S9(4) [USAGE IS] COMP-5】:

次に印刷するパーティションの先頭印刷行位置を返却します。復帰値の属性と同じ属性を持つデータ項目を記述してください。

先頭印刷行位置は、現在のプリンタヘッド位置、前回印刷したパーティションの印刷後制御情報および、次に印刷するパーティションの印刷前制御情報を加味した値です。

なお、明細/ボディ部印刷位置が改ページによって次ページになる場合は、復帰値に0を返します。また、ページ最終行に達して次ページに印刷するデータがない場合は、-1が返されます。

#### 例外エラー

例外エラーの発生はありません。

#### 注意事項

当メソッドは、[制御頭書きイベント](#)、[制御脚書きイベント](#)、[明細出力イベント](#)のイベント処理中でだけ呼び出すことが有効です。これら以外のタイミングで当メソッドを呼び出した場合の動作については保証しません。

また、これらのイベントの処理では、各パーティションの印刷処理はまだ行われていません。そのため、通知される先頭印刷行位置は、各イベント処理直後に処理するパーティション（発生した各イベントに対応するパーティション）の出力行位置になります。

### 8.1.5 SKIP-PARTITIONメソッド(パーティション抑止メソッド)

#### 説明

印刷対象のパーティションの印刷を抑止するためのメソッドです。

#### 記述形式

INVOKE [CONTROL-OBJECT-2](#) "SKIP-PARTITION"  
[RETURNING *BOOL-VAL*]

#### 引数

なし。

#### 復帰値

**BOOL-VAL** 【復帰値の属性: PIC 1(1) [USAGE IS] DISPLAY】:

以下の値を返却します。復帰値の属性と同じ属性を持つデータ項目を記述してください。

値	意味
B"1" (真)	メソッドの正常終了

値	意味
B"0" (偽)	メソッドの異常終了

### 例外エラー

例外エラーの発生はありません。

### 注意事項

集計表形式の帳票定義体でだけ使用できます。

抑止するパーティションに対応するイベントで、当メソッドを呼び出してください。(例: 報告書頭書きパーティション [報告書頭書きイベント](#)。明細パーティション [明細計算イベント](#)...)

### 使用例

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER-CLASS INHERITS  FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS  FJPRT-EVENT-HANDLER
    CLASS  FJPRT-CONTROL
:
IDENTIFICATION DIVISION.
METHOD-ID.  PAGE-FOOT OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01 W-RESULT  PIC 1(1)  DISPLAY.
LINKAGE SECTION.
    01 L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    INVOKE  L-CTRLOBJ  "SKIP-PARTITION"  RETURNING  W-RESULT  *>...[1]
END METHOD  PAGE-FOOT.
:
END CLASS  USER-HANDLER-CLASS.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

[例の説明]

[1]

ページ脚書きパーティションの印刷を常に抑止します (当メソッドがページ脚書きイベント処理メソッド(PAGE-FOOT)から呼び出されているため抑止されます)。

## 8.1.6 EXPORT-PRINTER- INFORMATIONメソッド(プリンタ情報生成メソッド)

### 説明

[印刷制御オブジェクト](#)が持つプロパティの値をプリンタ情報ファイルとして保存します。

このメソッドで生成したプリンタ情報ファイルを [プリンタ情報ファイル名プロパティ \(PRINTER- INFORMATION-FILENAME\)](#) で利用することで、印刷制御オブジェクトがもつ複数のプロパティの値を一括で変更することができます。

また、既存のプログラムで使用しているプロパティの値を当メソッドでプリンタ情報ファイルに

保存し、これをほかのアプリで流用することもできます。  
付加できるパスとして、フルパス、相対パスがあります。

## 記述形式

```
INVOKE CONTROL-OBJECT-3 "EXPORT-PRINTER- INFORMATION"
      USING      [BY REFERENCE] PRINTER-INF
      [RETURNING BOOL-VAL]
```

### 【補足】

#### **CONTROL-OBJECT-3 :**

操作の対象となる、[印刷制御オブジェクト](#)を持つ、オブジェクト参照一意名を記述します。  
具体的な記述は、当メソッドを呼び出す記述箇所によって、次のように異なります。

- a) イベント処理メソッド内から呼び出す場合:  
[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)を継承した子クラスのイベント処理メソッド内では、以下を記述します。
  - SELF
  - [帳票印刷イベント処理クラス\(FJPRT-EVENT-HANDLER\)](#)を継承した子クラスのイベント処理メソッド内では、以下を記述します。
    - [イベント処理メソッド](#)の第一引数として渡ってくる、印刷制御オブジェクト引数のオブジェクト参照一意名
- b) 業務処理から呼び出す場合:  
[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)または、それを継承したクラス(例:  
*USER-CONTROL-CLASS*)のオブジェクト参照一意名は、それぞれ次の属性で宣言してください。なお、オブジェクト参照一意名の名前は任意に名付けてください。
  - [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL
  - または、
  - [USAGE IS] OBJECT REFERENCE *USER-CONTROL-CLASS*

## 引数

#### **PRINTER-INF** 【引数の属性: PIC X ANY LENGTH】:

プリンタ情報ファイル名を渡します。引数には最大で259文字を格納できるだけの任意の長さの英数字項目を記述してください。

当引数は最大259文字までを認識しますが、実際に有効となるファイル名の長さはシステムに依存します。

## 復帰値

#### **BOOL-VAL** 【復帰値の属性: PIC 1(1) [USAGE IS] DISPLAY】:

以下の値を返却します。復帰値の属性と同じ属性を持つデータ項目を記述してください。

値	意味
B"1" (真)	メソッドの正常終了
B"0" (偽)	メソッドの異常終了

## 例外エラー

例外エラーの発生はありません。

## 注意事項

保存されるプリンタ情報ファイルは、動作システム依存となります。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
PROGRAM-ID.  MAIN.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS  FJPRT-CONTROL
DATA DIVISION.
WORKING-STORAGE SECTION.
    01 W-CTRLOBJ  OBJECT REFERENCE FJPRT-CONTROL.
    01 W-RESULT   PIC 1(1)  DISPLAY.
    01 W-FILENAME PIC X(20).
PROCEDURE DIVISION.
    INVOKE FJPRT-CONTROL "NEW" RETURNING W-CTRLOBJ *>....[1]
    MOVE "FORM-1" TO FORM-NAME OF W-CTRLOBJ *>.....[2]
    MOVE "pxd" TO FORM-SUFFIX OF W-CTRLOBJ
    MOVE 10 TO PRINT-COPIES OF W-CTRLOBJ
    :
    MOVE "OUTPUT.fil" TO W-FILENAME *>.....[3]
    INVOKE W-CTRLOBJ "EXPORT-PRINTER-INFORMATION"
        USING W-FILENAME *>.....[4]
    SET W-CTRLOBJ TO NULL *>.....[5]
    EXIT PROGRAM.
END PROGRAM  MAIN.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

## [例の説明]

- [1] [FJPRT-CONTROL](#)クラスのオブジェクトを作成します。
- [2] 各種プロパティに値を設定します。
- [3] プリンタ情報ファイルとして出力するファイルの名前をデータ項目に設定します。
- [4] プリンタ情報生成メソッド (EXPORT-PRINTER-INFORMATION) を呼び出し、プリンタ情報ファイルが作成されます。
- [5] 作成した[印刷制御オブジェクト](#)を削除します。

## 8.1.7 PRINTER-DEVICE-NUMBERメソッド(プリンタデバイス数参照メソッド)

## 説明

システムに登録されているプリンタの個数を参照する場合に使用します。  
[プリンタデバイス名参照メソッド \(PRINTER-DEVICE-NAMES\)](#) と組にして使われます。

## 記述形式

```

INVOKE CONTROL-OBJECT-3 "PRINTER-DEVICE-NUMBER"
RETURNING SHORT-VAL

```

## 引数

なし。

## 復帰値

**SHORT-VAL** 【復帰値の属性: PIC S9(4) [USAGE IS] COMP-5】:

システムに登録されているプリンタの個数を返します。復帰値の属性と同じ属性を持つデータ項目を記述してください。

## 例外エラー

例外エラーの発生はありません。

## 使用例

プリンタデバイス名参照メソッド(PRINTER-DEVICE-NAMES)の“[使用例](#)”を参照してください。

# 8.1.8 PRINTER-DEVICE-NAMESメソッド(プリンタデバイス名参照メソッド)

## 説明

システムに登録されているプリンタの名前を登録されている順番に参照します。

[プリンタデバイス数参照メソッド\(PRINTER-DEVICE-NUMBER\)](#)で返された個数分、当メソッドを呼び出してください。

## 記述形式

INVOKE [CONTROL-OBJECT-3](#) "PRINTER-DEVICE-NAMES"

USING [BY REFERENCE] *PRINTER-NAMES*

## 引数

**PRINTER-NAMES** 【引数の属性: PIC X ANY LENGTH】:

システムに登録されているプリンタの名前をデータ項目に返します。引数には最大で63文字を格納できるだけの任意の長さを持つ英数字項目を記述してください。

ひとつのプリンタ名の最大文字数は63文字です。ただし、実際の最大に有効となるプリンタ名の長さはシステムに依存します。

ローカル接続プリンタは、[スタート]メニュー [設定] [プリンタ]で表示されるプリンター一覧に表示されている名前が返されます。サーバ接続プリンタは共有名が返されず。

## 復帰値

なし。

## 例外エラー

例外エラーの発生はありません。



## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
PROGRAM-ID.  MAIN.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS  FJPRT-CONTROL
DATA DIVISION.
WORKING-STORAGE SECTION.
    01 W-CTROBJ  OBJECT REFERENCE FJPRT-CONTROL.
    01 W-RESULT  PIC 1(1)  DISPLAY.
    01 W-DEVICE-NUMBER  PIC S9(4)  COMP-5.
    01 W-DEVICE-NAME.
        02 W-DEVICE-NAMES  PIC X(30) OCCURS 20.
    01 W-I  PIC  S9(4)  COMP-5.
    01 W-NO  PIC  S9(4)  COMP-5.
PROCEDURE DIVISION.
    INVOKE FJPRT-CONTROL  "NEW"  RETURNING W-CTROBJ  *>.....[1]
    INVOKE W-CTROBJ  "PRINTER-DEVICE-NUMBER"
        RETURNING W-DEVICE-NUMBER  *>.....[2]
    PERFORM TEST BEFORE VARYING W-I
        FROM 1 BY 1 UNTIL W-I > W-DEVICE-NUMBER  *>.....[3]
        INVOKE W-CTROBJ  "PRINTER-DEVICE-NAMES"
            USING W-DEVICE-NAMES(W-I) RETURNING W-RESULT  *>...[4]
        DISPLAY "No." W-I  ": DEVICE-NAME: " W-DEVICE-NAMES(W-I)
    END-PERFORM
    DISPLAY "PLEASE ENTER No."
    ACCEPT W-NO  *>.....[5]
    MOVE W-DEVICE-NAMES(W-NO) TO PRINTER-NAME OF W-CTROBJ  *>...[6]
    :
END PROGRAM  MAIN.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

## [例の説明]

- [1]  
[FJPRT-CONTROL](#)クラスのオブジェクトを作成します。
- [2]  
プリンタデバイス数を参照し、データ項目に格納します。
- [3]  
[2]で参照できたプリンタデバイス数だけ繰り返し処理を行います。
- [4]  
繰り返し回数を持つデータ項目にプリンタデバイス名を格納します。
- [5]  
オペレータにプリンタを番号で選択させます。
- [6]  
オペレータが選択した番号のプリンタを出力先のプリンタとします。

## 8.2 イベント処理メソッド

ここでは、[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)が提供する以下のカテゴリのメソッドについての説明をします。

イベント処理メソッド

[OPEN-REPORTメソッド\(帳票開設イベント処理メソッド\)](#)  
[CLOSE-REPORTメソッド\(帳票閉鎖イベント処理メソッド\)](#)  
[OPEN-FORMメソッド\(帳票定義体開設イベント処理メソッド\)](#)  
[CLOSE-FORMメソッド\(帳票定義体閉鎖イベント処理メソッド\)](#)  
[READ-RECORDメソッド\(レコード読み込みイベント処理メソッド\)](#)  
[CALCULATE-DEメソッド\(明細計算イベント処理メソッド\)](#)  
[PRINT-DEメソッド\(明細出力イベント処理メソッド\)](#)  
[PAGE-HEADメソッド\(ページ頭書きイベント処理メソッド\)](#)  
[PAGE-FOOTメソッド\(ページ脚書きイベント処理メソッド\)](#)  
[CB-HEADメソッド\(制御頭書きイベント処理メソッド\)](#)  
[CB-FOOTメソッド\(制御脚書きイベント処理メソッド\)](#)  
[REPORT-HEADメソッド\(報告書頭書きイベント処理メソッド\)](#)  
[REPORT-FOOTメソッド\(報告書脚書きイベント処理メソッド\)](#)  
[ERROR-MESSAGEメソッド\(エラーメッセージイベント処理メソッド\)](#)  
[CANCEL-PRINTメソッド\(印刷キャンセルイベント処理メソッド\)](#)



注意

イベント処理メソッドの中では、STOP RUN文のようにプログラムの実行を中断することを行わないように注意してください。

### 8.2.1 OPEN-REPORTメソッド(帳票開設イベント処理メソッド)

#### イベント発生条件

[帳票印刷メソッド\(PRINT-CONTROL\)](#)実行中の、同メソッドが呼ばれた直後。

#### イベント処理後のクラスライブラリの動作

帳票印刷のための初期化処理および帳票定義体の開設を行います。

#### 記述形式

```

*>.1.*:.....2.....3.....4.....5.....6.....7.*:.....8
  IDENTIFICATION DIVISION.
  METHOD-ID.  OPEN-REPORT  OVERRIDE.
  DATA DIVISION.
  WORKING-STORAGE SECTION.
  LINKAGE SECTION.
  PROCEDURE DIVISION.

*>  SELFを使ってイベント処理を記述します。

      EXIT METHOD.
  END METHOD  OPEN-REPORT.
*>.1.*:.....2.....3.....4.....5.....6.....7.*:.....8

```

**引数**

なし。

**復帰値**

なし。

**注意事項**

このイベントは、[帳票印刷メソッド\(PRINT-CONTROL\)](#)実行中の同メソッドの呼出し直後に 1 回だけ発生します。ただし、[印刷種別プロパティ\(PRINT-TYPE\)](#)で“通常印刷 2 回読み”または“サマリ印刷 2 回読み”を設定した場合は、2 回発生します。

[印刷種別プロパティ\(PRINT-TYPE\)](#)に“通常印刷 2 回読み”または“サマリ印刷 2 回読み”が設定されている場合、1 回目に帳票開設イベントから帳票閉鎖イベントまでひとつおりのイベントが発生したあと、再び、帳票開設イベントから帳票閉鎖イベントまでのひとつおりのイベントが発生します。

また、2 回目の帳票開設イベントでは、COBOL ファイルやデータベースなどの読み込み対象レコード位置を初回の読み込み時と同じ位置にする処理を行うように注意してください。

**使用例**

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID. USING-FILE INHERITS FJPRT-CONTROL.
:
IDENTIFICATION DIVISION.
OBJECT.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT INPUT-FILE1 ASSIGN TO SYS006
    ORGANIZATION IS LINE SEQUENTIAL.
:
IDENTIFICATION DIVISION.
METHOD-ID. OPEN-REPORT OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.
    MOVE "ADDRLIST" TO FORM-NAME OF L-CTRLOBJ *>...[1]
    OPEN INPUT INPUT-FILE1 *>.....[2]
    EXIT METHOD.
END METHOD OPEN-REPORT.
:
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

**[例の説明]****[1]**

印刷で使用する帳票定義体の名前を指定します。

**[2]**

以降に発生するイベントで行順ファイルのレコード読み込みを行うために、ファイルを開いておきます。なお、このファイルのレコードを読み込む処理は通常は[レコード読み込みイベント\(READ-RECORDメソッド\)](#)で行うように設計します。またファイルのクローズは通常は[帳票閉鎖イベント\(CLOSE-REPORTメソッド\)](#)で行うように設計します。

## 8.2.2 CLOSE-REPORTメソッド(帳票閉鎖イベント処理メソッド)

### イベント発生条件

[帳票印刷メソッド\(PRINT-CONTROL\)](#)が終了する直前。  
帳票定義体閉鎖イベント発生のため。

### イベント処理後のクラスライブラリの動作

帳票印刷の後処理を行います。

### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  CLOSE-REPORT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.

*>  SELFを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  CLOSE-REPORT.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
```

### 引数

なし。

### 復帰値

なし。

### 注意事項

このイベントは、[帳票印刷メソッド\(PRINT-CONTROL\)](#)の実行終了直前に1回だけ発生します。ただし、帳票開設イベントで、[印刷種別プロパティ\(PRINT-TYPE\)](#)に“通常印刷 2回読み”または“サマリ印刷 2回読み”を設定した場合は、2回発生します。  
[印刷種別プロパティ\(PRINT-TYPE\)](#)に“通常印刷 2回読み”または“サマリ印刷 2回読み”が設定されているときは、1回目に帳票開設イベントから帳票閉鎖イベントまでのイベントがひととおり発生したあと、再び、帳票開設イベントから帳票閉鎖イベントまでのひと通りのイベントが発生します。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID. USING-FILE INHERITS FJPRT-CONTROL.
:
IDENTIFICATION DIVISION.
OBJECT.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT INPUT-FILE1 ASSIGN TO SYS006
    ORGANIZATION IS LINE SEQUENTIAL.
:
IDENTIFICATION DIVISION.
METHOD-ID. CLOSE-REPORT OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.
    CLOSE INPUT-FILE1 *>.....[1]
    EXIT METHOD.
END METHOD CLOSE-REPORT.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

[例の説明]

[1]

ほかのイベントで使われていた行順ファイルをクローズします。 帳票の印刷で使うファイルは通常は[帳票開設イベント \(OPEN-REPORTメソッド\)](#)でオープンするように設計します。また、ファイルのレコードを読み込む処理は通常は[レコード読み込みイベント \(READ-RECORDメソッド\)](#)で行うように設計します。

### 8.2.3 ERROR-MESSAGEメソッド(エラーメッセージイベント処理メソッド)

## イベント発生条件

[帳票印刷メソッド \(PRINT-CONTROL\)](#) 実行中に[印刷エラー](#)が発生したとき。

## イベント処理後のクラスライブラリの動作

ありません。

## 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  ERROR-MESSAGE  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.

*>  SELFを使ってイベント処理を記述します。

      EXIT METHOD.
END METHOD  ERROR-MESSAGE.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
```

## 引数

なし。

## 復帰値

なし。

## 注意事項

発生した[印刷エラー](#)に対応するエラーコードは[印刷ジョブエラーコードプロパティ \(PRINT-JOB-ERROR-CODE\)](#)で参照できます。必要に応じてこの値を使ってエラーメッセージの表示などの処理を記述してください。

印刷エラーの発生によって印刷の続行を中断したい場合は、[キャンセル要求プロパティ \(REQUEST-CANCEL\)](#)に真(B"1")を設定してください。

## 使用例

```
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
:
IDENTIFICATION DIVISION.
METHOD-ID.  ERROR-MESSAGE  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.
      DISPLAY  NC"エラーメッセージイベントが発生しました。"
      DISPLAY  PRINT-JOB-ERROR-CODE OF SELF  *>.....[1]
      MOVE  B"1" TO  REQUEST-CANCEL OF SELF  *>.....[2]
      EXIT METHOD.
END METHOD  ERROR-MESSAGE.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
```

[例の説明]

[1]

印刷エラーの種別を示す印刷ジョブエラーコードを表示します。

[2]

印刷をキャンセルします。

## 8.2.4 CANCEL-PRINTメソッド(印刷キャンセルイベント処理メソッド)

### イベント発生条件

印刷中のダイアログボックスから、オペレータが印刷中止を指示したとき。  
あるイベント処理メソッド内で[キャンセル要求プロパティ \(REQUEST-CANCEL\)](#)に真(B"1")を設定してそのイベント処理メソッドを抜けた後。

### イベント処理後のクラスライブラリの動作

印刷途中のデータをキャンセルします。

### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  CANCEL-PRINT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.

*>  SELFを使ってイベント処理を記述します。

      EXIT METHOD.
END METHOD  CANCEL-PRINT.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

### 引数

なし。

### 復帰値

なし。

### 注意事項

なし。

### 使用例

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
:
IDENTIFICATION DIVISION.
METHOD-ID.  CANCEL-PRINT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.
      DISPLAY  NC"印刷キャンセルイベントが発生しました。"  *>...[1]
      EXIT METHOD.
END METHOD  CANCEL-PRINT.
:
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

[例の説明]

[1]

印刷キャンセルイベントが発生したことを画面に表示し、帳票の印刷がキャンセルされた

ことをオペレータに通知します。

## 8.2.5 READ-RECORDメソッド(レコード読み込みイベント処理メソッド)

### イベント発生条件

帳票定義体開設イベント発生の際  
各明細/ボディ部/ページの印刷の繰返しごと

### イベント処理後のクラスライブラリの動作

[レコードEOFプロパティ \(RECORD-EOF\)](#)に真(B"1")を設定した場合は、制御脚書き、ページ脚書き、報告書脚書きを印刷し(このとき、各パーティションに対応するイベントも発生します)、続いて帳票定義体閉鎖を行います。

### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  READ-RECORD  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.

*>  SELFを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  READ-RECORD.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

### 引数

なし。

### 復帰値

なし。

### 注意事項

このイベントは、レコード/ファイルの終わり(EOF)をプログラムが検出したことを示すために[レコードEOFプロパティ \(RECORD-EOF\)](#)に真(B"1")を設定するまで、発生を繰り返します。EOFが通知されないと、[帳票印刷メソッド \(PRINT-CONTROL\)](#)内部で無限ループしますので必ず終了条件を設けて EOFを通知するように注意してください。

[レコード回数プロパティ \(RECORD-COUNT\)](#)を利用することで、レコードの読み込み回数を判断することができます。レコード回数プロパティ (RECORD-COUNT)は、初回のレコード読み込みイベント発生時には、1が設定されています。



## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID. USING-FILE INHERITS FJPRT-CONTROL.
:
IDENTIFICATION DIVISION.
OBJECT.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT INPUT-FILE1 ASSIGN TO SYS006
    ORGANIZATION IS LINE SEQUENTIAL.
DATA DIVISION.
FILE SECTION.
FD INPUT-FILE1.
01.
    02 F-NAME1      PIC X(28).
    02 FILLER       PIC X(1).
    02 F-AGE1       PIC 9(3).
    02 FILLER       PIC X(1).
    02 F-ADDRESS1   PIC X(45).
WORKING-STORAGE SECTION.
    COPY ADDRLIST OF XMDLIB.
01 W-ADDAGE        PIC 9(5).
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID. READ-RECORD OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.
    READ INPUT-FILE1 AT END *>.....[1]
    MOVE B"1" TO RECORD-EOF OF SELF
    NOT AT END *>.....[2]
    INVOKE SELF "GET-RECORD-CONTENT" USING ADDRLIST
    MOVE F-NAME1 TO NAME1 OF ADDRLIST
    MOVE F-AGE1 TO AGE1 OF ADDRLIST
    MOVE F-ADDRESS1 TO ADDRESS1 OF ADDRLIST
    INVOKE SELF "SET-RECORD-CONTENT" USING ADDRLIST
END-READ
EXIT METHOD.
END METHOD READ-RECORD.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

## [1]

ほかのイベントで既にオープンされている行順ファイルの 1レコードを読みます。読み込みでファイル終了条件が発生したら[レコードEOFプロパティ\(RECORD-EOF\)](#)を真にしてレコード読み込みが終了したことを帳票印刷クラスライブラリに通知します。

なお、帳票の印刷で使うファイルは通常は[帳票開設イベント\(OPEN-REPORTメソッド\)](#)でオ

ーブンするに設計します。また、ファイルのクローズは通常は[帳票閉鎖イベント \(CLOSE-REPORTメソッド\)](#)で行うように設計します。

[2]

正常にレコードが読み込めたら、読んだレコードの内容を帳票印刷クラスライブラリが管理しているレコードに転記します。当メソッドの終了後、別のイベント処理([明細出力イベント](#))後にこのレコードの内容がページ/明細/ボディ部のいずれとして(これは帳票定義体の種別に依存します)印刷されます。

## 8.2.6 CALCULATE-DEメソッド(明細計算イベント処理メソッド)

### イベント発生条件

レコードが読み込まれるたびに、その後に発生します。

### イベント処理後のクラスライブラリの動作

ありません。

### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  CALCULATE-DE  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.

*>  SELFを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  CALCULATE-DE.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

### 引数

なし。

### 復帰値

なし。

### 注意事項

なし。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-CONTROL  INHERITS  FJPRT-CONTROL.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  ADDRLIST OF XMDLIB.
    COPY  ADDRLIST OF XMDLIB
        REPLACING  ADDRLIST  BY  ADDRLIST-PRE.
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID.  CALCULATE-DE  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.
    INVOKE  SELF  "GET-RECORD-CONTENT"  USING  ADDRLIST  *>.....[1]
    INVOKE  SELF  "GET-PRE-RECORD-CONTENT"  USING  ADDRLIST-PRE  *>...[2]
    MOVE  NAME1  OF ADDRLIST-PRE  TO  NAME2  OF ADDRLIST  *>.....[3]
    MOVE  ADDRESS1 OF ADDRLIST-PRE  TO  ADDRESS2 OF ADDRLIST  *>.....[4]
    INVOKE  SELF  "SET-RECORD-CONTENT"  USING  ADDRLIST  *>.....[5]
    EXIT METHOD.
END METHOD  CALCULATE-DE.
:
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

## [例の説明]

- [1]  
レコードの内容を参照します。
- [2]  
前レコードの内容を参照します。
- [3]  
前レコードの NAME1 項目の値を NAME2 に転記します。
- [4]  
前レコードの ADDRESS1 項目の値を ADDRESS2 に転記します。
- [5]  
新たに転記された項目の値をレコードに反映します。

## 8.2.7 PRINT-DEメソッド(明細出力イベント処理メソッド)

### イベント発生条件

明細/ボディ部/ページの印刷の直前に発生します。

### イベント処理後のクラスライブラリの動作

明細/ボディ部/ページが存在し印刷が抑止されていなければ、明細/ボディ部/ページを印刷します。

## 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  PRINT-DE  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.

*>  SELFを使ってイベント処理を記述します。

      EXIT METHOD.
END METHOD  PRINT-DE.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
```

## 引数

なし。

## 復帰値

なし。

## 注意事項

印刷する明細の切替えは、複数の明細が定義された集計表形式の帳票定義体を利用する場合に有効となります。

明細の切替えについては “ [明細名プロパティ \(DE-NAME\)](#) ” を参照してください。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID. COLOR-CHANGE INHERITS FJPRT-CONTROL.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
    SYMBOLIC CONSTANT
    COPY FJPRTSC.
.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY FORM1 OF XMDLIB.
    01 SEIBETSU1-NAME PIC X(9) VALUE "SEIBETSU1".
    01 ACCUMULATOR1 PIC 9(10).
PROCEDURE DIVISION.
.
IDENTIFICATION DIVISION.
METHOD-ID. PRINT-DE OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01 W-COLOR PIC S9(4) COMP-5.
LINKAGE SECTION.
PROCEDURE DIVISION.
    INVOKE SELF "GET-RECORD-CONTENT"
        USING FORM1 *>.....[1]
    EVALUATE SEIBETSU1 OF FORM1 *>.....[2]
    WHEN NC"男"
        MOVE FJPRT-SC-COLOR-BLUE TO W-COLOR
    WHEN NC"女"
        MOVE FJPRT-SC-COLOR-RED TO W-COLOR
    WHEN OTHER *> 性別が未入力の場合
        MOVE FJPRT-SC-COLOR-BLACK TO W-COLOR
    END-EVALUATE
    INVOKE SELF "SET-TEXT-COLOR"
        USING SEIBETSU1-NAME W-COLOR *>....[3]
    ADD VALUE1 OF FORM1 TO ACCUMULATOR1 *>....[4]
    EXIT METHOD.
END METHOD PRINT-DE.
.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

## [例の説明]

- [1]  
レコードの内容を参照します。
- [2]  
SEIBETSU1 項目の値を調べ、色に対応する数値を変数に設定します。
- [3]  
SEIBETSU1 項目の色を設定します。

[4]

フッタに関するほかのイベントで使われる変数への加算を行います。なお、この変数の初期化はヘッダに関するイベントで行うように設計します。

## 8.2.8 OPEN-FORMメソッド(帳票定義体開設イベント処理メソッド)

### イベント発生条件

帳票定義体を開いたとき。  
帳票開設イベントのあと。

### イベント処理後のクラスライブラリの動作

ありません。

### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID. OPEN-FORM OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-FORMNAME PIC X(518).
PROCEDURE DIVISION USING L-FORMNAME.

*> SELFを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD OPEN-FORM.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

### 引数

**L-FORMNAME** 【引数の属性: PIC X(518)】:

イベントの処理対象となった帳票定義体名が渡ってきます。引数となるデータ項目は引数の属性で示した属性で宣言してください。データ項目の名前は任意に付けてください。当イベント処理中に[帳票定義体名プロパティ \(FORM-NAME\)](#)で印刷する帳票定義体を切り替えた後でも、イベントの処理対象となった帳票定義体を参照することができます。

### 復帰値

なし。

### 注意事項

各イベントでも同様の処理を記述できますが、帳票印刷の起動時にだけ処理をすればよい項目に関しては、このイベントで記述することによって性能を向上させることができます。ページ形式やタックシール形式の帳票定義体の印刷処理のための初期化に有効です。

帳票定義体名を変更した場合、帳票定義体開設イベントと帳票定義体閉鎖イベントが発生する場合があります。

複数の帳票定義体を利用する場合のイベント発生の流れについては、“[複数の帳票定義体を利用する場合のイベント発生の流れ](#)”を参照してください。

### 8.2.9 CLOSE-FORMメソッド(帳票定義体閉鎖イベント処理メソッド)

#### イベント発生条件

帳票定義体が閉じるとき。  
帳票閉鎖イベントの前。

#### イベント処理後のクラスライブラリの動作

ありません。

#### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID. CLOSE-FORM OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-FORMNAME PIC X(518).
PROCEDURE DIVISION USING L-FORMNAME.

*> SELFを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD CLOSE-FORM.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

#### 引数

**L-FORMNAME** 【引数の属性: PIC X(518)】:

イベントの処理対象の帳票定義体名が渡ってきます。引数となるデータ項目は引数の属性で示した属性で宣言してください。データ項目の名前は任意に付けてください。  
帳票定義体を[帳票定義体名プロパティ \(FORM-NAME\)](#)で設定し直した後に発生した当イベントでは、当引数には切り替え前の帳票定義体の名前が格納されています。しかし、このとき帳票定義体名プロパティ (FORM-NAME)の参照値は、まだ当イベントの処理対象となっていない切り替え後の帳票定義体の名前が返却されます。

#### 復帰値

なし。

#### 注意事項

当イベントで帳票の項目への設定を行っても、以前の設定値で帳票印刷が行われます。  
ただし、[印刷種別プロパティ \(PRINT-TYPE\)](#)で、“通常印刷2回読み”または“サマリ印刷2回読み”を指定した場合に限り、帳票の項目に設定した結果が1ページ目に印刷されます。  
帳票定義体を切り替えた場合、帳票定義体開設イベントと帳票定義体閉鎖イベントが発生する場合があります。  
複数の帳票定義体を利用する場合のイベント発生の流れについては、“[複数の帳票定義体を利用する場合のイベント発生の流れ](#)”を参照してください。

### 8.2.10 CB-HEADメソッド(制御頭書きイベント処理メソッド)

#### イベント発生条件

以下のいずれかの条件によって[コントロールブレーク](#)が発生したとき。

あるブレーク項目の値が変化して、対応する[制御脚書きイベント](#)が発生したあと。  
最初の[レコード読み込みイベント](#)が発生したあと。

## イベント処理後のクラスライブラリの動作

制御頭書きパーティションが存在し印刷が抑止されていなければ、制御頭書きを印刷します。

## 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  CB-HEAD  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-FIELDNAME PIC X(60).
PROCEDURE DIVISION  USING L-FIELDNAME.

*> SELFを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  CB-HEAD.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

## 引数

**L-FIELDNAME** 【引数の属性: PIC X(60)】:

制御ブレーク項目名が渡ってきます。これは、当イベントの発生対象となったコントロールブレークが発生した項目の名前を持ちます。引数のデータ項目の宣言ではデータ項目の名前を任意に付けてください。

当引数は、動作モードがUnicodeで、制御ブレーク項目名が日本語のとき、最大で20文字までしか受け取ることができませんので、注意してください。[参照] “[Unicode固有の留意事項](#)”

## 復帰値

なし。

## 注意事項

このイベントは、集計表形式の帳票定義体でだけ発生します。

ブレーク項目が定義されていない制御頭書きパーティションに対しても当イベントが発生します。この場合は、制御ブレーク項目名引数(L-FIELDNAME)には空白が渡されます。



## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  CB-USING INHERITS  FJPRT-CONTROL.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  FORM2 OF XMDLIB.
    01  KA-GOUKEI  PIC  9(4).
    01  BU-GOUKEI  PIC  9(4).
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID.  CB-HEAD  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-FIELDNAME  PIC  X(60).
PROCEDURE DIVISION  USING  L-FIELDNAME.
    INVOKE  SELF  "GET-RECORD-CONTENT"
                USING  FORM2  *>.....[1]
    EVALUATE  L-FIELDNAME  *>.....[2]
    WHEN      "部名"
        MOVE  ZERO  TO  BU-GOUKEI  *>.....[3]
        MOVE  部名 OF FORM2  TO  DIV-NAME OF FORM2  *>...[4]
    WHEN      "課名"
        MOVE  ZERO  TO  KA-GOUKEI  *>.....[3']
        MOVE  課名 OF FORM2  TO  SEC-NAME OF FORM2  *>...[4']
    END-EVALUATE
    INVOKE  SELF  "SET-RECORD-CONTENT"
                USING  FORM2  *>.....[5]
    EXIT METHOD.
END METHOD  CB-HEAD.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

[1]

レコードの内容を参照します。

[2]

制御ブレイク項目名を参照して、処理内容を切りわけます。ここでの“部名”という項目と“課名”という項目はブレイク項目として帳票定義体に定義しており、帳票印刷の始めと終わりと、これらの項目の値が変化したときに[コントロールブレイク](#)が発生してこのイベント処理メソッドに制御が渡ってきます。

[3],[3']

[制御脚書きイベント](#)で使う合計の値を初期化します。このような場合、[明細出力イベント](#)で合計の元となる値を加算するように設計します。

[4],[4']

ブレイク項目に対応する各パーティションの項目に文字を設定します。

[5]

変更された項目の内容をレコードに反映します。

### 8.2.11 CB-FOOTメソッド(制御脚書きイベント処理メソッド)

#### イベント発生条件

以下のいずれかの条件によって[コントロールブレイク](#)が発生したとき。

あるブレイク項目の値が変化したあと。

[レコード読み込みイベント](#)で[レコードEOFプロパティ \(RECORD-EOF\)](#)に真を設定したあと。

#### イベント処理後のクラスライブラリの動作

制御脚書きパーティションが存在し印刷が抑止されていなければ、制御脚書きを印刷します。

#### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  CB-FOOT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-FIELDNAME  PIC  X(60).
PROCEDURE DIVISION  USING  L-FIELDNAME.

*>  SELFを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  CB-FOOT.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

#### 引数

**L-FIELDNAME** 【引数の属性: PIC X(60)】:

制御ブレイク項目名が渡ってきます。これは、当イベントの発生対象となったコントロールブレイクが発生した項目の名前を持ちます。引数のデータ項目の宣言ではデータ項目の名前を任意に付けてください。

当引数は、動作モードがUnicodeで、制御ブレイク項目名が日本語のとき、最大で20文字までしか受け取ることができませんので、注意してください。[参照] “[Unicode固有の留意事項](#)”

#### 復帰値

なし。

#### 注意事項

このイベントは、集計表形式の帳票定義体でだけ発生します。

ブレイク項目が定義されていない制御脚書きパーティションに対しても当イベントが発生します。この場合は、制御ブレイク項目名引数(L-FIELDNAME)には空白が渡されます。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  CB-USING  INHERITS  FJPRT-CONTROL.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  FORM2 OF XMDLIB.
    01  KA-GOUKEI  PIC  9(4).
    01  BU-GOUKEI  PIC  9(4).
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID.  CB-FOOT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-FIELDNAME  PIC  X(60).
PROCEDURE DIVISION  USING  L-FIELDNAME.
    INVOKE  SELF  "GET-RECORD-CONTENT"
                USING  FORM2  *>.....[1]
    EVALUATE  L-FIELDNAME  *>.....[2]
    WHEN      "部名"
        MOVE  BU-GOUKEI  TO  BU-GOUKEI1 OF FORM2  *>...[3]
    WHEN      "課名"
        MOVE  KA-GOUKEI  TO  KA-GOUKEI1 OF FORM2  *>...[3']
    END-EVALUATE
    INVOKE  SELF  "SET-RECORD-CONTENT"
                USING  FORM2  *>.....[4]
    EXIT METHOD.
END METHOD  CB-FOOT.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

[1]

レコードの内容を参照します。

[2]

制御ブレイク項目名を参照して、処理内容を切りわけます。ここでの“部名”という項目と“課名”という項目はブレイク項目として帳票定義体に定義しており、帳票印刷の始めと終わりと、これらの項目の値が変化したときに[コントロールブレイク](#)が発生してこのイベント処理メソッドに制御が渡ってきます。

[3], [3']

ほかのイベントで求められた合計を各パーティションの項目に設定します。このような場合、[制御頭書きイベント](#)で合計用の変数を初期化し、[明細出カイベント](#)で合計の元となる値を加算するように設計します。

[4]

変更された項目の内容をレコードに反映します。

## 8.2.12 PAGE-HEADメソッド(ページ頭書きイベント処理メソッド)

### イベント発生条件

各ページの先頭から印刷が行われようとする前に発生します。

### イベント処理後のクラスライブラリの動作

ページ頭書きパーティションが存在し印刷が抑止されていなければ、ページ頭書きを印刷します。

### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID. PAGE-HEAD OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.

*> SELFを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD PAGE-HEAD.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

### 引数

なし。

### 復帰値

なし。

### 注意事項

このイベントは、[帳票定義体の種別](#)が段組み伝票形式または集計表形式の場合だけ発生します。

印刷で使用する帳票定義体のページ頭書きパーティションの有無にかかわらず、このイベントは発生します。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-CONTROL  INHERITS  FJPRT-CONTROL.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  FORM3 OF XMDLIB.
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID.  PAGE-HEAD  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.
    INVOKE  SELF  "GET-RECORD-CONTENT"  USING  ADDRLIST  *>...[1]
    MOVE  PRINTING-PAGE-NUMBER  OF  SELF
        TO  PH-PAGE-COUNT OF FORM3  *>.....[2]
    INVOKE  SELF  "SET-RECORD-CONTENT"  USING  ADDRLIST  *>...[3]
    EXIT METHOD.
END METHOD  PAGE-HEAD.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

[例の説明]

[1]

レコードの内容を参照します。

[2]

ページ頭書き/ヘッダ部パーティションに存在する項目に印刷中ページ番号を設定します。

[3]

新たに転記された項目の値をレコードに反映します。

### 8.2.13 PAGE-F00Tメソッド(ページ脚書きイベント処理メソッド)

#### イベント発生条件

印刷が各ページの最終行になる直前

レコード読みイベント処理メソッドで[レコードEOFプロパティ \(RECORD-EOF\)](#)に真(B"1")を設定してイベント処理メソッドを抜けた後。

#### イベント処理後のクラスライブラリの動作

ページ脚書きパーティションが存在し印刷が抑止されていない場合は、ページ脚書きを印刷します。

## 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  PAGE-FOOT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.

*>  SELFを使ってイベント処理を記述します。

      EXIT METHOD.
END METHOD  PAGE-FOOT.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
```

## 引数

なし。

## 復帰値

なし。

## 注意事項

このイベントは、[帳票定義体の種別](#)が伝票形式または集計表形式の場合だけ発生します。印刷で使用する帳票定義体のページ脚書きパーティションの有無にかかわらず、このイベントは発生します。

## 使用例

```
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-CONTROL  INHERITS  FJPRT-CONTROL.
      :
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
      COPY  FORM3 OF XMDLIB.
PROCEDURE DIVISION.
      :
IDENTIFICATION DIVISION.
METHOD-ID.  PAGE-FOOT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.
      INVOKE  SELF  "GET-RECORD-CONTENT"  USING  ADDRLIST  *>...[1]
      MOVE  PRINTING-PAGE-NUMBER  OF  SELF
            TO  PF-PAGE-COUNT OF FORM3  *>.....[2]
      INVOKE  SELF  "SET-RECORD-CONTENT"  USING  ADDRLIST  *>...[3]
      EXIT METHOD.
END METHOD  PAGE-FOOT.
      :
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
```

[例の説明]

- [1]  
レコードの内容を参照します。
- [2]  
ページ脚書き/フッタ部パーティションに存在する項目に印刷中ページ番号を設定します。
- [3]  
新たに転記された項目の値をレコードに反映します。

## 8.2.14 REPORT-HEADメソッド(報告書頭書きイベント処理メソッド)

### イベント発生条件

帳票印刷のため読み込まれた 1 件目のレコードを対象として、明細計算イベントが発生した後に発生します。

### イベント処理後のクラスライブラリの動作

報告書頭書きパーティションが存在し印刷が抑止されていなければ、報告書頭書きを印刷します。

### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  REPORT-HEAD  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION.

*>  SELFを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  REPORT-HEAD.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
```

### 引数

なし。

### 復帰値

なし。

### 注意事項

このイベントは、集計表形式の帳票定義体でだけ発生するイベントです。  
報告書頭書きパーティションの有無にかかわらず、このイベントは発生します。

## 使用例

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-CONTROL  INHERITS  FJPRT-CONTROL.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  FORM3 OF XMDLIB.
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID.  REPORT-HEAD  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01  W-DATE.
        02  W-YYYY  PIC  9(4).
        02  W-MM    PIC  9(2).
        02  W-DD    PIC  9(2).
        02  FILLER  PIC  X(13).
LINKAGE SECTION.
PROCEDURE DIVISION.
    INVOKE SELF "GET-RECORD-CONTENT" USING ADDRLIST *>...[1]
    MOVE FUNCTION CURRENT-DATE TO W-DATE *>.....[2]
    MOVE W-YYYY TO REP-HEAD-YYTT OF FORM3
    MOVE W-MM   TO REP-HEAD-MM   OF FORM3
    MOVE W-DD   TO REP-HEAD-DD   OF FORM3
    INVOKE SELF "SET-RECORD-CONTENT" USING ADDRLIST *>...[3]
    EXIT METHOD.
END METHOD  REPORT-HEAD.
:
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

[例の説明]

- [1]  
レコードの内容を参照します。
- [2]  
報告書頭書きパーティションに存在する項目に年、月、日の値を設定します。
- [3]  
新たに転記された項目の値をレコードに反映します。

### 8.2.15 REPORT-F00Tメソッド(報告書脚書きイベント処理メソッド)

#### イベント発生条件

印刷するすべてのレコードが処理されたあとに発生します。

#### イベント処理後のクラスライブラリの動作

報告書脚書きパーティションが存在し印刷が抑止されていなければ、報告書脚書きを印刷します。



**記述形式**

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
  IDENTIFICATION DIVISION.
  METHOD-ID.  REPORT-FOOT  OVERRIDE.
  DATA DIVISION.
  WORKING-STORAGE SECTION.
  LINKAGE SECTION.
  PROCEDURE DIVISION.

  *>  SELFを使ってイベント処理を記述します。

      EXIT METHOD.
  END METHOD  REPORT-FOOT.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

**引数**

なし。

**復帰値**

なし。

**注意事項**

このイベントは、集計表形式の帳票定義体でだけ発生します。  
 帳票定義体に報告書脚書きパーティションの有無にかかわらず、このイベントは発生します。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-CONTROL  INHERITS  FJPRT-CONTROL.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  FORM3 OF XMDLIB.
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID.  REPORT-FOOT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01  W-TIME.
        02  W-HH  PIC  9(2).
        02  W-MM  PIC  9(2).
        02  W-SS  PIC  9(2).
        02  W-MS  PIC  9(2).
LINKAGE SECTION.
PROCEDURE DIVISION.
    INVOKE  SELF  "GET-RECORD-CONTENT"  USING  ADDRLIST  *>...[1]
    ACCEPT  W-TIME  FROM TIME  *>.....[2]
    MOVE  W-HH  TO  REP-FOOT-HH OF FORM3
    MOVE  W-MM  TO  REP-FOOT-MM OF FORM3
    MOVE  W-SS  TO  REP-FOOT-SS OF FORM3
    INVOKE  SELF  "SET-RECORD-CONTENT"  USING  ADDRLIST  *>...[3]
    EXIT METHOD.
END METHOD  REPORT-FOOT.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

[1]

レコードの内容を参照します。

[2]

報告書脚書きパーティションに存在する項目に時、分、秒の値を設定します。

[3]

新たに転記された項目の値をレコードに反映します。

## 8.3 帳票項目のプロパティに関するメソッド

ここでは、[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)が提供する以下のカテゴリのメソッドについての説明をします。

帳票項目のプロパティに関するメソッド

[SET-RECORD-CONTENTメソッド\(レコード内容の設定メソッド\)](#)  
[GET-RECORD-CONTENTメソッド\(レコード内容の参照メソッド\)](#)  
[GET-PRE-RECORD-CONTENTメソッド\(前レコード内容の参照メソッド\)](#)  
[SET-TEXT-COLORメソッド\(文字色プロパティ設定メソッド\)](#)  
[GET-TEXT-COLORメソッド\(文字色プロパティ参照メソッド\)](#)  
[SET-DISABLE-FIELDメソッド\(項目抑止プロパティ設定メソッド\)](#)  
[GET-DISABLE-FIELDメソッド\(項目抑止プロパティ参照メソッド\)](#)  
[SET-EXTEND-CHARACTERメソッド\(文字引伸しプロパティ設定メソッド\)](#)  
[GET-EXTEND-CHARACTERメソッド\(文字引伸しプロパティ参照メソッド\)](#)  
[SET-FONT-STRIKE-OUTメソッド\(抹消線プロパティ設定メソッド\)](#)  
[GET-FONT-STRIKE-OUTメソッド\(抹消線プロパティ参照メソッド\)](#)  
[SET-FONT-ITALICメソッド\(イタリックプロパティ設定メソッド\)](#)  
[GET-FONT-ITALICメソッド\(イタリックプロパティ参照メソッド\)](#)  
[SET-FONT-BOLDメソッド\(ボールドプロパティ設定メソッド\)](#)  
[GET-FONT-BOLDメソッド\(ボールドプロパティ参照メソッド\)](#)  
[SET-FONT-UNDERLINEメソッド\(下線プロパティ設定メソッド\)](#)  
[GET-FONT-UNDERLINEメソッド\(下線プロパティ参照メソッド\)](#)  
[SET-FIELD-UNDERLINEメソッド\(項目下線プロパティ設定メソッド\)](#)  
[GET-FIELD-UNDERLINEメソッド\(項目下線プロパティ参照メソッド\)](#)

以下に、本文の引数/復帰値の項目を読む上での注意事項を挙げます。



注意

引数/復帰値の項目に書かれている“属性”は、メソッドの定義上のインタフェースを表すものです。

PIC X ANY LENGTH の属性を持つ引数には、任意の長さを持つ英数字のデータ項目を記述してください。なお、この引数には値が返却されることがあります。これについては各メソッドの仕様を確認してください。

- |       |                                       |
|-------|---------------------------------------|
| 8.3.1 | SET-RECORD-CONTENTメソッド(レコード内容の設定メソッド) |
| 8.3.2 | GET-RECORD-CONTENTメソッド(レコード内容の参照メソッド) |

### 説明

帳票印刷クラスライブラリが管理している“レコード”の内容(帳票項目の集団項目)を設定/参照することにより、[印刷する文字を操作](#)します。

レコードについては“MeFt 説明書”の“レコードとは”を参照してください。

## 記述形式

### 【設定】

```
INVOKE CONTROL-OBJECT-2 "SET-RECORD-CONTENT"
      USING [BY REFERENCE] RECORD-CONTENT
```

### 【参照】

```
INVOKE CONTROL-OBJECT-2 "GET-RECORD-CONTENT"
      USING [BY REFERENCE] RECORD-CONTENT
```

## 引数

**RECORD-CONTENT** 【引数の属性: PIC X ANY LENGTH】:

帳票定義体を COPY文で取り込んで展開した集団項目のデータ名を記述します。

設定の場合、COBOLの文で設定した帳票項目を持つ集団項目が帳票印刷クラスライブラリが管理しているレコードに渡され、印刷時に利用されます。

集団項目内の各項目は、帳票定義体をCOBOLの[登録集として取り込んだ](#)際に宣言されていたものと同じ型/領域長を持ちます。

参照の場合、帳票印刷クラスライブラリが管理しているレコード(帳票項目の集団項目)の値が渡されます。

## 例外エラー

設定/参照で、例外エラー(コード: [FJPRT-SCE-ILLEGAL-STATE](#))が発生することがあります。

## 注意事項

### 【例外エラーコード: FJPRT-SCE-ILLEGAL-STATEの発生】

イベント以外の処理記述からと、以下のイベント処理記述内から当メソッドを呼び出した場合、[例外エラー](#)(コード: [FJPRT-SCE-ILLEGAL-STATE](#))が発生します。下記以外のイベント処理メソッドでだけ当メソッドを呼び出すようにプログラムを修正してください。

[帳票開設イベント](#)(OPEN-REPORTメソッド)

[帳票閉鎖イベント](#)(CLOSE-REPORTメソッド)

[エラーメッセージイベント](#)(ERROR-MESSAGEメソッド)

[印刷キャンセルイベント](#)(CANCEL-PRINTメソッド)

## 使用例

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[レコード読み込みメソッド\(READ-RECORD\)の使用例](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[レコード読み込みメソッド\(READ-RECORD\)の使用例](#)

## 8.3.3 GET-PRE-RECORD-CONTENTメソッド(前レコード内容の参照メソッド)

## 説明

帳票印刷クラスライブラリがレコード読み込みイベント発生の前に内部的に保存したレコード([前レコード](#))の内容(帳票項目の集団項目)を参照します。

レコードについては“MeFt 説明書”の“レコードとは”を参照してください。

## 記述形式

### 【参照】

```
INVOKE CONTROL-OBJECT-2 "GET-PRE-RECORD-CONTENT"
      USING [BY REFERENCE] RECORD-CONTENT
```

## 引数

**RECORD-CONTENT** 【引数の属性: PIC X ANY LENGTH】:

帳票定義体を COPY文で取り込んで展開した集団項目のデータ名を記述します。

帳票印刷クラスライブラリ が前回の印刷で管理していたレコード(帳票項目の集団項目)の値が渡されます。  
集団項目内の各項目は、帳票定義体をCOBOLの登録集として取り込んだ際に宣言されていたものと同じ型/領域長を持ちます。

### 例外エラー

参照で、例外エラー(コード: [FJPRT-SCE-ILLEGAL-STATE](#))が発生することがあります。

### 注意事項

[参照] “ [例外エラーコード: FJPRT-SCE-ILLEGAL-STATEの発生](#) ”

### 使用例

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の  
[明細計算イベントメソッド\(CALCULATE-DE\)の使用例](#)  
帳票印刷制御クラス(FJPRT-CONTROL)の  
[明細計算イベントメソッド\(CALCULATE-DE\)の使用例](#)

## 8.3.4 SET-TEXT-COLORメソッド(文字色プロパティ設定メソッド)

## 8.3.5 GET-TEXT-COLORメソッド(文字色プロパティ参照メソッド)

### 説明

帳票項目の文字色を設定/参照します。  
文字色として、基本色の16色が帳票定義体のパレットで定義した色が指定できます。  
帳票定義体に定義された項目の状態が、デフォルト値となります。

記号定数	文字色
FJPRT-SC-COLOR-BLACK	黒色
FJPRT-SC-COLOR-BLUE	青色
FJPRT-SC-COLOR-RED	赤色
FJPRT-SC-COLOR-YELLOW	黄色
FJPRT-SC-COLOR-GREEN	緑色
FJPRT-SC-COLOR-PINK	桃色
FJPRT-SC-COLOR-SKYBLUE	水色
FJPRT-SC-COLOR-GRAY	灰色
FJPRT-SC-COLOR-DARKBLUE	暗青色
FJPRT-SC-COLOR-DARKRED	暗赤色
FJPRT-SC-COLOR-DARKGREEN	暗緑色
FJPRT-SC-COLOR-DARKYELLOW	暗黄色
FJPRT-SC-COLOR-DARKPINK	暗桃色
FJPRT-SC-COLOR-DARKSKYBLUE	暗水色
FJPRT-SC-COLOR-DARKGRAY	暗灰色
FJPRT-SC-COLOR-WHITE	白色
FJPRT-SC-COLOR-PALET0	パレット0
FJPRT-SC-COLOR-PALET1	パレット1
FJPRT-SC-COLOR-PALET2	パレット2
FJPRT-SC-COLOR-PALET3	パレット3
FJPRT-SC-COLOR-PALET4	パレット4
FJPRT-SC-COLOR-PALET5	パレット5
FJPRT-SC-COLOR-PALET6	パレット6
FJPRT-SC-COLOR-PALET7	パレット7
FJPRT-SC-COLOR-PALET8	パレット8
FJPRT-SC-COLOR-PALET9	パレット9
FJPRT-SC-COLOR-PALET10	パレット10
FJPRT-SC-COLOR-PALET11	パレット11
FJPRT-SC-COLOR-PALET12	パレット12
FJPRT-SC-COLOR-PALET13	パレット13
FJPRT-SC-COLOR-PALET14	パレット14
FJPRT-SC-COLOR-PALET15	パレット15

## 記述形式

### 【設定】

```
INVOKE CONTROL-OBJECT-2 "SET-TEXT-COLOR"
      USING [BY REFERENCE] FIELD-NAME
          TEXT-COLOR
```

### 【参照】

```
INVOKE CONTROL-OBJECT-2 "GET-TEXT-COLOR"
      USING [BY REFERENCE] FIELD-NAME
      RETURNING TEXT-COLOR
```

## 引数 / 復帰値

***FIELD-NAME*** 【引数の属性: PIC X ANY LENGTH】:

設定/参照する対象となる帳票項目の名前を渡します。引数には任意の長さの英数字項目を記述します。

***TEXT-COLOR*** 【引数/復帰値の属性: PIC S9(4) [USAGE IS] COMP-5】:

設定時には、指定した帳票項目に設定する文字色を渡します。引数の属性と同じ属性を持

つデータ項目が定数を記述します。

参照時には、指定した帳票項目に設定されている文字色が返ります。引数の属性と同じ属性を持つデータ項目を記述します。

### 例外エラー

設定で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-SET-VALUE](#)

[FJPRT-SCE-FIELD-NAME](#)

[FJPRT-SCE-ILLEGAL-STATE](#)

[FJPRT-SCE-ARRAY-INDEX](#)

参照で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)

[FJPRT-SCE-ILLEGAL-STATE](#)

[FJPRT-SCE-ARRAY-INDEX](#)

### 注意事項

[参照] “[例外エラーコード: FJPRT-SCE-ILLEGAL-STATEの発生](#)”

### 使用例

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[明細計算メソッド\(PRINT-DE\)の使用例](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[明細計算メソッド\(PRINT-DE\)の使用例](#)

**8.3.6 SET-DISABLE-FIELDメソッド(項目抑止プロパティ設定メソッド)**

**8.3.7 GET-DISABLE-FIELDメソッド(項目抑止プロパティ参照メソッド)**

### 説明

帳票項目の印刷を抑止するか否かを設定/参照します。

デフォルト値として“抑止しない”を持ちます。

値	項目抑止
B"1" (真)	抑止する
B"0" (偽)	抑止しない

### 記述形式

#### 【設定】

```
INVOKE CONTROL-OBJECT-2 "SET-DISABLE-FIELD"
      USING [BY REFERENCE] FIELD-NAME
           BOOL-VAL
```

#### 【参照】

```
INVOKE CONTROL-OBJECT-2 "GET-DISABLE-FIELD"
      USING [BY REFERENCE] FIELD-NAME
      RETURNING BOOL-VAL
```

### 引数 / 復帰値

*FIELD-NAME* 【引数の属性: PIC X ANY LENGTH】:

設定/参照する対象となる帳票項目の名前を渡します。引数には任意の長さの英数字項目を記述します。

**BOOL-VAL** 【引数/復帰値の属性: PIC 1(1) [USAGE IS] DISPLAY】:

設定時には、指定した帳票項目を抑止するか否かを渡します。引数の属性と同じ属性を持つデータ項目か定数を記述します。

参照時には、指定した帳票項目が抑止されるか否かが返ります。引数の属性と同じ属性を持つデータ項目を記述します。

### 例外エラー

設定で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)

[FJPRT-SCE-ILLEGAL-STATE](#)

[FJPRT-SCE-ARRAY-INDEX](#)

参照で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)

[FJPRT-SCE-ILLEGAL-STATE](#)

[FJPRT-SCE-ARRAY-INDEX](#)

### 注意事項

[参照] “ [例外エラーコード: FJPRT-SCE-ILLEGAL-STATEの発生](#) ”

**8.3.8 SET-EXTEND-CHARACTERメソッド(文字引伸しプロパティ設定メソッド)**

**8.3.9 GET-EXTEND-CHARACTERメソッド(文字引伸しプロパティ参照メソッド)**

### 説明

帳票項目の文字引伸しを設定/参照します。

文字引伸しとは、指定された文字を項目領域長分繰り返して印刷する機能です。

文字引伸しには、1文字だけの英数字を受け渡します。

文字引伸しに空白を指定した場合、または該当項目のレコードにデータが格納されていない場合は、文字引伸しは行われません。

デフォルト値として“空白”を持ちます。

### 記述形式

【設定】

```
INVOKE CONTROL-OBJECT-2 "SET-EXTEND-CHARACTER"
      USING [BY REFERENCE] FIELD-NAME
           [BY REFERENCE] EXT-CHAR
```

【参照】

```
INVOKE CONTROL-OBJECT-2 "GET-EXTEND-CHARACTER"
      USING [BY REFERENCE] FIELD-NAME
           [BY REFERENCE] EXT-CHAR
```

### 引数 / 復帰値

**FIELD-NAME** 【引数の属性: PIC X ANY LENGTH】:

設定/参照する対象となる帳票項目の名前を渡します。引数には任意の長さの英数字項目を記述します。

**EXT-CHAR** 【引数の属性: PIC X ANY LENGTH】:



設定時には、指定した帳票項目に対して文字引伸しで使う文字を渡します。引数には任意の長さの英数字項目を記述します。ただし、実際に有効になるのは先頭の 1 文字です。参照時には、指定した帳票項目に対する文字引伸しで使われる文字が返ります。引数には任意の長さの英数字項目を記述します。

### 例外エラー

設定で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)

[FJPRT-SCE-ILLEGAL-STATE](#)

[FJPRT-SCE-ARRAY-INDEX](#)

参照で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)

[FJPRT-SCE-ILLEGAL-STATE](#)

[FJPRT-SCE-ARRAY-INDEX](#)

### 注意事項

文字引伸しは、数字項目の帳票項目でだけ有効な属性です。

[参照] “ [例外エラーコード: FJPRT-SCE-ILLEGAL-STATEの発生](#) ”

**8.3.10 SET-FONT-STRIKE-OUTメソッド(抹消線プロパティ設定メソッド)**

**8.3.11 GET-FONT-STRIKE-OUTメソッド(抹消線プロパティ参照メソッド)**

### 説明

帳票項目の文字に掛ける抹消線を設定/参照します。

デフォルト値として“ 抹消線なし ”を持ちます。

記号定数	抹消線
FJPRT-SC-FTSTRK-OFF	抹消線なし
FJPRT-SC-FTSTRK-LINE1	一重抹消線
FJPRT-SC-FTSTRK-LINE2	二重抹消線

### 記述形式

#### 【設定】

```
INVOKE CONTROL-OBJECT-2 "SET-FONT-STRIKE-OUT"
      USING [BY REFERENCE] FIELD-NAME
          FONT-SO
```

#### 【参照】

```
INVOKE CONTROL-OBJECT-2 "GET-FONT-STRIKE-OUT"
      USING [BY REFERENCE] FIELD-NAME
      RETURNING FONT-SO
```

### 引数 / 復帰値

*FIELD-NAME* 【引数の属性: PIC X ANY LENGTH】:

設定/参照する対象となる帳票項目の名前を渡します。引数には任意の長さの英数字項目を記述します。

*FONT-SO* 【引数/復帰値の属性: PIC S9(4) [USAGE IS] COMP-5】:

設定時には、指定した帳票項目に掛ける抹消線の種類を渡します。引数の属性と同じ属性

を持つデータ項目か定数を記述します。

参照時には、指定した帳票項目に掛けられる抹消線の種類が返ります。引数の属性と同じ属性を持つデータ項目を記述します。

### 例外エラー

設定で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-SET-VALUE](#)

[FJPRT-SCE-FIELD-NAME](#)

[FJPRT-SCE-ILLEGAL-STATE](#)

[FJPRT-SCE-ARRAY-INDEX](#)

参照で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)

[FJPRT-SCE-ILLEGAL-STATE](#)

[FJPRT-SCE-ARRAY-INDEX](#)

### 注意事項

[参照] “ [例外エラーコード: FJPRT-SCE-ILLEGAL-STATEの発生](#) ”

8.3.12 SET-FONT-ITALICメソッド(イタリックプロパティ設定メソッド)

8.3.13 GET-FONT-ITALICメソッド(イタリックプロパティ参照メソッド)

### 説明

帳票項目の文字をイタリックで出力するか否かを設定/参照します。

帳票定義体に定義された項目の状態が、デフォルト値となります。

値	イタリック
B"1" (真)	あり
B"0" (偽)	なし

### 記述形式

#### 【設定】

```
INVOKE CONTROL-OBJECT-2 "SET-FONT-ITALIC"
      USING [BY REFERENCE] FIELD-NAME
           BOOL-VAL
```

#### 【参照】

```
INVOKE CONTROL-OBJECT-2 "GET-FONT-ITALIC"
      USING [BY REFERENCE] FIELD-NAME
      RETURNING BOOL-VAL
```

### 引数 / 復帰値

***FIELD-NAME*** 【引数の属性: PIC X ANY LENGTH】:

設定/参照する対象となる帳票項目の名前を渡します。引数には任意の長さの英数字項目を記述します。

***BOOL-VAL*** 【引数/復帰値の属性: PIC 1(1) [USAGE IS] DISPLAY】:

設定時には、指定した帳票項目をイタリックにするか否かを渡します。引数の属性と同じ属性を持つデータ項目か定数を記述します。

参照時には、指定した帳票項目がイタリックであるか否かが返ります。引数の属性と同じ

属性を持つデータ項目を記述します。

### 例外エラー

設定で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)  
[FJPRT-SCE-ILLEGAL-STATE](#)  
[FJPRT-SCE-ARRAY-INDEX](#)

参照で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)  
[FJPRT-SCE-ILLEGAL-STATE](#)  
[FJPRT-SCE-ARRAY-INDEX](#)

### 注意事項

[参照] “ [例外エラーコード：FJPRT-SCE-ILLEGAL-STATEの発生](#) ”

8.3.14 SET-FONT-BOLDメソッド(ボールドプロパティ設定メソッド)

8.3.15 GET-FONT-BOLDメソッド(ボールドプロパティ参照メソッド)

### 説明

帳票項目の文字をボールドで出力するか否かを設定/参照します。

帳票定義体に定義された項目の状態が、デフォルト値となります。

値	ボールド
B"1" (真)	あり
B"0" (偽)	なし

### 記述形式

#### 【設定】

INVOKE [CONTROL-OBJECT-2](#) "SET-FONT-BOLD"  
 USING [BY REFERENCE] *FIELD-NAME*  
*BOOL-VAL*

#### 【参照】

INVOKE [CONTROL-OBJECT-2](#) "GET-FONT-BOLD"  
 USING [BY REFERENCE] *FIELD-NAME*  
 RETURNING *BOOL-VAL*

### 引数 / 復帰値

***FIELD-NAME*** 【引数の属性: PIC X ANY LENGTH】:

設定/参照する対象となる帳票項目の名前を渡します。引数には任意の長さの英数字項目を記述します。

***BOOL-VAL*** 【引数/復帰値の属性: PIC 1(1) [USAGE IS] DISPLAY】:

設定時には、指定した帳票項目をボールドにするか否かを渡します。引数の属性と同じ属性を持つデータ項目が定数を記述します。

参照時には、指定した帳票項目がボールドであるか否かが返ります。引数の属性と同じ属性を持つデータ項目を記述します。

## 例外エラー

設定で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)

[FJPRT-SCE-ILLEGAL-STATE](#)

[FJPRT-SCE-ARRAY-INDEX](#)

参照で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)

[FJPRT-SCE-ILLEGAL-STATE](#)

[FJPRT-SCE-ARRAY-INDEX](#)

## 注意事項

[参照] “ [例外エラーコード: FJPRT-SCE-ILLEGAL-STATEの発生](#) ”

8.3.16 SET-FONT-UNDERLINEメソッド(下線プロパティ設定メソッド)

8.3.17 GET-FONT-UNDERLINEメソッド(下線プロパティ参照メソッド)

## 説明

帳票項目の文字に下線を引くか否かを設定/参照します。

帳票定義体に定義された項目の状態が、デフォルト値となります。

値	下線
B"1" (真)	あり
B"0" (偽)	なし

## 記述形式

### 【設定】

```
INVOKE CONTROL-OBJECT-2 "SET-FONT-UNDERLINE"
      USING [BY REFERENCE] FIELD-NAME
           BOOL-VAL
```

### 【参照】

```
INVOKE CONTROL-OBJECT-2 "GET-FONT-UNDERLINE"
      USING [BY REFERENCE] FIELD-NAME
      RETURNING BOOL-VAL
```

## 引数 / 復帰値

***FIELD-NAME*** 【引数の属性: PIC X ANY LENGTH】:

設定/参照する対象となる帳票項目の名前を渡します。引数には任意の長さの英数字項目を記述します。

***BOOL-VAL*** 【引数/復帰値の属性: PIC 1(1) [USAGE IS] DISPLAY】:

設定時には、指定した帳票項目に下線を引くか否かを渡します。引数の属性と同じ属性を持つデータ項目が定数を記述します。

参照時には、指定した帳票項目に下線が引かれるか否かが返ります。引数の属性と同じ属性を持つデータ項目を記述します。

## 例外エラー

設定で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)

[FJPRT-SCE-ILLEGAL-STATE](#)[FJPRT-SCE-ARRAY-INDEX](#)

参照で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)[FJPRT-SCE-ILLEGAL-STATE](#)[FJPRT-SCE-ARRAY-INDEX](#)

### 注意事項

[参照] “[例外エラーコード: FJPRT-SCE-ILLEGAL-STATEの発生](#)”

**8.3.18 SET-FIELD-UNDERLINEメソッド(項目下線プロパティ設定メソッド)**

**8.3.19 GET-FIELD-UNDERLINEメソッド(項目下線プロパティ参照メソッド)**

### 説明

帳票項目に項目下線を引くか否かを設定/参照します。  
デフォルト値として“なし”を示す値を持ちます。

値	項目下線
B"1" (真)	あり
B"0" (偽)	なし

### 記述形式

#### 【設定】

```
INVOKE CONTROL-OBJECT-2 "SET-FIELD-UNDERLINE"
      USING [BY REFERENCE] FIELD-NAME
           BOOL-VAL
```

#### 【参照】

```
INVOKE CONTROL-OBJECT-2 "GET-FIELD-UNDERLINE"
      USING [BY REFERENCE] FIELD-NAME
      RETURNING BOOL-VAL
```

### 引数 / 復帰値

***FIELD-NAME*** 【引数の属性: PIC X ANY LENGTH】:

設定/参照する対象となる帳票項目の名前を渡します。引数には任意の長さの英数字項目を記述します。

***BOOL-VAL*** 【引数/復帰値の属性: PIC 1(1) [USAGE IS] DISPLAY】:

設定時には、指定した帳票項目に項目下線を引くか否かを渡します。引数の属性と同じ属性を持つデータ項目が定数を記述します。

参照時には、指定した帳票項目に項目下線が引かれるか否かが返ります。引数の属性と同じ属性を持つデータ項目を記述します。

### 例外エラー

設定で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)[FJPRT-SCE-ILLEGAL-STATE](#)[FJPRT-SCE-ARRAY-INDEX](#)

参照で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-FIELD-NAME](#)

[FJPRT-SCE-ILLEGAL-STATE](#)

[FJPRT-SCE-ARRAY-INDEX](#)

### 注意事項

[参照] “[例外エラーコード: FJPRT-SCE-ILLEGAL-STATEの発生](#)”

## 8.4 帳票定義体に関するプロパティ

ここでは、[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)が提供する以下のカテゴリのプロパティについての説明をします。

帳票定義体に関するプロパティ

[FORM-NAMEプロパティ\(帳票定義体名プロパティ\)](#)

[DE-NAMEプロパティ\(明細名プロパティ\)](#)

[CONTROL-PARTITION-NAMEプロパティ\(制御パーティション名プロパティ\)](#)

[OVERLAY-NAMEプロパティ\(オーバーレイ名プロパティ\)](#)

[FORM-DIRECTORYプロパティ\(帳票定義体格納ディレクトリプロパティ\)](#)

[OVERLAY-DIRECTORYプロパティ\(オーバーレイ格納ディレクトリプロパティ\)](#)

[MEDIA-DIRECTORYプロパティ\(メディア格納ディレクトリプロパティ\)](#)

[FORM-SUFFIXプロパティ\(帳票定義体拡張子プロパティ\)](#)

[OVERLAY-SUFFIXプロパティ\(オーバーレイ拡張子プロパティ\)](#)

### 8.4.1 FORM-NAMEプロパティ(帳票定義体名プロパティ)

#### 説明

[帳票の印刷](#)で使用する帳票定義体名を設定/参照します。

帳票の印刷中に当プロパティを使用して、[複数の帳票定義体を利用した帳票印刷](#)を行うことができます。

当プロパティの設定値に拡張子を付加しなかった場合、印刷に使われる帳票定義体は、[帳票定義体拡張子プロパティ\(FORM-SUFFIX\)](#)の参照値を付加したものとなります。

当プロパティの設定値の前方にパスを付加しなかった場合、印刷に使われる帳票定義体は、[帳票定義体格納ディレクトリプロパティ\(FORM-DIRECTORY\)](#)の参照値の位置にあるものとなります。

付加できるパスとして、フルパス、相対パスがあります。

デフォルト値として空白を持ちますが、帳票の印刷を行うためには必ず帳票定義体名を設定する必要があります。

#### プロパティの属性

PIC X(518)

#### 記述形式

##### 【設定】

MOVE *ALPHANUM-VAL* TO **FORM-NAME** OF [CONTROL-OBJECT-3](#)

##### 【補足】

**ALPHANUM-VAL :**

帳票定義体名を渡します。英数字の属性を持つ語を記述してください。

動作モードがシフトJISのときに最大で 259文字まで設定することができます。ただし、実際に有効となる文字列の長さはシステムの制限に依存します。

##### 【参照】

**FORM-NAME** OF [CONTROL-OBJECT-3](#)

##### 【補足】

当プロパティのデフォルト値または設定された値を英数字の属性で返します。返却される文字列の最大文字数は、動作モードがシフトJISのときに259文字です。

#### 例外エラー

設定/参照とも例外エラーの発生はありません。

#### 注意事項

帳票開設イベントより後に発生するイベントで設定し直した場合、次のページの印刷から設定が

反映されます。

## 使用例

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER-CLASS INHERITS  FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
    ENVIRONMENT-NAME  IS  ENV-NAME
    ENVIRONMENT-VALUE IS  ENV-VALUE
.
REPOSITORY.
    CLASS  FJPRT-EVENT-HANDLER
    CLASS  FJPRT-CONTROL
:
IDENTIFICATION DIVISION.
METHOD-ID.  OPEN-REPORT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01  W-DIR PIC X(30).
LINKAGE SECTION.
    01  L-CTRLOBJ OBJECT REFERENCE FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    DISPLAY "FORMDIR" UPON ENV-NAME *>.....[1]
    ACCEPT  W-DIR      FROM ENV-VALUE *>.....[2]
    MOVE  W-DIR TO FORM-DIRECTORY OF L-CTRLOBJ *>....[3]
    MOVE  W-DIR TO OVERLAY-DIRECTORY OF L-CTRLOBJ *>....[4]
    MOVE  W-DIR TO MEDIA-DIRECTORY OF L-CTRLOBJ *>....[5]
    MOVE "FORM1" TO FORM-NAME OF L-CTRLOBJ *>....[6]
    MOVE "OVERLAY1" TO OVERLAY-NAME OF L-CTRLOBJ *>....[7]
    MOVE "pxd" TO FORM-SUFFIX OF L-CTRLOBJ *>....[8]
    MOVE "ovd" TO OVERLAY-SUFFIX OF L-CTRLOBJ *>...[9]
    EXIT METHOD.
END METHOD  OPEN-REPORT.
:
END CLASS  USER-HANDLER-CLASS.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

### [例の説明]

[1]

アクセスする環境変数の名前を "FORMDIR" とします。

[2]

環境変数 "FORMDIR" の設定値を W-DIR 変数に取り出します。

[3]

[2]で取り出した値を[帳票定義体格納ディレクトリプロパティ \(FORM-DIRECTORY\)](#)に設定します。

[4]

[2]で取り出した値を[オーバーレイ格納ディレクトリプロパティ \(OVERLAY-DIRECTORY\)](#)に設定します。

[5]

[2]で取り出した値を[メディア格納ディレクトリプロパティ \(MEDIA-DIRECTORY\)](#)に設定し



ます。

- [6] 帳票の印刷で使用する帳票定義体を指定します。
- [7] 帳票の印刷で使用するオーバーレイ定義体を明に指定します。
- [8] 帳票定義体の拡張子を明に指定します。
- [9] オーバレイ定義体の拡張子を明に指定します。

## 8.4.2 DE-NAMEプロパティ(明細名プロパティ)

### 説明

集計表形式の帳票定義体を使った帳票の印刷で使用する明細名(明細パーティション名)を設定/参照します。

このプロパティを操作することで、帳票定義体に複数定義してある明細を切り替えて帳票の印刷を行うことができます。

デフォルト値として空白を持ちます。このプロパティへの設定を省略したときは、帳票の印刷時に帳票定義体上の主明細名が指定されたものとみなします。

### プロパティの属性

PIC X(60)

### 記述形式

#### 【設定】

MOVE *ALPHANUM-VAL* TO **DE-NAME** OF [CONTROL-OBJECT-3](#)

#### 【補足】

**ALPHANUM-VAL** :

明細名を渡します。英数字の属性を持つ語を記述します。

動作モードがシフトJISのときに最大で 30文字まで設定することができます。

#### 【参照】

**DE-NAME** OF [CONTROL-OBJECT-3](#)

#### 【補足】

当プロパティのデフォルト値または設定された値を英数字の属性で返します。返却される文字列の最大文字数は、動作モードがシフトJISのときに30文字です。

### 例外エラー

設定/参照とも例外エラーの発生はありません。

### 注意事項

当プロパティは集計表形式の帳票定義体でだけ有効です。

[明細出力イベント](#)で明細名を変更する場合、変更した明細がページ内に収まる場合にだけ、明細名を変更できます。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER-CLASS INHERITS  FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS  FJPRT-EVENT-HANDLER
    CLASS  FJPRT-CONTROL
:
IDENTIFICATION DIVISION.
METHOD-ID.  PRINT-DE  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.
    INVOKE  L-CTRLOBJ  "GET-RECORD-CONTENT"  USING  PAGE1  *>...[1]
    IF  VALUE1 OF  PAGE1 < 10  THEN  *>.....[2]
        MOVE  "DE1"  TO  DE-NAME OF  L-CTRLOBJ
    ELSE
        MOVE  "DE"  TO  DE-NAME OF  L-CTRLOBJ
    END-IF
    EVALUATE  DE-NAME OF  L-CTRLOBJ  *>.....[3]
    WHEN      "DE1"
        MOVE  NAME1 OF  PAGE1  TO  NAME-DE1 OF  PAGE1
    END-EVALUATE
    INVOKE  L-CTRLOBJ  "SET-RECORD-CONTENT"  USING  PAGE1  *>...[4]
    EXIT METHOD.
END METHOD  PRINT-DE.
:
END CLASS  USER-HANDLER-CLASS.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

- [1]  
レコード内容を取り出します。
- [2]  
ある帳票項目の値を調べて、ある条件で明細名を切り替えます。
- [3]  
明細名が切り替えられているとき、切り替えた明細パーティションが持つ項目に値を転記し直します。
- [4]  
レコード内容へ変更を反映します。

### 8.4.3 CONTROL-PARTITION-NAME プロパティ (制御パーティション名プロパティ)

#### 説明

現在、イベントで処理対象となっているパーティション名を参照します。  
当プロパティは集計表形式の帳票定義体を使った帳票の印刷で発生する[制御頭書きイベント](#)と[制御脚書きイベント](#)の処理中でだけ有効です。

#### プロパティの属性

PIC X(60)

#### 記述形式

##### 【参照】

CONTROL-PARTITION-NAME OF [CONTROL-OBJECT-2](#)

##### 【補足】

現在、イベントで処理の対象となっている制御頭書きパーティション名または制御脚書きパーティション名を英数字の属性で返します。制御頭書きイベントまたは制御脚書きイベント以外で呼び出した場合は、空白が返されます。返却される文字列の最大文字数は、動作モードがシフトJISのときに30文字です。

#### [例外エラー](#)

例外エラーの発生はありません。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER-CLASS  INHERITS  FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS  FJPRT-EVENT-HANDLER
    CLASS  FJPRT-CONTROL
:
IDENTIFICATION DIVISION.
METHOD-ID.  CB-FOOT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01  W-DIR  PIC  X(30).
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.
    INVOKE  L-CTRLOBJ  "GET-RECORD-CONTENT"  USING  PAGE1  *>...[1]
    EVALUATE  CONTROL-PARTITION-NAME OF L-CTRLOBJ  *>.....[2]
    WHEN      "CF1"
        MOVE  "VALUE-FOR-CF1-FIELD" TO CF1-FIELD OF PAGE1
    WHEN      "CF2"
        MOVE  "VALUE-FOR-CF2-FIELD" TO CF2-FIELD OF PAGE1
    END-EVALUATE
    INVOKE  L-CTRLOBJ  "SET-RECORD-CONTENT"  USING  PAGE1  *>...[3]
    EXIT METHOD.
END METHOD  CB-FOOT.
:
END CLASS  USER-HANDLER-CLASS.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

- [1]  
レコード内容を取り出します。
- [2]  
イベントの処理対象のパーティションが持つ項目に値を設定します。
- [3]  
レコード内容へ変更を反映します。

#### 8.4.4 OVERLAY-NAMEプロパティ(オーバレイ名プロパティ)

## 説明

帳票定義体上のオーバレイ指定を無視して、印刷時のオーバレイ定義体を変更したい場合にオーバレイ名を設定します。

当プロパティの設定値に拡張子を付加しなかった場合、印刷に使われるオーバレイ定義体は、[オーバレイ拡張子プロパティ\(OVERLAY-SUFFIX\)](#)の参照値を付加したものとなります。オーバレイ定義体として、OVD、BMP、WMF、EMFの各形式が指定可能です。

当プロパティの設定値の前方にパスを付加しなかった場合、印刷に使われるオーバレイ定義体は、[オーバレイ格納ディレクトリプロパティ\(OVERLAY-DIRECTORY\)](#)の参照値の位置にあるものとなります。

ます。

付加できるパスとして、フルパス、相対パスがあります。

オーバーレイ名は、帳票の印刷の途中で変更が可能です。

デフォルト値として空白を持ちます。このプロパティへの設定を省略したときは、帳票定義体上のオーバーレイ指定が有効になります。

### プロパティの属性

PIC X(518)

### 記述形式

#### 【設定】

MOVE *ALPHANUM-VAL* TO **OVERLAY-NAME** OF [CONTROL-OBJECT-3](#)

#### 【補足】

**ALPHANUM-VAL** :

オーバーレイ名を渡します。英数字の属性を持つ語を記述します。

動作モードがシフトJISのときに最大で 259文字まで設定することができます。ただし、実際に有効となる文字の長さはシステムの制限に依存します。

#### 【参照】

**OVERLAY-NAME** OF [CONTROL-OBJECT-3](#)

#### 【補足】

当プロパティのデフォルト値または設定された値を英数字の属性で返します。返却される文字列の最大文字数は、動作モードがシフトJISのときに259文字です。

### 例外エラー

設定/参照とも例外エラーの発生はありません。

### 使用例

“ [帳票定義体名プロパティ \(FORM-NAME\)の使用例](#) ” を参照してください。

## 8.4.5 FORM-DIRECTORYプロパティ (帳票定義体格納ディレクトリプロパティ)

### 説明

帳票定義体が格納されているディレクトリを設定/参照します。[帳票定義体名プロパティ \(FORM-NAME\)](#) に、フルパスで帳票定義体名が指定されていない場合に、このプロパティの参照値が先頭に付加されたフルパス名を使用して、帳票定義体を読み込みます。

指定できるパスとして、フルパス、相対パスがあります。

デフォルト値として空白を持ちます。このプロパティへの設定を省略したときは、カレントディレクトリが指定されたものとみなします。

### プロパティの属性

PIC X(518)

### 記述形式

#### 【設定】

MOVE *ALPHANUM-VAL* TO **FORM-DIRECTORY** OF [CONTROL-OBJECT-3](#)

#### 【補足】

**ALPHANUM-VAL** :

帳票定義体格納ディレクトリを渡します。英数字の属性を持つ語を記述します。

動作モードがシフトJISのときに最大で 259文字まで設定することができます。ただし、実際に有効となる文字の長さはシステムの制限に依存します。

#### 【参照】

**FORM-DIRECTORY** OF [CONTROL-OBJECT-3](#)

【補足】

当プロパティのデフォルト値または設定された値を英数字の属性で返します。返却される文字列の最大文字数は、動作モードがシフトJISのときに259文字です。

例外エラー

設定/参照とも例外エラーの発生はありません。

**注意事項**

このプロパティは、帳票開設イベントより後に発生するイベントで変更した内容は無効になります。

**使用例**

“ [帳票定義体名プロパティ \(FORM-NAME\)の使用例](#) ” を参照してください。

## 8.4.6 OVERLAY-DIRECTORYプロパティ(オーバレイ格納ディレクトリプロパティ)

**説明**

オーバレイ定義体が格納されているディレクトリを設定/参照します。[オーバレイ名プロパティ \(OVERLAY-NAME\)](#)に、フルパスでオーバレイが指定されていない場合に、このプロパティの参照値が先頭に付加されたフルパス名を使用して、オーバレイ定義体を読み込みます。  
指定できるパスとして、フルパス、相対パスがあります。  
デフォルト値として空白を持ちます。このプロパティへの設定を省略したときは、カレントディレクトリが指定されたものとみなします。

**プロパティの属性**

PIC X(518)

**記述形式**

【設定】

MOVE *ALPHANUM-VAL* TO OVERLAY-DIRECTORY OF [CONTROL-OBJECT-3](#)

【補足】

*ALPHANUM-VAL* :

オーバレイ格納ディレクトリを渡します。英数字の属性を持つ語を記述します。  
動作モードがシフトJISのときに最大で 259文字まで設定することができます。ただし、実際に有効となる文字の長さはシステムの制限に依存します。

【参照】

OVERLAY-DIRECTORY OF [CONTROL-OBJECT-3](#)

【補足】

当プロパティのデフォルト値または設定された値を英数字の属性で返します。返却される文字列の最大文字数は、動作モードがシフトJISのときに259文字です。

例外エラー

設定/参照とも例外エラーの発生はありません。

**注意事項**

このプロパティは、帳票開設イベントより後に変更した内容は無効になります。

**使用例**

“ [帳票定義体名プロパティ \(FORM-NAME\)の使用例](#) ” を参照してください。

### 8.4.7 MEDIA-DIRECTORYプロパティ(メディア格納ディレクトリプロパティ)

#### 説明

帳票定義体で使用しているメディアおよび、[コンパクト印刷用オーバーレイ名プロパティ \(OVERLAY-NAME-TO-COMPACT-PRINT\)](#) で使用するオーバーレイが格納されているディレクトリを設定/参照します。

指定できるパスとして、フルパス、相対パスがあります。

デフォルト値として空白を持ちます。このプロパティへの設定を省略したときは、カレントディレクトリが指定されたものとみなします。

#### プロパティの属性

PIC X(518)

#### 記述形式

##### 【設定】

MOVE ALPHANUM-VAL TO MEDIA-DIRECTORY OF [CONTROL-OBJECT-3](#)

##### 【補足】

##### ALPHANUM-VAL :

メディア格納ディレクトリを渡します。英数字の属性を持つ語を記述します。

動作モードがシフトJISのときに最大で 259文字まで設定することができます。ただし、実際に有効となる文字の長さはシステムの制限に依存します。

##### 【参照】

MEDIA-DIRECTORY OF [CONTROL-OBJECT-3](#)

##### 【補足】

当プロパティのデフォルト値または設定された値を英数字の属性で返します。返却される文字列の最大文字数は、動作モードがシフトJISのときに259文字です。

#### 例外エラー

設定/参照とも例外エラーの発生はありません。

#### 注意事項

このプロパティは、帳票開設イベントより後に変更した内容は無効になります。

#### 使用例

“ [帳票定義体名プロパティ \(FORM-NAME\)の使用例](#) ” を参照してください。

### 8.4.8 FORM-SUFFIXプロパティ(帳票定義体拡張子プロパティ)

#### 説明

[帳票の印刷](#) で使用する帳票定義体の拡張子を設定/参照します。[帳票定義体名プロパティ \(FORM-NAME\)](#) に拡張子付きの名前が指定されてない場合、このプロパティで指定した拡張子が付加されて帳票定義体を読み込みます。

デフォルト値として "PXD" を持ちます。

#### プロパティの属性

PIC X(518)

#### 記述形式

##### 【設定】

MOVE ALPHANUM-VAL TO FORM-SUFFIX OF [CONTROL-OBJECT-3](#)

##### 【補足】

##### ALPHANUM-VAL :

帳票定義体の拡張子を渡します。英数字の属性を持つ語を記述します。  
動作モードがシフトJISのときに最大で 259文字まで設定することができます。ただし、実際に有効となる文字の長さはシステムの制限に依存します。

【参照】

FORM-SUFFIX OF [CONTROL-OBJECT-3](#)

【補足】

当プロパティのデフォルト値または設定された値を英数字の属性で返します。返却される文字列の最大文字数は、動作モードがシフトJISのときに259文字です。ただし、実際に有効となる文字の長さはシステムの制限に依存します。

### 例外エラー

設定/参照とも例外エラーの発生はありません。

### 使用例

“ [帳票定義体名プロパティ \(FORM-NAME\)の使用例](#) ” を参照してください。

## 8.4.9 OVERLAY-SUFFIXプロパティ(オーバレイ拡張子プロパティ)

### 説明

帳票の印刷で使用するオーバレイ定義体の拡張子を設定/参照します。[オーバレイ名プロパティ \(OVERLAY-NAME\)](#)に拡張子付きの名前が指定されていない場合、このプロパティで指定した拡張子が末尾に付加されてオーバレイを読み込みます。

デフォルト値として "OVD" を持ちます。

SystemWalker/ListWORKS連携時は、"OVD" だけが有効となります。

### プロパティの属性

PIC X(518)

### 記述形式

【設定】

MOVE ALPHANUM-VAL TO OVERLAY-SUFFIX OF [CONTROL-OBJECT-3](#)

【補足】

ALPHANUM-VAL :

オーバレイ定義体の拡張子を渡します。英数字の属性を持つ語を記述します。  
動作モードがシフトJISのときに最大で 259文字まで設定することができます。ただし、実際に有効となる文字の長さはシステムの制限に依存します。

【参照】

OVERLAY-SUFFIX OF [CONTROL-OBJECT-3](#)

【補足】

当プロパティのデフォルト値または設定された値を英数字の属性で返します。返却される文字列の最大文字数は、動作モードがシフトJISのときに259文字です。

### 例外エラー

設定/参照とも例外エラーの発生はありません。

### 使用例

“ [帳票定義体名プロパティ \(FORM-NAME\)の使用例](#) ” を参照してください。



## 8.5 印刷エラーに関するプロパティ

ここでは、[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)が提供する以下のカテゴリのプロパティについての説明をします。

印刷エラーに関するプロパティ

[PRINT-JOB-ERROR-CODEプロパティ \(印刷ジョブエラーコードプロパティ\)](#)

### 8.5.1 PRINT-JOB-ERROR-CODEプロパティ (印刷ジョブエラーコードプロパティ)

#### 説明

[帳票印刷メソッド\(PRINT-CONTROL\)](#)から制御が戻ってきた時点で参照するとき、帳票印刷の終了時のエラーコードを通知します。

また、帳票の印刷中に[印刷エラー](#)が発生した場合に帳票印刷クラスライブラリは、このプロパティでエラーコードを参照できるようにしてから、エラーメッセージイベントを発生させます。

このプロパティは参照だけ可能で、設定はできません。

デフォルト値として 0(エラーなし)を持ちます。

#### プロパティの属性

PIC S9(9) [USAGE IS] COMP-5

#### 記述形式

【参照】

PRINT-JOB-ERROR-CODE OF [CONTROL-OBJECT-3](#)

【補足】

印刷時に発生したエラーコードを参照します。このコードが示す意味については、“[印刷ジョブエラーコード](#)”を参照してください。

#### 例外エラー

例外エラーの発生はありません。

#### 使用例

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[エラーメッセージメソッド\(ERROR-MESSAGE\)の使用例](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[エラーメッセージメソッド\(ERROR-MESSAGE\)の使用例](#)

## 8.6 印刷行/ページ/タックに関するプロパティ

ここでは、[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)が提供する以下のカテゴリのプロパティについての説明をします。

印刷行/ページ/タックに関するプロパティ

[PRINTING-PAGE-NUMBERプロパティ\(印刷中ページ番号プロパティ\)](#)

[PRINTING-LINE-NUMBERプロパティ\(印刷中行番号プロパティ\)](#)

[INIT-PAGE-NUMBERプロパティ\(最初のページ番号プロパティ\)](#)

[PAGE-RANGEプロパティ\(ページ範囲プロパティ\)](#)

[START-PAGE-NUMBERプロパティ\(開始ページ番号プロパティ\)](#)

[END-PAGE-NUMBERプロパティ\(終了ページ番号プロパティ\)](#)

[START-TACK-NUMBERプロパティ\(タック開始番号プロパティ\)](#)

### 8.6.1 PRINTING-PAGE-NUMBERプロパティ(印刷中ページ番号プロパティ)

#### 説明

印刷中のページ番号を参照します。

このプロパティは参照することだけが可能で、設定することはできません。

デフォルト値として 1 を持ちます。

#### プロパティの属性

PIC S9(9) [USAGE IS] COMP-5

#### 記述形式

【参照】

PRINTING-PAGE-NUMBER OF [CONTROL-OBJECT-2](#)

【補足】

参照した時点での印刷中のページ番号を返します。通知される値の範囲は 1 ~ 32767 です。

#### 例外エラー

例外エラーの発生はありません。

#### 使用例

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[ページ頭書きメソッド\(PAGE-HEAD\)の使用例](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[ページ頭書きメソッド\(PAGE-HEAD\)の使用例](#)

### 8.6.2 PRINTING-LINE-NUMBERプロパティ(印刷中行番号プロパティ)

#### 説明

印刷中の行番号を参照します。

このプロパティは参照することだけが可能で、設定することはできません。

デフォルト値として 1 を持ちます。

## プロパティの属性

PIC S9(4) [USAGE IS] COMP-5

## 記述形式

### 【参照】

PRINTING-LINE-NUMBER OF [CONTROL-OBJECT-2](#)

### 【補足】

参照した時点での印刷中の行番号を返します。通知される値の範囲は 1～512 です。

## 例外エラー

例外エラーの発生はありません。

## 注意事項

参照した行番号は、現時点で印刷が完了した行の行番号です。例えば、[明細出力イベント](#)で参照した場合、対象となる明細/ボディ部を印刷する行番号ではなく、その一つ手前の行番号が通知されます。

例外として、先頭行の印刷時に参照した場合にはデフォルト値として 1が通知されますので注意してください。

# 8.6.3 INIT-PAGE-NUMBERプロパティ(最初のページ番号プロパティ)

## 説明

帳票の印刷時の最初のページ番号を指定します。

集計表形式の帳票定義体の場合、報告書頭書きページに対しては、ここで指定したページ番号から 1を引いたものが開始ページとして扱われます。

デフォルト値として 1を持ちます。

## プロパティの属性

PIC S9(4) [USAGE IS] COMP-5

## 記述形式

### 【設定】

MOVE *SHORT-VAL* TO INIT-PAGE-NUMBER OF [CONTROL-OBJECT-3](#)

### 【補足】

### *SHORT-VAL:*

最初のページ番号を渡します。数字の属性を持つ語を記述します。

設定できる値の範囲は 1～999 です。

### 【参照】

INIT-PAGE-NUMBER OF [CONTROL-OBJECT-3](#)

### 【補足】

現在設定されている最初のページ番号を返します。通知される値の範囲は 1～999 です。

## 例外エラー

設定で例外エラー(コード: [FJPRT-SCE-SET-VALUE](#))が発生することがあります。

参照では例外エラーの発生はありません。

## 注意事項

このプロパティは、帳票開設イベントより後に変更した内容は無効になります。

## 8.6.4 PAGE-RANGE プロパティ (ページ範囲プロパティ)

### 説明

印刷するページ範囲を示す値を設定/参照します。

“範囲指定”を指定した場合は、[開始ページ番号プロパティ \(START-PAGE-NUMBER\)](#)、[終了ページ番号プロパティ \(END-PAGE-NUMBER\)](#) で指定した範囲のページを印刷します。

デフォルト値として“全ページ指定”を持ちます。

記号定数	意味
FJPRT-SC-PGRANG-ALL	全ページ指定
FJPRT-SC-PGRANG-RANGE	範囲指定

### プロパティの属性

PIC S9(4) [USAGE IS] COMP-5

### 記述形式

#### 【設定】

MOVE *SHORT-VAL* TO **PAGE-RANGE** OF [CONTROL-OBJECT-3](#)

#### 【補足】

**SHORT-VAL:**

ページ範囲を示す値を渡します。

#### 【参照】

**PAGE-RANGE** OF [CONTROL-OBJECT-3](#)

#### 【補足】

現在設定されているページ範囲を示す値を返します。

### 例外エラー

設定で例外エラー (コード: [FJPRT-SCE-SET-VALUE](#)) が発生することがあります。

参照では例外エラーの発生はありません。

### 注意事項

このプロパティは、帳票開設イベントより後に変更した内容は無効になります。

印刷プレビューをする場合は、当プロパティへの設定が無効となります。印刷プレビューについては [SET-PRINT-PREVIEWメソッド\(印刷プレビュープロパティ設定\)](#) を参照してください。

## 8.6.5 START-PAGE-NUMBER プロパティ (開始ページ番号プロパティ)

### 説明

印刷を開始するページのページ番号を指定します。

“最初のページ番号”プロパティを指定した場合は、最初のページ番号 + 開始ページ番号の値を指定します。

“ページ範囲”プロパティが“範囲指定”の場合に有効になります。

デフォルト値として 1 を持ちます。

### プロパティの属性

PIC S9(4) [USAGE IS] COMP-5

### 記述形式

#### 【設定】

MOVE *SHORT-VAL* TO **START-PAGE-NUMBER** OF [CONTROL-OBJECT-3](#)

## 【補足】

**SHORT-VAL:**

設定する開始ページ番号。  
値は1～999まで指定可能です。

## 【参照】

START-PAGE-NUMBER OF [CONTROL-OBJECT-3](#)

## 【補足】

参照時には、現在設定されている開始ページ番号を返します。値の範囲は1～999です。

**例外エラー**

設定で例外エラー(コード: [FJPRT-SCE-SET-VALUE](#))が発生することがあります。  
参照では例外エラーの発生はありません。

**注意事項**

このプロパティは、帳票開設イベントより後に変更した内容は無効になります。  
印刷プレビューをする場合は、当プロパティへの設定が無効となります。印刷プレビューについては、[SET-PRINT-PREVIEWメソッド\(印刷プレビュープロパティ設定\)](#)を参照してください。

## 8.6.6 END-PAGE-NUMBERプロパティ(終了ページ番号プロパティ)

**説明**

終了ページ番号を指定します。  
“最初のページ番号”プロパティを指定した場合は、最初のページ番号 + 終了ページ番号の値を指定します。  
“ページ範囲”プロパティが“範囲指定”の場合に有効になります。  
デフォルト値として 1を持ちます。

**プロパティの属性**

PIC S9(4) [USAGE IS] COMP-5

**記述形式**

## 【設定】

MOVE SHORT-VAL TO END-PAGE-NUMBER OF [CONTROL-OBJECT-3](#)

## 【補足】

**SHORT-VAL:**

設定する終了ページ番号。  
値は 1～999 まで指定可能です。

## 【参照】

END-PAGE-NUMBER OF [CONTROL-OBJECT-3](#)

## 【補足】

参照時には、現在設定されている終了ページ番号を返します。値の範囲は 1～999 です。

**例外エラー**

設定で例外エラー(コード: [FJPRT-SCE-SET-VALUE](#))が発生することがあります。  
参照では例外エラーの発生はありません。

**注意事項**

このプロパティは、帳票開設イベントより後に変更した内容は無効になります。  
印刷プレビューをする場合は、当プロパティへの設定が無効となります。印刷プレビューについては、[SET-PRINT-PREVIEWメソッド\(印刷プレビュープロパティ設定\)](#)を参照してください。

### 8.6.7 START-TACK-NUMBER プロパティ (タック開始番号プロパティ)

#### 説明

タック印刷を行う場合、先頭のタックから印刷するのではなく、途中のタックから印刷を行いたい場合に指定します。

デフォルト値として 1 を持ちます。

#### プロパティの属性

PIC S9(4) [USAGE IS] COMP-5

#### 記述形式

##### 【設定】

MOVE *SHORT-VAL* TO START-TACK-NUMBER OF [CONTROL-OBJECT-3](#)

##### 【補足】

##### **SHORT-VAL:**

タック開始番号。

値は 1～99 まで指定可能です。

##### 【参照】

START-TACK-NUMBER OF [CONTROL-OBJECT-3](#)

##### 【補足】

現在設定されているタック開始番号を返します。値の範囲は1～99です。

#### [例外エラー](#)

設定で例外エラー (コード: [FJPRT-SCE-SET-VALUE](#)) が発生することがあります。

参照では例外エラーの発生はありません。

#### 注意事項

当プロパティはタックシール形式の[帳票定義体](#)でだけ有効です。

## 8.7 入力データに関するプロパティ

ここでは、[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)が提供する以下のカテゴリのプロパティについての説明をします。

入力データに関するプロパティ

[AUTOMATIC-RECORD-CHANGEプロパティ\(レコード自動切替えプロパティ\)](#)

[RECORD-EOFプロパティ\(レコードEOFプロパティ\)](#)

[RECORD-COUNTプロパティ\(レコード回数プロパティ\)](#)

### 8.7.1 AUTOMATIC-RECORD-CHANGEプロパティ(レコード自動切替えプロパティ)

#### 説明

[複数の帳票定義体を切り替えて使用](#)する帳票の印刷で、帳票印刷クラスライブラリが管理するレコードが持つ各項目の値を、帳票定義体を切り替えた後のレコード内の項目の並びに従って、並び替えるか否かを設定/参照します。この並び替えの動作のことを“レコードの自動切替え”と言います。

帳票定義体を切り替えるとき、切り替える前のレコードが持つ項目の値を切り替えた後のイベントで利用したい場合は、“組立て直す”を指定します。

“組立て直す”が指定されると、切替え後の帳票定義体が持つレコードの項目の並びに従ってデータの並び替えをします。

“組立て直さない”が指定されると、切替え後の帳票定義体が持つレコードの項目の並びに従ったデータ並び替えを行いません。

いずれの場合も、切替え後のレコードの項目の並びは切り替えられます。

切替え前後のレコードが持つ項目の名前や属性が同一の場合には真を設定し、出力対象のパーティションに対応したイベント内で、出力データの設定を行う場合(明細出力イベントで明細パーティションの印刷文字を設定する場合)には偽を設定します。

デフォルト値として“組立て直す”を持ちます。

値	意味
B"1" (真)	組み立て直す
B"0" (偽)	組み立て直さない

以下に“組立て直す”である場合の、レコード切替えの例を示します。

#### [レコードの切替えの例]

(1) 帳票定義体: FORM-A 処理時のレコード内の項目の並び

オフセット	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
データ	V	W	X	Y	Z	あ		1	2	3
項目名	ALN					NAT		NUM		

(2) 帳票定義体: FORM-B 処理時のレコード内の項目の並び

オフセット	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
データ	1	2	3	あ		V	W	X	Y	Z
項目名	NUM			NAT		ALN				

帳票定義体: FORM-Aには、ALN、NAT、NUMという順でレコード内に項目が定義してあり、帳票印刷クラスライブラリのレコードは(1)のようになります。

この状態で、NUM、NAT、ALNというレコード上の順序で項目が定義してある帳票定義体:

FORM-Bに切り替えると、帳票印刷クラスライブラリが管理するレコードは、(2)のようにレコードの自動切替えがなされます。

### プロパティの属性

PIC 1(1) [USAGE IS] DISPLAY

### 記述形式

#### 【設定】

MOVE *BOOL-VAL* TO AUTOMATIC-RECORD-CHANGE OF [CONTROL-OBJECT-3](#)

#### 【補足】

**BOOL-VAL:**

レコード自動切替えをするか否かを渡します。

#### 【参照】

AUTOMATIC-RECORD-CHANGE OF [CONTROL-OBJECT-3](#)。

#### 【補足】

レコード自動切替えをするか否かを示す値が返ります。

### 例外エラー

設定/参照とも例外エラーの発生はありません。

### 注意事項

このプロパティは、帳票開設イベントより後に変更した内容は無効になります。

## 8.7.2 RECORD-EOFプロパティ(レコードEOFプロパティ)

### 説明

[レコード読み込みイベント](#)において、処理対象のレコードの有無を設定します。

“処理対象レコードなし”の場合、最終レコードまで処理したと判断して、帳票印刷の終了処理を行います。

デフォルト値として“処理対象レコードあり”を持ちます。

値	意味
B"1" (真)	処理対象レコードなし
B"0" (偽)	処理対象レコードあり

### プロパティの属性

PIC 1(1) [USAGE IS] DISPLAY

### 記述形式

#### 【設定】

MOVE *BOOL-VAL* TO RECORD-EOF OF [CONTROL-OBJECT-2](#)

#### 【補足】

**BOOL-VAL:**

処理対象のレコードの有無を渡します。

#### 【参照】

RECORD-EOF OF [CONTROL-OBJECT-3](#)

#### 【補足】

処理対象のレコードの有無が返ります。

### 例外エラー

設定/参照とも例外エラーの発生はありません。

### 注意事項

[帳票印刷メソッド\(PRINT-CONTROL\)](#)の“印刷モード引数”に0を指定した場合、最終レコードを処理した後に発生したレコード読み込みイベントで当プロパティに真(B"1")を設定しないと帳票



印刷が無限ループしますので注意してください。

### 使用例

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[レコード読み込みメソッド\(READ-RECORD\)の使用例](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[レコード読み込みメソッド\(READ-RECORD\)の使用例](#)

## 8.7.3 RECORD-COUNTプロパティ(レコード回数プロパティ)

### 説明

レコード読み込み回数を参照します。

レコードを処理した回数を利用して制御したい場合に使用します。

一番最初の[レコード読み込みイベント](#)が発生した時の参照値は、1になります。

このプロパティは参照することだけが可能です。設定をすることはできません。

### プロパティの属性

PIC S9(9) [USAGE IS] COMP-5

### 記述形式

【参照】

RECORD-COUNT OF [CONTROL-OBJECT-2](#)

【補足】

現在のレコード読み込み回数を返します。

### 例外エラー

設定/参照とも例外エラーの発生はありません。

### 注意事項

[印刷種別プロパティ\(PRINT-TYPE\)](#)が“通常印刷 2回読み”または“サマリ印刷 2回読み”の場合、[帳票開設イベント](#)の直後に発生する[レコード読み込みイベント](#)でレコード読み込み回数が必ず 1になります。

### 使用例

“[改ページメソッド\(PAGE-FEED\)の使用例](#)”を参照してください。

## 8.8 印刷制御に関するプロパティ

ここでは、[帳票印刷制御クラス\(FJPRT-CONTROL\)](#) が提供する以下のカテゴリのプロパティについての説明をします。

印刷制御に関するプロパティ

[PRINT-TYPEプロパティ\(印刷種別プロパティ\)](#)  
[PRINT-COUNTプロパティ\(印刷回数プロパティ\)](#)  
[REQUEST-CANCELプロパティ\(キャンセル要求プロパティ\)](#)  
[PRINTER-NAMEプロパティ\(プリンタ名プロパティ\)](#)  
[PRINT-JOB-NAMEプロパティ\(印刷ジョブ名プロパティ\)](#)  
[PRINT-JOB-CONDITIONプロパティ\(印刷ジョブ状態プロパティ\)](#)  
[PAPER-TYPEプロパティ\(用紙種別プロパティ\)](#)  
[PRINT-COPIESプロパティ\(印刷部数プロパティ\)](#)  
[PAPER-SIZEプロパティ\(用紙サイズプロパティ\)](#)  
[PAPER-SIZE-HEIGHTプロパティ\(用紙縦サイズプロパティ\)](#)  
[PAPER-SIZE-WIDTHプロパティ\(用紙横サイズプロパティ\)](#)  
[PRINT-POSITIONプロパティ\(印刷原点プロパティ\)](#)  
[PRINT-POSITION-TOPプロパティ\(印刷原点縦位置プロパティ\)](#)  
[PRINT-POSITION-LEFTプロパティ\(印刷原点横位置プロパティ\)](#)  
[REDUCTION-TYPEプロパティ\(縮小種別プロパティ\)](#)  
[COMPACT-PRINT-TYPEプロパティ\(コンパクト印刷種別プロパティ\)](#)  
[OVERLAY-NAME-TO-COMPACT-PRINTプロパティ\(コンパクト印刷用オーバーレイ名プロパティ\)](#)  
[PRINTING-DIALOGプロパティ\(印刷中ダイアログプロパティ\)](#)  
[PRINTING-DIALOG-CANCEL-BUTTONプロパティ\(印刷中ダイアログのキャンセルボタンプロパティ\)](#)  
[PRINTING-DIALOG-MINIMIZE-BOXプロパティ\(印刷中ダイアログ最小化ボックスプロパティ\)](#)  
[PRINTER- INFORMATION-FILENAMEプロパティ\(プリンタ情報ファイル名プロパティ\)](#)

### 8.8.1 PRINT-TYPEプロパティ(印刷種別プロパティ)

#### 説明

印刷種別を設定/参照します。

“サマリ印刷 1回読み”または“サマリ印刷 2回読み”は、集計表形式の帳票定義体を持つ明細の出力を抑止します。

“通常印刷 2回読み”または“サマリ印刷 2回読み”は、報告書頭書きパーティションに全体の集計結果を印刷する場合や、ページ頭書きにページ内のデータの集計結果を出す場合などに使用します。

印刷種別の設定によるイベントの発生への影響については、“[イベントの発生に影響する事項](#)”を参照してください。

デフォルト値として“通常印刷 1回読み”を持ちます。

記号定数	印刷種別
FJPRT-SC-PRTTYP-NORMAL1	通常印刷 1回読み
FJPRT-SC-PRTTYP-SUMMARY1	サマリ印刷 1回読み
FJPRT-SC-PRTTYP-NORMAL2	通常印刷 2回読み
FJPRT-SC-PRTTYP-SUMMARY2	サマリ印刷 2回読み

**プロパティの属性**

PIC S9(4) [USAGE IS] COMP-5

**記述形式****【設定】**MOVE *SHORT-VAL* TO **PRINT-TYPE** OF [CONTROL-OBJECT-3](#)**【補足】***SHORT-VAL* :

印刷種別を渡します。プロパティの属性と同じ属性の語を記述してください。

**【参照】****PRINT-TYPE** OF [CONTROL-OBJECT-3](#)**【補足】**

現在設定されている印刷種別が返ります。

**例外エラー**設定で例外エラー(コード: [FJPRT-SCE-SET-VALUE](#))が発生することがあります。

参照では例外エラーの発生はありません。

**注意事項**

帳票開設イベントより後に発生したイベントで設定しても無効になります。

“[イベントの発生に影響する事項](#)”を参照してください。**8.8.2 PRINT-COUNTプロパティ(印刷回数プロパティ)****説明**

[印刷種別プロパティ\(PRINT-TYPE\)](#)で“通常印刷 2回読み”または“サマリ印刷 2回読み”を設定した場合に、帳票開設イベントから帳票閉鎖イベントまでの一連のイベントの発生が 2回起こりますが、この発生が何回目であることを示します。

このプロパティは参照だけ可能で、設定/変更はできません。

デフォルト値として“1回目”を持ちます。

値	印刷回数
1	1回目
2	2回目

**プロパティの属性**

PIC S9(4) [USAGE IS] COMP-5

**記述形式****【参照】****PRINT-COUNT** OF [CONTROL-OBJECT-2](#)**【補足】**

現在、一連のイベントの発生が 1回目であるか、2回目であるかが返ります。

**例外エラー**

例外エラーの発生はありません。

**8.8.3 REQUEST-CANCELプロパティ(キャンセル要求プロパティ)****説明**

帳票の印刷の途中で印刷処理を中止したい場合に“あり”を設定します。

デフォルト値として“なし”を持ちます。

値	キャンセル要求
B"1" (真)	あり
B"0" (偽)	なし

## プロパティの属性

PIC 1(1) [USAGE IS] DISPLAY

## 記述形式

### 【設定】

MOVE *BOOL-VAL* TO REQUEST-CANCEL OF [CONTROL-OBJECT-2](#)

### 【補足】

***BOOL-VAL*** :

キャンセル要求をするか否かを渡します。プロパティの属性と同じ属性を持つ語を記述してください。

### 【参照】

REQUEST-CANCEL OF [CONTROL-OBJECT-2](#)

### 【補足】

現在のキャンセル要求の状態が返ります。

## 例外エラー

設定/参照とも例外エラーの発生はありません。

## 注意事項

[イベント](#)を処理するイベント処理メソッドからの参照だけが有効です。

## 使用例

帳票印刷イベント処理クラス(FJPRT-EVENT-HANDLER)の

[エラーメッセージメソッド\(ERROR-MESSAGE\)の使用例](#)

帳票印刷制御クラス(FJPRT-CONTROL)の

[エラーメッセージメソッド\(ERROR-MESSAGE\)の使用例](#)

# 8.8.4 PRINTER-NAMEプロパティ(プリンタ名プロパティ)

## 説明

印刷先のプリンタ名を設定します。設定値の内容は、動作システム依存となります。  
デフォルト値として空白を持ちます。

## プロパティの属性

PIC X(128)

## 記述形式

### 【設定】

MOVE *ALPHANUM-VAL* TO PRINTER-NAME OF [CONTROL-OBJECT-3](#)

### 【補足】

***ALPHANUM-VAL*** :

印刷先プリンタ名を渡します。英数字の属性を持つ語を記述します。

動作モードがシフトJISのときに最大で 63文字まで設定することができます。ただし、実際に有効となる文字の長さの範囲はシステムの制限に依存します。

ローカル接続プリンタは、[スタート]メニュー [設定] [プリンタ]で表示されるプリンター一覧に表示されている名前を設定します。サーバ接続プリンタはネットワーク共有名を設定します。

### 【参照】

PRINTER-NAME OF [CONTROL-OBJECT-3](#)

## 【補足】

現在設定されている印刷先プリンタ名が英数字の属性で返ります。返却されるプリンタ名の最大文字数は、動作モードがシフトJISのときに63文字です。

**例外エラー**

設定/参照とも例外エラーの発生はありません。

**注意事項**

SystemWalker/ListWORKSのプリンタ名を設定することはできません。この場合は“[電子帳票宛先名プロパティ設定メソッド](#)”を使います。

電子帳票を出力する場合は、設定値が無効になります。“[電子帳票種別プロパティ設定メソッド](#)”の注意事項も参照してください。

帳票開設イベントより後に発生したイベントで設定しても無効になります。

**使用例**

“[プリンタデバイス名参照メソッド\(PRINTER-DEVICE-NAMES\)の使用例](#)”を参照してください。

## 8.8.5 PRINT-JOB-NAMEプロパティ(印刷ジョブ名プロパティ)

**説明**

印刷中ダイアログボックスおよびスプールの状態画面に表示する印刷ジョブ名を設定します。デフォルト値として空白を持ちます。

**プロパティの属性**

PIC X(518)

**記述形式**

## 【設定】

MOVE *ALPHANUM-VAL* TO PRINT-JOB-NAME OF [CONTROL-OBJECT-3](#)

## 【補足】

*ALPHANUM-VAL* :

印刷ジョブ名を渡します。英数字の属性を持つ語を記述します。

動作モードがシフトJISのときに最大で 259文字まで設定できます。ただし、実際に有効となる文字の長さはシステムの制限に依存します。

## 【参照】

PRINT-JOB-NAME OF [CONTROL-OBJECT-3](#)

## 【補足】

現在設定されている印刷ジョブ名が英数字の属性で返ります。返却される印刷ジョブ名の最大文字数は、動作モードがシフトJISのときに259文字です。

**例外エラー**

設定/参照とも例外エラーの発生はありません。

**注意事項**

帳票開設イベントより後に発生したイベントで設定しても無効になります。

“PDF形式”の電子帳票を出力する場合は、設定値が無効になります。この場合、指定は文書情報ファイルで行ってください。“[電子帳票種別プロパティ設定メソッド](#)”の注意事項も参照してください。

## 8.8.6 PRINT-JOB-CONDITIONプロパティ (印刷ジョブ状態プロパティ)

### 説明

印刷ジョブの状態を示します。

印刷ジョブの起動中は、新しい印刷ジョブの起動ができないため、このプロパティを利用して制御します。

このプロパティは参照だけ可能で、設定 / 変更はできません。

デフォルト値として “ 停止中 ” を持ちます。

値	印刷ジョブ状態
B"1" (真)	起動中
B"0" (偽)	停止中

### プロパティの属性

PIC 1(1) [USAGE IS] DISPLAY

### 記述形式

【参照】

PRINT-JOB-CONDITION OF [CONTROL-OBJECT-3](#)

【補足】

現在の印刷ジョブ状態が返ります。

### 例外エラー

例外エラーの発生はありません。

## 8.8.7 PAPER-TYPEプロパティ (用紙種別プロパティ)

### 説明

連帳印刷可能なプリンタ装置を利用する場合は、連帳印刷を行うか単票印刷を行うかを設定します。

連帳印刷ができないプリンタ装置に印刷を行う場合は、“ 単票 ” を設定します。

デフォルト値として “ 連帳 ” を持ちます。

記号定数	用紙種別
FJPRT-SC-PAPTYP-SERIAL	連帳
FJPRT-SC-PAPTYP-SINGLE	単票

### プロパティの属性

PIC S9(4) [USAGE IS] COMP-5

### 記述形式

【設定】

MOVE *SHORT-VAL* TO PAPER-TYPE OF [CONTROL-OBJECT-3](#)

【補足】

*SHORT-VAL* :

用紙種別を渡します。プロパティの属性と同じ属性を持つ語を記述してください。

【参照】

PAPER-TYPE OF [CONTROL-OBJECT-3](#)

【補足】

現在設定されている用紙種別が返ります。

### 例外エラー

設定で例外エラー(コード: [FJPRT-SCE-SET-VALUE](#))が発生することがあります。  
参照では例外エラーの発生はありません。

### 注意事項

帳票開設イベントより後に発生したイベントで設定しても無効になります。  
電子帳票を出力する場合は、設定値が無効になります。“[電子帳票種別プロパティ設定メソッド](#)”の注意事項も参照してください。

## 8.8.8 PRINT-COPIESプロパティ(印刷部数プロパティ)

### 説明

印刷部数を設定します。  
デフォルト値として 1を持ちます。

### プロパティの属性

PIC S9(4) [USAGE IS] COMP-5

### 記述形式

#### 【設定】

MOVE *SHORT-VAL* TO PRINT-COPIES OF [CONTROL-OBJECT-3](#)

#### 【補足】

##### *SHORT-VAL* :

印刷部数を渡します。プロパティの属性と同じ属性を持つ語を記述してください。  
設定可能な値の範囲は 1～99 です。

#### 【参照】

PRINT-COPIES OF [CONTROL-OBJECT-3](#)

#### 【補足】

現在設定されている印刷部数が返ります。値の範囲は 1～99 です。

### 例外エラー

設定で例外エラー(コード: [FJPRT-SCE-SET-VALUE](#))が発生することがあります。  
参照では例外エラーの発生はありません。

### 注意事項

帳票開設イベントより後に発生したイベントで設定しても無効になります。  
電子帳票を出力する場合は、設定値が無効になります。“[電子帳票種別プロパティ設定メソッド](#)”の注意事項も参照してください。

### 使用例

“[プリンタ情報生成メソッド\(EXPORT-PRINTER-INFORMATION\)の使用例](#)”を参照してください。

## 8.8.9 PAPER-SIZEプロパティ(用紙サイズプロパティ)

### 説明

帳票定義体上の用紙サイズの定義を無視して任意の用紙サイズの帳票印刷を行うか否かを設定/参照します。  
印刷対象の帳票定義体をすべて同一の用紙サイズで印刷を行いたい場合などに使用します。  
デフォルト値として任意の用紙サイズの“指定なし”を持ちます。

値	用紙サイズ指定
B"1" (真)	指定あり
B"0" (偽)	指定なし

## プロパティの属性

PIC 1(1) [USAGE IS] DISPLAY

### 記述形式

#### 【設定】

MOVE *BOOL-VAL* TO **PAPER-SIZE** OF [CONTROL-OBJECT-3](#)

#### 【補足】

**BOOL-VAL** :

用紙サイズを設定するか否かを渡します。プロパティの属性と同じ属性を持つ語を記述してください。

#### 【参照】

**PAPER-SIZE** OF [CONTROL-OBJECT-3](#)

#### 【補足】

現在設定されている用紙サイズの設定の状態が返ります。

### 例外エラー

設定/参照とも例外エラーの発生はありません。

### 使用例

```
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID. USER-HANDLER-CLASS INHERITS FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS FJPRT-EVENT-HANDLER
    CLASS FJPRT-CONTROL
    :
IDENTIFICATION DIVISION.
METHOD-ID. OPEN-REPORT OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-CTRLOBJ OBJECT REFERENCE FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    MOVE B"1" TO PAPER-SIZE OF L-CTRLOBJ *>.....[1]
    MOVE 2970 TO PAPER-SIZE-HEIGHT OF L-CTRLOBJ *>.....[2]
    MOVE 2100 TO PAPER-SIZE-WIDTH OF L-CTRLOBJ *>.....[3]
    :
END METHOD OPEN-REPORT.
:
END CLASS USER-HANDLER-CLASS.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
```

#### [例の説明]

- [1]  
用紙サイズを任意に設定できるようにします。
- [2]  
用紙縦サイズを 297.0mm に設定します。
- [3]  
用紙横サイズを 210.0mm に設定します。



### 8.8.10 PAPER-SIZE-HEIGHTプロパティ(用紙縦サイズプロパティ)

#### 説明

印刷を行う用紙の縦サイズを0.1mm単位で設定します。

設定可能なサイズは、プリンタドライバによって異なりますので、プリンタドライバの仕様を確認してサイズを設定してください。

[用紙サイズプロパティ\(PAPER-SIZE\)](#)が“指定あり”の場合にだけ有効です。

デフォルト値として 1を持ちます。

#### プロパティの属性

PIC S9(9) [USAGE IS] COMP-5

#### 記述形式

##### 【設定】

MOVE *LONG-VAL* TO PAPER-SIZE-HEIGHT OF [CONTROL-OBJECT-3](#)

##### 【補足】

*LONG-VAL* :

用紙縦サイズを渡します。プロパティの属性と同じ属性を持つ語を記述してください。

##### 【参照】

PAPER-SIZE-HEIGHT OF [CONTROL-OBJECT-3](#)

##### 【補足】

現在設定されている用紙縦サイズが返ります。

#### 例外エラー

設定/参照とも例外エラーの発生はありません。

#### 注意事項

帳票定義体の“印刷方向”に“横”を設定した場合、このプロパティの情報は用紙の横サイズになります。

#### 使用例

“[用紙サイズプロパティ\(PAPER-SIZE\)の使用例](#)”を参照してください。

### 8.8.11 PAPER-SIZE-WIDTHプロパティ(用紙横サイズプロパティ)

#### 説明

印刷を行う用紙の横サイズを0.1mm単位で設定します。

設定可能なサイズは、プリンタドライバによって異なりますので、プリンタドライバの仕様を確認してサイズを設定してください。

[用紙サイズプロパティ\(PAPER-SIZE\)](#)が“指定あり”の場合にだけ有効です。

デフォルト値として 1を持ちます。

#### プロパティの属性

PIC S9(9) [USAGE IS] COMP-5

#### 記述形式

##### 【設定】

MOVE *LONG-VAL* TO PAPER-SIZE-WIDTH OF [CONTROL-OBJECT-3](#)

##### 【補足】

*LONG-VAL* :

用紙横サイズを渡します。プロパティの属性と同じ属性を持つ語を記述してください。

##### 【参照】

PAPER-SIZE-WIDTH OF [CONTROL-OBJECT-3](#)

【補足】

現在設定されている用紙横サイズが返ります。

### 例外エラー

設定/参照とも例外エラーの発生はありません。

### 注意事項

帳票定義体の“印刷方向”に“横”を設定した場合、このプロパティの情報は用紙の縦サイズになります。

### 使用例

“[用紙サイズプロパティ\(PAPER-SIZE\)の使用例](#)”を参照してください。

## 8.8.12 PRINT-POSITIONプロパティ(印刷原点プロパティ)

### 説明

帳票定義体上の印刷原点の定義を無視して任意に印刷原点での帳票印刷を行うか否かを設定/参照します。

印刷対象の帳票定義体をすべて同一の原点で印刷を行いたい場合などに使用します。

デフォルト値として“指定なし”を持ちます。印刷原点を任意に与える設定をしない場合は、プリンタ装置の印字可能域の左上を原点として印刷が行われます。

値	印刷原点指定
B"1" (真)	指定あり
B"0" (偽)	指定なし

### プロパティの属性

PIC 1(1) [USAGE IS] DISPLAY

### 記述形式

【設定】

MOVE *BOOL-VAL* TO PRINT-POSITION OF [CONTROL-OBJECT-3](#)

【補足】

*BOOL-VAL* :

印刷原点を任意に与えるか否かを渡します。プロパティの属性と同じ属性を持つ語を記述してください。

【参照】

PRINT-POSITION OF [CONTROL-OBJECT-3](#)

【補足】

現在設定されている印刷原点を任意に与えるか否かの状態が返ります。

### 例外エラー

設定/参照とも例外エラーの発生はありません。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER-CLASS  INHERITS  FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS  FJPRT-EVENT-HANDLER
    CLASS  FJPRT-CONTROL
    :
IDENTIFICATION DIVISION.
METHOD-ID.  OPEN-REPORT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.
    MOVE  B"1"  TO  PRINT-POSITION OF L-CTRLOBJ  *>.....[1]
    MOVE  2000  TO  PRINT-POSITION-TOP OF L-CTRLOBJ  *>.....[2]
    MOVE  1000  TO  PRINT-POSITION-LEFT OF L-CTRLOBJ  *>.....[3]
    :
END METHOD  OPEN-REPORT.
:
END CLASS  USER-HANDLER-CLASS.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

- [1]  
印刷原点を任意に設定できるようにします。
- [2]  
印刷原点縦位置を 20.00mm に設定します。
- [3]  
印刷原点横位置を 10.00mm に設定します。

### 8.8.13 PRINT-POSITION-TOPプロパティ(印刷原点縦位置プロパティ)

## 説明

印刷原点の縦位置を用紙の上端からの絶対位置を0.01mm単位で設定します。  
[印刷原点プロパティ\(PRINT-POSITION\)](#)が“指定あり”の場合にだけ有効です。  
 デフォルト値として 0を持ちます。

## プロパティの属性

PIC S9(9) [USAGE IS] COMP-5

## 記述形式

## 【設定】

MOVE *LONG-VAL* TO **PRINT-POSITION-TOP** OF [CONTROL-OBJECT-3](#)

## 【補足】

**LONG-VAL :**

印刷原点縦位置を渡します。プロパティの属性と同じ属性を持つ語を記述してくだ

さい。

【参照】

PRINT-POSITION-TOP OF [CONTROL-OBJECT-3](#)

【補足】

現在設定されている印刷原点縦位置が返ります。

例外エラー

設定/参照とも例外エラーの発生はありません。

使用例

“ [印刷原点プロパティ \(PRINT-POSITION\)](#) の使用例 ” を参照してください。

## 8.8.14 PRINT-POSITION-LEFTプロパティ (印刷原点横位置プロパティ)

説明

印刷原点の横位置を用紙の左端からの絶対位置を0.01mm単位で設定します。

[印刷原点プロパティ \(PRINT-POSITION\)](#) が “ 指定あり ” の場合にだけ有効です。

デフォルト値として 0を持ちます。

プロパティの属性

PIC S9(9) [USAGE IS] COMP-5

記述形式

【設定】

MOVE LONG-VAL TO PRINT-POSITION-LEFT OF [CONTROL-OBJECT-3](#)

【補足】

LONG-VAL :

印刷原点横位置を渡します。プロパティの属性と同じ型を持つ語を記述してください。

【参照】

PRINT-POSITION-LEFT OF [CONTROL-OBJECT-3](#)

【補足】

現在設定されている印刷原点横位置が返ります。

例外エラー

設定/参照とも例外エラーの発生はありません。

使用例

“ [印刷原点プロパティ \(PRINT-POSITION\)](#) の使用例 ” を参照してください。

## 8.8.15 REDUCTION-TYPEプロパティ (縮小種別プロパティ)

説明

帳票定義体で定義したサイズから、設定した用紙範囲に縮小して印刷を行うか否かを設定します。

“ 用紙範囲 ” とは、[用紙サイズプロパティ \(PAPER-SIZE\)](#) が “ 指定あり ” の場合に[用紙縦サイズプロパティ \(PAPER-SIZE-HEIGHT\)](#)、[用紙横サイズプロパティ \(PAPER-SIZE-WIDTH\)](#)に設定した範囲です。[用紙サイズプロパティ \(PAPER-SIZE\)](#) が “ 指定なし ” の場合は無効になります。

また、等方性を守って印刷を行う / 行わないの設定も可能です。“ 等方性を守る ” とは、帳票定義体の縦横の比率を維持したまま縮小を行います。

帳票定義体の拡大 / 縮小設定は無視されます。

デフォルト値として “ 縮小なし ” を持ちます。

記号定数	縮小種別
FJPRT-SC-RDCTYP-OFF	縮小なし
FJPRT-SC-RDCTYP-ON1	縮小あり(等方性有効)
FJPRT-SC-RDCTYP-ON2	縮小あり(等方性無効)

### プロパティの属性

PIC S9(4) [USAGE IS] COMP-5

### 記述形式

#### 【設定】

MOVE *SHORT-VAL* TO REDUCTION-TYPE OF [CONTROL-OBJECT-3](#)

#### 【補足】

**SHORT-VAL :**

縮小種別を渡します。プロパティの属性と同じ属性を持つ語を記述してください。

#### 【参照】

REDUCTION-TYPE OF [CONTROL-OBJECT-3](#)

#### 【補足】

現在設定されている縮小種別が返ります。

### 例外エラー

設定で例外エラー(コード: [FJPRT-SCE-SET-VALUE](#))が発生することがあります。

参照では例外エラーの発生はありません。

### 注意事項

“ ListWORKSの電子帳票 ” を出力する場合は、設定値が無効になります。“ [電子帳票種別プロパティ設定メソッド](#) ” の注意事項も参照してください。

## 8.8.16 COMPACT-PRINT-TYPEプロパティ(コンパクト印刷種別プロパティ)

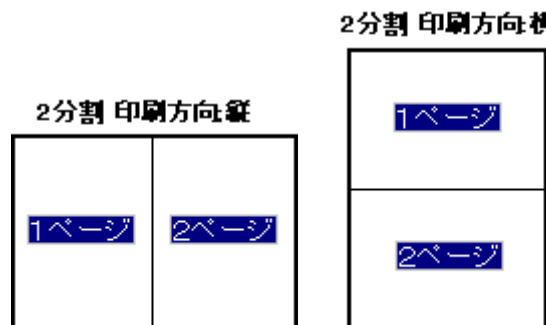
### 説明

コンパクト印刷を行うか否かを設定します。

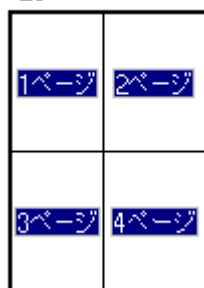
デフォルト値として“ コンパクト印刷なし ” を持ちます。

記号定数	コンパクト印刷種別
FJPRT-SC-CPTTYP-OFF	コンパクト印刷なし
FJPRT-SC-CPTTYP-ON2	2分割
FJPRT-SC-CPTTYP-ON4W	4分割(横方向)
FJPRT-SC-CPTTYP-ON4H	4分割(縦方向)

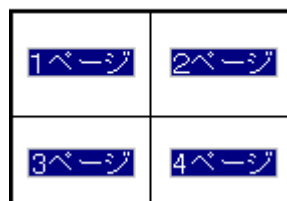
以下にコンパクト印刷を設定した場合の印刷例を示します。



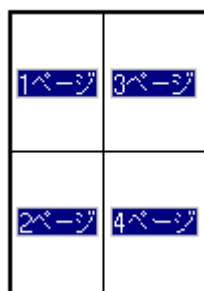
4分割(横) 印刷方向:縦



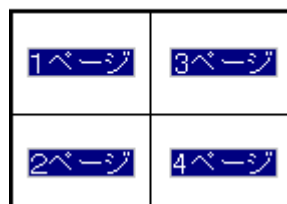
4分割(横) 印刷方向:横



4分割(縦) 印刷方向:縦



4分割(縦) 印刷方向:横



図：コンパクト印刷設定時の印刷例

## プロパティの属性

PIC S9(4) [USAGE IS] COMP-5

## 記述形式

### 【設定】

MOVE *SHORT-VAL* TO COMPACT-PRINT-TYPE OF [CONTROL-OBJECT-3](#)

### 【補足】

*SHORT-VAL* :

コンパクト印刷種別を渡します。プロパティの属性と同じ属性を持つ語を記述してください。

### 【参照】

COMPACT-PRINT-TYPE OF [CONTROL-OBJECT-3](#)

### 【補足】

現在設定されているコンパクト印刷種別が返ります。

## 例外エラー

設定で例外エラー(コード: [FJPR-T-SCE-SET-VALUE](#))が発生することがあります。

参照では例外エラーの発生はありません。

## 注意事項

帳票開設イベントより後に発生したイベントで設定しても無効になります。

タックシール形式の帳票定義体では、このプロパティの設定は無効になります。

電子帳票を出力する場合は、設定値が無効になります。“[電子帳票種別プロパティ設定メソッド](#)”の注意事項も参照してください。

### 8.8.17 OVERLAY-NAME-TO-COMPACT-PRINTプロパティ(コンパクト印刷用オーバーレイ名プロパティ)

#### 説明

コンパクト印刷時に分割された領域でなく、用紙全体に対しての背景となるオーバーレイの名前を設定します。

[コンパクト印刷種別プロパティ \(COMPACT-PRINT-TYPE\)](#)で“コンパクト印刷なし”以外を設定した場合に有効です。

オーバーレイ定義体として、OVD、BMP、WMF、EMFの各形式が設定可能です。

付加できるパスとして、フルパス、相対パスがあります。

デフォルト値として空白を持ちます。

#### プロパティの属性

PIC X(518)

#### 記述形式

##### 【設定】

MOVE *ALPHANUM-VAL* TO OVERLAY-NAME-TO-COMPACT-PRINT OF [CONTROL-OBJECT-3](#)

##### 【補足】

*ALPHANUM-VAL* :

オーバーレイ名を渡します。英数字の属性を持つ語を記述します。

動作モードがシフトJISのときに最大で 259文字を設定可能です。ただし、実際に有効となる文字の長さはシステムの制限に依存します。

##### 【参照】

OVERLAY-NAME-TO-COMPACT-PRINT OF [CONTROL-OBJECT-3](#)

##### 【補足】

現在設定されているコンパクト印刷用オーバーレイ名が英数字の属性で返ります。返却されるコンパクト印刷用オーバーレイ名の最大文字数は、動作モードがシフトJISのときに259文字です。

#### 例外エラー

設定/参照とも例外エラーの発生はありません。

#### 注意事項

帳票開設イベントより後に発生したイベントで設定しても無効になります。

タックシール形式の帳票定義体では、このプロパティの設定は無効になります。

電子帳票を出力する場合は、設定値が無効になります。“[電子帳票種別プロパティ設定メソッド](#)”の注意事項も参照してください。

### 8.8.18 PRINTING-DIALOGプロパティ(印刷中ダイアログプロパティ)

#### 説明

印刷中ダイアログを表示するか否かを設定します。

デフォルト値として“表示する”を持ちます。

表示される印刷中ダイアログの例を以下に示します。



図：印刷中ダイアログの例

“ダイアログのタイトル”には、帳票定義体で設定されたものを表示します。設定が省略された場合は、“帳票印刷”と表示されます。

“印刷名”には、[印刷ジョブ名プロパティ \(PRINT-JOB-NAME\)](#) で設定した文字列が表示されます。

“出力先”には、印刷を行うプリンタ名が表示されます。このプリンタ名は、[プリンタ名プロパティ \(PRINTER-NAME\)](#) で設定したプリンタ名が表示されます。設定が省略された場合は、システムのデフォルトプリンタの名前が表示されます。

“印刷中のページ数”の表示は、帳票開設イベントから帳票閉鎖イベントまでに、通算何ページを印刷を行ったかを表示します。そのため、[最初のページ番号プロパティ \(INIT-PAGE-NUMBER\)](#) で設定したページ番号や、[印刷範囲プロパティ \(PAGE-RANGE\)](#) で範囲が設定されても、表示は 1 ページから始まります。

値	印刷中ダイアログ表示指定
B"1" (真)	表示する
B"0" (偽)	表示しない

## プロパティの属性

PIC 1(1) [USAGE IS] DISPLAY

## 記述形式

### 【設定】

MOVE *BOOL-VAL* TO PRINTING-DIALOG OF [CONTROL-OBJECT-3](#)

### 【補足】

*BOOL-VAL* :

印刷中ダイアログを表示するか否かを渡します。プロパティの属性と同じ属性の語を記述してください。

### 【参照】

PRINTING-DIALOG OF [CONTROL-OBJECT-3](#)

### 【補足】

現在設定されている印刷中ダイアログ表示の状態が返ります。

## 例外エラー

設定/参照とも例外エラーの発生はありません。

## 注意事項

帳票開設イベントより後に発生したイベントで設定しても無効になります。

### 8.8.19 PRINTING-DIALOG-CANCEL-BUTTONプロパティ (印刷中ダイアログのキャンセルボタンプロパティ)

#### 説明

印刷中ダイアログに対して、キャンセルボタンを表示するか否かを設定します。デフォルト値として“表示する”を持ちます。



値	印刷中ダイアログのキャンセルボタン指定
B"1" (真)	表示する
B"0" (偽)	表示しない

“表示しない”を設定した場合の印刷中ダイアログを以下に示します。



図：キャンセルボタンを表示しない印刷中ダイアログ

### プロパティの属性

PIC 1(1) [USAGE IS] DISPLAY

### 記述形式

#### 【設定】

MOVE *BOOL-VAL* TO PRINTING-DIALOG-CANCEL-BUTTON OF [CONTROL-OBJECT-3](#)

#### 【補足】

**BOOL-VAL :**

印刷中ダイアログのキャンセルボタンを表示するか否かを渡します。プロパティの属性と同じ属性の語を記述してください。

#### 【参照】

PRINTING-DIALOG-CANCEL-BUTTON OF [CONTROL-OBJECT-3](#)

#### 【補足】

現在設定されている印刷中ダイアログのキャンセルボタンの状態が返ります。

### 例外エラー

設定/参照とも例外エラーの発生はありません。

### 注意事項

帳票開設イベントより後に発生したイベントで設定しても無効になります。

## 8.8.20 PRINTING-DIALOG-MINIMIZE-BOXプロパティ (印刷中ダイアログ最小化ボックスプロパティ)

### 説明

印刷中ダイアログに最小化ボックスをつけるか否かを設定します。

デフォルト値として“つけない”を持ちます。

値	最小化ボックス付加指定
B"1" (真)	つける
B"0" (偽)	つけない

“つける”を設定した場合の印刷中ダイアログを以下に示します。



図：最小化ボックスを表示した印刷中ダイアログ

## プロパティの属性

PIC 1(1) [USAGE IS] DISPLAY

## 記述形式

### 【設定】

MOVE *BOOL-VAL* TO PRINTING-DIALOG-MINIMIZE-BOX OF [CONTROL-OBJECT-3](#)

### 【補足】

#### *BOOL-VAL*:

最小化ボックスを付加するか否かを設定します。プロパティの属性と同じ属性を持つ語を記述してください。

### 【参照】

PRINTING-DIALOG-MINIMIZE-BOX OF [CONTROL-OBJECT-3](#)

### 【補足】

現在設定されている最小化ボックス付加の状態が返ります。

## 例外エラー

設定/参照とも例外エラーの発生はありません。

## 注意事項

帳票開設イベントより後に発生したイベントで設定しても無効になります。

# 8.8.21 PRINTER- INFORMATION-FILENAME プロパティ (プリンタ情報ファイル名プロパティ)

## 説明

印刷時に使用するプリンタ情報ファイル名を設定します。  
プリンタ情報ファイルを利用することで、[印刷制御オブジェクト](#)が持つ複数のプロパティ値を一括で変更することができます。また、既存のプログラムのプロパティ値を参照して、印刷制御オブジェクトのプロパティを設定することができます。  
付加できるパスとして、フルパス、相対パスがあります。  
プリンタ情報ファイルの内容は、このプロパティを設定したタイミングで、帳票印刷クラスライブラリのプロパティに反映されます。  
デフォルト値として空白を持ちます。

## プロパティの属性

PIC X(518)

## 記述形式

### 【設定】

MOVE *ALPHANUM-VAL* TO PRINTER- INFORMATION-FILENAME OF [CONTROL-OBJECT-3](#)

### 【補足】

#### *ALPHANUM-VAL* :

プリンタ情報ファイル名を渡します。英数字の属性を持つ語を記述します。

動作モードがシフトJISのときに最大で 259文字まで設定することができます。ただし、実際に有効となる文字の長さはシステムの制限に依存します。

【参照】

PRINTER- INFORMATION-FILENAME OF [CONTROL-OBJECT-3](#)

【補足】

現在設定されているプリンタ情報ファイル名が英数字の属性で返ります。返却されるプリンタ情報ファイル名の最大文字数は、動作モードがシフトJISのときに259文字です。

### 例外エラー

設定/参照とも例外エラーの発生はありません。

### **注意事項**

帳票開設イベントより後に発生したイベントで設定しても無効になります。

## 8.9 イベント処理に関するプロパティ

ここでは、[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)が提供する以下のカテゴリのプロパティについての説明をします。

イベント処理に関するプロパティ

[EVENT-HANDLERプロパティ\(イベント処理オブジェクトプロパティ\)](#)

### 8.9.1 EVENT-HANDLERプロパティ(イベント処理オブジェクトプロパティ)

#### 説明

帳票の印刷時に発生する[イベント](#)を処理するオブジェクト([イベント処理オブジェクト](#))を設定([登録](#)/登録の解消)/参照します。

デフォルト値は NULL(イベント処理オブジェクトが登録されていない状態)です。

#### プロパティの属性

[USAGE IS] OBJECT REFERENCE

#### 記述形式

【設定 (登録/登録の解消)】

SET **EVENT-HANDLER** OF [CONTROL-OBJECT-1](#) TO [HANDLER-OBJECT-1](#)

【補足】

**HANDLER-OBJECT-1 :**

[イベント処理オブジェクト](#)または NULLオブジェクトを渡します。

イベント処理オブジェクトを登録するときは、[帳票印刷イベント処理クラス\(FJPRT-EVENT-HANDLER\)](#)を継承したクラス(例: *USER-HANDLER-CLASS*)のオブジェクト参照一意名を記述します。オブジェクト参照一意名は次の属性で宣言します。

[USAGE IS] OBJECT REFERENCE *USER-HANDLER-CLASS*

イベント処理オブジェクトの登録を解消するときは、NULLを記述します。

【参照】

SET [HANDLER-OBJECT-2](#) TO **EVENT-HANDLER** OF [CONTROL-OBJECT-1](#)

【補足】

**HANDLER-OBJECT-2 :**

[イベント処理オブジェクト](#)または NULLオブジェクトが返ります。

型付けなしのオブジェクト参照一意名を記述してください。具体的には次の属性でオブジェクト参照一意名を宣言します。

[USAGE IS] OBJECT REFERENCE

なお、イベント処理オブジェクトが登録されていない場合、オブジェクト参照一意名には NULLオブジェクトが格納されます。

#### 例外エラー

設定で例外エラー(コード: [FJPRT-SCE-ILLEGAL-STATE](#))が発生することがあります。詳細については注意事項を参照してください。

参照では例外エラーの発生はありません。

#### 注意事項

[帳票印刷メソッド\(PRINT-CONTROL\)](#)を呼び出してから発生する[イベント](#)に対応するイベント処理メソッド内からは、当プロパティにオブジェクトを登録/登録解消をすることは

できません。この状態で登録/登録解除を行った場合、例外エラー(コード:  
[FJPRT-SCE-ILLEGAL-STATE](#))が発生します。  
[印刷制御オブジェクトが削除される](#)とき、\_FINALIZEメソッドによってイベント処理オブ  
ジェクトの登録が解除されます。

### 使用例

帳票印刷制御クラス(FJPRT-CONTROL)の  
[帳票印刷メソッド\(PRINT-CONTROL\)の使用例](#)

## 8.10 電子帳票制御のプロパティに関するメソッド

ここでは、[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)が提供する以下のカテゴリのメソッドについての説明をします。

電子帳票制御のプロパティに関するメソッド

[\\_SET-E-REPORT-TYPEメソッド\(電子帳票種別プロパティ設定\)](#)

[\\_GET-E-REPORT-TYPEメソッド\(電子帳票種別プロパティ参照\)](#)

[\\_SET-E-REPORT-CONTROL-FILENAMEメソッド\(電子帳票制御ファイル名プロパティ設定\)](#)

[\\_GET-E-REPORT-CONTROL-FILENAMEメソッド\(電子帳票制御ファイル名プロパティ参照\)](#)

[\\_SET-E-REPORT-DESTINATIONメソッド\(電子帳票宛先名プロパティ設定\)](#)

[\\_GET-E-REPORT-DESTINATIONメソッド\(電子帳票宛先名プロパティ参照\)](#)

[\\_SET-E-REPORT-FILENAMEメソッド\(電子帳票ファイル名プロパティ設定\)](#)

[\\_GET-E-REPORT-FILENAMEメソッド\(電子帳票ファイル名プロパティ参照\)](#)

以下に、本文の引数/復帰値の項目を読む上での注意事項を挙げます。



**注意**

引数/復帰値の項目に書かれている“属性”は、メソッドの定義上のインタフェースを表すものです。

PIC X ANY LENGTH の属性を持つ引数には、任意の長さを持つ英数字のデータ項目を記述してください。なお、この引数には値が返却されることがあります。これについては各メソッドの仕様を確認してください。

### 8.10.1 \_SET-E-REPORT-TYPEメソッド(電子帳票種別プロパティ設定メソッド)

### 8.10.2 \_GET-E-REPORT-TYPEメソッド(電子帳票種別プロパティ参照メソッド)

#### 説明

帳票印刷の結果を電子帳票として出力するかどうかを設定/参照します。

帳票を電子帳票として出力するには、当プロパティに以下に示す電子帳票種別を設定してください。電子帳票種別を“指定なし”にした場合は、電子帳票の出力は行われず、プリンタに出力されます。

デフォルト値として“指定なし”を持ちます。

記号定数	電子帳票種別	電子帳票関連ソフトウェア
FJPRT-SC-EREPTYP-OFF	指定なし	なし
FJPRT-SC-EREPTYP-SPL	ListWORKSの電子帳票	SystemWalker/ListWORKS
FJPRT-SC-EREPTYP-PDF	PDF形式	SystemWalker/e-DocGenerator

#### 記述形式

【設定】

```
INVOKE CONTROL-OBJECT-3 "_SET-E-REPORT-TYPE"
      USING [BY REFERENCE] E-REPORT-TYPE
```

## 【参照】

INVOKE [CONTROL-OBJECT-3](#) "\_GET-E-REPORT-TYPE"  
RETURNING *E-REPORT-TYPE*

## 引数 / 復帰値

*E-REPORT-TYPE* 【引数/復帰値の属性: PIC S9(4) [USAGE IS] COMP-5】:

設定時には、電子帳票種別を渡します。引数の属性と同じ属性を持つデータ項目か定数を記述します。

参照時には、現在設定されている電子帳票種別が返ります。引数の属性と同じ属性を持つデータ項目を記述します。

## 例外エラー

設定で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-SET-VALUE](#)

## 注意事項

このプロパティは、帳票開設イベントより後に変更した内容は無効になります。

電子帳票の出力を選択した場合、以下のプロパティ設定メソッドを呼び出してプロパティを設定してください。設定しなかった場合は、印刷エラーが発生します。

[電子帳票制御ファイル名プロパティ設定メソッド  
\(\[\\\_SET-E-REPORT-CONTROL-FILENAME\]\(#\)\)](#)

[電子帳票宛先名プロパティ設定メソッド\(\[\\\_SET-E-REPORT-DESTINATION\]\(#\)\)](#)  
(“ListWORKSの電子帳票”の場合のみ)

[電子帳票ファイル名プロパティ設定メソッド\(\[\\\_SET-E-REPORT-FILENAME\]\(#\)\)](#) (“PDF形式”の場合のみ)

電子帳票の出力を選択した場合、以下のプロパティに影響があります。それぞれのプロパティの注意事項を参照してください。

[プリンタ名プロパティ\(\[PRINTER-NAME\]\(#\)\)](#)

[印刷ジョブ名プロパティ\(\[PRINT-JOB-NAME\]\(#\)\)](#)

[用紙種別プロパティ\(\[PAPER-TYPE\]\(#\)\)](#)

[印刷部数プロパティ\(\[PRINT-COPIES\]\(#\)\)](#)

[縮小種別プロパティ\(\[REDUCTION-TYPE\]\(#\)\)](#)

[コンパクト印刷種別プロパティ\(\[COMPACT-PRINT-TYPE\]\(#\)\)](#)

[コンパクト印刷用オーバーレイ名プロパティ\(\[OVERLAY-NAME-TO-COMPACT-PRINT\]\(#\)\)](#)

## 使用例

“[電子帳票宛先名プロパティ設定メソッド\(\[\\\_SET-E-REPORT-DESTINATION\]\(#\)\)の使用例](#)”および  
“[電子帳票ファイル名プロパティ設定メソッド\(\[\\\_SET-E-REPORT-FILENAME\]\(#\)\)の使用例](#)”を参照してください。

**8.10.3    [\\_SET-E-REPORT-CONTROL-FILENAME](#)メソッド(電子帳票制御  
ファイル名プロパティ設定メソッド)**

**8.10.4    [\\_GET-E-REPORT-CONTROL-FILENAME](#)メソッド(電子帳票制御  
ファイル名プロパティ参照メソッド)**

## 説明

電子帳票制御ファイル名を設定/参照します。電子帳票の出力を行うには、当設定メソッドで電子帳票制御ファイル名を設定してください。

当プロパティは、電子帳票種別プロパティで“指定なし”以外を指定した場合だけ有効となります。

す。

電子帳票制御ファイル名とは、具体的には以下のものです。

電子帳票関連ソフトウェア	電子帳票種別	電子帳票制御ファイル名
SystemWalker/ListWORKS	ListWORKSの電子帳票	電子帳票情報ファイル名
SystemWalker/e-DocGenerator	PDF形式	文書情報ファイル名

## 記述形式

### 【設定】

```
INVOKE CONTROL-OBJECT-3 "_SET-E-REPORT-CONTROL-FILENAME"
      USING [BY REFERENCE] CTRL-FILENAME
```

### 【参照】

```
INVOKE CONTROL-OBJECT-3 "_GET-E-REPORT-CONTROL-FILENAME"
      USING [BY REFERENCE] CTRL-FILENAME
```

## 引数 / 復帰値

**CTRL-FILENAME** 【引数の属性: PIC X ANY LENGTH】:

設定時には、電子帳票種別に対応する電子帳票制御ファイル名を渡します。引数には任意の長さの英数字項目を記述します。動作モードがシフトJISのときに最大で 259文字(PIC X(518))まで設定することができます。ただし、実際に有効となる文字列の長さはシステムの制限に依存します。

参照時には、現在設定されている電子帳票制御ファイル名が返ります。引数には任意の長さの英数字項目を記述します。返却される文字列の最大文字数は、動作モードがシフトJISのときに259文字です。

## 注意事項

このプロパティは、帳票開設イベントより後に変更した内容は無効になります。

## 使用例

“ [電子帳票宛先名プロパティ設定メソッド\(\\_SET-E-REPORT-DESTINATION\)の使用例](#) ” および  
“ [電子帳票ファイル名プロパティ設定メソッド\(\\_SET-E-REPORT-FILENAME\)の使用例](#) ”を参照してください。

### 8.10.5 \_SET-E-REPORT-DESTINATIONメソッド(電子帳票宛先名プロパティ設定メソッド)

### 8.10.6 \_GET-E-REPORT-DESTINATIONメソッド(電子帳票宛先名プロパティ参照メソッド)

## 説明

“ ListWORKSの電子帳票 ” の作成時の出力先を設定/参照します。電子帳票種別が “ ListWORKSの電子帳票 ” である電子帳票の出力を行うには、当設定メソッドで電子帳票宛先名を設定してください。

電子帳票宛先名は、具体的にはSystemWalker/ListWORKSのプリンタ名を設定します。これは、動作システム依存の形式となります。

## 記述形式

### 【設定】

```
INVOKE CONTROL-OBJECT-3 "_SET-E-REPORT-DESTINATION"
      USING [BY REFERENCE] PRINTER-NAME
```



## 【参照】

```
INVOKE CONTROL-OBJECT-3 "_GET-E-REPORT-DESTINATION"  
      USING [BY REFERENCE] PRINTER-NAME
```

**引数 / 復帰値**

***PRINTER-NAME*** 【引数の属性: PIC X ANY LENGTH】:

設定時には、電子帳票宛先名を渡します。引数には任意の長さの英数字項目を記述します。動作モードがシフトJISのときに最大で 63文字(PIC X(126))まで設定することができます。ただし、実際に有効となる文字列の長さはシステムの制限に依存します。

ローカル接続プリンタの場合は、[ スタート ] メニュー [ 設定 ] [ プリンタ ] で表示されるプリンター一覧に表示されている名前を渡します。サーバ接続プリンタの場合は、ネットワーク共有名が付加された名前を渡します。

参照時には、現在設定されている電子帳票宛先名が返ります。引数には任意の長さの英数字項目を記述します。返却される文字列の最大文字数は、動作モードがシフトJISのときに63文字です。

**注意事項**

このプロパティは、帳票開設イベントより後に変更した内容は無効になります。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER-CLASS  INHERITS  FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS  FJPRT-EVENT-HANDLER
    CLASS  FJPRT-CONTROL
.
SPECIAL-NAMES.
    SYMBOLIC CONSTANT
    COPY  FJPRTSC.
.
:
IDENTIFICATION DIVISION.
METHOD-ID.  OPEN-REPORT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01  W-ETYPE  PIC  S9(4) COMP-5.
    01  W-CFILE  PIC  X(518).
    01  W-PDEST  PIC  X(126).
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.

    MOVE  "FORM1"  TO  FORM-NAME  OF  L-CTRLOBJ

    MOVE  FJPRT-SC-EREPTYP-SPL  TO  W-ETYPE
    INVOKE  L-CTRLOBJ  " _SET-E-REPORT-TYPE "  USING  W-ETYPE  *>.....[1]

    MOVE  "listworks.env"  TO  W-CFILE
    INVOKE  L-CTRLOBJ  " _SET-E-REPORT-CONTROL-FILENAME "  USING  W-CFILE  *>...[2]

    MOVE  "ListWORKS-Printer"  TO  W-PDEST
    INVOKE  L-CTRLOBJ  " _SET-E-REPORT-DESTINATION "  USING  W-PDEST  *>.....[3]

    EXIT METHOD.
END METHOD  OPEN-REPORT.
:
END CLASS  USER-HANDLER-CLASS.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

[1]

電子帳票種別として、“ListWORKSの電子帳票”を設定します。

[2]

電子帳票情報ファイル名 "listworks.env" を、電子帳票制御ファイル名として設定します。

[3]

SystemWalker/ListWORKSのプリンタ名 "ListWORKS-Printer" を、電子帳票宛先名として設定します。

### 8.10.7 \_SET-E-REPORT-FILENAMEメソッド(電子帳票ファイル名プロパティ設定メソッド)

### 8.10.8 \_GET-E-REPORT-FILENAMEメソッド(電子帳票ファイル名プロパティ参照メソッド)

#### 説明

作成する電子帳票ファイル名を設定/参照します。電子帳票種別がPDF形式の電子帳票の出力を行うには、当設定メソッドで電子帳票ファイル名を設定してください。

電子帳票ファイル名は、具体的にはSystemWalker/e-DocGenerator連携で作成するPDF形式の電子帳票ファイル名を設定します。

#### 記述形式

##### 【設定】

```
INVOKE CONTROL-OBJECT-3 "_SET-E-REPORT-FILENAME"
      USING [BY REFERENCE] PDF-FILENAME
```

##### 【参照】

```
INVOKE CONTROL-OBJECT-3 "_GET-E-REPORT-FILENAME"
      USING [BY REFERENCE] PDF-FILENAME
```

#### 引数 / 復帰値

**PDF-FILENAME** 【引数の属性: PIC X ANY LENGTH】:

設定時には、電子帳票ファイル名を渡します。引数には任意の長さの英数字項目を記述します。動作モードがシフトJISのときに最大で 259文字(PIC X(518))まで設定することができます。ただし、実際に有効となる文字列の長さはシステムの制限に依存します。

参照時には、現在設定されている電子帳票ファイル名が返ります。引数には任意の長さの英数字項目を記述します。返却される文字列の最大文字数は、動作モードがシフトJISのときに259文字です。

#### 注意事項

このプロパティは、帳票開設イベントより後に変更した内容は無効になります。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER-CLASS  INHERITS  FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS  FJPRT-EVENT-HANDLER
    CLASS  FJPRT-CONTROL
.
SPECIAL-NAMES.
    SYMBOLIC CONSTANT
    COPY  FJPRTSC.
.
:
IDENTIFICATION DIVISION.
METHOD-ID.  OPEN-REPORT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01  W-ETYPE  PIC  S9(4) COMP-5.
    01  W-CFILE  PIC  X(518).
    01  W-PFILE  PIC  X(518).
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.

    MOVE  "FORM1"  TO  FORM-NAME  OF  L-CTRLOBJ

    MOVE  FJPRT-SC-EREPTYP-PDF  TO  W-ETYPE
    INVOKE  L-CTRLOBJ  " _SET-E-REPORT-TYPE"  USING  W-ETYPE  *>.....[1]

    MOVE  "generator.env"  TO  W-CFILE
    INVOKE  L-CTRLOBJ  " _SET-E-REPORT-CONTROL-FILENAME"  USING  W-CFILE  *>...[2]

    MOVE  "reportform.pdf"  TO  W-PFILE
    INVOKE  L-CTRLOBJ  " _SET-E-REPORT-FILENAME"  USING  W-PFILE  *>.....[3]

    EXIT METHOD.
END METHOD  OPEN-REPORT.
:
END CLASS  USER-HANDLER-CLASS.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

[1]

電子帳票種別として、“PDF形式”を設定します。

[2]

文書情報ファイル名 "generator.env" を、電子帳票制御ファイル名として設定します。

[3]

作成するPDF形式の電子帳票ファイル名 "reportform.pdf" を、電子帳票ファイル名として設定します。

## 8.11 印刷制御のプロパティに関するメソッド

ここでは、[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)が提供する以下のカテゴリのメソッドについての説明をします。

印刷制御のプロパティに関するメソッド電子帳票

[\\_SET-PRINT-PREVIEWメソッド\(印刷プレビュープロパティ設定\)](#)

[\\_GET-PRINT-PREVIEWメソッド\(印刷プレビュープロパティ参照\)](#)

以下に、本文の引数/復帰値の項目を読む上での注意事項を挙げます。



**注意**

引数/復帰値の項目に書かれている“属性”は、メソッドの定義上のインタフェースを表すものです。

### 8.11.1 \_SET-PRINT-PREVIEWメソッド(印刷プレビュープロパティ設定メソッド)

### 8.11.2 \_GET-PRINT-PREVIEWメソッド(印刷プレビュープロパティ参照メソッド)

#### 説明

印刷プレビューウィンドウを表示するか否かを設定/参照します。

印刷プレビューとは、印刷処理を行った結果を、プリンタへの印刷を行う代わりに、プレビューウィンドウ内に表示するものです。このため、印刷プレビューをする場合にはプリンタへの印刷は行われません。ただし、印刷プレビューウィンドウのメニュー上からプリンタへの印刷は可能となります。以下に印刷プレビューウィンドウの表示例を挙げます。印刷プレビューウィンドウの操作方法や詳細については“MeFt説明書”を参照してください。

段組み集計表形式 コントロールプレーク印刷

Print(P)

◀

Page

1

/ 6

▶

▶

🔍

🔍

Page Whole

▼

Help(H)

Close(C)

部門別売上実績 - 平成 11 年 10 月

(子四)

部	課	売上高		利益
		目標	実績	一人当
第一課 実績	第一課	1,000.0		110.0
		1,200.0	20.0%	22.0
第二課 実績	第二課	1,000.0		110.0
		1,200.0	20.0%	22.0
第三課 実績	第三課	1,000.0		110.0
		1,200.0	20.0%	22.0

(子四)

部	課	売上高		利益
		目標	実績	一人当
第二課 実績	第一課	1,000.0		110.0
		1,200.0	20.0%	22.0
第二課 実績	第二課	1,000.0		110.0
		1,200.0	20.0%	22.0
第二課 実績	第三課	1,000.0		110.0
		1,200.0	20.0%	22.0

(子四)

部	課	売上高		利益
		目標	実績	一人当
第三課 実績	第一課	1,000.0		110.0
		1,200.0	20.0%	22.0
第三課 実績	第二課	1,000.0		110.0
		1,200.0	20.0%	22.0

部合計	3,000.0	330.0
	3,160.0	20.0%
全部合計	3,000.0	330.0
	10,000.0	20.0%

部合計	3,000.0	330.0
	3,160.0	20.0%

部合計	3,000.0	330.0
	3,160.0	20.0%

図：印刷プレビューウィンドウの表示例

印刷プレビューウィンドウの表示は、印刷処理が完了したタイミングで、帳票印刷メソッド (PRINT-CONTROL) から復帰する直前に行われます。このとき、オペレータが印刷プレビューウィンドウを閉じてから帳票印刷メソッド (PRINT-CONTROL) から復帰するか、閉じる前に復帰するかを選択できます。

前者は、一つのプロセス上で印刷プレビューウィンドウの処理を行いますが、後者は新しくプロセスを一つ生成し、そのプロセス上で印刷プレビューウィンドウの処理がなされることとなります。また後者は、印刷プレビューウィンドウを閉じずにいれば、次回以降の印刷プレビューウィンドウと同時に表示が可能となります。

印刷プレビューを行うには、当プロパティに以下に示すプレビュー種別を設定してください。プレビュー種別を“指定なし”にした場合は、印刷プレビューウィンドウの表示は行われず、プリンタへの印刷がなされます。

デフォルト値として“指定なし”を持ちます。

記号定数	プレビュー種別
FJPRT-SC-PVIEW-OFF	指定なし。印刷プレビューを行わない。
FJPRT-SC-PVIEW-WAIT	プレビュー表示あり(同期待ち合わせあり)。 印刷プレビューウィンドウの処理完了まで、帳票印刷メソッド (PRINT-CONTROL) 内で同期待ち合わせを行います。
FJPRT-SC-PVIEW-THROW	プレビュー表示あり(同期待ち合わせなし)。 印刷プレビューウィンドウの処理完了を待たずに帳票印刷メソッド (PRINT-CONTROL) から復帰します。

## 記述形式

【設定】

INVOKE [CONTROL-OBJECT-3](#) "\_SET-PRINT-PREVIEW"

USING [BY REFERENCE] *E-PREVIEW-TYPE*

【参照】

INVOKE [CONTROL-OBJECT-3](#) "\_GET-PRINT-PREVIEW "  
RETURNING *E-PREVIEW-TYPE*

## 引数 / 復帰値

*E-PREVIEW-TYPE* 【引数/復帰値の属性: PIC S9(4) [USAGE IS] COMP-5】:

設定時には、プレビュー種別を渡します。引数の属性と同じ属性を持つデータ項目か定数を記述します。

参照時には、現在設定されているプレビュー種別が返ります。引数の属性と同じ属性を持つデータ項目を記述します。

## 例外エラー

設定で、以下の例外エラーが発生することがあります。

[FJPRT-SCE-SET-VALUE](#)

## 注意事項

このプロパティは、帳票開設イベントより後に変更した内容は無効になります。

電子帳票の出力を選択した場合には、当プロパティの設定が無視され、電子帳票の出力の設定が有効となります。

印刷プレビューが有効なときは、以下のプロパティに影響があります。詳細については以下の“注意事項”を参照してください。

[PAGE-RANGEプロパティ\(ページ範囲プロパティ\)](#)

[START-PAGE-NUMBERプロパティ\(開始ページ番号プロパティ\)](#)

[END-PAGE-NUMBERプロパティ\(終了ページ番号プロパティ\)](#)





---

## 第9章 帳票印刷イベント処理クラスのメソッド

---

[帳票印刷イベント処理クラス\(FJPRT-EVENT-HANDLER\)](#)には以下のカテゴリのメソッドがあります。  
[イベント処理メソッド](#)

---

## 9.1 イベント処理メソッド

ここでは、[帳票印刷イベント処理クラス\(FJPRT-EVENT-HANDLER\)](#)が提供する以下のカテゴリのメソッドについての説明をします。

イベント処理メソッド

[OPEN-REPORTメソッド\(帳票開設イベント処理メソッド\)](#)  
[CLOSE-REPORTメソッド\(帳票閉鎖イベント処理メソッド\)](#)  
[OPEN-FORMメソッド\(帳票定義体開設イベント処理メソッド\)](#)  
[CLOSE-FORMメソッド\(帳票定義体閉鎖イベント処理メソッド\)](#)  
[READ-RECORDメソッド\(レコード読み込みイベント処理メソッド\)](#)  
[CALCULATE-DEメソッド\(明細計算イベント処理メソッド\)](#)  
[PRINT-DEメソッド\(明細出力イベント処理メソッド\)](#)  
[PAGE-HEADメソッド\(ページ頭書きイベント処理メソッド\)](#)  
[PAGE-FOOTメソッド\(ページ脚書きイベント処理メソッド\)](#)  
[CB-HEADメソッド\(制御頭書きイベント処理メソッド\)](#)  
[CB-FOOTメソッド\(制御脚書きイベント処理メソッド\)](#)  
[REPORT-HEADメソッド\(報告書頭書きイベント処理メソッド\)](#)  
[REPORT-FOOTメソッド\(報告書脚書きイベント処理メソッド\)](#)  
[ERROR-MESSAGEメソッド\(エラーメッセージイベント処理メソッド\)](#)  
[CANCEL-PRINTメソッド\(印刷キャンセルイベント処理メソッド\)](#)



注意

イベント処理メソッドの中では、STOP RUN文のようにプログラムの実行を中断することを行わないように注意してください。

### 9.1.1 OPEN-REPORTメソッド(帳票開設イベント処理メソッド)

#### イベント発生条件

[帳票印刷メソッド\(PRINT-CONTROL\)](#)実行中で、同メソッドが呼ばれた直後。

#### イベント処理後のクラスライブラリの動作

帳票印刷のための初期化処理および帳票定義体の開設を行います。

#### 記述形式

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
  IDENTIFICATION DIVISION.
  METHOD-ID.  OPEN-REPORT  OVERRIDE.
  DATA DIVISION.
  WORKING-STORAGE SECTION.
  LINKAGE SECTION.
    01 L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
  PROCEDURE DIVISION  USING  L-CTRLOBJ.

*>  L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

      EXIT METHOD.
  END METHOD  OPEN-REPORT.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

## 引数

**L-CTRLOBJ** 【引数の属性: [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL】:

[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

## 復帰値

なし。

## 注意事項

このイベントは、[帳票印刷メソッド\(PRINT-CONTROL\)](#)実行中の同メソッドの呼出し直後に 1 回だけ発生します。ただし、[印刷種別プロパティ\(PRINT-TYPE\)](#)で“通常印刷 2 回読み”または“サマリ印刷 2 回読み”を設定した場合は、2 回発生します。

[印刷種別プロパティ\(PRINT-TYPE\)](#)に“通常印刷 2 回読み”または“サマリ印刷 2 回読み”が設定されている場合、1 回目に帳票開設イベントから帳票閉鎖イベントまでひとつのイベントが発生したあと、再び、帳票開設イベントから帳票閉鎖イベントまでのひとつのイベントが発生します。

また、2 回目の帳票開設イベントでは、COBOL ファイルやデータベースなどの読み込み対象レコード位置を初回の読み込み時と同じ位置にする処理を行うように注意してください。

## 使用例

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID. USING-FILE INHERITS FJPRT-EVENT-HANDLER.
:
IDENTIFICATION DIVISION.
OBJECT.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT INPUT-FILE1 ASSIGN TO SYS006
    ORGANIZATION IS LINE SEQUENTIAL.
:
IDENTIFICATION DIVISION.
METHOD-ID. OPEN-REPORT OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-CTRLOBJ OBJECT REFERENCE FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    MOVE "ADDRLIST" TO FORM-NAME OF L-CTRLOBJ *>...[1]
    OPEN INPUT INPUT-FILE1 *>.....[2]
    EXIT METHOD.
END METHOD OPEN-REPORT.
:
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

[例の説明]

[1]

印刷で使用する帳票定義体の名前を指定します。

[2]

以降に発生するイベントで行順ファイルのレコード読み込みを行うために、ファイルをオープンしておきます。なお、このファイルのレコードを読み込む処理は通常は[レコード読](#)

[込みイベント \(READ-RECORDメソッド\)](#)で行うように設計します。またファイルのクローズは通常は[帳票閉鎖イベント \(CLOSE-REPORTメソッド\)](#)で行うように設計します。

### 9.1.2 CLOSE-REPORTメソッド(帳票閉鎖イベント処理メソッド)

#### イベント発生条件

[帳票印刷メソッド \(PRINT-CONTROL\)](#)が終了する直前。

帳票定義体閉鎖イベント発生のおと。

#### イベント処理後のクラスライブラリの動作

帳票印刷の後処理を行います。

#### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  CLOSE-REPORT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.

*>  L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  CLOSE-REPORT.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

#### 引数

**L-CTRLOBJ** 【引数の属性: [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL】:

[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

#### 復帰値

なし。

#### 注意事項

このイベントは、[帳票印刷メソッド \(PRINT-CONTROL\)](#)の実行終了直前に1回だけ発生します。ただし、帳票開設イベントで、[印刷種別プロパティ \(PRINT-TYPE\)](#)に“通常印刷 2回読み”または“サマリ印刷 2回読み”を設定した場合は、2回発生します。

[印刷種別プロパティ \(PRINT-TYPE\)](#)に“通常印刷 2回読み”または“サマリ印刷 2回読み”が設定されているときは、1回目に帳票開設イベントから帳票閉鎖イベントまでのイベントがひととおり発生したあと、再び、帳票開設イベントから帳票閉鎖イベントまでのひととおりのイベントが発生します。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID. USING-FILE INHERITS FJPRT-EVENT-HANDLER.
:
IDENTIFICATION DIVISION.
OBJECT.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT INPUT-FILE1 ASSIGN TO SYS006
    ORGANIZATION IS LINE SEQUENTIAL.
:
IDENTIFICATION DIVISION.
METHOD-ID. CLOSE-REPORT OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-CTRLOBJ OBJECT REFERENCE FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    CLOSE INPUT-FILE1 *>.....[1]
    EXIT METHOD.
END METHOD CLOSE-REPORT.
:
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

[例の説明]

[1]

ほかのイベントで使われていた行順ファイルをクローズします。

帳票の印刷で使うファイルは通常は[帳票開設イベント \(OPEN-REPORTメソッド\)](#)でオープンするように設計します。また、ファイルのレコードを読み込む処理は通常は[レコード読み込みイベント \(READ-RECORDメソッド\)](#)で行うように設計します。

### 9.1.3 ERROR-MESSAGEメソッド(エラーメッセージイベント処理メソッド)

#### イベント発生条件

[帳票印刷メソッド \(PRINT-CONTROL\)](#) 実行中に[印刷エラー](#)が発生したとき。

#### イベント処理後のクラスライブラリの動作

ありません。

## 記述形式

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  ERROR-MESSAGE  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.

*>   L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

      EXIT METHOD.
      END METHOD  ERROR-MESSAGE.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## 引数

**L-CTRLOBJ** 【引数の属性: [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL】:

[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

## 復帰値

なし。

## 注意事項

発生した[印刷エラー](#)に対応するエラーコードは[印刷ジョブエラーコードプロパティ \(PRINT-JOB-ERROR-CODE\)](#)で参照できます。必要に応じてこの値を使ってエラーメッセージの表示などの処理を記述してください。

印刷エラーの発生によって印刷の続行を中断したい場合は、[キャンセル要求プロパティ \(REQUEST-CANCEL\)](#)に真(B"1")を設定してください。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
:
IDENTIFICATION DIVISION.
METHOD-ID.  ERROR-MESSAGE  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.
    DISPLAY  NC"エラーメッセージイベントが発生しました。 "
    DISPLAY  PRINT-JOB-ERROR-CODE OF L-CTRLOBJ  *>.....[1]
    MOVE  B"1" TO  REQUEST-CANCEL OF L-CTRLOBJ  *>.....[2]
    EXIT METHOD.
    END METHOD  ERROR-MESSAGE.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

[例の説明]

[1]

印刷エラーの種別を示す印刷ジョブエラーコードを表示します。

[2]

印刷をキャンセルします。

### 9.1.4 CANCEL-PRINTメソッド(印刷キャンセルイベント処理メソッド)

#### イベント発生条件

印刷中のダイアログボックスから、オペレータが印刷中止を指示したとき。

あるイベント処理メソッド内で[キャンセル要求プロパティ \(REQUEST-CANCEL\)](#)に真("1")

を設定してそのイベント処理メソッドを抜けた後。

#### イベント処理後のクラスライブラリの動作

印刷途中のデータをキャンセルします。

#### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  CANCEL-PRINT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.

*>  L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  CANCEL-PRINT.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

#### 引数

**L-CTRLOBJ** 【引数の属性: [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL】:

[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

#### 復帰値

なし。

#### 注意事項

なし。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
:
IDENTIFICATION DIVISION.
METHOD-ID.  CANCEL-PRINT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    DISPLAY NC"印刷キャンセルイベントが発生しました。"  *>...[1]
    EXIT METHOD.
END METHOD  CANCEL-PRINT.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

[例の説明]

[1]

印刷キャンセルイベントが発生したことを画面に表示し、帳票の印刷がキャンセルされたことをオペレータに通知します。

### 9.1.5 READ-RECORDメソッド(レコード読み込みイベント処理メソッド)

## イベント発生条件

帳票定義体開設イベント発生の次  
各明細/ボディ部/ページの印刷の繰返しごと

## イベント処理後のクラスライブラリの動作

[レコードEOFプロパティ \(RECORD-EOF\)](#) に真(B"1")を設定した場合は、制御脚書き、ページ脚書き、報告書脚書きを印刷し(このとき、各パーティションに対応するイベントも発生します)、続いて帳票定義体閉鎖を行います。

## 記述形式

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  READ-RECORD  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.

*>  L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  READ-RECORD.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## 引数

**L-CTRLOBJ** 【引数の属性: [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL】:



[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

**復帰値**

なし。

**注意事項**

このイベントは、レコード/ファイルの終わり (EOF) をプログラムが検出したことを示すために [レコードEOFプロパティ \(RECORD-EOF\)](#) に真 (B"1") を設定するまで、発生を繰り返します。EOF が通知されないと、[帳票印刷メソッド \(PRINT-CONTROL\)](#) 内部で **無限ループしますの**で必ず終了条件を設けて EOF を通知するように注意してください。

[レコード回数プロパティ \(RECORD-COUNT\)](#) を利用することで、レコードの読み込み回数を判断することができます。レコード回数プロパティ (RECORD-COUNT) は、初回のレコード読み込みイベント発生時には、1 が設定されています。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID. USING-FILE INHERITS FJPRT-EVENT-HANDLER.
:
IDENTIFICATION DIVISION.
OBJECT.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT INPUT-FILE1 ASSIGN TO SYS006
    ORGANIZATION IS LINE SEQUENTIAL.
DATA DIVISION.
FILE SECTION.
FD INPUT-FILE1.
01.
    02 F-NAME1      PIC X(28).
    02 FILLER       PIC X(1).
    02 F-AGE1       PIC 9(3).
    02 FILLER       PIC X(1).
    02 F-ADDRESS1   PIC X(45).
WORKING-STORAGE SECTION.
    COPY ADDRLIST OF XMDLIB.
01 W-ADDAGE        PIC 9(5).
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID. READ-RECORD OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-CTRLOBJ OBJECT REFERENCE FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    READ INPUT-FILE1 AT END *>.....[1]
        MOVE B"1" TO RECORD-EOF OF SELF
    NOT AT END *>.....[2]
        INVOKE L-CTRLOBJ "GET-RECORD-CONTENT" USING ADDRLIST
        MOVE F-NAME1      TO NAME1      OF ADDRLIST
        MOVE F-AGE1       TO AGE1       OF ADDRLIST
        MOVE F-ADDRESS1   TO ADDRESS1 OF ADDRLIST
        INVOKE L-CTRLOBJ "SET-RECORD-CONTENT" USING ADDRLIST
    END-READ
    EXIT METHOD.
END METHOD READ-RECORD.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

[例の説明]

[1]

ほかのイベントで既にオープンされている行順ファイルの 1レコードを読みます。読み込みでファイル終了条件が発生したら[レコードEOFプロパティ \(RECORD-EOF\)](#)を真にしてレコード読み込みが終了したことを帳票印刷クラスライブラリに通知します。

なお、帳票の印刷で使うファイルは通常は[帳票開設イベント](#) ([OPEN-REPORTメソッド](#)) でオープンするに設計します。また、ファイルのクローズは通常は[帳票閉鎖イベント](#) ([CLOSE-REPORTメソッド](#)) で行うように設計します。

[2]

正常にレコードが読み込めたら、読んだレコードの内容を帳票印刷クラスライブラリが管理しているレコードに転記します。当メソッドの終了後、別のイベント処理 ([明細出力イベント](#)) 後にこのレコードの内容がページ/明細/ボディ部のいずれとして (これは帳票定義体の種別に依存します) 印刷されます。

### 9.1.6 CALCULATE-DEメソッド(明細計算イベント処理メソッド)

#### イベント発生条件

レコードが読み込まれるたびに、その後に発生します。

#### イベント処理後のクラスライブラリの動作

ありません。

#### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  CALCULATE-DE  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.

*>   L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

      EXIT METHOD.
      END METHOD  CALCULATE-DE.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

#### 引数

**L-CTRLOBJ** 【引数の属性: [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL】:

[印刷制御オブジェクト](#) が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

#### 復帰値

なし。

#### 注意事項

なし。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER  INHERITS  FJPRT-EVENT-HANDLER.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  ADDRLIST OF XMDLIB.
    COPY  ADDRLIST OF XMDLIB
        REPLACING  ADDRLIST  BY  ADDRLIST-PRE.
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID.  CALCULATE-DE  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.
    INVOKE  L-CTRLOBJ  "GET-RECORD-CONTENT"  USING  ADDRLIST  *>.....[1]
    INVOKE  L-CTRLOBJ  "GET-PRE-RECORD-CONTENT"  USING  ADDRLIST-PRE  *>...[2]
    MOVE  NAME1  OF ADDRLIST-PRE  TO  NAME2  OF ADDRLIST  *>.....[3]
    MOVE  ADDRESS1 OF ADDRLIST-PRE  TO  ADDRESS2 OF ADDRLIST  *>.....[4]
    INVOKE  L-CTRLOBJ  "SET-RECORD-CONTENT"  USING  ADDRLIST  *>.....[5]
    EXIT METHOD.
END METHOD  CALCULATE-DE.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

- [1]  
レコードの内容を参照します。
- [2]  
前レコードの内容を参照します。
- [3]  
前レコードの NAME1 項目の値を NAME2 に転記します。
- [4]  
前レコードの ADDRESS1 項目の値を ADDRESS2 に転記します。
- [5]  
新たに転記された項目の値をレコードに反映します。

## 9.1.7 PRINT-DEメソッド(明細出力イベント処理メソッド)

### イベント発生条件

明細/ボディ部/ページの印刷の直前に発生します。

### イベント処理後のクラスライブラリの動作

明細/ボディ部/ページが存在し印刷が抑止されていなければ、明細/ボディ部/ページを印刷します。

## 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  PRINT-DE  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.

*>   L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

      EXIT METHOD.
      END METHOD  PRINT-DE.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
```

## 引数

***L-CTRLOBJ*** 【引数の属性: [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL】:

[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

## 復帰値

なし。

## 注意事項

印刷する明細の切替えは、複数の明細が定義された集計表形式の帳票定義体を利用する場合に有効となります。

明細の切替えについては“ [明細名プロパティ \(DE-NAME\)](#) ”を参照してください。

## 使用例

```
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID. COLOR-CHANGE INHERITS FJPRT-EVENT-HANDLER.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
    SYMBOLIC CONSTANT
    COPY FJPRTSC.
.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY FORM1 OF XMDLIB.
    01 SEIBETSU1-NAME PIC X(9) VALUE "SEIBETSU1".
    01 ACCUMULATOR1 PIC 9(10).
PROCEDURE DIVISION.
.
IDENTIFICATION DIVISION.
METHOD-ID. PRINT-DE OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01 W-COLOR PIC S9(4) COMP-5.
LINKAGE SECTION.
    01 L-CTRLOBJ OBJECT REFERENCE FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    INVOKE L-CTRLOBJ "GET-RECORD-CONTENT"
        USING FORM1 *>.....[1]
    EVALUATE SEIBETSU1 OF FORM1 *>.....[2]
    WHEN NC"男"
        MOVE FJPRT-SC-COLOR-BLUE TO W-COLOR
    WHEN NC"女"
        MOVE FJPRT-SC-COLOR-RED TO W-COLOR
    WHEN OTHER *> 性別が未入力の場合
        MOVE FJPRT-SC-COLOR-BLACK TO W-COLOR
    END-EVALUATE
    INVOKE L-CTRLOBJ "SET-TEXT-COLOR"
        USING SEIBETSU1-NAME W-COLOR *>....[3]
    ADD VALUE1 OF FORM1 TO ACCUMULATOR1 *>....[4]
    EXIT METHOD.
END METHOD PRINT-DE.
.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
```

### [例の説明]

[1]

レコードの内容を参照します。

[2]

SEIBETSU1 項目の値を調べ、色に対応する数値を変数に設定します。

[3]

SEIBETSU1 項目の色を設定します。

[4]

フッタに関するほかのイベントで使われる変数への加算を行います。なお、この変数の初期化はヘッダに関するイベントで行うように設計します。

### 9.1.8 OPEN-FORMメソッド(帳票定義体開設イベント処理メソッド)

#### イベント発生条件

帳票定義体を開いたとき。

帳票開設イベントのあと。

#### イベント処理後のクラスライブラリの動作

ありません。

#### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID. OPEN-FORM OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-CTRLOBJ    OBJECT REFERENCE FJPRT-CONTROL.
    01 L-FORMNAME  PIC X(518).
PROCEDURE DIVISION USING L-CTRLOBJ L-FORMNAME.

*>  L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD OPEN-FORM.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

#### 引数

**L-CTRLOBJ** 【引数の属性: [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL】:

[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

**L-FORMNAME** 【引数の属性: PIC X(518)】:

イベントの処理対象となった帳票定義体名が渡ってきます。引数となるデータ項目は引数の属性で示した属性で宣言してください。データ項目の名前は任意に付けてください。当イベント処理中に[帳票定義体名プロパティ \(FORM-NAME\)](#)で印刷する帳票定義体を切り替えた後でも、イベントの処理対象となった帳票定義体を参照することができます。

#### 復帰値

なし。

#### 注意事項

各イベントでも同様の処理を記述できますが、帳票印刷の起動時にだけ処理をすればよい項目に関しては、このイベントで記述することによって性能を向上させることができます。ページ形式やタックシール形式の帳票定義体の印刷処理のための初期化に有効です。

帳票定義体名を変更した場合、帳票定義体開設イベントと帳票定義体閉鎖イベントが発生する場

合があります。

複数の帳票定義体を利用する場合のイベント発生の流れについては、“[複数の帳票定義体を利用する場合のイベント発生の流れ](#)”を参照してください。

### 9.1.9 CLOSE-FORMメソッド(帳票定義体閉鎖イベント処理メソッド)

#### イベント発生条件

帳票定義体が閉じるとき。

帳票閉鎖イベントの前。

#### イベント処理後のクラスライブラリの動作

ありません。

#### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID. CLOSE-FORM OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-CTRLOBJ    OBJECT REFERENCE FJPRT-CONTROL.
    01 L-FORMNAME  PIC X(518).
PROCEDURE DIVISION    USING L-CTRLOBJ L-FORMNAME.

*>  L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD    CLOSE-FORM.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

#### 引数

**L-CTRLOBJ** 【引数の属性: [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL】:

[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

**L-FORMNAME** 【引数の属性: PIC X(518)】:

イベントの処理対象の帳票定義体名が渡ってきます。引数となるデータ項目は引数の属性で示した属性で宣言してください。データ項目の名前は任意に付けてください。帳票定義体を[帳票定義体名プロパティ \(FORM-NAME\)](#)で設定し直した後に発生した当イベントでは、当引数には切り替え前の帳票定義体の名前が格納されています。しかし、このとき帳票定義体名プロパティ (FORM-NAME)の参照値は、まだ当イベントの処理対象となっていない切り替え後の帳票定義体の名前が返却されます。

#### 復帰値

なし。

#### 注意事項

当イベントで帳票の項目への設定を行っても、以前の設定値で帳票印刷が行われます。

ただし、[印刷種別プロパティ \(PRINT-TYPE\)](#)で、“通常印刷2回読み”または“サマリ印刷2回読み”を指定した場合に限り、帳票の項目に設定した結果が1ページ目に印刷されます。

帳票定義体を切り替えた場合、帳票定義体開設イベントと帳票定義体閉鎖イベントが発生する場



合があります。

複数の帳票定義体を利用する場合のイベント発生の流れについては、“[複数の帳票定義体を利用する場合のイベント発生の流れ](#)”を参照してください。

### 9.1.10 CB-HEADメソッド(制御頭書きイベント処理メソッド)

#### イベント発生条件

以下のいずれかの条件によって[コントロールブレーク](#)が発生したとき。

あるブレーク項目の値が変化して、対応する[制御脚書きイベント](#)が発生したあと。

最初の[レコード読み込みイベント](#)が発生したあと。

#### イベント処理後のクラスライブラリの動作

制御頭書きパーティションが存在し印刷が抑止されていなければ、制御頭書きを印刷します。

#### 記述形式

```
*>.1.*.:...2....3.....4.....5.....6.....7.*.:...8
IDENTIFICATION DIVISION.
METHOD-ID.  CB-HEAD  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ    OBJECT REFERENCE  FJPRT-CONTROL.
    01  L-FIELDNAME  PIC  X(60).
PROCEDURE DIVISION  USING  L-CTRLOBJ  L-FIELDNAME.

*>   L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  CB-HEAD.
*>.1.*.:...2....3.....4.....5.....6.....7.*.:...8
```

#### 引数

**L-CTRLOBJ** 【引数の属性: [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL】:

[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

**L-FIELDNAME** 【引数の属性: PIC X(60)】:

制御ブレーク項目名が渡ってきます。これは、当イベントの発生対象となったコントロールブレークが発生した項目の名前を持ちます。引数のデータ項目の宣言ではデータ項目の名前を任意に付けてください。

当引数は、動作モードがUnicodeで、制御ブレーク項目名が日本語のとき、最大で20文字までしか受け取ることができませんので、注意してください。[参照] “[Unicode固有の留意事項](#)”

#### 復帰値

なし。

#### 注意事項

このイベントは、集計表形式の帳票定義体でだけ発生します。

ブレーク項目が定義されていない制御頭書きパーティションに対しても当イベントが発生します。この場合は、制御ブレーク項目名引数(L-FIELDNAME)には空白が渡されます。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  CB-USING  INHERITS  FJPRT-EVENT-HANDLER.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  FORM2 OF XMDLIB.
    01  KA-GOUKEI  PIC  9(4).
    01  BU-GOUKEI  PIC  9(4).
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID.  CB-HEAD  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ   OBJECT REFERENCE FJPRT-CONTROL.
    01  L-FIELDNAME PIC  X(60).
PROCEDURE DIVISION USING  L-CTRLOBJ  L-FIELDNAME.
    INVOKE  L-CTRLOBJ  "GET-RECORD-CONTENT"
        USING  FORM2  *>.....[1]
    EVALUATE  L-FIELDNAME  *>.....[2]
    WHEN      "部名"
        MOVE  ZERO TO BU-GOUKEI  *>.....[3]
        MOVE  部名 OF FORM2 TO DIV-NAME OF FORM2  *>...[4]
    WHEN      "課名"
        MOVE  ZERO TO KA-GOUKEI  *>.....[3']
        MOVE  課名 OF FORM2 TO SEC-NAME OF FORM2  *>...[4']
    END-EVALUATE
    INVOKE  L-CTRLOBJ  "SET-RECORD-CONTENT"
        USING  FORM2  *>.....[5]
    EXIT METHOD.
END METHOD  CB-HEAD.
:
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

## [例の説明]

[1]

レコードの内容を参照します。

[2]

制御ブレイク項目名を参照して、処理内容を切りわけます。

ここでの“部名”という項目と“課名”という項目はブレイク項目として帳票定義体に定義しており、帳票印刷の始めと終わりと、これらの項目の値が変化したときに[コントロールブレイク](#)が発生してこのイベント処理メソッドに制御が渡ってきます。

[3],[3']

[制御脚書きイベント](#)で使う合計の値を初期化します。このような場合、[明細出力イベント](#)で合計の元となる値を加算するように設計します。

[4],[4']

ブレイク項目に対応する各パーティションの項目に文字を設定します。

[5]

変更された項目の内容をレコードに反映します。

### 9.1.11 CB-FOOTメソッド(制御脚書きイベント処理メソッド)

#### イベント発生条件

以下のいずれかの条件によって[コントロールブレイク](#)が発生したとき。

あるブレイク項目の値が変化したあと。

[レコード読み込みイベント](#)で[レコードEOFプロパティ \(RECORD-EOF\)](#)に真を設定したあと。

#### イベント処理後のクラスライブラリの動作

制御脚書きパーティションが存在し印刷が抑止されていなければ、制御脚書きを印刷します。

#### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  CB-FOOT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ    OBJECT REFERENCE  FJPRT-CONTROL.
    01  L-FIELDNAME  PIC  X(60).
PROCEDURE DIVISION  USING  L-CTRLOBJ  L-FIELDNAME.

*>  L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  CB-FOOT.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
```

#### 引数

**L-CTRLOBJ** 【引数の属性: [USAGE IS] OBJECT REFERENCE [FJPRT-CONTROL](#)】:

[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

**L-FIELDNAME** 【引数の属性: PIC X(60)】:

制御ブレイク項目名が渡ってきます。これは、当イベントの発生対象となったコントロールブレイクが発生した項目の名前を持ちます。引数のデータ項目の宣言ではデータ項目の名前を任意に付けてください。

当引数は、動作モードがUnicodeで、制御ブレイク項目名が日本語のとき、最大で20文字までしか受け取ることができませんので、注意してください。[参照] “[Unicode固有の留意事項](#)”

#### 復帰値

なし。

#### 注意事項

このイベントは、集計表形式の帳票定義体でだけ発生します。

ブレイク項目が定義されていない制御脚書きパーティションに対しても当イベントが発生します。この場合は、制御ブレイク項目名引数(L-FIELDNAME)には空白が渡されます。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  CB-USING  INHERITS  FJPRT-EVENT-HANDLER.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  FORM2 OF XMDLIB.
    01  KA-GOUKEI  PIC  9(4).
    01  BU-GOUKEI  PIC  9(4).
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID.  CB-FOOT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ   OBJECT REFERENCE FJPRT-CONTROL.
    01  L-FIELDNAME PIC  X(60).
PROCEDURE DIVISION USING  L-CTRLOBJ  L-FIELDNAME.
    INVOKE  L-CTRLOBJ  "GET-RECORD-CONTENT"
        USING  FORM2  *>.....[1]
    EVALUATE  L-FIELDNAME  *>.....[2]
    WHEN      "部名"
        MOVE  BU-GOUKEI  TO  BU-GOUKEI1 OF FORM2  *>...[3]
    WHEN      "課名"
        MOVE  KA-GOUKEI  TO  KA-GOUKEI1 OF FORM2  *>...[3']
    END-EVALUATE
    INVOKE  L-CTRLOBJ  "SET-RECORD-CONTENT"
        USING  FORM2  *>.....[4]
    EXIT METHOD.
END METHOD  CB-FOOT.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

[1]

レコードの内容を参照します。

[2]

制御ブレイク項目名を参照して、処理内容を切りわけます。

ここでの“部名”という項目と“課名”という項目はブレイク項目として帳票定義体に定義しており、帳票印刷の始めと終わりと、これらの項目の値が変化したときに[コントロールブレイク](#)が発生してこのイベント処理メソッドに制御が渡ってきます。

[3],[3']

ほかのイベントで求められた合計を各パーティションの項目に設定します。このような場合、[制御頭書きイベント](#)で合計用の変数を初期化し、[明細出力イベント](#)で合計の元となる値を加算するように設計します。

[4]

変更された項目の内容をレコードに反映します。

### 9.1.12 PAGE-HEADメソッド(ページ頭書きイベント処理メソッド)

#### イベント発生条件

各ページの先頭から印刷が行われようとする前に発生します。

#### イベント処理後のクラスライブラリの動作

ページ頭書きパーティションが存在し印刷が抑止されていなければ、ページ頭書きを印刷します。

#### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID. PAGE-HEAD OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01 L-CTRLOBJ OBJECT REFERENCE FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.

*> L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD PAGE-HEAD.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

#### 引数

**L-CTRLOBJ** 【引数の属性: [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL】:

[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

#### 復帰値

なし。

#### 注意事項

このイベントは、[帳票定義体の種別](#)が段組み伝票形式または集計表形式の場合だけ発生します。

印刷で使用する帳票定義体のページ頭書きパーティションの有無にかかわらず、このイベントは発生します。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER  INHERITS  FJPRT-EVENT-HANDLER.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  FORM3 OF XMDLIB.
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID.  PAGE-HEAD  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    INVOKE  L-CTRLOBJ  "GET-RECORD-CONTENT"  USING  ADDRLIST  *>...[1]
    MOVE    PRINTING-PAGE-NUMBER  OF  L-CTRLOBJ
            TO  PH-PAGE-COUNT OF FORM3  *>.....[2]
    INVOKE  L-CTRLOBJ  "SET-RECORD-CONTENT"  USING  ADDRLIST  *>...[3]
    EXIT METHOD.
END METHOD  PAGE-HEAD.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

- [1]  
レコードの内容を参照します。
- [2]  
ページ頭書き/ヘッダ部パーティションに存在する項目に印刷中ページ番号を設定します。
- [3]  
新たに転記された項目の値をレコードに反映します。

### 9.1.13 PAGE-F00Tメソッド(ページ脚書きイベント処理メソッド)

## イベント発生条件

印刷が各ページの最終行になる直前  
レコード読みイベント処理メソッドで[レコードEOFプロパティ \(RECORD-EOF\)](#)に真(B"1")を設定してイベント処理メソッドを抜けた後。

## イベント処理後のクラスライブラリの動作

ページ脚書きパーティションが存在し印刷が抑止されていなければ、ページ脚書きを印刷します。

## 記述形式

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  PAGE-FOOT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.

*>   L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  PAGE-FOOT.
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## 引数

***L-CTRLOBJ*** 【引数の属性: [USAGE IS] OBJECT REFERENCE [FJPRT-CONTROL](#)】:

[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

## 復帰値

なし。

## 注意事項

このイベントは、[帳票定義体の種別](#)が段組み伝票形式または集計表形式の場合だけ発生します。

印刷で使用する帳票定義体のページ脚書きパーティションの有無にかかわらず、このイベントは発生します。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7..*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER  INHERITS  FJPRT-EVENT-HANDLER.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  FORM3 OF XMDLIB.
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID.  PAGE-FOOT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    INVOKE  L-CTRLOBJ  "GET-RECORD-CONTENT"  USING  ADDRLIST  *>...[1]
    MOVE    PRINTING-PAGE-NUMBER  OF L-CTRLOBJ
            TO  PF-PAGE-COUNT OF FORM3  *>.....[2]
    INVOKE  L-CTRLOBJ  "SET-RECORD-CONTENT"  USING  ADDRLIST  *>...[3]
    EXIT METHOD.
END METHOD  PAGE-FOOT.
:
*>.1.*.....2.....3.....4.....5.....6.....7..*.....8

```

## [例の説明]

- [1]  
レコードの内容を参照します。
- [2]  
ページ脚書き/フッタ部パーティションに存在する項目に印刷中ページ番号を設定します。
- [3]  
新たに転記された項目の値をレコードに反映します。

### 9.1.14 REPORT-HEADメソッド(報告書頭書きイベント処理メソッド)

## イベント発生条件

帳票印刷のため読み込まれた 1件目のレコードを対象として、明細計算イベントが発生した後に発生します。

## イベント処理後のクラスライブラリの動作

報告書頭書きパーティションが存在し印刷が抑止されていなければ、報告書頭書きを印刷します。



## 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  REPORT-HEAD  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.

*>   L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  REPORT-HEAD.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

## 引数

***L-CTRLOBJ*** 【引数の属性: [USAGE IS] OBJECT REFERENCE [FJPRT-CONTROL](#)】:

[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

## 復帰値

なし。

## 注意事項

このイベントは、集計表形式の帳票定義体でだけ発生するイベントです。

報告書頭書きパーティションの有無にかかわらず、このイベントは発生します。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER  INHERITS  FJPRT-EVENT-HANDLER.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  FORM3 OF XMDLIB.
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID.  REPORT-HEAD  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01  W-DATE.
        02  W-YYYY  PIC  9(4).
        02  W-MM    PIC  9(2).
        02  W-DD    PIC  9(2).
        02  FILLER  PIC  X(13).
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    INVOKE L-CTRLOBJ  "GET-RECORD-CONTENT"  USING  ADDRLIST  *>...[1]
    MOVE  FUNCTION CURRENT-DATE  TO  W-DATE  *>.....[2]
    MOVE  W-YYYY  TO  REP-HEAD-YYYY OF FORM3
    MOVE  W-MM    TO  REP-HEAD-MM   OF FORM3
    MOVE  W-DD    TO  REP-HEAD-DD   OF FORM3
    INVOKE L-CTRLOBJ  "SET-RECORD-CONTENT"  USING  ADDRLIST  *>...[3]
    EXIT METHOD.
END METHOD REPORT-HEAD.
:
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

## [例の説明]

- [1]  
レコードの内容を参照します。
- [2]  
報告書頭書きパーティションに存在する項目に年、月、日の値を設定します。
- [3]  
新たに転記された項目の値をレコードに反映します。

### 9.1.15 REPORT-FOOTメソッド(報告書脚書きイベント処理メソッド)

## イベント発生条件

印刷するすべてのレコードが処理されたあとに発生します。

## イベント処理後のクラスライブラリの動作

報告書脚書きパーティションが存在し印刷が抑止されていなければ、報告書脚書きを印刷します。

### 記述形式

```
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
METHOD-ID.  REPORT-FOOT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION  USING  L-CTRLOBJ.

*>  L-CTRLOBJ が示す印刷制御オブジェクトを使ってイベント処理を記述します。

EXIT METHOD.
END METHOD  REPORT-FOOT.
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
```

### 引数

**L-CTRLOBJ** 【引数の属性: [USAGE IS] OBJECT REFERENCE FJPRT-CONTROL】:

[印刷制御オブジェクト](#)が渡ってきます。引数となるオブジェクト参照一意名は引数の属性で示した属性で宣言してください。オブジェクト参照一意名の名前は任意に付けてください。

### 復帰値

なし。

### 注意事項

このイベントは、集計表形式の帳票定義体でだけ発生します。

帳票定義体に報告書脚書きパーティションの有無にかかわらず、このイベントは発生します。

## 使用例

```

*>.1.*.....2.....3.....4.....5.....6.....7.*.....8
IDENTIFICATION DIVISION.
CLASS-ID.  USER-HANDLER  INHERITS  FJPRT-EVENT-HANDLER.
:
IDENTIFICATION DIVISION.
OBJECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY  FORM3 OF XMDLIB.
PROCEDURE DIVISION.
:
IDENTIFICATION DIVISION.
METHOD-ID.  REPORT-FOOT  OVERRIDE.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01  W-TIME.
        02  W-HH  PIC  9(2).
        02  W-MM  PIC  9(2).
        02  W-SS  PIC  9(2).
        02  W-MS  PIC  9(2).
LINKAGE SECTION.
    01  L-CTRLOBJ  OBJECT REFERENCE  FJPRT-CONTROL.
PROCEDURE DIVISION USING L-CTRLOBJ.
    INVOKE  L-CTRLOBJ  "GET-RECORD-CONTENT"  USING  ADDRLIST  *>...[1]
    ACCEPT  W-TIME  FROM TIME  *>.....[2]
    MOVE  W-HH  TO  REP-FOOT-HH OF FORM3
    MOVE  W-MM  TO  REP-FOOT-MM OF FORM3
    MOVE  W-SS  TO  REP-FOOT-SS OF FORM3
    INVOKE  L-CTRLOBJ  "SET-RECORD-CONTENT"  USING  ADDRLIST  *>...[3]
    EXIT METHOD.
END METHOD  REPORT-FOOT.
:
*>.1.*.....2.....3.....4.....5.....6.....7.*.....8

```

## [例の説明]

- [1]  
レコードの内容を参照します。
- [2]  
報告書脚書きパーティションに存在する項目に時、分、秒の値を設定します。
- [3]  
新たに転記された項目の値をレコードに反映します。

---

## 第10章 帳票印刷例外クラスのメソッド

---

[帳票印刷例外クラス\(FJPRT-EXCEPTION\)](#)には以下のカテゴリのメソッドがあります。

[例外エラー情報参照メソッド](#)

---

## 10.1 例外エラー情報参照メソッド

ここでは、[帳票印刷例外クラス\(FJPRT-EXCEPTION\)](#)が提供する以下のカテゴリのメソッドについての説明をします。

例外エラー情報参照メソッド

[GET-CLASS-NAMEメソッド\(クラス名の参照メソッド\)](#)

[GET-METHOD-NAMEメソッド\(メソッド名の参照メソッド\)](#)

[GET-CODEメソッド\(コードの参照メソッド\)](#)

[GET-MESSAGEメソッド\(メッセージの参照メソッド\)](#)

### 10.1.1 GET-CLASS-NAMEメソッド(クラス名の参照メソッド)

#### 説明

[例外エラー](#)が発生したオブジェクトのクラス名を英数字で返します。

#### 記述形式

```
INVOKE EXCEPTION-OBJECT-1 "GET-CLASS-NAME"
      USING      [BY REFERENCE] CLASS-NAME
      [RETURNING CLASS-NAME-LENGTH]
```

【補足】

***EXCEPTION-OBJECT-1*** :

操作の対象となる[帳票印刷例外クラス\(FJPRT-EXCEPTION\)](#)のオブジェクトを持つオブジェクト参照一意名を記述します。

USE文を使った例外オブジェクト発生 of 例外処理の記述箇所では、例外オブジェクトを示す予約語の EXCEPTION-OBJECT を記述します。詳細については“[例外エラー](#)”を参照してください。

#### 引数

***CLASS-NAME*** 【引数の属性: PIC X ANY LENGTH】:

[例外エラー](#)が発生したオブジェクトのクラス名が格納される任意の長さの英数字データ項目を記述します。

#### 復帰値

***CLASS-NAME-LENGTH*** 【復帰値の属性: PIC S9(9) [USAGE IS] COMP-5】:

*CLASS-NAME*引数に格納されたクラス名の長さ(バイト数)が返却されます。復帰値の属性と同じ属性を持つデータ項目を記述してください。

#### 解説

[例外エラー](#)が発生したオブジェクトのクラス名およびそのクラス名の長さを返却します。

#### 使用例

“[プログラム定義で例外処理を記述する場合の例](#)”を参照してください。

### 10.1.2 GET-METHOD-NAMEメソッド(メソッド名の参照メソッド)

#### 使用例

[例外エラー](#)が発生したメソッドの名前を英数字で返します。

#### 記述形式

```
INVOKE EXCEPTION-OBJECT-1 "GET-METHOD-NAME"
      USING      [BY REFERENCE] METHOD-NAME
```

[RETURNING *METHOD-NAME-LENGTH*]**引数*****METHOD-NAME*** 【引数の属性: PIC X ANY LENGTH】:

[例外エラー](#)が発生したメソッドの名前が格納される任意の長さの英数字データ項目を記述します。

**復帰値*****METHOD-NAME-LENGTH*** 【復帰値の属性: PIC S9(9) [USAGE IS] COMP-5】:

*METHOD-NAME*引数に格納されたメソッドの名前の長さ(バイト数)が返却されます。復帰値の属性と同じ属性を持つデータ項目を記述してください。

**解説**

[例外エラー](#)が発生したメソッドの名前およびそのメソッド名の長さを返却します。

**使用例**

[プログラム定義で例外処理を記述する場合の例](#)  
[イベント処理メソッドで例外処理を記述する場合の例](#)

### 10.1.3 GET-CODEメソッド(コードの参照メソッド)

**説明**

発生した[例外エラー](#)の種類を示す[例外エラーコード](#)を返します。

**記述形式**

```
INVOKE EXCEPTION-OBJECT-1 "GET-CODE"
      RETURNING ERROR-CODE
```

**復帰値*****ERROR-CODE*** 【復帰値の属性: PIC S9(9) [USAGE IS] COMP-5】:

[例外エラーコード](#)が返却されます。

**解説**

[例外エラー](#)が発生したときの例外エラーの種類を示す[例外エラーコード](#)が返却されます。

**使用例**

“ [プログラム定義で例外処理を記述する場合の例](#) ” を参照してください。

### 10.1.4 GET-MESSAGEメソッド(メッセージの参照メソッド)

**説明**

[例外エラー](#)を簡単に説明する内容を持つメッセージ(日本語)を英数字の属性で返します。

**記述形式**

```
INVOKE EXCEPTION-OBJECT-1 "GET-MESSAGE"
      USING [BY REFERENCE] MESSAGE
      [RETURNING MESSAGE-LENGTH]
```

**引数*****MESSAGE*** 【引数の属性: PIC X ANY LENGTH】:

メッセージが格納される任意の長さの英数字データ項目を記述します。

**復帰値*****MESSAGE-LENGTH*** 【復帰値の属性: PIC S9(9) [USAGE IS] COMP-5】:

*MESSAGE*引数に格納されたエラーメッセージの長さ(バイト数)が返却されます。復帰値の

属性と同じ属性を持つデータ項目を記述してください。

### 解説

メッセージおよびそのメッセージの長さが返却されます。返却されたメッセージは表示用だけに使用してください。

### 使用例

[プログラム定義で例外処理を記述する場合の例](#)

[イベント処理メソッドで例外処理を記述する場合の例](#)



---

## 第11章 マルチスレッド

---

ここでは帳票印刷クラスライブラリのマルチスレッドに関する説明をします。

[マルチスレッド環境下での動作](#)

[マルチスレッドプログラムの作成](#)

[マルチスレッド固有の留意事項](#)

---

## 11.1 マルチスレッド環境下での動作

帳票印刷クラスライブラリは、他のマルチスレッドプログラムと組み合わせることにより、マルチスレッド環境下で動作可能です。



注意

SystemWalker/e-DocGeneratorはマルチスレッドに対応していません。このため、当クラスライブラリと組み合わせてマルチスレッド環境下で動作させないでください。

## 11.2 マルチスレッドプログラムの作成

帳票印刷クラスライブラリを利用したCOBOLプログラムを、翻訳オプションTHREAD(MULTI) を指定して構築します。

マルチスレッドプログラムの作成方法の詳細については、“NetCOBOL使用手引書”を参照してください。

## 11.3 マルチスレッド固有の留意事項

以下のクラスまたは以下のクラスを継承したクラスのオブジェクトは、複数のスレッドで共有しないでください。

[帳票印刷制御クラス\(FJPRT-CONTROL\)](#)

[帳票印刷イベント処理クラス\(FJPRT-EVENT-HANDLER\)](#)



**注意**

オブジェクトをスレッド間で共有した場合、帳票印刷クラスライブラリは正常に動作しません。

---

## 第12章   Unicode

---

ここでは帳票印刷クラスライブラリのUnicodeに関する説明をします。

[Unicode環境下での動作](#)

[Unicodeプログラムの作成](#)

[Unicode固有の留意事項](#)

## 12.1 Unicode環境下での動作

帳票印刷クラスライブラリは、Unicodeを動作モードとするCOBOLプログラムから利用することにより、Unicode環境下で動作可能です。

## 12.2 Unicodeプログラムの作成

帳票印刷クラスライブラリを利用したCOBOLプログラムを作成します。

このとき、インタフェースに英数字項目を持つプロパティやメソッドでは、実際に受け渡す文字数と利用する英数字項目の長さに注意する必要があります。英数字項目の長さと文字数の関係は、シフトJISのときとは異なります。1文字に対して必要な英数字項目の領域長(バイト数)は1~3バイトとなりますので注意してください。詳細については、“NetCOBOL 使用手引書”の“表現形式”のUTF-8の項を参照してください。

帳票印刷クラスライブラリを利用したCOBOLプログラムを作成した後は、翻訳オプションRCS(UCS2)を指定して構築します。

## 12.3 Unicode固有の留意事項

NetCOBOLが英数字項目に採用しているエンコード方式のUTF-8は、文字種によって1文字に必要な領域長が可変になります。1文字を表現するために最小で1バイトから最大で3バイトが必要になります。このため、英数字項目をインタフェースとしているプロパティやメソッドには次の影響があります。

英数字項目の型を持つプロパティ

下表のプロパティは、動作モードがシフトJISのときとUnicodeのときでは設定/参照時に保証される最大の文字数が異なります。このため印刷処理の設計には注意してください。

表：帳票印刷制御クラス(FJPRT-CONTROL)のプロパティの文字数

カテゴリ	プロパティ	プロパティの属性	シフトJISで保証される最大の文字数	Unicodeで保証される最大の文字数
帳票定義体に関するプロパティ	<a href="#">FORM-NAME</a>	PIC X(518)	259文字	172文字
	<a href="#">DE-NAME</a>	PIC X(60)	30文字	20文字
	<a href="#">CONTROL-PARTITION-NAME</a>			
	<a href="#">OVERLAY-NAME</a>	PIC X(518)	259文字	172文字
	<a href="#">FORM-DIRECTORY</a>			
	<a href="#">OVERLAY-DIRECTORY</a>			
	<a href="#">MEDIA-DIRECTORY</a>			
	<a href="#">FORM-SUFFIX</a>			
	<a href="#">OVERLAY-SUFFIX</a>			
印刷制御に関するプロパティ	<a href="#">PRINTER-NAME</a>	PIC X(128)	63文字	42文字
	<a href="#">PRINT-JOB-NAME</a>	PIC X(518)	259文字	172文字
	<a href="#">OVERLAY-NAME-TO-COMPACT-PRINT</a>			
	<a href="#">PRINTER-INFORMATION-FILENAME</a>			

英数字項目の引数を持つイベント処理メソッド

下表のメソッドの英数字項目属性の引数は、動作モードがシフトJISのときとUnicodeのときでは参照により保証される最大の文字数が異なります。このため印刷処理の設計には注意してください。

クラス	メソッド	引数の属性	シフトJISで保証される最大の文字数	Unicodeで保証される最大の文字数
帳票印刷イベント処理クラス (FJPRT-EVENT-HANDLER)	<a href="#">OPEN-FORM</a>	PIC X(518)	259文字	172文字
	<a href="#">CLOSE-FORM</a>			
	<a href="#">CB-HEAD</a>	PIC X(60)	30文字	20文字
	<a href="#">CB-FOOT</a>			
帳票印刷制御クラス (FJPRT-CONTROL)	<a href="#">OPEN-FORM</a>	PIC X(518)	259文字	172文字
	<a href="#">CLOSE-FORM</a>			
	<a href="#">CB-HEAD</a>	PIC X(60)	30文字	20文字
	<a href="#">CB-FOOT</a>			



---

## 第13章 電子帳票

---

ここでは帳票印刷クラスライブラリの電子帳票に関する説明をします。

[電子帳票の概要](#)

[SystemWalker/ListWORKS連携](#)

[SystemWalker/e-DocGenerator連携](#)

## 13.1 電子帳票の概要

帳票印刷クラスライブラリは、帳票の電子化に対応しています。帳票の電子化とは、プリンタによる紙への帳票印刷を行わず、代わりに印刷結果のイメージをコンピュータ内に蓄積することを行います。また、電子化された帳票のことを、電子帳票といいます。

電子帳票には、帳票の電子的な管理およびペーパーレス化による帳票管理業務の効率化、コスト削減などの利点があります。

帳票印刷クラスライブラリは、以下のいずれかの帳票関連ソフトウェアと連携することによって、電子帳票の出力が実現できます。

SystemWalker/ListWORKS

SystemWalker/e-DocGenerator

SystemWalker/ListWORKSとの連携で出力した電子帳票は、SystemWalker/ListWORKSの帳票管理/活用機能によって、管理や印刷、メモの記入、帳票データ内の絞り込み / 検索、グループウェア連携などを行うことができます。

SystemWalker/e-DocGeneratorとの連携では、PDF ( Adobe Portable Document Format ) 形式の電子帳票を出力します。作成したPDFファイルは、Adobe Acrobat Readerを利用して印刷/画面表示などが可能です。

## 13.2 SystemWalker/ListWORKS連携

ここではSystemWalker/ListWORKSとの連携について説明します。

SystemWalker/ListWORKSと連携する場合、“ListWORKSの電子帳票”が出力されます。この場合は、以下のプロパティ設定メソッドによってプロパティに対して値を設定します。詳細については、各メソッドの説明を参照してください。

[電子帳票種別プロパティ設定](#)

[電子帳票制御ファイル名プロパティ設定](#)

[電子帳票宛先名プロパティ設定](#)

使用するサーバによってSystemWalker/ListWORKSの提供機能が異なります。詳しくは、“SystemWalker/ListWORKS運用手引書”を参照してください。

電子帳票情報ファイルの詳細は“SystemWalker/ListWORKS運用手引書”を参照してください。

作成した“ListWORKSの電子帳票”の表示または印刷については、“SystemWalker/ListWORKS操作手引書”を参照してください。

SystemWalker/ListWORKS連携での注意事項は、以下のとおりです。

縮小種別プロパティおよびコンパクト印刷種別プロパティの設定は無効になります。

オーバレイ拡張子プロパティには、“OVD”を設定する必要があります。デフォルト時には“OVD”が設定されています。

利用者プログラムの動作モードがUnicodeの場合、ADJUSTまたはSystemWalker/CharsetMGRをインストールする必要があります。

出力可能なバーコード種別に制限があります。JAN標準、JAN短縮、CODE 3 OF 9、INDUSTRIAL 2 OUT OF 5、INTERLEAVED 2 OUT OF 5、NW7、およびカスタムバーコードが出力可能です。ただし、カスタムバーコードの印刷方向については、0度または270度のみが有効になります。

バーコード項目の矩形の指定範囲が狭い場合、その範囲を超えて印刷されます。

罫線の長破線、長鎖線、長二点鎖線、二重線、および破線は、いずれも実線で出力されます。

網がけパターンは、プリンタによる帳票印刷の結果とは見栄えが異なります。

SystemWalker/ListWORKS V5.1以降でのみ組込みメディア項目の印刷をサポートしています。組込みメディア項目にはBMP形式のファイルのみ出力可能です。

2ページ目以降の印刷処理中に、帳票のサイズの変更や印刷方向の変更を行った場合、その操作は無効になります。

## 13.3 SystemWalker/e-DocGenerator連携

ここではSystemWalker/e-DocGeneratorとの連携について説明します。

SystemWalker/e-DocGeneratorと連携する場合、PDF形式の電子帳票が出力されます。この場合は、以下のプロパティ設定メソッドによってプロパティに対して値を設定します。詳細については、各メソッドの説明を参照してください。

[電子帳票種別プロパティ設定](#)

[電子帳票制御ファイル名プロパティ設定](#)

[電子帳票ファイル名プロパティ設定](#)

帳票定義体のフォントの指定では、SystemWalker/e-DocGeneratorがサポートしているフォントを指定する必要があります。サポートしているフォントについては、“SystemWalker/e-DocGenerator 説明書 UNIX・PC編”を参照してください。

文書情報ファイルの詳細は“SystemWalker/e-DocGenerator 説明書 UNIX・PC編”を参照してください。

作成したPDFファイルを表示または印刷する場合は、“SystemWalker/e-DocGenerator 説明書 UNIX・PC編”を参照してください。

SystemWalker/e-DocGenerator連携での注意事項は、以下のとおりです。

SystemWalker/e-DocGeneratorはマルチスレッドに対応していません。このため、当クラスライブラリと組み合わせてマルチスレッド環境下で動作させないでください。

綴じ代幅の指定は無効となります。

帳票定義体で文字ピッチの指定を省略した場合、文字出力位置はプリンタによる帳票印刷の結果と異なります。

帳票定義体でフォントの横幅を変更した場合、および文字配置で圧縮を指定した場合、出力される文字サイズはプリンタによる帳票印刷の結果と異なります。

出力可能なバーコード種別に制限があります。詳しくは、“SystemWalker/e-DocGenerator 説明書 UNIX・PC編”を参照してください。

網がけパターンは、プリンタによる帳票印刷の結果とは見栄えが異なります。

罫線の二重線と波線は、いずれも実線で出力されます。

組込みメディア項目には BMP形式のファイルのみ出力可能です。

コンパクト印刷種別プロパティおよびコンパクト印刷用オーバーレイ名プロパティの設定は無効になります。

オーバーレイを出力する場合、ADJUSTまたはSystemWalker/CharsetMGRをインストールする必要があります。

バーコード項目の矩形の指定範囲が狭い場合、その範囲を超えて印刷されます。

---

## 付録A エラーコード一覧

ここでは、帳票印刷クラスライブラリを使う上で発生するエラーを識別する例外エラーコードと印刷ジョブエラーコードの一覧を記します。

[例外エラーコード](#)

[印刷ジョブエラーコード](#)

### A.1 例外エラーコード

ここでは、[例外エラー](#)の発生で帳票印刷クラスライブラリによって作られた[帳票例外クラス \(FJPRT-EXCEPTION\)](#)のオブジェクトが持つ[コードの取得メソッド \(GET-CODE\)](#)により返却される“例外エラーコード”について説明します。

なお、本書では例外エラーコードを記号定数で説明しています。詳しくは“[COPY文による記号定数の取込みの説明](#)”を参照してください。

#### FJPRT-SCE-SET-VALUE

プロパティに設定しようとした値に誤りがあります。

[意味]

プロパティに設定しようとした値がそのプロパティに設定可能な値の範囲を超えています。

[プログラマの処置]

プロパティに設定しようとした値が、そのプロパティに設定可能な値かどうか確認し、修正してください。

#### FJPRT-SCE-FIELD-NAME

項目名に誤りがあります。

[意味]

指定された帳票項目名が帳票定義体に存在しません。帳票項目名の指定に関しては“[帳票項目名によるプロパティの設定/参照](#)”を参照してください。

[プログラマの処置]

帳票定義体を確認、またはメソッドの引数に渡した帳票項目名の値を確認し、修正してください。

#### FJPRT-SCE-ILLEGAL-STATE

要求された操作の実行ができない状況です。

[意味]

以下のような状況のため、メソッド/プロパティにアクセスできません。

- イベント処理中でだけ有効なメソッド/プロパティを、イベント処理以外のタイミングで実行しました。
- イベント処理中は有効でないメソッド/プロパティを、イベント処理中に実行しました。

[プログラマの処置]

メソッド/プロパティの仕様を確認し、修正してください。

#### FJPRT-SCE-ARRAY-INDEX

項目に対する配列の要素番号に誤りがあります。

[意味]

メソッドの帳票項目名引数の集団項目の中に定義した繰返し回数に誤りがあります。帳票項目名が持つ繰返し回数の与え方については“[帳票項目が繰返し回数を持つ](#)”を参照してください。

[プログラマの処置]

メソッドに指定した帳票項目名引数の集団項目の中にある配列の要素番号を確認

---

し、修正してください。

#### FJPRT-SCE-INTERNAL-ERROR

内部エラーが発生しました。

[意味]

帳票印刷クラスライブラリの障害と考えられます。

[プログラマの処置]

技術員(SE)に連絡してください。

#### FJPRT-SCE-INVALID-OBJECT

オブジェクトの作成または操作に失敗しました。

[意味]

以下のいずれかの状況です。

- [印刷制御オブジェクトのNEWメソッド](#)の呼出しで、メモリ不足が発生しました。
- プログラムから明に \_FINALIZEメソッドが呼ばれた印刷制御オブジェクトに対して、[メソッド/プロパティ](#)を扱おうとしました。

[プログラマの処置]

状況によって以下のいずれかの対処をしてください。

- NEWメソッドによる場合:  
実行中のその他のアプリケーションを終了するか、OSが使用する仮想メモリに割り当てる領域を増やし、再度実行してください。
- その他のメソッド/プロパティによる場合:  
印刷制御オブジェクトが持つ \_FINALIZEメソッドは、明に呼び出さないようにプログラムを修正してください。[参照] “[オブジェクトの削除](#)”。また、オブジェクトを削除してからは、そのメソッド/プロパティを扱わないようにプログラムを修正してください。

## A.2 印刷ジョブエラーコード

印刷ジョブエラーコードは、[帳票印刷メソッド \(PRINT-CONTROL\)](#)の呼出しから復帰した後や、[エラーメッセージイベント \(ERROR-MESSAGEメソッド\)](#)の処理中に、[印刷ジョブエラーコードプロパティ \(PRINT-JOB-ERROR-CODE\)](#)を参照することによって通知されます。

コード	意味		対処方法
0	<a href="#">印刷エラー</a> は発生していません。	I	特にありません。
90409	内部矛盾が発生しました。処理を中止します。	U	MeFtの障害と考えられます。技術員(SE)に連絡してください。
90410	メモリ不足のため実行できません。	U	ほかのアプリケーションを終了してから、再度実行してください。
90411	ディスク空き容量不足のため実行できません。	U	環境変数 TEMP または TMP で指定したフォルダが存在するドライブに空き領域を作ってから、再度実行してください。
90412	印刷に使用する資源へのアクセス権がないため実行できません。	U	印刷に使用する資源に対するアクセス権の設定を確認し、再度実行してください。
90430	MeFtの環境に問題があります。	U	MeFtをインストールしなおしてください。
90432	電子帳票制御ファイル名、電子帳票ファイル名、または電子帳票宛先名が設定されていないため、電子帳票を出力できません。	S	電子帳票制御ファイル名プロパティ、電子帳票ファイル名プロパティ、または電子帳票宛先名プロパティの設定を正しく行うように処理を見直してください。

コード	意味		対処方法
90433	SystemWalker/ListWORKS連携中、またはSystemWalker/e-DocGenerator連携中にエラーが発生しました。	S	SystemWalker/ListWORKS または SystemWalker/e-DocGenerator の環境に問題があります。これらをインストールしなおしてください。
90434	SystemWalker/e-DocGenerator 連携中にエラーが発生したため、PDFファイルが生成できません。	S	以下のいずれかの状態であり、状態により異なる対処が必要です。 SystemWalker/e-DocGeneratorでサポートしていない機能を使用しました。使用している機能を見直してください。 SystemWalker/e-DocGenerator連携でオーバーレイを出力する場合には、ADJUSTまたはSystemWalker/CharsetMGRが必要です。どちらかをインストールしてください。 SystemWalker/e-DocGeneratorの環境に問題があります。 SystemWalker/e-DocGeneratorをインストールしなおしてください。
90440	プリンタのオープン処理に失敗しました。	S	プリンタドライバの状態を確認してください。
90445	オーバーレイライブラリのロードに失敗しました。	S	オーバーレイライブラリが存在しているか確認してください。
90447	印刷キャンセルが行われました。	I	オペレータによって印刷がキャンセルされました。再度印刷してください。
90448	指定したオーバーレイが読み込めません。	S	指定したオーバーレイが、オーバーレイ格納ディレクトリに存在しているか確認してください。
90449	オープン中のファイルが多すぎるため帳票定義体をオープンできません。	S	ファイルオープン数を減らすように、アプリケーションの処理を見直してください。
90450	プリンタ装置が使用中です。	W	プリンタ装置の使用が終了するまでお待ちください。
90451	プリンタ装置に異常があります。	S	プリンタ装置を確認してください。
90452	プリンタ装置に用紙がありません。	W	プリンタ装置に用紙を補給してください。
90453	プリンタ装置がオフライン状態か、電源が切れています。	W	プリンタ装置をオンラインにするか電源を入れてください。
90456	ネットワーク印刷環境に誤りがあります。	S	ネットワーク印刷環境を見直してください。
90457	ネットワーク通信中に異常が発生しました。	S	ネットワーク印刷環境を見直してください。
90460	不完全な帳票定義体を指定しました。	S	帳票定義体の定義エラーを修正して再度実行してください。
90463	指定した帳票定義体が見つかりません。	S	指定した帳票定義体が、帳票定義体格納ディレクトリに存在しているか確認してください。
90464	クラスインタフェース形式以外の帳票定義体が指定されました。	S	帳票定義体名プロパティに設定した帳票定義体名およびその種別を確認してください。
90465	未サポートの帳票定義体を指定しました。	S	帳票定義体名プロパティに設定した帳票定義体名およびその種別を確認してください。

コード	意味		対処方法
90466	帳票定義体の切替えができません。	S	帳票定義体名プロパティに設定した帳票定義体名と種別を確認し、アプリケーションの処理を見直してください。
90468	出力処理で予期しないエラーが発生しました。	S	MeFtの障害と考えられます。技術員(SE)に連絡してください。
90470	メディアデータのロードに失敗しました。	S	指定したメディアデータが、メディア格納ディレクトリに存在しているか確認してください。
90471	メディアデータの出力に失敗しました。	S	メディアのデータ形式およびメディア格納ディレクトリが正しいか確認してください。
90478	バーコードの出力が失敗しました。エラーの発生したバーコード項目を出力しません。	W	バーコード項目の矩形の大きさおよびデータを見直してください。
90479	オーバーレイの切替えに失敗しました。オーバーレイの指定を無視します。	W	帳票定義体名プロパティに設定した帳票定義体に定義されているオーバーレイ名を確認し、アプリケーションを見直してください。
90480	SystemWalker/ListWORKS連携時に、文字コードの変換ができません。	S	SystemWalker/ListWORKS連携時かつ利用者プログラムの動作モードがUnicodeの場合は、ADJUSTまたはSystemWalker/CharsetMGRをインストールしてください。
90483	帳票項目に不当なデータが指定されました。	W	帳票定義体で指定した項目種別、項目長、混在項目データ領域長、および出力データを確認してください。
90484	帳票項目に不当なデータが指定されました。または編集処理中にエラーが発生しました。	W	帳票定義体で指定した項目種別、項目長、混在項目データ領域長、および出力データを確認してください。
90485	装置への出力に失敗したか、出力が取り消されました。	S	プリンタ装置が使用できない、または、システムが印刷できない状況にあります。システムの環境を見直してください。
90499	集計表形式の帳票定義体の設計に矛盾があり、印刷を続行できません。	S	使用する帳票定義体の明細開始位置、終了位置、パーティション縦幅、改ページ時印刷、行ピッチの指定などを見直してエラーが発生しないように修正してください。

[記号の説明]

：エラーレベルを表します。記号の意味は以下のとおりです。

I：状況を通知します。

W：実行環境かプログラムに誤りがありましたが、印刷処理は続行します。

S：重度の誤りがありました。印刷処理を終了します。

U：回復不可能なエラーです。印刷処理を終了します。



---

# 索引

—  
\_GET-E-REPORT-CONTROL-FILENAMEメソッド .172, 91  
\_GET-E-REPORT-DESTINATIONメソッド .173, 92  
\_GET-E-REPORT-FILENAMEメソッド .176, 95  
\_GET-E-REPORT-TYPEメソッド .171, 90  
\_GET-PRINT-PREVIEWメソッド .178, 97  
\_SET-E-REPORT-CONTROL-FILENAMEメソッド .172, 91  
\_SET-E-REPORT-DESTINATIONメソッド .173, 92  
\_SET-E-REPORT-FILENAMEメソッド .176, 95  
\_SET-E-REPORT-TYPEメソッド .171, 90  
\_SET-PRINT-PREVIEWメソッド .178, 97

## A

AUTOMATIC-RECORD-CHANGEプロパティ .148, 67

## C

CALCULATE-DEメソッド .103, 190, 22, 11  
CANCEL-PRINTメソッド .100, 186, 19, 7  
CB-FOOTメソッド .111, 198, 30, 19  
CB-HEADメソッド .108, 196, 27, 17  
CLOSE-FORMメソッド .108, 195, 27, 16  
CLOSE-REPORTメソッド .97, 183, 16, 4  
COMPACT-PRINT-TYPEプロパティ .162, 81  
CONTROL-PARTITION-NAMEプロパティ .136, 55  
COPY文による記号定数の取込み .20, 9  
COPY文の記述 .52, 2

## D

DE-NAMEプロパティ .134, 53

## E

END-PAGE-NUMBERプロパティ .146, 65  
ERROR-MESSAGEメソッド .98, 184, 17, 5  
EVENT-HANDLERプロパティ .169, 88  
EXPORT-PRINTER-INFORMATIONメソッド .90, 9

## F

FJPRT-CONTROLクラス .72, 2  
FJPRT-EVENT-HANDLERクラス .78, 8  
FJPRT-EXCEPTIONクラス .81, 11  
FORM-DIRECTORYプロパティ .138, 57  
FORM-NAMEプロパティ .132, 51  
FORM-SUFFIXプロパティ .140, 59

## G

GET-CLASS-NAMEメソッド .209, 2  
GET-CODEメソッド .210, 3  
GET-DISABLE-FIELDメソッド .124, 43  
GET-EXTEND-CHARACTERメソッド .125, 44  
GET-FIELD-UNDERLINEメソッド .130, 49  
GET-FONT-BOLDメソッド .128, 47  
GET-FONT-ITALICメソッド .127, 46  
GET-FONT-STRIKE-OUTメソッド .126, 45  
GET-FONT-UNDERLINEメソッド .129, 48  
GET-MESSAGEメソッド .210, 3  
GET-METHOD-NAMEメソッド .209, 2

GET-PRE-RECORD-CONTENTメソッド .121, 40  
GET-RECORD-CONTENTメソッド .120, 39  
GET-TEXT-COLORメソッド .122, 41

## I

INIT-PAGE-NUMBERプロパティ .144, 63

## L

LIBオプション .9, 4

ListWORKSの電子帳票162, 171, 173, 222, 81, 90, 92, 3

## M

MEDIA-DIRECTORYプロパティ .140, 59

## N

NEXT-DE-PRINT-LINEメソッド .86, 5

NEXT-PRINT-LINEメソッド .89, 8

## O

OPEN-FORMメソッド .107, 194, 26, 15  
OPEN-REPORTメソッド .95, 181, 14, 2  
OVERLAY-DIRECTORYプロパティ .139, 58  
OVERLAY-NAME-TO-COMPACT-PRINTプロパティ .164, 83  
OVERLAY-NAMEプロパティ .137, 56  
OVERLAY-SUFFIXプロパティ .141, 60

## P

PAGE-FEEDメソッド .85, 4  
PAGE-FOOTメソッド .114, 201, 33, 22  
PAGE-HEADメソッド .113, 200, 32, 21  
PAGE-RANGEプロパティ .145, 64  
PAPER-SIZE-HEIGHTプロパティ .158, 77  
PAPER-SIZE-WIDTHプロパティ .158, 77  
PAPER-SIZEプロパティ .156, 75  
PAPER-TYPEプロパティ .155, 74  
PDF形式 .171, 173, 223, 90, 92, 4  
PRINT-CONTROLメソッド .83, 2  
PRINT-COPIESプロパティ .156, 75  
PRINT-COUNTプロパティ .152, 71  
PRINT-DEメソッド .104, 191, 23, 12  
PRINTER-DEVICE-NAMESメソッド .93, 12  
PRINTER-DEVICE-NUMBERメソッド .92, 11  
PRINTER-INFORMATION-FILENAMEプロパティ .167, 86  
PRINTER-NAMEプロパティ .153, 72  
PRINTING-DIALOG-CANCEL-BUTTONプロパティ .165, 84  
PRINTING-DIALOG-MINIMIZE-BOXプロパティ .166, 85  
PRINTING-DIALOGプロパティ .164, 83  
PRINTING-LINE-NUMBERプロパティ .143, 62  
PRINTING-PAGE-NUMBERプロパティ .143, 62  
PRINT-JOB-CONDITIONプロパティ .155, 74  
PRINT-JOB-ERROR-CODEプロパティ .142, 61  
PRINT-JOB-NAMEプロパティ .154, 73  
PRINT-POSITION-LEFTプロパティ .161, 80  
PRINT-POSITION-TOPプロパティ .160, 79  
PRINT-POSITIONプロパティ .159, 78  
PRINT-TYPEプロパティ .151, 70

---

R	
RCS(UCS2)オプション	218, 3
READ-RECORDメソッド	101, 187, 20, 8
RECORD-COUNTプロパティ	150, 69
RECORD-EOFプロパティ	149, 68
REDUCTION-TYPEプロパティ	161, 80
REPINオプション	9, 4
REPORT-FOOTメソッド	117, 205, 36, 26
REPORT-HEADメソッド	116, 203, 35, 24
REQUEST-CANCELプロパティ	152, 71
S	
SET-DISABLE-FIELDメソッド	124, 43
SET-EXTEND-CHARACTERメソッド	125, 44
SET-FIELD-UNDERLINEメソッド	130, 49
SET-FONT-BOLDメソッド	128, 47
SET-FONT-ITALICメソッド	127, 46
SET-FONT-STRIKE-OUTメソッド	126, 45
SET-FONT-UNDERLINEメソッド	129, 48
SET-RECORD-CONTENTメソッド	120, 39
SET-TEXT-COLORメソッド	122, 41
SKIP-PARTITIONメソッド	89, 8
START-PAGE-NUMBERプロパティ	145, 64
START-TACK-NUMBERプロパティ	147, 66
T	
THREAD(MULTI)オプション	214, 3
U	
Unicode	216, 1
あ	
アプリケーションの実行	11, 6
アプリケーションの設計	13, 2
い	
イタリックプロパティ参照メソッド	127, 46
イタリックプロパティ設定メソッド	127, 46
イベント	30, 1
イベント処理オブジェクト	79, 9
イベント処理オブジェクトプロパティ	169, 88
印刷エラー	70, 8
印刷回数プロパティ	152, 71
印刷キャンセルイベント処理メソッド	100, 186, 19, 7
印刷原点縦位置プロパティ	160, 79
印刷原点プロパティ	159, 78
印刷原点横位置プロパティ	161, 80
印刷種別プロパティ	151, 70
印刷ジョブエラーコード	225, 2
印刷ジョブエラーコードプロパティ	142, 61
印刷ジョブ状態プロパティ	155, 74
印刷ジョブ名プロパティ	154, 73
印刷処理	12, 1
印刷制御オブジェクト	72, 2
印刷中行番号プロパティ	143, 62
印刷中ダイアログ最小化ボックスプロパティ	166, 85
印刷中ダイアログのキャンセルボタンプロパティ	165, 84
印刷中ダイアログプロパティ	164, 83
印刷中ページ番号プロパティ	143, 62
印刷部数プロパティ	156, 75
印刷プレビュー	145, 146, 178, 64, 65, 97
印刷プレビューウィンドウ	178, 97
印刷プレビュープロパティ参照メソッド	178, 97
印刷プレビュープロパティ設定メソッド	178, 97
え	
エラーコード一覧	224, 1
エラーメッセージイベント処理メソッド	98, 184, 17, 5
お	
オーバーレイ拡張子プロパティ	141, 60
オーバーレイ格納ディレクトリプロパティ	139, 58
オーバーレイ定義体	7, 13, 2
オーバーレイ定義体の作成	7, 2
オーバーレイ名プロパティ	137, 56
オブジェクトの削除	73, 79, 3, 9
オブジェクトの作成	72, 79, 2, 9
オブジェクトファイルのリンク	10, 5
か	
開始ページ番号プロパティ	145, 64
改ページメソッド	85, 4
下線プロパティ参照メソッド	129, 48
下線プロパティ設定メソッド	129, 48
き	
記号定数	20, 9
キャンセル要求プロパティ	152, 71
業務処理	14, 3
く	
クラス名の参照メソッド	209, 2
こ	
項目下線プロパティ参照メソッド	130, 49
項目下線プロパティ設定メソッド	130, 49
項目抑止プロパティ参照メソッド	124, 43
項目抑止プロパティ設定メソッド	124, 43
コードの参照メソッド	210, 3
コントロールブレークの制御方法	40, 11
コンパクト印刷種別プロパティ	162, 81
コンパクト印刷用オーバーレイ名プロパティ	164, 83
さ	
最初のページ番号プロパティ	144, 63
し	
次印刷行参照メソッド	89, 8
次明細印刷行参照メソッド	86, 5
終了ページ番号プロパティ	146, 65
縮小種別プロパティ	161, 80
準備するもの	4
せ	
制御脚書きイベント処理メソッド	111, 198, 30, 19
制御頭書きイベント処理メソッド	108, 196, 27, 17
制御パーティション名プロパティ	136, 55

た		文書情報ファイル..... 223, 4
タック開始番号プロパティ.....147, 66		文書情報ファイル名..... 173, 92
段組み集計表形式.....3		へ
ち		ページ脚書きイベント処理メソッド114, 201, 33, 22
帳票印刷イベント処理クラス.....78, 8		ページ頭書きイベント処理メソッド113, 200, 32, 21
帳票印刷クラスライブラリとは.....2		ページ範囲プロパティ..... 145, 64
帳票印刷クラスライブラリの概要.....1		ほ
帳票印刷制御クラス.....72, 2		報告書脚書きイベント処理メソッド117, 205, 36, 26
帳票印刷メソッド.....83, 2		報告書頭書きイベント処理メソッド116, 203, 35, 24
帳票印刷例外クラス.....81, 11		ボールドプロパティ参照メソッド..... 128, 47
帳票開設イベント処理メソッド....95, 181, 14, 2		ボールドプロパティ設定メソッド..... 128, 47
帳票項目.....51, 1		ま
帳票定義体.....7, 13, 43, 45, 2, 14, 16		前レコード..... 38, 9
帳票定義体開設イベント処理メソッド107, 194, 26, 15		前レコード内容の参照メソッド..... 121, 40
帳票定義体拡張子プロパティ.....140, 59		抹消線プロパティ参照メソッド..... 126, 45
帳票定義体格納ディレクトリプロパティ...138, 57		抹消線プロパティ設定メソッド..... 126, 45
帳票定義体の作成.....7, 2		マルチスレッド..... 212, 1
帳票定義体閉鎖イベント処理メソッド108, 195, 27, 16		め
帳票定義体名プロパティ.....132, 51		明細計算イベント処理メソッド. 103, 190, 22, 11
帳票の電子化.....221, 2		明細出力イベント処理メソッド. 104, 191, 23, 12
帳票閉鎖イベント処理メソッド....97, 183, 16, 4		明細名プロパティ..... 134, 53
て		メソッド名の参照メソッド..... 209, 2
電子帳票4, 8, 154, 156, 163, 164, 171, 179, 220, i, 3, 73		メッセージの参照メソッド..... 210, 3
電子帳票宛先名プロパティ参照メソッド...173, 92		メディア格納ディレクトリプロパティ.... 140, 59
電子帳票宛先名プロパティ設定メソッド...173, 92		も
電子帳票種別プロパティ参照メソッド.....171, 90		文字色プロパティ参照メソッド..... 122, 41
電子帳票種別プロパティ設定メソッド.....171, 90		文字色プロパティ設定メソッド..... 122, 41
電子帳票情報ファイル.....222, 3		文字引伸しプロパティ参照メソッド..... 125, 44
電子帳票情報ファイル名.....173, 92		文字引伸しプロパティ設定メソッド..... 125, 44
電子帳票制御ファイル名.....172, 91		よ
電子帳票制御ファイル名プロパティ参照メソッド172, 91		用紙サイズプロパティ..... 156, 75
電子帳票制御ファイル名プロパティ設定メソッド172, 91		用紙種別プロパティ..... 155, 74
電子帳票ファイル名プロパティ参照メソッド176, 95		用紙縦サイズプロパティ..... 158, 77
電子帳票ファイル名プロパティ設定メソッド176, 95		用紙横サイズプロパティ..... 158, 77
は		れ
パーティション抑止メソッド.....89, 8		例外エラー..... 64, 2
ふ		例外エラーコード..... 224, 1
プリンタ情報生成メソッド.....90, 9		レコード..... 52, 2
プリンタ情報ファイル名プロパティ.....167, 86		レコードEOFプロパティ..... 149, 68
プリンタデバイス数参照メソッド.....92, 11		レコード回数プロパティ..... 150, 69
プリンタデバイス名参照メソッド.....93, 12		レコード自動切替えプロパティ..... 148, 67
プリンタ名プロパティ.....153, 72		レコード内容の参照メソッド..... 120, 39
ブレイク項目.....39, 10		レコード内容の設定メソッド..... 120, 39
プログラムの作成.....8, 3		レコードの切替えの例..... 148, 67
プログラムの翻訳.....9, 4		レコード読み込みイベント処理メソッド101, 187, 20, 8







このマニュアルはエコマーク認定の再生紙を使用しています。