

uCosminexus Batch Job Execution Server 使用の手引

解説・手引・文法・操作書

3000-3-E02-F0

前書き

■ 対象製品

P-1M12-A291 uCosminexus Batch Job Execution Server 09-10 (適用 OS : AIX)

P-8412-A291 uCosminexus Batch Job Execution Server 09-10 (適用 OS : Red Hat Enterprise Linux Server 7 (64-bit x86_64), Red Hat Enterprise Linux Server 8 (64-bit x86_64), Red Hat Enterprise Linux Server 9 (64-bit x86_64))

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

HITACHI, Cosminexus, EUR, JP1, uCosminexus は、株式会社 日立製作所の商標または登録商標です。

AMD は、Advanced Micro Devices, Inc.の商標です。

AIX は、世界の多くの国で登録された International Business Machines Corporation の商標です。

Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標です。

Microsoft は、マイクロソフト 企業グループの商標です。

Oracle(R), Java, MySQL 及び NetSuite は、Oracle, その子会社及び関連会社の米国及びその他の国における登録商標です。

Red Hat, and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the United States and other countries. Linux(R) is the registered trademark of Linus Torvalds in the U.S. and other countries.

Red Hat, および Red Hat Enterprise Linux は、米国およびその他の国における Red Hat, Inc.の登録商標です。Linux(R)は、米国およびその他の国における Linus Torvalds 氏の登録商標です。

RHEL is a trademark or a registered trademark of Red Hat, Inc. in the United States and other countries.

RHEL は、米国およびその他の国における Red Hat, Inc.の商標または登録商標です。

UNIX は、The Open Group の登録商標です。

Windows は、マイクロソフト 企業グループの商標です。

その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。



■ 発行

2023 年 12 月 3000-3-E02-F0

■ 著作権

All Rights Reserved. Copyright (C) 2007, 2023, Hitachi, Ltd.

変更内容

変更内容 (3000-3-E02-F0) uCosminexus Batch Job Execution Server 09-10

追加・変更内容	変更箇所
適用 OS に次の OS を追加した。 <ul style="list-style-type: none">• AIX7.2• AIX7.3• Red Hat Enterprise Linux Server 8• Red Hat Enterprise Linux Server 9 また、適用 OS から次の OS を削除した。 <ul style="list-style-type: none">• AIX6.1• Red Hat Enterprise Linux Server 6	1.3.1

単なる誤字・脱字などはお断りなく訂正しました。

変更内容 (3000-3-E02-E0) uCosminexus Batch Job Execution Server 09-02

追加・変更内容
JOBLOG の出力形式に AJSJOBNAME を追加した。
一時ファイルを作成するディレクトリをジョブ定義 XML ファイルで指定できるようにした。
ディレクトリに対する排他/共用制御で、ディレクトリ配下のファイルおよびディレクトリを再帰的に排他/共用制御するか指定できるようにした。
スプールジョブ保管時のディレクトリ名を変更できるようにした。
bjexu_ismut および bjexu_ismut64 でのファイルの複写と変換を高速化できるようにした。
bjexcollect コマンドで JP1 独自の hosts 情報を指定できるようにした。

はじめに

このマニュアルは、バッチジョブ実行システムの機能、および操作方法について説明したものです。

■ 対象読者

UNIX および JP1/AJS の知識がある方を前提とし、バッチジョブの運用をオープンシステムで実現するシステムを構築したい方を対象としています。

■ マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

なお、このマニュアルは、AIX, Linux の各 OS に共通のマニュアルです。OS ごとに差異がある場合は、本文中でそのつど内容を書き分けています。

第 1 章 概要

バッチジョブ実行システムの導入の目的、特長、前提条件、および全体構成について説明しています。

第 2 章 インストール

バッチジョブ実行システムのインストールの準備と手順、および見積もりについて説明しています。

第 3 章 機能

バッチジョブ実行システムの機能について説明しています。

第 4 章 運用

バッチジョブ実行システムの運用について説明しています。

第 5 章 操作

バッチジョブ実行システムと JP1 プログラムや Prinfina MANAGER SP および PDE (PRINT DATA EXCHANGE) を連携して使用するための、設定および操作方法について説明しています。

第 6 章 障害対策

バッチジョブ実行システムに必要な障害対策について説明しています。

第 7 章 コマンド

バッチジョブ実行システムのコマンドについて説明しています。

第 8 章 設定ファイル

バッチジョブ実行システムの設定ファイルについて説明しています。

第 9 章 ジョブ定義 XML ファイル

バッチジョブ実行システムのジョブ定義 XML ファイルについて説明しています。

第 10 章 メッセージと終了コード

バッチジョブ実行システムが出力するメッセージおよび終了コードについて説明しています。

付録 A ジョブ定義 XML ファイル仕様

ジョブ定義 XML ファイルの XML 仕様について説明しています。

付録 B プロセス一覧

バッチジョブ実行システムのプロセス一覧について説明しています。

付録 C bjexcollect コマンド (障害調査のための資料の一括収集)

bjexcollect コマンドについて説明しています。

付録 D メッセージ制御機能のエラー情報

メッセージ制御機能のエラー情報について説明しています。

付録 E このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報について説明しています。

付録 F 用語解説

このマニュアルで使用する用語について説明しています。

■ 読書手順

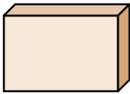
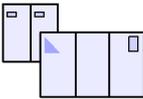
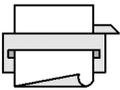
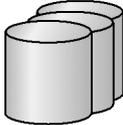
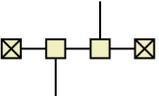
このマニュアルは、利用目的に合わせて章を選択して読むことができます。利用目的別にお読みいただくことをお勧めします。

マニュアルを読む目的	記述箇所
バッチジョブ実行システムの概要について知りたい。	1 章
バッチジョブ実行システムのインストールについて知りたい。	2 章
バッチジョブ実行システムの機能について知りたい。	3 章
バッチジョブ実行システムの運用について知りたい。	4 章
バッチジョブ実行システムの操作について知りたい。	5 章
バッチジョブ実行システムの障害対策について知りたい。	6 章

マニュアルを読む目的	記述箇所
バッチジョブ実行システムのコマンドについて知りたい。	7章
バッチジョブ実行システムの設定ファイルについて知りたい。	8章
バッチジョブ実行システムのジョブ定義 XML ファイルについて知りたい。	9章
バッチジョブ実行システムのメッセージについて知りたい。	10章
ジョブ定義 XML ファイル仕様, プロセサー一覧, bjexcollect コマンド, メッセージ制御機能のエラー情報, このマニュアルの参考情報, または用語解説について知りたい。	付録

■ 図中で使用する記号

このマニュアルの図中で使用する記号を、次のように定義します。

●工程, 作業項目の 流れ	●プログラム	●コンピュータ	●ホストセンタ
			
●端末プリンタ	●ファイル	●ファイル(複数)	●ジョブまたは ジョブステップ
			
●ネットワーク			
			

■ このマニュアルで使用している記号

このマニュアルの文法説明で使用する記号を次に示します。

記号	意味
[]	<p>[] で囲まれている項目は、任意に指定できる (省略してもよい)。 複数の項目が記述されている場合、すべてを省略するか、どれか 1 つを選択する。</p> <p>例</p> <ul style="list-style-type: none"> • [A] 「何も指定しない」か「A を指定する」ことを示す。 • [B C]

記号	意味
[]	「何も指定しない」か「BまたはCを指定する」ことを示す。
< >	各項目を記述するときに従わなくてはならないシンタクス記号を示す。
{ }	この記号で囲まれている複数の項目の中から、一組の項目を必ず選択する。項目と項目の区切りは「 」で示す。 例 {A B C} 「A, B, またはCのどれかを必ず指定する」ことを示す。
 (ストローク)	複数の項目に対し、項目間の区切りを示し、「または」の意味を示す。 例 [A B C] 「A, B または C」を示す。
+	この記号の直前に示された項目を繰り返して複数指定ができる。 例 {A B} + 「A または B を任意の順序で1つ以上指定する」ことを示す。
*	この記号の直前に示された項目を指定しないか、繰り返して複数指定ができる。 例 「{A B} *」は、「A または B を指定しないか、任意の順序で1つ以上指定する」ことを示す。
~	この記号の直前に示されている項目を、この記号に続く< >, 《 》, (())などの文法規則に従って記述する必要があることを示す。
《 》	項目を省略したときにシステムがとる仮定値を示す。
(())	指定できる値の範囲を示す。 ((n 文字))と記述している場合、特に説明がない限り、1バイトコードの文字で定義できる文字数を意味する (n: 整数)。
__ (下線)	選択記号で囲まれている項目を省略したときにシステムが取る仮定値であることを示す。
… (点線)	この記号の直前に示された項目を繰り返して複数個指定できることを示す。 例 A, B, … 「A のあとに B を必要個数指定する」ことを示す。

このマニュアルのシンタクス記号を次に示します。

シンタクス記号	指定できる文字の内容
<数字>	0 1 2 3 4 5 6 7 8 9

シンタクス記号	指定できる文字の内容
<英大文字>	A B C … Z
<英小文字>	a b c … z
<英字>	<英大文字> <英小文字>
<特殊文字>※	, . / ' () * & + - = △(空白) ¥
< 8 進数>	< 0 1 2 3 4 5 6 7 > +
< 10 進数>	< 数字 > +
< 16 進数字>	0 1 2 3 4 5 6 7 8 9 A B C D E F
<整数>	符号が付いてもよい数字または符号のない数字の集まり。
<符号なし整数>	<数字> +
<記号名称>, <クラス名>	{ <英字> <数字> @ # _ (アンダースコア) } + 対象：ジョブ名, ステップ名, およびクラス名など
<特殊文字列>※	英数字, 特殊文字, または特殊文字の集まり。
<引用文字列>※	記号名称または特殊文字列をアポストロフィ (') で囲んだもの。 文字列中に空白などの区切り文字を含めるときにアポストロフィで囲んで指定する。ただし、特殊文字列中のアポストロフィは、¥ (エスケープ文字) を使い、¥¥で指定する。また、特殊文字列中の¥は、¥¥で指定する。¥を文字として指定するとき、¥と¥¥の指定は、どちらも 1 文字の¥として扱う。 引用文字列の文字数は、アポストロフィやエスケープ文字を含まない数を表す。
<環境変数名>	{ <英字> _ (アンダースコア) } { <英字> _ (アンダースコア) <数字> } * 対象：DD 名, パス名置換識別子 BJEX_DIR_任意名など
<パス名>	UNIX または Windows のファイルパス名規則に従った文字列
<任意文字列>	任意の文字による文字列 <ul style="list-style-type: none"> • バッチジョブ実行システムは文字種別をチェックしない。 • 指定個所に応じた適切な意味のある文字列とする必要がある。 • 記号名称の範囲での利用を推奨する。 対象 ：パス名, DSN 属性の値, EXEC 要素, および DD 要素の内容の文字列など
< ASCII 文字列>	ASCII 文字コード範囲中の、制御文字を除いた範囲 (0x20~0x7e) の文字によって構成される文字列。
<名前トークン>	{ <英字> <数字> . - _ : } +
<グループ名>	{ <英字> <数字> _ (アンダースコア) } +

注※ <特殊文字>, <特殊文字列>, および<引用文字列>の定義は、従来のシンタクス記号に追加となる。

目次

前書き	2
変更内容	4
はじめに	5

1 概要 21

1.1	バッチジョブ実行システムとは	22
1.2	バッチジョブ実行システムの特長	24
1.2.1	バッチジョブ定義の定型化	24
1.2.2	バッチジョブの実行結果の一元管理	25
1.2.3	PDE との連携による帳票出力	26
1.3	前提条件	28
1.3.1	前提プログラムおよび関連プログラム	28
1.3.2	ファイルシステム	31
1.3.3	2GB 超えのファイル (ラージファイル) の扱い	33
1.4	バッチジョブ実行システムの全体構成	34

2 インストール 35

2.1	バッチジョブ実行システムのセットアップ手順の概要	36
2.1.1	基本システム構成の概要とセットアップ手順	36
2.2	インストール	39
2.2.1	バッチジョブ実行システムのインストール	39
2.2.2	バッチジョブ実行システムのアンインストール	42
2.2.3	バッチジョブ実行システムのバージョン情報の表示	46
2.2.4	インストール後の設定	46
2.2.5	旧バージョンから移行時の注意事項	50
2.3	セットアップ	51
2.3.1	セットアップ前の作業	51
2.3.2	セットアップ	52
2.3.3	bjexcollect コマンドの設定	54
2.3.4	JP1/AJS 環境設定	54
2.3.5	JP1/FTP 環境設定	55
2.3.6	メッセージ制御機能を使用する場合の環境設定	55
2.3.7	排他/共用制御機能および実行中のジョブ情報表示機能を使用する場合	56
2.4	ストレージ設計	57

3	機能 58
3.1	バッチジョブ実行システムの機能 59
3.1.1	バッチジョブ実行システムの機能の特長と処理手順 59
3.1.2	ジョブ定義 XML ファイル 59
3.1.3	ジョブコントローラ 61
3.1.4	スプール 71
3.2	カタログドプロシジャと入力ストリームプロシジャ 75
3.2.1	プロシジャの概要 75
3.2.2	入力ストリームプロシジャの作成方法 75
3.2.3	カタログドプロシジャの使用法 76
3.2.4	入力ストリームプロシジャの使用法 77
3.2.5	プロシジャの修正方法 78
3.2.6	プロシジャ内の EXEC 要素の修正方法 78
3.2.7	プロシジャ内の DD 要素の修正方法 81
3.2.8	プロシジャ内への DD 要素の追加方法 83
3.2.9	プロシジャの記号パラメータによる修正方法 84
3.2.10	プロシジャライブラリの定義方法 85
3.3	プログラムのシステムへの登録と使用 87
3.3.1	システムのプログラム使用法 87
3.3.2	ユーザプログラムの使用法 87
3.4	ファイルの割り当て 90
3.4.1	ファイルの準備 90
3.4.2	ファイルの状態と後処理 91
3.4.3	ファイルの連結処理 99
3.4.4	一時ファイル 102
3.4.5	一時 ISAM ファイル 104
3.4.6	ダミーファイル 108
3.4.7	プログラムへのファイル割り当て機能 109
3.4.8	逆方向参照 114
3.5	ジョブステップ 117
3.5.1	プログラムコマンドを実行 117
3.5.2	シェルスクリプトを実行 117
3.5.3	PARM 属性値が空白を含む場合 118
3.5.4	ジョブステップの正常終了／異常終了 119
3.5.5	注意事項 120
3.6	ジョブステップの条件付き実行 (COND 属性) 121
3.6.1	条件付き実行の指定方法 121
3.6.2	条件付き実行 122
3.6.3	ジョブステップを条件付きで実行するときのステップ名の検索範囲 127

3.7	ジョブステップの実行制御 (IF 要素)	131
3.7.1	ジョブステップの実行制御の各要素の使用規則	131
3.7.2	ジョブステップの実行制御の使用例	132
3.8	ステップリスタート	134
3.8.1	ステップリスタートを実行するための条件	134
3.8.2	JP1/AJS - View でのステップリスタートの実行	134
3.8.3	bjexec コマンドでのステップリスタートの実行	137
3.8.4	ステップリスタートを実行する上でのファイルの扱い	137
3.8.5	ステップリスタートでの注意事項	138
3.9	世代ファイル	139
3.9.1	世代ファイルの概要	139
3.9.2	世代データ群の作成	142
3.9.3	世代ファイルの作成と参照	143
3.9.4	世代ファイルおよび世代データ群の削除	146
3.9.5	世代ファイルの保存日数超過による一括削除機能	148
3.9.6	世代ファイルの世代管理ファイルへの登録タイミング変更機能	150
3.9.7	世代ファイルの注意事項	154
3.10	SYSIN データの読み込み	159
3.10.1	SYSIN データの書き方	159
3.10.2	SYSIN データとなる文字列	159
3.10.3	SYSIN データを順編成固定長ファイルとしてプログラムに渡す方法	161
3.11	SYSOUT の機能	164
3.11.1	出力クラス	165
3.11.2	出力情報の複写	165
3.11.3	出力形式	165
3.11.4	出力保留	165
3.11.5	出力後保存	165
3.11.6	プリンタへの出力要求	166
3.11.7	レベル 1 外字名指定	168
3.11.8	0 バイトの SYSOUT ファイル転送機能	168
3.12	ジョブステップを並列実行する機能	169
3.12.1	JSPEC 機能	169
3.12.2	PREST 機能	176
3.13	ジョブ単位のファイルおよびディレクトリに対する排他/共用制御機能	182
3.13.1	排他/共用制御の指定方法と適用範囲	186
3.13.2	ディレクトリ割り当て時の排他/共用要求の範囲	188
3.13.3	排他/共用制御による実行待ち状態の確認方法	192
3.13.4	排他/共用制御に関する注意事項	195
3.14	実行中のジョブ情報表示機能	196

- 3.14.1 表示されるジョブ情報 196
- 3.14.2 設定ファイルの指定 197
- 3.14.3 ジョブの多重度制御 197
- 3.15 メッセージ制御機能 198
- 3.15.1 メッセージ制御機能の種類 198
- 3.15.2 応答要求メッセージ 205
- 3.15.3 JP1/IM - View との関係 205
- 3.15.4 JP1/IM - Manager との関係 206
- 3.15.5 JP1/IM - View からの応答入力 206
- 3.15.6 メッセージ制御機能でのデータ置き換え機能 208
- 3.15.7 DISPLAY 文による JP1 イベント出力の抑止機能 208
- 3.15.8 メッセージ制御機能でのイベント到達確認 209
- 3.15.9 メッセージ制御機能に関する注意事項 210
- 3.16 スプールジョブ保管機能 214
- 3.16.1 スプールジョブ保管機能の概要 214
- 3.16.2 スプールジョブ保管機能の設定方法 215
- 3.16.3 日付や時間ごとのサブディレクトリへの保管 216
- 3.16.4 保管先ディレクトリの名称 216
- 3.16.5 スプールジョブ保管機能使用時の注意事項 217
- 3.17 PDE ファイル形式 2 219
- 3.18 ファイル操作ユティリティ 220
- 3.18.1 ファイル操作ユティリティの概要 220
- 3.18.2 ファイル操作ユティリティのサポートするファイル 220
- 3.18.3 ファイル操作ユティリティの実行方法 223
- 3.18.4 ファイル操作ユティリティの制御文の形式 224
- 3.18.5 bjexu_dummy (ファイルおよびディレクトリ操作用ダミー処理の実行) 225
- 3.18.6 bjexu_pcopy (ファイルの複写) 226
- 3.18.7 bjexu_scopy (ファイルのレコード単位の複写) 237
- 3.18.8 bjexu_ismut および bjexu_ismut64 (索引順編成ファイルの作成, 削除, 索引順編成ファイルと順編成ファイルの複写, 変換および併合) 262
- 3.18.9 メッセージの出力先 297
- 3.19 Cosminexus との連携によるバッチアプリケーション実行 298
- 3.19.1 LANG="Java"を指定した場合の Java バッチアプリケーションを実行する方法 299
- 3.19.2 LANG="Java"を指定しないで Java バッチアプリケーションを実行する方法 303
- 3.19.3 Java バッチアプリケーションの強制終了 303
- 3.20 JP1/Script の実行 304
- 3.20.1 JP1/Script を実行する方法 304
- 3.21 環境変数 306
- 3.21.1 ジョブコントローラが使用する環境変数 306

- 3.21.2 ジョブコントローラから実行されるプログラムに設定される環境変数 306
- 3.21.3 ファイル操作ユティリティの機能を定義する環境変数 307
- 3.21.4 XML ファイルへの環境変数指定機能 308
- 3.21.5 環境変数一括設定機能 310
- 3.21.6 環境変数の優先順位 313

4 運用 314

- 4.1 概要 315
- 4.2 利用者の役割 316
 - 4.2.1 システム管理者 316
 - 4.2.2 一般ユーザ 316
- 4.3 バッチジョブ実行システムのシステム設計 319
 - 4.3.1 動作環境の確認 319
 - 4.3.2 スプールジョブ転送サービス 320
 - 4.3.3 出力クラスの設定 321
 - 4.3.4 セキュリティ情報の設定 322
 - 4.3.5 バッチジョブ業務の自動化の検討 323
 - 4.3.6 排他/共用制御機能の設定 323
 - 4.3.7 メッセージ制御機能の設定 324
- 4.4 JP1/AJS によるジョブスケジュール 326
 - 4.4.1 バッチジョブ業務の自動化 326
 - 4.4.2 バッチジョブの実行環境 329
 - 4.4.3 バッチジョブ定義の記述方法 330
 - 4.4.4 バッチジョブ実行システム実行ユーザ管理 332
 - 4.4.5 注意事項 332
- 4.5 スプールジョブ転送サービスの運用 334
 - 4.5.1 スプールジョブ転送デーモンの運用 334
 - 4.5.2 帳票処理の概要 335
 - 4.5.3 出力クラスによる制御 336
 - 4.5.4 出力保留属性および出力非保留属性 337
 - 4.5.5 スプールジョブの削除 338
 - 4.5.6 印刷ジョブの削除 338
 - 4.5.7 スプールジョブの回復 339
 - 4.5.8 スプールジョブを転送しない運用 339
- 4.6 JP1/IM による実行状態の監視と実行結果の確認 340
 - 4.6.1 統合コンソールによる集中監視 340
 - 4.6.2 ジョブ名などの情報をイベント一覧に表示 341
 - 4.6.3 応答要求メッセージの監視 341
- 4.7 EUR を使用した帳票印刷 343

4.7.1	機能概要	343
4.7.2	EUR 定義ファイルの準備	344
4.7.3	設定ファイルの定義	347
4.7.4	ジョブ定義 XML ファイルでの指定	348
4.8	EUC 言語環境でのジョブ実行【AIX 限定】	350
4.8.1	機能概要	350
4.8.2	言語環境の設定	350
4.8.3	ジョブ定義 XML ファイルの指定	350
4.8.4	EUC 言語環境でのメッセージ制御機能の注意事項	351
4.8.5	設定ファイル	351
4.9	バッチジョブ実行システム的环境を複数構築する場合の設定	352
4.9.1	複数のバッチジョブ実行システムの運用の前提条件	352
4.9.2	論理ホストのバッチジョブ実行システム的环境設定	354
4.9.3	複数のバッチジョブ実行システムの運用の注意事項	357
4.10	クラスタシステムで運用する場合の設定	358
4.10.1	クラスタシステムおよびクラスタ運用の概要	358
4.10.2	クラスタ運用の前提条件とサポート範囲	360
4.10.3	クラスタ運用の環境設定	361
4.10.4	クラスタ運用に関する注意事項	363
4.11	バッチジョブ実行システムの運用時の注意事項	364
4.11.1	システム停止時の注意事項	364
4.11.2	ウイルス対策ソフト実行時の注意事項	364
4.12	PREST 機能を使用するジョブの定義	365
4.13	COBOL プログラムで使用する文字コードを変更する場合の設定	366
4.13.1	機能概要	366
4.13.2	ジョブ定義 XML ファイルの指定	366
4.13.3	設定ファイル	366
4.14	バッチジョブ実行システムにおける SystemV セマフォ、共有メモリの管理	368
4.14.1	使用する共有メモリ	368
4.14.2	管理方法	368
4.14.3	セマフォ、共有メモリ重複時のメッセージ	368
4.14.4	セマフォ、共有メモリ重複時の対処方法	369
5	操作	370
5.1	バッチジョブ実行システムの使用概要	371
5.1.1	バッチジョブ実行システムを使用するためのシステム構成	371
5.2	バッチジョブの実行	372
5.2.1	バッチジョブの起動	372
5.2.2	JP1/AJS からバッチジョブを実行する場合のジョブ定義	374

- 5.2.3 実行結果の参照 377
- 5.2.4 バッチジョブの強制終了 377
- 5.3 Prinfina MANAGER SP および PDE との連携による帳票印刷 379
 - 5.3.1 帳票印刷 379
 - 5.3.2 JP1/FTP を使った印刷データ転送 379
 - 5.3.3 SYSOUT 属性の PDE および Prinfina MANAGER SP での扱い 383
 - 5.3.4 印刷データの転送契機 386
- 5.4 JP1/IM - Manager との連携による応答要求メッセージへの応答入力 388
 - 5.4.1 応答の入力 388
 - 5.4.2 JP1/IM - Manager との連携 388
 - 5.4.3 応答要求メッセージの同時出力数 388
 - 5.4.4 応答入力 388

6 障害対策 390

- 6.1 バックアップとリカバリー 391
 - 6.1.1 バックアップ 391
 - 6.1.2 リカバリー 391
- 6.2 トラブルシューティング 392
 - 6.2.1 対処の方法 392
 - 6.2.2 ログ情報の種類 392
 - 6.2.3 トラブル発生時に採取が必要な資料 395
 - 6.2.4 資料の採取方法 400
- 6.3 PREST ファイルが存在して、新たに PREST ファイルを使用するジョブを実行できない場合の対処方法 401
 - 6.3.1 ジョブ間 PREST ファイルの場合 401
 - 6.3.2 ジョブステップ間 PREST ファイルの場合 401
- 6.4 ジョブが長時間待ち状態となった場合の対処方法 402
- 6.5 メッセージ制御機能での障害対応 403
- 6.6 JP1/FTP を使った印刷データ転送での障害対応 405
- 6.7 実行ログ出力用ディスクの容量不足の対処方法 406
- 6.8 スプールジョブ転送デーモンの障害対策 407

7 コマンド 408

- コマンドの記述形式 409
- コマンド一覧 410
 - bjex_msg_setup (メッセージ制御機能の初期設定 (バッチジョブ実行システムの設定用)) 411
 - bjexchjob (スプールジョブの属性変更 (一般ユーザ用)) 412
 - bjexchmsg (応答待ちメッセージへの応答入力 (管理者用)) 417
 - bjexec (バッチジョブの実行 (一般ユーザ用)) 419
 - bjexgdg (世代データ群の生成および更新 (一般ユーザ用)) 421

bjexhkgdg (保存日数に指定した日数を超過した世代ファイルの一括削除を行う (一般ユーザ用)) 427

bjexlsexjob (実行中のジョブ情報の表示 (一般ユーザ用)) 429

bjexlsjob (スプールジョブの一覧表示 (一般ユーザ用)) 435

bjexlslock (ファイルの排他状態の表示 (一般ユーザ用)) 443

bjexlsmmsg (応答待ちメッセージおよび受信待ちメッセージ一覧の表示 (管理者用)) 452

bjexm_message (メッセージまたは応答要求メッセージの出力 (一般ユーザ用)) 454

bjexpd (スプールジョブ転送デーモン (管理者用)) 459

bjexpdctl (スプールジョブ転送デーモンの起動, 停止 (管理者用)) 461

bjexrmjob (HOLD 状態またはエラー状態のスプールジョブの削除 (一般ユーザ用)) 463

bjextrjob (転送ディレクトリ下の印刷ジョブ一覧の表示, 削除, および再転送 (管理者用)) 468

8 設定ファイル 472

設定ファイルの記述形式 473

設定ファイルのパラメーター一覧 474

BJEX_DIR_任意名 (パス名置換識別子とディレクトリのパス名定義) 478

COBOL_LANG (COBOL プログラムを実行するときの LANG 環境変数値の指定) 479

DD_DEALLOC_ON_ALLOCERR (ファイル割り当て処理中にエラーが発生した場合の後処理の方法の定義) 480

DD_EXISTCHECK (DD 要素に指定したファイルパスの存在確認要否を実施するための定義) 481

DD_EXISTCHECK_COBOL (DD 要素に指定したファイルパスの存在確認要否を実施するための定義) 483

DD_SAMF (DD 要素の SAMF 属性での省略値定義) 485

DISPLAY_REPLY_MESSAGE (応答メッセージを JP1/IM - View の画面に出力するための定義) 486

DISPMOD_NOFILE (DD 要素の DISP 属性に MOD を指定したファイルが存在しない場合の処理の定義) 487

DUMMY_FILE_TYPE (DD 要素に TYPE="DUMMY"を指定した場合の割り当てファイルの定義) 488

EUR_DEFINITION_FILE_DIR (EUR 定義ファイル格納ディレクトリの定義) 490

EUR_LANG (EUR のエンコード種別の定義) 491

EUR_PROGRAM (EUR プログラムのパス定義) 492

EXEC_COND_NOSTEPNAME (COND 属性ステップ名未指定時の演算対象定義) 493

EXEC_COND_STEPSEARCH (COND 属性および逆方向参照の検索範囲の定義) 【AIX 限定】 494

EXEC_PARMSPLIT (EXEC 要素の PARMSPLIT 属性での省略値定義) 495

GDG_REGISTER_JOBRC_THRESHOLD (世代ファイルの世代管理ファイルへの登録タイミングの変更と登録させられる条件の定義) 496

ISAMLIB_MODE (ISAM ライブラリ (32 ビット版または 64 ビット版) の定義) 【AIX 限定】 497

JOB_RCDEFINE (ジョブコントローラの終了コードの決定方法定義) 498

JOB_STATUS_DISPLAY (実行中のジョブ情報表示機能を使用するかどうかの定義) 500

JOBLOG_CONVERT (ジョブログ出力時の文字コード変換定義) 501

JOBPARM_JSPECAN (ジョブステップ異常時の打ち切り定義) 503

JP1_HOSTNAME (JP1 が稼働する論理ホスト名定義) 504

JP1EVENT_JOBINFO (JP1 イベントのジョブ情報の指定) 505

JP1EVENT_SUPPRESS_ID (出力抑止する JP1 イベントのイベント ID 定義) 506

JP1FTP_RETRY_INTERVAL (印刷ジョブ転送のリトライ間隔の指定) 507

JP1FTP_RETRY_TIMES (印刷ジョブ転送のリトライ回数の指定) 508

JP1IM_MANAGER_CHKDELAY (イベント到達確認までの待ち時間の指定)	509
JP1IM_MANAGER_HOST (JP1/IM - Manager が稼働するホスト名定義)	510
LOCK_DIR_RECURSIVE (排他制御機能適用時に再帰的にロックを確保するかの定義)	511
LOCK_JOBMPLEX (同時に実行できるジョブ数の定義)	512
LOCK_MAXFILE (排他/共用制御で使用する排他資源管理テーブル数の定義)	513
LOCK_OPTION (排他/共用制御を使用するかの定義)	514
LOCK_KEY_ID (排他制御機能および実行中のジョブ情報表示機能で使用するセマフォ, 共有メモリのキー識別情報の定義)	515
LOCK_RELEASE (排他制御機能適用時の排他解除契機定義)	516
LOG_DIR (実行ログ出力ディレクトリのパス名定義)	517
LOG_FILE_CNT (実行ログをバックアップする面数定義)	518
LOG_FILE_SIZE (実行ログを出力するファイルサイズ定義)	519
MAX_RESPONSE_REQUEST (JP1/IM - Manager に同時に受け付けられる応答待ちイベント数定義)	520
MESSAGE_REPLACE_SPACE (半角空白に置き換える 1 バイトコードの定義)	521
MESSAGE_RETRY_TIMEOUT (メッセージ制御機能でリトライを中断する定義)	522
MSG_KEY_ID (メッセージ制御機能で使用するセマフォ, 共有メモリのキー識別情報の定義)	523
MSGCLASS_DEFAULT (JOB 要素の MSGCLASS 属性での省略値定義)	524
PERMANENT_FILE_DIR (恒久ファイル用ディレクトリのパス名定義)	525
PERMISSION_DIR (PERMISSION 要素の DIR 属性での省略値定義)	526
PERMISSION_FILE (PERMISSION 要素の FILE 属性での省略値定義)	527
PERMISSION_SPOOL_STORE_JOB (保管するスプールジョブの権限の定義)	528
PGM_RCABEND (ジョブステップを異常終了とするユーザプログラムの終了コードの定義)	529
PROCEDURE_DIR (プロシジャライブラリのパス名定義)	530
PROCGROUP_QUEUE_NAME (JP1/AJS のキュー名をプロシジャグループ名として使用する定義)	531
RNW_FILE_METHOD (ファイル再作成の定義)	532
SPOOL_DIR (スプールディレクトリのパス名定義)	533
SPOOL_JOB_STORE (ジョブ終了時にスプールジョブを保管する機能の定義)	534
SPOOL_JOB_TRANSFER (スプールジョブの転送定義)	535
SPOOL_STORE_DIR (スプールジョブの保管先の定義)	536
SPOOL_STORE_DIRNAME (スプールジョブの保管先ディレクトリ名称の定義)	537
SPOOL_STORE_SUBDIR (スプールジョブの保管先のサブディレクトリの定義)	538
STDERRCLASS_DEFAULT (標準エラー出力を格納するファイルの出力クラスの省略値定義)	539
STDOUTCLASS_DEFAULT (標準出力を格納するファイルの出力クラスの省略値定義)	540
SYSOUT_APPEND_WRITE (SYSOUT データファイルへの書き込みモード定義)	541
SysoutClass (出力クラスの属性値定義)	542
TEMP_FILE_DIR (一時ファイルディレクトリのパス名定義)	546
TEMP_FILE_DIR_XML_USE (一時ファイルディレクトリのパス名を DD 要素の DSN 属性で指定可能とする定義)	547
TRACE_DIR (トレースを出力するディレクトリのパス名定義)	548
TRACE_FILE_CNT (トレース面数の定義)	549
TRACE_FILE_SIZE (トレースファイルサイズの定義)	550
TRACE_LEVEL (トレース出力レベルの定義)	551
設定ファイルの使用例	552

9 ジョブ定義 XML ファイル 553

- ジョブ定義 XML ファイルの文法 554
- prolog (XML の宣言) 562
- XML ジョブ制御要素一覧 564
- CALL 要素 (プロシジャの呼び出し) 566
- DD 要素 (ファイルの定義) 567
- ELSE 要素 (IF 要素に対する分岐) 584
- ENDIF 要素 (ジョブステップの実行制御の終了) 585
- EXEC 要素 (プログラムの実行) 586
- HitachiBatchJobExec 要素 (ジョブ定義 XML ファイルの宣言) 593
- IF 要素 (ジョブステップの実行制御の開始) 595
- JOB 要素 (ジョブの属性と終了条件の定義) 599
- JOBPARM 要素 (ジョブ単位の制御情報の定義) 602
- PERMISSION 要素 (ファイルまたはディレクトリのパーミションの定義) 603
- PROC 要素 (プロシジャの定義) 604
- REPLACE 要素 (DD 要素または EXEC 要素の属性の変更内容の定義) 606
- REPLACESTEP 要素 (DD 要素または EXEC 要素の属性を変更対象とするジョブステップの定義) 608
- SETENV 要素 (環境変数の定義) 609
- SETENVFILE 要素 (環境変数格納ファイルの定義) 610
- SETMAXRC 要素 (ジョブステップの終了コードの最大値, または最終ジョブステップの終了コードの変更) 611
- SETSymbOL 要素 (記号パラメータの値の定義) 613
- STEP 要素 (ジョブステップの定義) 614
- SYMBOL 要素 (記号パラメータの定義) 615

10 メッセージと終了コード 616

- 10.1 メッセージの形式 617
 - 10.1.1 メッセージの出力形式 617
 - 10.1.2 メッセージの記載形式 617
 - 10.1.3 メッセージ番号の割り当て 618
 - 10.1.4 システム管理者への指示 619
 - 10.1.5 メッセージの出力 619
- 10.2 メッセージの出力先一覧 620
 - 10.2.1 KAKC で始まるメッセージの出力先一覧 620
 - 10.2.2 KAKD で始まるメッセージの出力先一覧 625
- 10.3 syslog に出力するメッセージの一覧 627
- 10.4 システムコールのエラー情報 628
- 10.5 メッセージ一覧 630
- 10.6 JP1 イベント一覧 801
- 10.7 JP1 イベントの詳細 803
- 10.8 終了コード 823
 - 10.8.1 ジョブステップの終了コード 823

- 10.8.2 ジョブコントローラの終了コード 823
- 10.8.3 スプールジョブ転送サービスの終了コード 827
- 10.8.4 世代コマンドの終了コード 827

付録 828

- 付録 A ジョブ定義 XML ファイル仕様 829
- 付録 A.1 ジョブ定義 XML ファイルの DTD 829
- 付録 B プロセス一覧 834
- 付録 C bjexcollect コマンド (障害調査のための資料の一括収集) 835
 - 付録 C.1 形式 835
 - 付録 C.2 機能 835
 - 付録 C.3 オプション 835
 - 付録 C.4 使用方法 835
 - 付録 C.5 定義ファイルの設定と指定例 836
 - 付録 C.6 注意事項 837
 - 付録 C.7 戻り値 837
 - 付録 C.8 メッセージ 837
- 付録 D メッセージ制御機能のエラー情報 839
- 付録 E このマニュアルの参考情報 844
 - 付録 E.1 関連マニュアル 844
 - 付録 E.2 このマニュアルでの表記 846
 - 付録 E.3 KB (キロバイト) などの単位表記について 847
- 付録 F 用語解説 848

索引 857

1

概要

この章では、バッチジョブ実行システムの導入の目的、特長、前提条件、および全体構成について説明する。

1.1 バッチジョブ実行システムとは

バッチジョブ実行システムは、一定期間のデータを集めて、一括処理をするバッチ処理を対象としたシステムである。今までメインフレームなどでバッチジョブの実行をしていた場合でも、ジョブ制御言語によるバッチジョブイメージを継承したジョブ制御を実現できる。

従来、オープンシステムでのバッチジョブの運用は、OSなどが提供する多彩なコマンド群とシェルプログラミング技術を使って一括処理の手順を作成していたため、プログラム作成と同等の作業量が必要であった。バッチジョブ実行システムを導入することで、バッチジョブの手続きをXML (eXtensible Markup Language) 形式でジョブ定義XMLファイル (定型ドキュメント) として、プログラムとファイルを定義することができ、バッチジョブの定義を簡素化できる。

ジョブ定義XMLファイルでは、ファイルの管理方法が付加できる。ファイルの管理方法には、世代管理や、異常終了時の削除などがある。これによって、シェルプログラムによるファイルの事前準備や後処理が不要となる。また、プログラム間の関連を持たせることで、従来のジョブ制御では実現が難しかったプログラム間のジョブステップ制御を実現でき、バッチジョブの運用が容易になる。

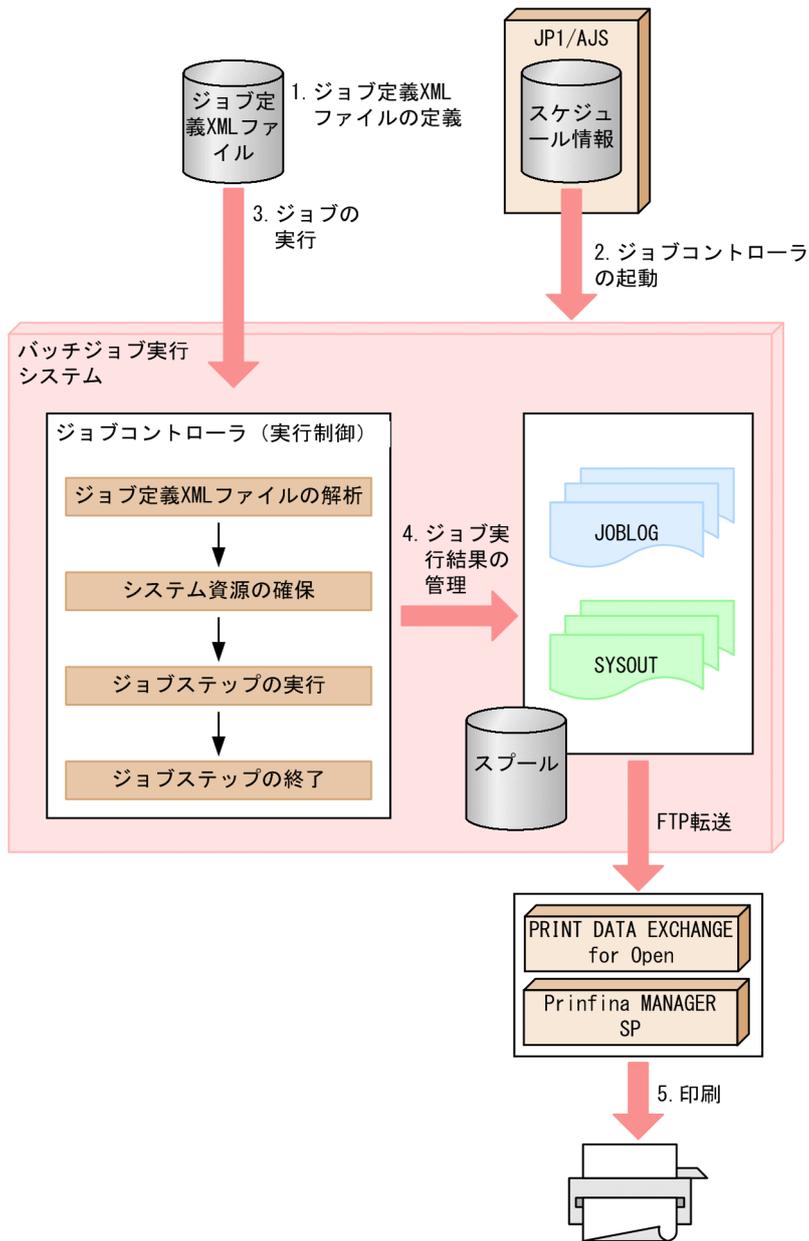
定義されたジョブ定義XMLファイルは、ジョブコントローラで解析される。ジョブコントローラは、入出力装置や各種システム資源の割り当て、および解放処理をして、ジョブの実行・終了を制御する。

さらに、バッチジョブ実行システムは、この定型ドキュメントを実行し、実行結果をスプールに集めて、一元管理できる。また、JP1/AJS および PDE と連携することによって、次に示す運用が実現できる。

- JP1/AJS と連携することによって、ジョブを実行するスケジュールを登録できる。
- PDE と連携することによって、スプールに集めたジョブの実行結果を帳票印刷できる。

バッチジョブ実行システムの処理の流れを次の図に示す (図中の番号は、次に示す説明の項番と対応している)。バッチジョブ実行システムが処理する内容は 1, 3, および 4 が該当する。

図 1-1 バッチジョブ実行システムの処理の流れ



1. バッチジョブ実行システムにジョブの実行を依頼するために、ジョブ定義 XML ファイルを定義する。
2. JP1/AJS で登録されているスケジュールに従い、ジョブコントローラ（実行制御）が起動される。
3. 1 で定義したジョブ定義 XML ファイルの内容に従い、次に示す順序でジョブコントローラがジョブを実行する。
ジョブ定義 XML ファイルの解析→システム資源の確保→ジョブステップの実行→ジョブステップの終了
4. ジョブの実行結果をスプールに集めて、一元管理する。
5. スプールに管理されているジョブの実行結果を PDE と連携し、帳票出力する。

1.2 バッチジョブ実行システムの特長

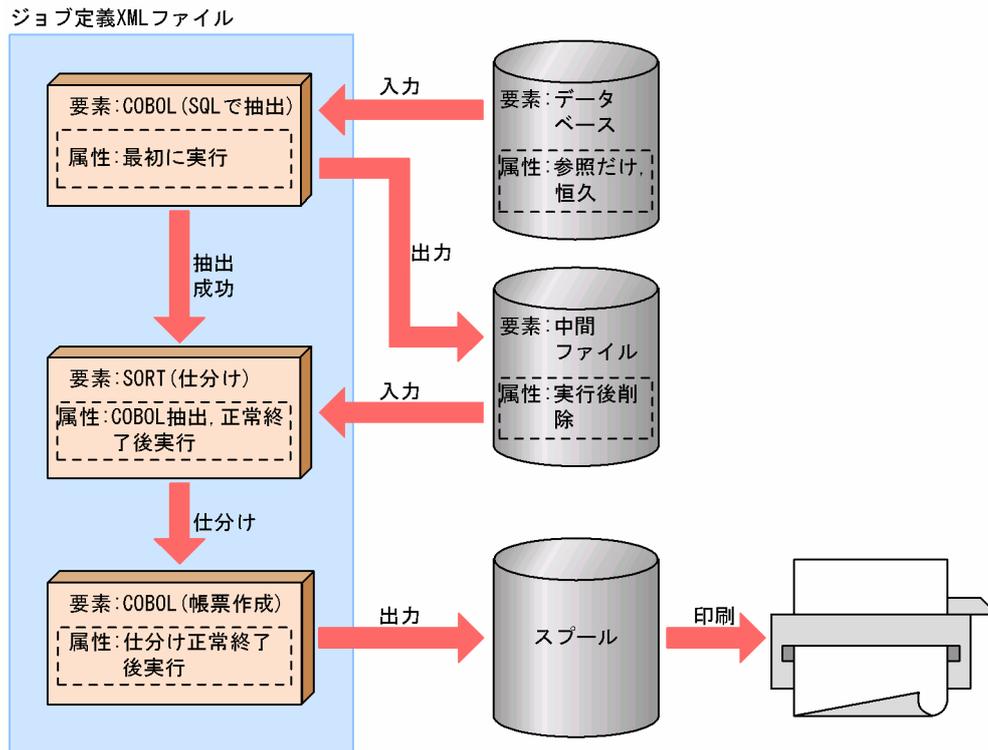
バッチジョブ実行システムの特長を次に示す。

- 多様なバッチジョブ業務を定型化して幅広く対応できる。
- バッチジョブ業務として実行結果を管理できる。
- バッチジョブ業務の結果を帳票印刷運用と連携できる。

1.2.1 バッチジョブ定義の定型化

バッチジョブ業務は次の図に示すように、「プログラム」と「ファイル」の組み合わせによって一括処理を実現する。それぞれのプログラムとファイルには順序処理だけではなく、前処理、後処理があり、このことをプログラムやファイルなどのバッチジョブでの個々の要素ごとに属性として定義すれば、バッチジョブ業務をプログラミングすることなく容易に実現できる。このバッチジョブ業務を標準的なドキュメントの表現方式であるXML形式（ジョブ定義XMLファイル）でジョブの定義を行うことで、定型化を図ることができる。

図 1-2 プログラムとファイルの組み合わせによるバッチジョブの一括処理



• ファイルの要素への属性定義

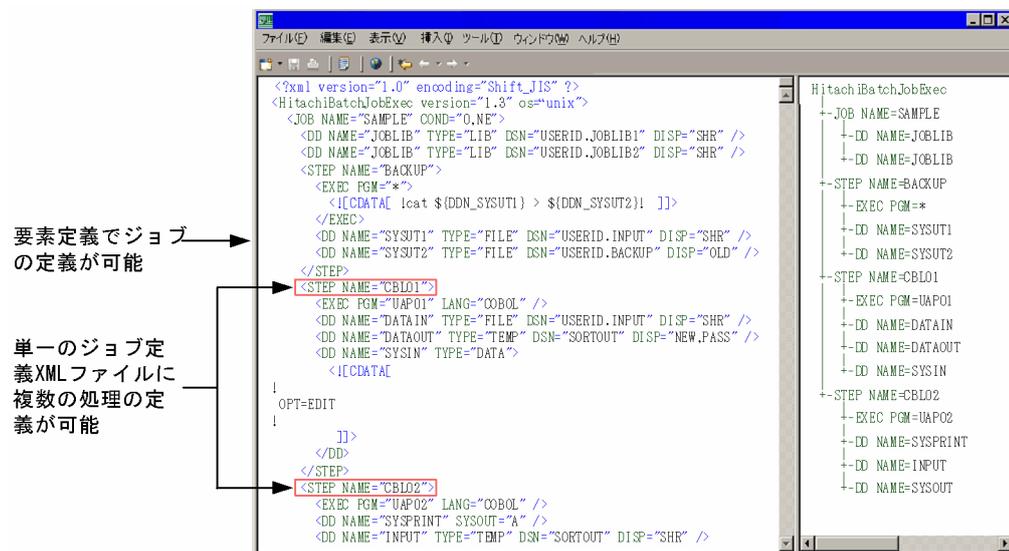
ファイルの要素に対して、世代ファイルおよび一時ファイルなどの属性を持たせることができる。実行終了後のファイルの削除なども要素の定義だけで実行できるため、シェルスクリプトの記述も必要なく、運用の変更やメンテナンスが容易となる。

- 複数のプログラム要素への属性定義

プログラムの要素に対して、実行したプログラムの結果を判断して実行可否属性の定義ができる。これによって、1つのバッチジョブ内で複数（ジョブステップ）の処理が制御できる。

複数のプログラム要素への属性定義を次の図に示す。

図 1-3 複数のプログラム要素への属性定義

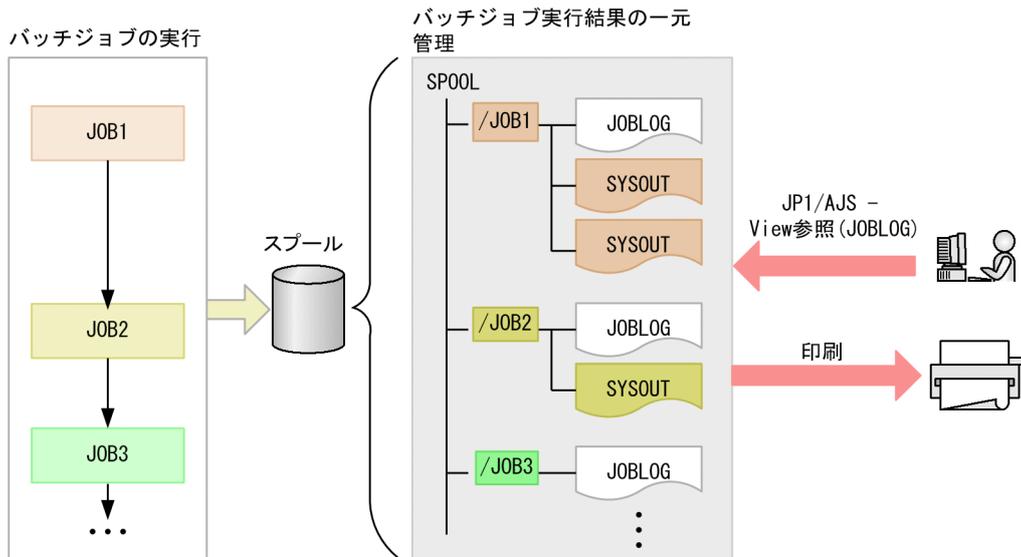


1.2.2 バッチジョブの実行結果の一元管理

従来、オープンシステムでのバッチジョブの実行結果は、格納先が一元化されていなかったため管理が煩雑であった。バッチジョブ実行システムでは、バッチジョブの実行結果をスプールに集めることができる。これによってバッチジョブの実行結果を一元管理できる。また、JP1/AJS - View を使用することで、バッチジョブの実行結果も参照できる。

バッチジョブの実行結果の一元管理を次の図に示す。

図 1-4 バッチジョブの実行結果の一元管理



(凡例)

JP1/AJS - Viewで参照できるJOBLOGデータ

1.2.3 PDE との連携による帳票出力

メインフレームの帳票印刷業務をオープン環境上で実現する PDE と、バッチジョブ実行システムが連携することで、書式、FCB (Forms Control Buffer) を使用した帳票出力ができる。

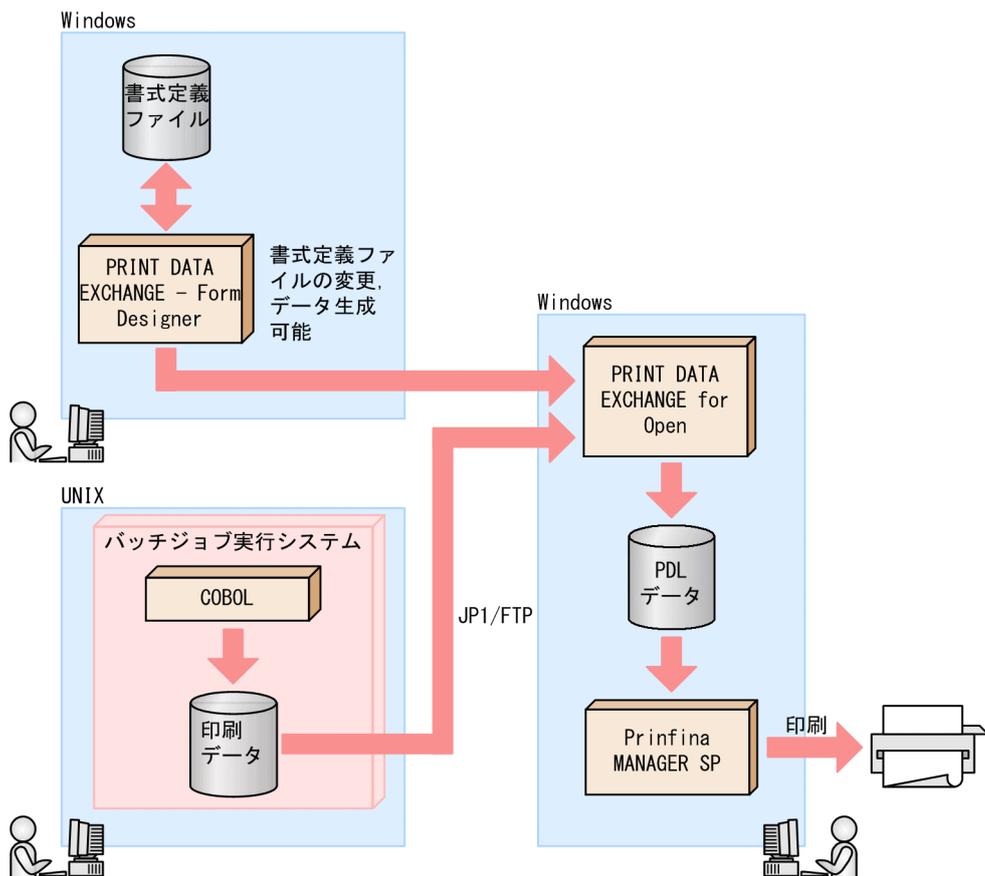
PRINT DATA EXCHANGE - Form Designer で書式や FCB を定義する書式定義ファイルを作成[※]し、PRINT DATA EXCHANGE for Open が PRINT DATA EXCHANGE - Form Designer で作成した書式定義ファイルとバッチジョブ実行システムの実行結果である印刷データを結合する。結合されたデータを PDL (Page Description Language) データに変換することで、プリンタ管理ソフトウェアである Prinfina MANAGER SP を通して帳票出力できる。

注※ PC 上での書式定義の変更もできる。

なお、PRINT DATA EXCHANGE - Form Designer の詳細については、マニュアル「PRINT DATA EXCHANGE - Form Designer」、PRINT DATA EXCHANGE for Open の詳細については、マニュアル「PRINT DATA EXCHANGE for Open」を参照のこと。

PDE との連携による帳票出力を次の図に示す。

図 1-5 PDE との連携による帳票出力



1.3 前提条件

1.3.1 前提プログラムおよび関連プログラム

バッチジョブ実行システムの前提プログラムおよび関連プログラムを次の表に示す。

表 1-1 バッチジョブ実行システムの前提プログラム【AIX の場合】

プログラム	組み込むシステム
AIX V7.1, AIX V7.2, または AIX V7.3	バッチジョブ実行システムと同じサーバ
JP1/Base	
JP1/Automatic Job Management System 3 - Agent*	
JP1/Automatic Job Management System 3 - Manager*	運用管理サーバ

注※ JP1/AJS - Manager には、JP1/AJS - Agent の機能が含まれているため、バッチジョブ実行システムと同一サーバ上に JP1/AJS - Manager があれば、JP1/AJS - Agent は不要である。

表 1-2 バッチジョブ実行システムの前提プログラム【Linux の場合】

プログラム	組み込むシステム
Red Hat Enterprise Linux Server 7, Red Hat Enterprise Linux Server 8, または Red Hat Enterprise Linux Server 9	バッチジョブ実行システムと同じサーバ
JP1/Base	
JP1/Automatic Job Management System 3 - Agent*	
JP1/Automatic Job Management System 3 - Manager*	運用管理サーバ

注※ JP1/AJS - Manager には、JP1/AJS - Agent の機能が含まれているため、バッチジョブ実行システムと同一サーバ上に JP1/AJS - Manager があれば、JP1/AJS - Agent は不要である。

表 1-3 バッチジョブ実行システムの関連プログラム【AIX の場合】

プログラム	組み込むシステム
COBOL2002 Net Server Suite または COBOL2002 Net Server Suite(64)*	バッチジョブ実行システムと同じサーバ
COBOL2002 Net Server Runtime または COBOL2002 Net Server Runtime(64)*	
ISAM Version7 または ISAM Version7(64)	
SORT Version9 または SORT Version9(64)	
SORT Version9 - Extended Edition または SORT Version9 - Extended Edition(64)	

プログラム	組み込むシステム
JP1/File Transmission Server/FTP	バッチジョブ実行システムと同じサーバ
JP1/Script	
uCosminexus Application Server Enterprise	
Hitachi Code Converter - Runtime for C/COBOL(64)	
EUR Print Service	プリントサーバ (バッチジョブ実行システムと同じサーバでも別のサーバでも良い)
EUR Print Manager	
JP1/Integrated Management 2 - Manager または JP1/Integrated Management 3 - Manager	JP1 イベント管理サーバ
JP1/Automatic Job Management System 3 - View	ジョブネットおよびスケジュール情報管理サーバ
JP1/Integrated Management 2 - View または JP1/Integrated Management 3 - View	JP1 イベント監視サーバ
PRINT DATA EXCHANGE - Electronic Document Option	プリントサーバ
PRINT DATA EXCHANGE - Enterprise Printer Option	
PRINT DATA EXCHANGE - Form Designer	
PRINT DATA EXCHANGE for Open	
Prinfina MANAGER SP	
JP1/File Transmission Server/FTP	

注※ 別サーバの COBOL2002 でコンパイル済みであれば、Runtime の導入で実行できる。

表 1-4 バッチジョブ実行システムの関連プログラム 【Linux の場合】

プログラム	組み込むシステム
COBOL2002 Net Server Suite(64) [※]	バッチジョブ実行システムと同じサーバ
COBOL2002 Net Server Runtime(64) [※]	
ISAM Version7(64)	
SORT Version9(64)	
SORT Version9 - Extended Edition(64)	
JP1/File Transmission Server/FTP	
uCosminexus Application Server Enterprise	
Hitachi Code Converter - Runtime for C/COBOL(64)	
EUR Print Service	プリントサーバ (バッチジョブ実行システムと同じサーバでも別のサーバでも良い)
EUR Print Manager	

プログラム	組み込むシステム
JP1/Integrated Management 2 - Manager または JP1/Integrated Management 3 - Manager	JP1 イベント管理サーバ
JP1/Automatic Job Management System 3 - View	ジョブネットおよびスケジュール情報管理サーバ
JP1/Integrated Management 2 - View または JP1/Integrated Management 3 - View	JP1 イベント監視サーバ
PRINT DATA EXCHANGE - Electronic Document Option	プリントサーバ
PRINT DATA EXCHANGE - Enterprise Printer Option	
PRINT DATA EXCHANGE - Form Designer	
PRINT DATA EXCHANGE for Open	
Prinfina MANAGER SP	
JP1/File Transmission Server/FTP	

注※ 別サーバの COBOL2002 でコンパイル済みであれば、Runtime の導入で実行できる。

各プログラムの役割を次に示す。

- **AIX V7.1, AIX V7.2, AIX V7.3, Red Hat Enterprise Linux 7, Red Hat Enterprise Linux 8, Red Hat Enterprise Linux 9**
バッチジョブなどを実行するプラットフォームの OS である。
- **JP1/Base**
ユーザの管理、バッチジョブなどのイベントの収集・管理ができる。
- **JP1/Automatic Job Management System 3 - Agent**
JP1/AJS - Manager から実行依頼を受けたバッチジョブなどの処理を実行できる。
- **JP1/Automatic Job Management System 3 - Manager**
ジョブネットの定義情報やスケジュール情報を管理し、バッチジョブなどの処理の実行をエージェント (JP1/AJS - Agent または JP1/AJS - Manager) に依頼できる。また、JP1/AJS - Manager 自身で処理の実行もできる。
- **COBOL2002**
COBOL は、事務処理用に最も使用されているプログラミング言語である。
- **ISAM**
索引順編成ファイルを作成できる。
- **SORT EE**
選択、ソート、マージ、および集約などの一連の処理を一度に実行できる。
- **JP1/File Transmission Server/FTP**
運用管理機能を備えたファイル伝送プログラムである。

- **JP1/Script**
JP1/Script を使用する場合に必要なプログラムである。
- **uCosminexus Application Server Enterprise**
Java のバッチアプリケーションを使用する場合に必要なプログラムである。
- **Hitachi Code Converter**
ジョブログを出力する際、実行環境に合わせた文字コードに変換できる。
- **EUR Print Service**
EUR Print Service と連携して使用する場合に必要なプログラムである。
- **EUR Print Manager**
EUR Print Manager と連携して使用する場合に必要なプログラムである。
- **JP1/Automatic Job Management System 3 - View**
JP1/AJS - Manager を GUI で操作できる。
- **JP1/IM - Manager**
統合コンソールおよび統合スコープのマネージャ機能を備えていて、バッチジョブなどで発生したイベントを管理できる。
- **JP1/IM - View**
統合コンソールの操作および統合スコープの操作画面を使用して、システムの監視・操作ができる。
- **PRINT DATA EXCHANGE - Electronic Document Option**
印刷データの PDF 変換および電子帳票を生成できる。
- **PRINT DATA EXCHANGE - Enterprise Printer Option**
Prinfina MANAGER SP との連携ができる。
- **PRINT DATA EXCHANGE - Form Designer**
書式オーバーレイの設計、FCB 定義による用紙サイズ、両面指定、行データの定義、および行データの印刷で適用する文字サイズや書体などの初期値を定義できる。
- **PRINT DATA EXCHANGE for Open**
バッチジョブの帳票出力に対し、書式、FCB などのプリンタ制御イメージを付加したオープン環境の印刷データを作成できる。

1.3.2 ファイルシステム

バッチジョブを実行する場合に、ファイルは信頼性の高いファイルシステムを使用する必要がある。

また、バッチジョブ実行システムは、NFS 上に配置できるファイルが限られている。NFS 上に配置できるファイルおよび指定できる設定ファイル中のパラメータを次に示す。また、別サーバから同じファイルまたはディレクトリを共用できるかを示す。

表 1-5 NFS のサポート内容

項目	NFS 設置 可否	共用 可否	不可, 非推奨の理由
ジョブ定義 XML ファイル	○	○	
設定ファイル	×	×	<ul style="list-style-type: none"> インストールディレクトリ固定のため設置できない。 バッチジョブ実行システムごとに別々の設定ファイルとなるため共用できない。
ジョブ定義 XML ファイルの EXEC 要素に指定する実行プログラム, および実行プログラムがアクセスするユーザファイル	○	○	
LOG_DIR パラメータ, および実行ログファイル	×	×	<ul style="list-style-type: none"> 障害調査でバッチジョブ実行システム環境ごとの実行ログファイルが必要なため共用できない。
PERMANENT_FILE_DIR パラメータ, および DD 要素で割り当てられるファイル	○	○*	
PROCEDURE_DIR パラメータ, およびプロシジャ XML ファイル	○	○	
SPOOL_DIR パラメータ, および SYSOUT データファイル	×	×	<ul style="list-style-type: none"> 複数のバッチジョブ実行システムで同じディレクトリを指定した場合スプールジョブ転送サービスの起動時にエラーにする。
SPOOL_STORE_DIR パラメータ	○	○	
TEMP_FILE_DIR パラメータ	△	×	<ul style="list-style-type: none"> 共用するファイルではないため指定を推奨しない。 一時ファイルが重複する可能性があるため共用できない。
TRACE_DIR パラメータ, およびトレースファイル	×	×	<ul style="list-style-type: none"> 障害調査でバッチジョブ実行システム環境ごとのトレースファイルが必要なため共用できない。

凡例

設置可否

- ：設置または指定できる
- △：設置または指定できるが推奨しない
- ×

共用可否

- ：複数のバッチジョブ実行システムで共用できる
- ×

注※ 恒久ファイル, 世代ファイルを NFS に設置する場合, 次の注意事項がある。

- バッチジョブ実行システムの排他制御機能は別サーバ間では有効にならないため別サーバで同じファイルにアクセスする場合, 注意が必要である

上記以外のファイル, ディレクトリについては動作を保証しない。

1.3.3 2GB 超えのファイル（ラージファイル）の扱い

バッチジョブ実行システムでは、ラージファイルを扱うことができる。

バッチジョブ実行システムで利用できるファイルのうち、ラージファイルに対応するファイルを次に示す。

- DD 要素で割り当てるファイル※（TYPE=TEMPISAM,DATA 以外）
- SYSOUT データファイル※

注※ UAP がラージファイルを扱えることが前提である。

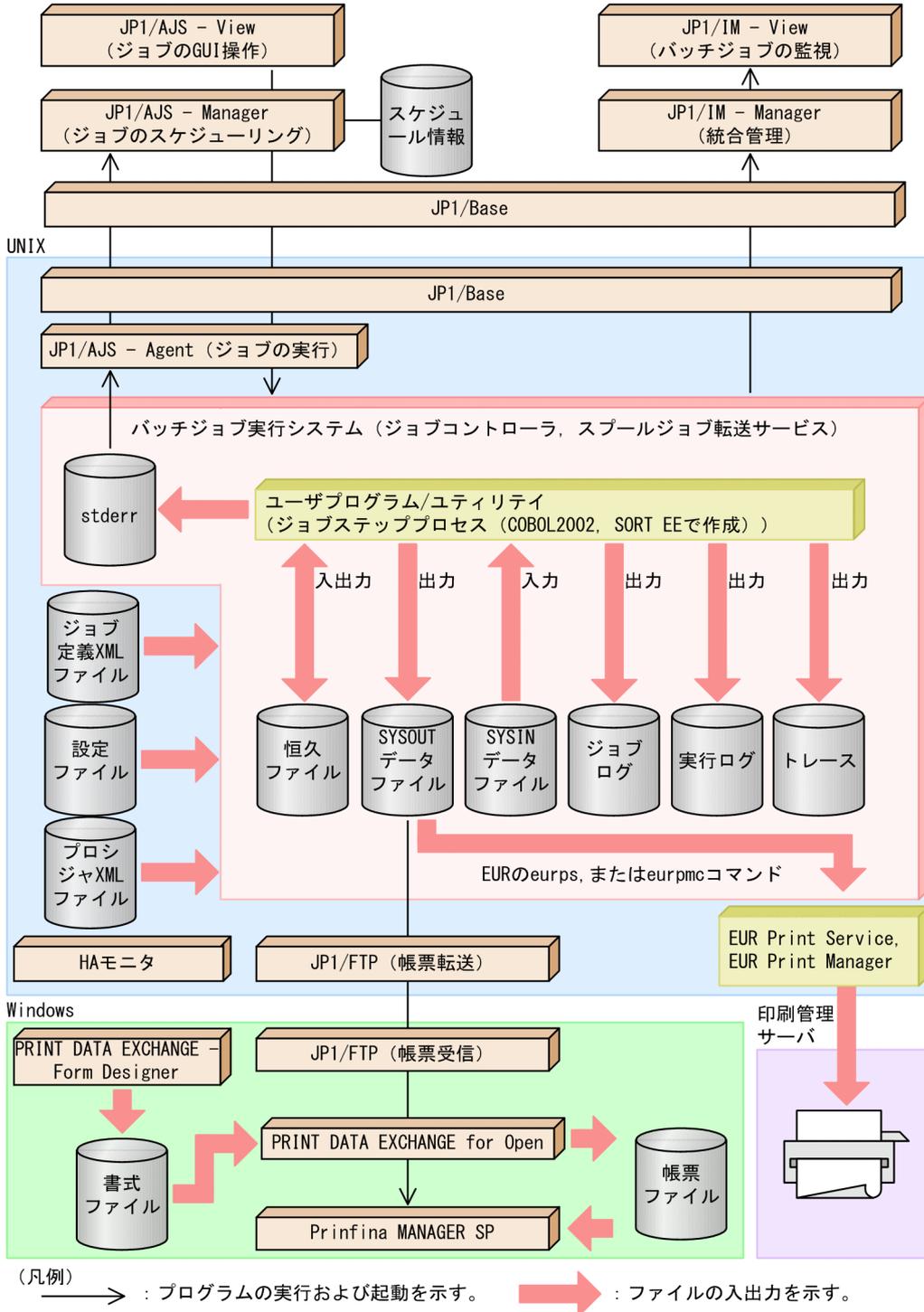
ラージファイルの使用可否はファイルシステムの種類や OS の設定（例: ulimit の設定）によって異なるため、ラージファイルが利用できる環境であることを運用設計前に確認すること。

1.4 バッチジョブ実行システムの全体構成

バッチジョブ実行システムは、JP1/AJS から起動されジョブを実行するジョブコントローラおよび、スプールの管理とジョブの出力する帳票をプリントサーバに転送するスプールジョブ転送サービスから成る。スプールジョブ転送サービスは、スプールジョブ転送デーモンとスプールジョブ管理コマンドから成る。

バッチジョブ実行システムの全体構成を次の図に示す。

図 1-6 バッチジョブ実行システムの全体構成



2

インストール

この章では、インストールの準備と手順、および見積もりについて説明する。

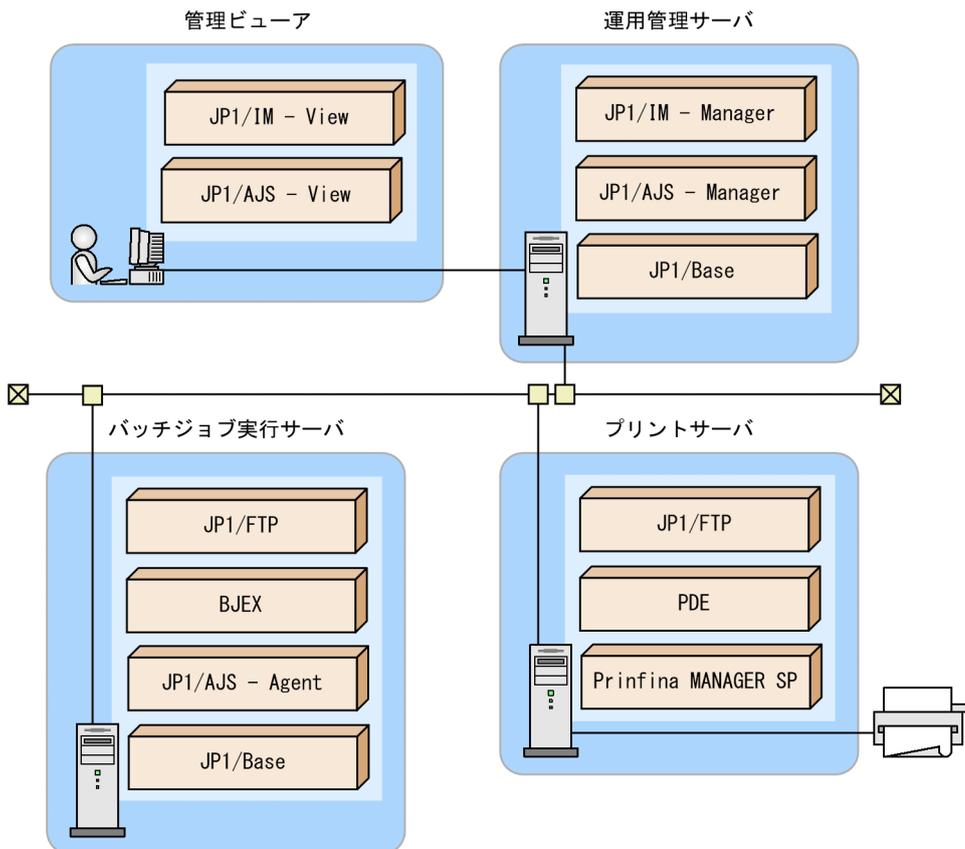
2.1 バッチジョブ実行システムのセットアップ手順の概要

ここでは、バッチジョブを実行して帳票出力する基本的なシステム構成とセットアップ手順の概要について説明する。

2.1.1 基本システム構成の概要とセットアップ手順

JP1/AJS からバッチジョブを実行して、帳票出力する基本的なシステム構成例を次に示す。

図 2-1 バッチジョブを実行して帳票出力する基本的なシステム構成例



JP1/AJS からバッチジョブを実行して、帳票出力するためのセットアップを次に示す。

(1) 運用管理サーバでのセットアップ作業

1. 運用管理サーバに JP1/Base, JP1/AJS - Manager, および JP1/IM - Manager のインストールとセットアップをする

JP1/Base, JP1/AJS - Manager, および JP1/IM - Manager のインストールとセットアップについては、JP1 のマニュアルを参照のこと。

(2) バッチジョブ実行サーバでのセットアップ作業

1. バッチジョブ実行サーバに JP1/Base, JP1/AJS - Agent のインストールとセットアップをする
JP1/Base, および JP1/AJS - Agent のインストールとセットアップについては, JP1 のマニュアルを参照のこと。
2. バッチジョブ実行サーバに JP1/FTP のインストールとセットアップをする
JP1/FTP のインストールとセットアップについては, JP1 のマニュアルを参照のこと。
3. バッチジョブ実行サーバにバッチジョブ実行システムのインストールとセットアップをする
バッチジョブ実行システムのインストールとセットアップについては, 「2.2 インストール」, および「2.3 セットアップ」を参照のこと。

(3) プリントサーバでのセットアップ作業

(a) 高性能プリンタへの Prinfina MANAGER SP 連携出力の場合

1. プリントサーバに Prinfina MANAGER SP および PDE のインストールとセットアップをする
2. プリントサーバに JP1/FTP のインストールとセットアップをする
JP1 イベントを使用した統合管理実施時には, JP1/Base のインストール・セットアップも必要である。
JP1/FTP のインストールとセットアップについては, JP1 のマニュアルを参照のこと。
3. バッチジョブ実行サーバから印刷データを転送するための受信フォルダを仮想ディレクトリとして設定する
4. JP1/FTP を起動する
JP1 イベントを使用した統合管理実施時には, JP1/Base の起動も必要である。
5. Prinfina MANAGER SP サービスを起動する

(b) PDF ファイル出力および Windows 接続プリンタへの GDI (Graphic Device Interface) 出力の場合

1. プリントサーバに PDE のインストールとセットアップをする
2. プリントサーバに JP1/FTP のインストールとセットアップをする
JP1 イベントを使用した統合管理実施時には, JP1/Base のインストールとセットアップも必要である。
JP1/FTP のインストールとセットアップについては, JP1 のマニュアルを参照のこと。
3. バッチジョブ実行サーバから印刷データを転送するための受信フォルダを仮想ディレクトリとして設定する
4. JP1/FTP を起動する

JP1 イベントを使用した統合管理実施時には、JP1/Base の起動も必要である。

5. PDE サービスを起動する

(4) 管理ビューアの設定

1. 管理ビューアに JP1/AJS - View および JP1/IM - View のインストールとセットアップをする
JP1/AJS - View と JP1/IM - View のインストールについては、JP1 のマニュアルを参照のこと。

2.2 インストール

ここでは、バッチジョブ実行システムのインストール方法について説明する。

2.2.1 バッチジョブ実行システムのインストール

(1) AIX の場合

AIX へのインストール手順を次に示す。

なお、CD-ROM のディレクトリ名やファイル名は、ハードウェア環境などによって記述した内容と見え方が異なる。ls コマンドで確認し、表示されたファイル名をそのまま入力すること。

1. ユーザ権限を設定する

バッチジョブ実行システムをインストールするホストに、スーパーユーザでログインする。または、su コマンドでユーザ権限をスーパーユーザに変更する。

2. すべてのプログラムを終了する

既存の JP1 シリーズのプログラム、およびバッチジョブ実行システムのプログラムが動作している場合、必ず停止する。

3. バッチジョブ実行システムの媒体をセットする

4. 次のコマンドを実行して、CD-ROM 装置をマウントする

```
/usr/sbin/mount -r -v cdrfs /dev/cd0 /cdrom
```

/cdrom は CD-ROM デバイススペシャルファイルのマウントポイントである。マウントポイントディレクトリがない場合は、作成すること。なお、デバイススペシャルファイル名、およびマウントポイントは、使用する環境によって異なる場合がある。

5. 次のコマンドを実行して、Hitachi PP Installer を起動する

```
/cdrom/AIX/SETUP /cdrom※
```

注※ ここでは、マウントポイントに/cdrom を仮定する。

Hitachi PP Installer が起動し、初期画面が表示される。

Hitachi PP Installer の初期画面の例を次に示す。

```
Hitachi PP Installer 05-34

L) List Installed Software.
I) Install Software.
D) Delete Software.
Q) Quit.
```

```
Select Procedure ==>
```

6. Hitachi PP Installer の初期画面で 「I」 を入力する

インストールできるプログラムの一覧が表示される。

7. バッチジョブ実行システムを選択して 「I」 を入力する

バッチジョブ実行システムがインストールされる。なお、プログラムを選択するには、カーソルを移動させてスペースバーで選択する。

インストール画面の例を次に示す。

```
      PP-No.          VR   PP-NAME
<@>001 P-1M12-A291    0910 uCosminexus Batch Job Execution Server
:
:
F) Forward B) Backward J) Down K) Up Space) Select/Unselect I) Install Q) Quit
```

選択した PP の左側に、 "<@>" が表示される。続いて 「I」 を入力すると、最下行に次に示すメッセージが表示される。

```
Install PP? (y: install, n: cancel)==>
```

ここで、 [y] または [Y] を選択するとインストールが開始される。 [n] または [N] を選択すると、インストールが中止され、PP インストール画面に戻る。

8. インストールが正常終了したら、「Q」を入力する

Hitachi PP Installer の初期画面に戻る。

なお、インストール時にインストーラーのログとして次のファイルが作成される。

```
/opt/hitachi/bjex/instlog/BJEX_INST_LOG , /opt/hitachi/bjex/instlog/BJEX_INST_USERLOG
```

インストーラーのログファイルが作成されていない場合、次に示す問題が考えられる。

- インストーラーのログファイルが通常のファイルでない。
- インストーラーのログファイルを作成するディレクトリに書き込み権限がない。
- インストーラーのログファイルのパス名を構成する各ファイルパスに、同一のファイルが存在する。同一のファイルが存在する状態を次に示す。

"/opt"がディレクトリでない。

"/opt/hitachi"がディレクトリでない。

"/opt/hitachi/bjex"がディレクトリでない。

"/opt/hitachi/bjex/instlog"がディレクトリでない。

インストールが完了すると、デフォルトの環境が設定されている。デフォルトから設定を変更する場合には、「[2.3 セットアップ](#)」を参照のこと。

(2) Linux の場合

Linux へのインストール手順を次に示す。

なお、CD-ROM のディレクトリ名やファイル名は、ハードウェア環境などによって記述した内容と見え方が異なる。ls コマンドで確認し、表示されたファイル名をそのまま入力すること。

1. ユーザ権限を設定する

バッチジョブ実行システムをインストールするホストに、スーパーユーザでログインする。または、su コマンドでユーザ権限をスーパーユーザに変更する。

2. すべてのプログラムを終了する

既存の JP1 シリーズのプログラム、およびバッチジョブ実行システムのプログラムが動作している場合、必ず停止する。

3. バッチジョブ実行システムの媒体をセットする

4. 次のコマンドを実行して、CD-ROM 装置をマウントする

```
/bin/mount -r -o mode=0544 /dev/cdrom /mnt/cdrom
```

/mnt/cdrom は CD-ROM デバイススペシャルファイルのマウントポイントである。マウントポイントディレクトリがない場合は、作成すること。なお、デバイススペシャルファイル名、およびマウントポイントは、使用する環境によって異なる場合がある。

5. 次のコマンドを実行して、Hitachi PP Installer を起動する

```
/cdrom/X64LIN/SETUP /cdrom※
```

注※ ここでは、マウントポイントに/cdrom を仮定する。

Hitachi PP Installer が起動し、初期画面が表示される。

Hitachi PP Installer の初期画面の例を次に示す。

```
Hitachi PP Installer 05-34

L) List Installed Software.
I) Install Software.
D) Delete Software.
Q) Quit.

Select Procedure ==>
```

6. Hitachi PP Installer の初期画面で「I」を入力する

インストールできるプログラムの一覧が表示される。

7. バッチジョブ実行システムを選択して「I」を入力する

バッチジョブ実行システムがインストールされる。なお、プログラムを選択するには、カーソルを移動させてスペースバーで選択する。

インストール画面の例を次に示す。

```
      PP-No.          VR    PP-NAME
<@>001 P-8412-A291    0910  uCosminexus Batch Job Execution Server
:
:
F) Forward B) Backward J) Down K) Up Space) Select/Unselect I) Install Q) Quit
```

選択した PP の左側に、"<@>"が表示される。続いて [I] を入力すると、最下行に次に示すメッセージが表示される。

```
Install PP? (y: install, n: cancel)==>
```

ここで、[y] または [Y] を選択するとインストールが開始される。[n] または [N] を選択すると、インストールが中止され、PP インストール画面に戻る。

8. インストールが正常終了したら、「Q」を入力する

Hitachi PP Installer の初期画面に戻る。

なお、インストール時にインストーラーのログとして次のファイルが作成される。

```
/opt/hitachi/bjex/instlog/BJEX_INST_LOG , /opt/hitachi/bjex/instlog/BJEX_INST_USERLOG
```

インストーラーのログファイルが作成されていない場合、次に示す問題が考えられる。

- インストーラーのログファイルが通常のファイルでない。
- インストーラーのログファイルを作成するディレクトリに書き込み権限がない。
- インストーラーのログファイルのパス名を構成する各ファイルパスに、同一のファイルが存在する。同一のファイルが存在する状態を次に示す。
 - "/opt"がディレクトリでない。
 - "/opt/hitachi"がディレクトリでない。
 - "/opt/hitachi/bjex"がディレクトリでない。
 - "/opt/hitachi/bjex/instlog"がディレクトリでない。

インストールが完了すると、デフォルトの環境が設定されている。デフォルトから設定を変更する場合には、「[2.3 セットアップ](#)」を参照のこと。

2.2.2 バッチジョブ実行システムのアンインストール

(1) AIX の場合

アンインストール手順を次に示す。Hitachi PP Installer の指示に従ってバッチジョブ実行システムをアンインストールする。

バッチジョブ実行システムをアンインストールする場合は、ジョブコントローラを利用するプログラムとスプールジョブ転送サービスをすべて終了すること。アンインストールでは、ユーザが変更を行うファイル、および新規に作成したファイルは削除されない。したがって、完全に環境を削除するには、ユーザが自分で削除する必要がある。

1. 次のコマンドを実行して、Hitachi PP Installer を起動する

```
/etc/hitachi_setup
```

Hitachi PP Installer が起動し、初期画面が表示される。

Hitachi PP Installer の初期画面の例を次に示す。

```
Hitachi PP Installer 05-34

L) List Installed Software.
I) Install Software.
D) Delete Software.
Q) Quit.

Select Procedure ==>
```

2. Hitachi PP Installer の初期画面で「D」を入力する

アンインストールできるソフトウェアの一覧が表示される。

3. バッチジョブ実行システムを選択して「D」を入力する

バッチジョブ実行システムがアンインストールされる。なお、プログラムを選択するには、カーソルを移動させてスペースバーで選択する。

アンインストール画面の例を次に示す。

```
      PP-No.          VR    PP-NAME
001 P-CC1M12-43DL    1300  JP1/AJS3 - Agent
002 P-1M12-A291      0910  uCosminexus Batch Job Execution Server
:
:
F) Forward B) Backward J) Down K) Up Space) Select/Unselect D) Delete Q) Quit
```

続いて [D] を入力すると、最下行に次に示すメッセージが表示される。

```
Delete PP? (y: delete, n: cancel) ==>
```

ここで、[y] または [Y] を選択するとアンインストールが開始される。[n] または [N] を選択すると、アンインストールが中止され、PP アンインストール画面に戻る。

4. アンインストールが正常終了したら、「Q」を入力する

Hitachi PP Installer の初期画面に戻る。

なお、アンインストール時にインストーラーのログとして次のファイルが作成される。

```
/opt/hitachi/bjex/instlog/BJEX_INST_LOG , /opt/hitachi/bjex/instlog/BJEX_INST_USERLOG
```

インストーラーのログファイルが作成されていない場合、次に示す問題が考えられる。

- インストーラーのログファイルが通常のファイルでない。
- インストーラーのログファイルを作成するディレクトリに書き込み権限がない。
- インストーラーのログファイルのパス名を構成する各ファイルパスに、同一のファイルが存在する。同一のファイルが存在する状態を次に示す。
 - "/opt"がディレクトリでない。
 - "/opt/hitachi"がディレクトリでない。
 - "/opt/hitachi/bjex"がディレクトリでない。
 - "/opt/hitachi/bjex/instlog"がディレクトリでない。

アンインストールが完了したあと、スプールジョブ転送デーモンの自動起動および自動終了の設定を解除する。

1. 次のコマンドを実行して、スプールジョブ転送デーモンの自動起動の設定を解除する

```
rmitab bjexpd
```

2. 論理ホスト用のスプールジョブ転送デーモンの自動起動を設定している場合は、論理ホスト用スプールジョブ転送デーモンのレコードを指定して rmitab コマンドを実行する

3. システム終了時の自動終了機能を解除するには、/etc/rc.shutdown の次に示す記述を削除する

```
test -x /opt/hitachi/bjex/sbin/bjexpdctl && /opt/hitachi/bjex/sbin/bjexpdctl stop
```

4. 論理ホスト用のスプールジョブ転送デーモンの自動起動を設定している場合は、論理ホスト用スプールジョブ転送デーモンの自動停止の記述を/etc/rc.shutdown から削除する

(2) Linux の場合

アンインストール手順を次に示す。Hitachi PP Installer の指示に従ってバッチジョブ実行システムをアンインストールする。

バッチジョブ実行システムをアンインストールする場合は、ジョブコントローラを利用するプログラムとスプールジョブ転送サービスをすべて終了すること。アンインストールでは、ユーザが変更を行うファイル、および新規に作成したファイルは削除されない。したがって、完全に環境を削除するには、ユーザが自分で削除する必要がある。

1. 次のコマンドを実行して、Hitachi PP Installer を起動する

```
/etc/hitachi_setup
```

Hitachi PP Installer が起動し、初期画面が表示される。

Hitachi PP Installer の初期画面の例を次に示す。

```
Hitachi PP Installer 05-34
```

```
L) List Installed Software.  
I) Install Software.  
D) Delete Software.  
Q) Quit.
```

```
Select Procedure ==>
```

2. Hitachi PP Installer の初期画面で「D」を入力する

アンインストールできるソフトウェアの一覧が表示される。

3. バッチジョブ実行システムを選択して「D」を入力する

バッチジョブ実行システムがアンインストールされる。なお、プログラムを選択するには、カーソルを移動させてスペースバーで選択する。

アンインストール画面の例を次に示す。

```
      PP-No.          VR   PP-NAME  
001 P-CC8412-4KDL   1300  JP1/AJS3 - Manager  
002 P-8412-A291    0910  uCosminexus Batch Job Execution Server  
:  
:  
F) Forward B) Backward J) Down K) Up Space) Select/Unselect D) Delete Q) Quit
```

続いて「D」を入力すると、最下行に次に示すメッセージが表示される。

```
Delete PP? (y: delete, n: cancel) ==>
```

ここで、「y」または「Y」を選択するとアンインストールが開始される。「n」または「N」を選択すると、アンインストールが中止され、PP アンインストール画面に戻る。

4. アンインストールが正常終了したら、「Q」を入力する

Hitachi PP Installer の初期画面に戻る。

なお、アンインストール時にインストーラーのログとして次のファイルが作成される。

```
/opt/hitachi/bjex/instlog/BJEX_INST_LOG , /opt/hitachi/bjex/instlog/BJEX_INST_USERLOG
```

インストーラーのログファイルが作成されていない場合、次に示す問題が考えられる。

- インストーラーのログファイルが通常のファイルでない。
- インストーラーのログファイルを作成するディレクトリに書き込み権限がない。
- インストーラーのログファイルのパス名を構成する各ファイルパスに、同一のファイルが存在する。同一のファイルが存在する状態を次に示す。

"/opt"がディレクトリでない。

"/opt/hitachi"がディレクトリでない。

"/opt/hitachi/bjex"がディレクトリでない。

"/opt/hitachi/bjex/instlog"がディレクトリでない。

アンインストールが完了したあと、スプールジョブ転送デーモンの自動起動および自動終了の設定を解除する。

1. 次のコマンドを実行して、スプールジョブ転送デーモンの自動起動の設定を解除する

```
systemctl disable bjex.service
```

2. 論理ホスト用のスプールジョブ転送デーモンの自動起動を設定している場合は、論理ホスト用スプールジョブ転送デーモンの Unit 名を指定して systemctl コマンドを実行する
3. /opt/hitachi/bjex/sample ディレクトリからコピーした Unit ファイル bjex.service をコピー先のディレクトリから削除する
4. 論理ホスト用のスプールジョブ転送デーモンの自動起動を設定している場合は、論理ホスト用スプールジョブ転送デーモンの Unit ファイルを削除する

自動起動・停止するためのスクリプトファイルのコピー先ディレクトリについては、「2.2.4(3) スプールジョブ転送デーモンの自動起動」を参照のこと。

2.2.3 バッチジョブ実行システムのバージョン情報の表示

バッチジョブ実行システムのバージョン情報の表示手順を説明する。

1. 次のコマンドを実行して、Hitachi PP Installer を起動する

```
/etc/hitachi_setup
```

2. 初期画面で「L」を入力する

インストール済みの日立製品の一覧が表示される。

2.2.4 インストール後の設定

インストール後の設定について説明する。なお、論理ホスト環境でバッチジョブ実行システムを運用する場合は、「4.9.2 論理ホストのバッチジョブ実行システムの環境設定」を参照のこと。

(1) 設定ファイルの確認

ジョブを実行する前に/opt/hitachi/bjex/conf/bjex.conf をシステムの環境に合わせて変更する。

bjex.conf に指定する各種ディレクトリをデフォルトから変更した場合、あらかじめディレクトリを作成しておくこと。

その後、次に示す手順で設定ファイルの確認を必ず実施すること。

1. bjex.conf のチェックを実施する

2. インストール

設定ファイル変更後、必ず `bjexpdctl` コマンドの `confstest` オプションを実行してエラーのないことを確認する。設定ファイルで指定する各種ディレクトリには、存在するディレクトリを指定すること。

2. スプールジョブ転送デーモンの起動確認を実施する

`bjexpdctl` コマンドの `start` オプションで起動する。

3. 必要に応じて `bjexec` コマンドによるバッチジョブの実行確認を実施する

設定ファイルは、スプールジョブ転送サービスと各バッチジョブで共用する。バッチジョブ実行時にスプールジョブに設定した属性を記録している。そのため設定ファイルは、すべてのスプールジョブを出力および削除したあとに変更する必要がある。また、新たにスプールジョブが生成されないように、すべてのプロセスを停止してから設定ファイルを変更すること。変更後は、上記の手順で設定ファイルの変更後の確認を必ず実施すること。

(2) JP1/IM の設定

バッチジョブ実行システムが通知する JP1 イベントの詳細情報を JP1/IM で表示させるためには、バッチジョブ実行システムが提供する拡張属性定義ファイルを JP1/IM のディレクトリにコピーし、設定ファイルに `JP1IM_MANAGER_HOST` パラメータを指定する必要がある。

(a) 拡張属性定義ファイルのコピー

拡張属性定義ファイルはバッチジョブ実行システムのインストール先ディレクトリの `system` ディレクトリに格納されている。

<拡張属性定義ファイルの名称>

- `hitachi_bjex_attr_ja.conf`

拡張属性定義ファイルのコピーは次に示す手順で実行する。

<手順>

1. 拡張属性定義ファイルを JP1/IM のディレクトリにコピーする※
2. JP1/IM - Manager を再起動する

注※ JP1/IM - Manager がインストールされたマシンの定義ファイル格納ディレクトリを次に示す。

- UNIX の場合

`/etc/opt/jplcons/conf/console/attribute/`

または

共有ディレクトリ `/jplcons/conf/console/attribute/`

- Windows の場合

Console パス `¥conf¥console¥attribute¥`

または

共有フォルダ `¥jplcons¥conf¥console¥attribute¥`

また、拡張属性定義ファイルはシフト JIS 形式であるため、OS の種類に応じて次の表に示すとおり拡張属性定義ファイルをコード変換してから、JP1/IM のディレクトリにコピーすること。

表 2-1 バッチジョブ実行システムとコピー先の OS 間で実施するコード変換

バッチジョブ実行システムの種類	コピー先の OS の種類	
	Linux	Windows
AIX 版	UTF-8	—
Linux 版	UTF-8	—

(凡例)

- ：コード変換不要。シフト JIS 形式のまま使用できる。
- UTF-8：シフト JIS 形式から UTF-8 形式にコード変換する必要がある。

(b) JP1IM_MANAGER_HOST パラメータの設定

JP1 イベントの送信先である JP1/IM-Manager が稼働しているホストを、設定ファイルの JP1IM_MANAGER_HOST パラメータで設定する必要がある。このパラメータを設定しない場合、バッチジョブ実行システムでは、バッチジョブ実行システムが稼働しているサーバ上で hostname コマンドを実行したときに表示されるホスト名が、JP1 イベントの送信先として仮定される。

JP1IM_MANAGER_HOST パラメータの詳細については、「8. 設定ファイル」の「JP1IM_MANAGER_HOST (JP1/IM - Manager が稼働するホスト名定義)」を参照のこと。

(3) スプールジョブ転送デーモンの自動起動

スプールジョブ転送デーモンを、システムの起動時および終了時に自動的に実行させる設定方法を次に示す。

(a) AIX の場合

システム起動時の自動起動機能

システム起動時の自動起動機能を設定するには、mkitab コマンドを使用して、自動起動の設定をする。設定後、次のシステム開始時から有効になる。

```
mkitab "bjexpd:2:wait:/opt/hitachi/bjex/sbin/bjexpdctl start"
```

スプールジョブ転送デーモンは、連携する JP1 シリーズのあとに起動するよう設定すること。JP1/Base, JP1/IM, JP1/AJS, JP1/FTP の順に自動起動を設定する場合、次に示す指定をしてコマンドを実行する。

```
mkitab -i hntr2mon "jp1base:2:wait:/etc/opt/jp1base/jbs_start"
mkitab -i jp1base "jp1cons:2:wait:/etc/opt/jp1cons/jco_start"
mkitab -i jp1cons "jp1ajs2:2:wait:/etc/opt/jp1ajs2/jajs_start"
mkitab -i jp1ajs2 "jp1_fts:2:wait:/etc/opt/jp1_fts/startup"
mkitab -i jp1_fts "bjexpd:2:wait:/opt/hitachi/bjex/sbin/bjexpdctl start"
```

設定後、lsitab コマンドを使用して、設定内容を確認すること。lsitab コマンドは、次に示す指定をして実行する。

```
lsitab -a
```

次に出力例を示す。

```
init:2:initdefault:
brc::sysinit:/sbin/rc.boot 3 >/dev/console 2>&1 # Phase 3 of system boot
:
hntr2mon:2:once:/opt/hitachi/HNTRLib2/etc/D002start
jp1base:2:wait:/etc/opt/jp1base/jbs_start
jp1cons:2:wait:/etc/opt/jp1cons/jco_start
jp1ajs2:2:wait:/etc/opt/jp1ajs2/jajs_start
jp1_fts:2:wait:/etc/opt/jp1_fts/startup
bjexpd:2:wait:/opt/hitachi/bjex/sbin/bjexpdctl start
```

システム終了時の自動終了機能

システム終了時の自動終了機能を設定するには、/etc/rc.shutdown を編集して、連携する JP1 シリーズより先に停止させるように次に示す記述を追加のこと。

```
test -x /opt/hitachi/bjex/sbin/bjexpdctl && /opt/hitachi/bjex/sbin/bjexpdctl stop
:
連携するJP1シリーズの停止処理
:
```

(b) Linux の場合

自動起動・停止するための Unit ファイル bjex.service を /opt/hitachi/bjex/sample ディレクトリに格納している。これを利用して次の手順で設定する。

/usr/lib/systemd/system ディレクトリへの追加

/opt/hitachi/bjex/sample ディレクトリに格納されている bjex.service を /usr/lib/systemd/system に追加する。bjex.service を追加するには次のように指定する。

```
cp /opt/hitachi/bjex/sample/bjex.service /usr/lib/systemd/system
chmod u=rw,go=r /usr/lib/systemd/system/bjex.service
chown root:root /usr/lib/systemd/system/bjex.service
```

bjex.service ファイルの編集

Unit ファイル bjex.service は、次に示す条件を満たす最低限必要な設定だけが記述されている。

- システムの起動時に、JP1/BASE の後にスプールジョブ転送デーモンが起動される。
 - システムの終了時に、JP1/BASE の前にスプールジョブ転送デーモンが停止される。
- 必要に応じて bjex.service ファイルの各項目を編集すること。

自動起動の有効化

bjex.service に対して自動起動を有効化する。自動起動を有効化するには次のように指定する。

```
systemctl --system enable bjex.service
```

2.2.5 旧バージョンから移行時の注意事項

旧バージョンからの移行時に次の項目について注意が必要である。

(1) スプールデータについて

現在使用しているバッチジョブ実行システムのバージョンによって、バージョンアップ後のスプールデータの扱いが異なる。スプールデータの扱いを次の表に示す。

表 2-2 バージョンアップ後のスプールデータの扱い

バッチジョブ実行システムのバージョン	バージョンアップ後のスプールデータの扱い
07-50 以前	<ul style="list-style-type: none">07-50 以前のスプールデータは、07-60 以降のバージョンに引き継ぐことはできない。07-50 以前のバージョンで作成したスプールデータは、バージョンアップ前にスプールディレクトリを削除または再作成する必要がある。
07-60 以降	07-60 以降のスプールデータは、07-61, 08-00 以降のバージョンに引き継いで使用できる。*

注※

- スプールディレクトリにあるデータだけ保証する。転送ディレクトリにあるデータは削除すること。
- ジョブ識別子そのまま引き継ぐことができる。
- 設定ファイルのホスト名、SysoutClass, および SPOOL_DIR を変更しないこと。また、バッチジョブ実行システムが稼働するホスト名を変更しないこと。変更した場合、動作を保証できない。
- スプールデータは下位バージョンのスプールデータと互換性がない。そのため、一度新バージョンに移行したあと、旧バージョンに戻してスプールデータをそのまま使用することはできない。

2.3 セットアップ

ここでは、バッチジョブ実行システムのセットアップ方法について説明する。

2.3.1 セットアップ前の作業

バッチジョブ実行システムのセットアップ前の作業について説明する。

(1) 環境変数の設定

バッチジョブ実行システムでは、言語種別およびローカルタイム情報を環境変数から参照して情報を入力するため、それらを環境変数で事前に設定しておく必要がある。

(a) 言語種別の設定

言語種別を環境変数 LANG で設定する。

- AIX の場合
 - 文字コード種別：シフト JIS
環境変数 LANG の値：Ja_JP
 - 文字コード種別：EUC コード
環境変数 LANG の値：ja_JP
- Linux の場合
 - 文字コード種別：UTF-8 コード
環境変数 LANG の値：ja_JP.UTF-8
 - 文字コード種別：ASCII コード
環境変数 LANG の値：C

(b) ローカルタイムの設定

バッチジョブ実行システムが提供するコマンドは、環境変数 TZ に従って情報を入力する。環境変数 TZ は、次に示すどれかの方法で指定すること。

- JP1/AJS のジョブ定義、環境変数の定義で指定
- システムプロファイル (/etc/profile) に指定
- ユーザプロファイル (\$HOME/.profile) に指定

(2) JP1 環境を確認する

JP1 の環境は、/etc/opt/jplbase/conf/jplbs_param.conf で決定する。

- AIX の場合
 - 文字コード種別：シフト JIS
エディタで"LANG=SJIS"の設定か確認すること。
 - 文字コード種別：EUC コード
エディタで"LANG=EUCJIS"の設定か確認すること。
- Linux の場合
 - 文字コード種別：UTF-8 コード
エディタで"LANG=UTF-8"の設定か確認すること。
 - 文字コード種別：ASCII コード
エディタで"LANG=C"の設定か確認すること。

(3) シェルの設定

- AIX の場合
JP1/AJS からジョブを起動する場合に使用するログインシェル，ジョブ定義 XML ファイルで指定するシェルは ksh だけ使用できる。/etc/passwd，ログインスクリプトファイルなど，ksh が使用できるように設定すること。
- Linux の場合
JP1/AJS からジョブを起動する場合に使用するログインシェル，ジョブ定義 XML ファイルで指定するシェルは bash だけ使用できる。/etc/passwd，ログインスクリプトファイルなど，bash が使用できるように設定すること。

2.3.2 セットアップ

バッチジョブ実行システムのインストールが完了後，バッチジョブ実行システムをセットアップする。

バッチジョブ実行システムの実行に必要なディレクトリを作成し，設定ファイルに指定する。設定ファイルに指定するディレクトリを次に示す。なお，バッチジョブ実行システムを実行するユーザは，これらのディレクトリに対して，書き込みおよび読み込み権限を付与すること。

- カタログドプロシジャ
定形のジョブ定義 XML ファイルを登録するディレクトリを指定する。
- 一時ファイル
ジョブ内だけで利用するファイルを一時的に作成するディレクトリを指定する。
- スプール
ジョブログや SYSOUT データを格納するディレクトリを指定する。
- 恒久ファイル

恒久ファイルとは、ジョブが終了後にジョブの結果として残すファイルのことである。恒久ファイルの指定で、相対パスを指定した場合に利用するカレントディレクトリを指定する。

- 実行ログ

システム管理者がジョブの実行を監視するために、ジョブの履歴を実行ログとして保存するディレクトリを指定する。

- トレース

システム障害時に、トラブルシュートとして状況を保存するためのディレクトリを指定する。

なお、ここではバッチジョブ実行システムを動作させるための基本となるセットアップ方法を説明する。業務に合わせて設定を変更したい場合は、必要に応じて次に示す設定ファイルのパラメータを参照すること。

表 2-3 設定が必要なファイルディレクトリ

ディレクトリ	設定ファイルのパラメータ名	デフォルト	デフォルトのパーミッション (8進数)
カタログドプロシジャ用	PROCEDURE_DIR	/var/opt/bjex/proc	AIX : 1777 Linux : 0777
一時ファイル用	TEMP_FILE_DIR	/tmp	なし※
スプール用	SPOOL_DIR	/var/opt/bjex/spool	AIX : 1777 Linux : 0777
恒久ファイル用	PERMANENT_FILE_DIR	/var/opt/bjex/perm	AIX : 1777 Linux : 0777
実行ログ用	LOG_DIR	/opt/hitachi/bjex/log	0777
トレース用	TRACE_DIR	/opt/hitachi/bjex/trace	AIX : 1777 Linux : 0777

注※ バッチジョブ実行システムのインストールでは設定しない。

バッチジョブを実行する OS ユーザに対して、各ディレクトリには次に示すファイルパーミッションが必要である。

表 2-4 ディレクトリのファイルパーミッション

ディレクトリ	読み込み許可 (r)	書き込み許可 (w)	実行許可 (x)	sticky ビット (t)
カタログドプロシジャ用	○	△	○	△
一時ファイル用	○	○	○	△
スプール用	○	○	○	AIX : △ Linux : ×
恒久ファイル用	○	○	○	△

ディレクトリ	読み込み許可 (r)	書き込み許可 (w)	実行許可 (x)	sticky ビット (t)
実行ログ用	○	○	○	×
トレース用	○	○	○	AIX : △ Linux : ×

(凡例)

- ：設定が必須である。
- △：システムの運用方針に従い，設定する。
- ×：設定しないこと。

スプール，恒久ファイルなどは，業務によって大容量となる場合があるため，専用のファイルシステムを作成して使用するのが望ましい。

2.3.3 bjexcollect コマンドの設定

bjexcollect コマンドは，バッチジョブ実行システムが異常終了および無応答となった場合などに，障害調査を実施するための資料を一括して収集するツールである。迅速な障害対応を行うため，bjexcollect の定義ファイル/opt/hitachi/bjex/maintenance/bjexcollect.conf にある次の表に示すパラメータの指定値を，「2.3.2 セットアップ」でセットアップした内容に従って書き換える。bjexcollect コマンドの詳細については，「付録 C bjexcollect コマンド (障害調査のための資料の一括収集)」を参照のこと。

表 2-5 bjexcollect の定義ファイルに設定する指定値

パラメータ	指定値の内容
SPOOL_DIR	スプールディレクトリを指定する。
LOG_DIR	ログディレクトリを指定する。
TRACE_DIR	トレースディレクトリを指定する。
LOGICAL_HOST	バッチジョブ実行システムが論理ホストで稼働している場合，論理ホスト名を指定する。

2.3.4 JP1/AJS 環境設定

JP1/AJS からバッチジョブ実行システムを実行するために，事前に実行環境を設定する必要がある。ここでは，JP1/AJS の環境設定の留意点について説明する。

(1) JP1/AJS のログ容量の見積もり

バッチジョブ実行システムを使用した場合，バッチジョブ実行システムを使用しない場合に比べて，次のログに出力するログ出力量が 1 ジョブ当たり約 350 バイト増加する。

ジョブ実行内部ログ

```
/var/opt/jp1ajs2/log/jpqagent/jpqagt_{00|01|02|03|04|05|06|07}.log
```

そのため、ログサイズを見積もる場合には、JP1 のマニュアルに記載されている計算式に上記の値を追加すること。

2.3.5 JP1/FTP 環境設定

バッチジョブ実行システムでは、スプール出力を Windows 上のプリントサーバへ転送して印刷できる。プリントサーバとの印刷ファイルの転送は、JP1/FTP を利用する。バッチジョブ実行システムから印刷する場合、事前に実行環境を設定する必要がある。ここでは、JP1/FTP を使用するための留意点について説明する。

(1) JP1/FTP ログインユーザの登録

ファイル伝送時に JP1/FTP にログインするログインユーザ[※]を、プリントサーバが動作する Windows 上の JP1/FTP に登録する。これによって、登録したユーザ以外から着信の要求が来た場合、JP1/FTP は着信を拒否できる。

注※ ここで登録するログインユーザは、Windows のログインユーザとは関係ない。

(2) 転送先ディレクトリの作成

出力クラスごとに、Windows 上のプリントサーバへ、スプールジョブ管理ファイルと SYSOUT データファイルの転送先ディレクトリをそれぞれ作成する。

出力クラスについては、「3.11.1 出力クラス」、および「4.3.3 出力クラスの設定」を参照のこと。転送先ディレクトリについては、「5.3.2 JP1/FTP を使った印刷データ転送」を参照のこと。

2.3.6 メッセージ制御機能を使用する場合の環境設定

メッセージ制御機能を使用して JP1/IM - View から応答を入力するためには、事前に実行環境を設定する必要がある。

メッセージ制御機能を使用する場合は、「4.3.7 メッセージ制御機能の設定」に示す設定を行ってから、次の作業を実施すること。

(1) bjex_msg_setup コマンドの実行

bjex_msg_setup コマンドを実行して、JP1/IM - Manager との連携を有効にする。bjex_msg_setup コマンドの詳細については、「7. コマンド」の「bjex_msg_setup (メッセージ制御機能の初期設定 (バッチジョブ実行システムの設定用))」を参照のこと。

bjex_msg_setup コマンドは次に示す手順で実行する。

1. JP1/Base がインストール済みであることを確認する。
2. root 権限で bjex_msg_setup コマンドを実行する。
3. KAKC5100-I メッセージが出力されることを確認する。

bjex_msg_setup コマンド実行後に JP1/Base を再インストールした場合は、bjex_msg_setup コマンドを再実行すること。

(2) JP1/IM - Manager の設定方法

JP1/IM - Manager および JP1/IM - View で、応答待ちイベントに関する機能を有効にする。有効にしていない場合、JP1/IM - View による応答の入力ができない。

JP1/IM - Manager および JP1/IM - View の設定方法ならびにバッチジョブ実行システムと JP1/IM - Manager との通信の設定については、JP1 のマニュアルを参照のこと。

(3) メッセージ出力用カタログドプロシジャを使用するための設定

メッセージ出力用カタログドプロシジャは uCosminexus Batch Job Execution Server のインストールディレクトリの sample ディレクトリにある。これらのカタログドプロシジャを、設定ファイルで定義したプロシジャライブラリのディレクトリに複製して使用する。メッセージ出力用カタログドプロシジャを次の表に示す。

表 2-6 メッセージ出力用カタログドプロシジャ

ファイル名	説明
bjex_PAUSE.xml	ジョブの実行を応答待ちにすることで、一時停止状態にする。
bjex_WRITE.xml	メッセージを出力する。
bjex_WRITER.xml	応答要求メッセージを出力する。応答されたメッセージが未定義の場合、エラーのリターンコードを設定して終了する。
bjex_WRITERX.xml	応答要求メッセージを出力する。応答されたメッセージが未定義の場合、再度応答入力メッセージを出力する。

2.3.7 排他/共用制御機能および実行中のジョブ情報表示機能を使用する場合

排他/共用制御機能および実行中のジョブ情報表示機能を使用する場合は、「4.3.6 排他/共用制御機能の設定」の設定をすること。

排他/共用制御機能および実行中のジョブ情報表示機能では、共用メモリおよびセマフォを使用する。OS によっては使用できる共用メモリおよびセマフォ値が低く設定されている場合がある。このため、排他/共用制御機能および実行中のジョブ情報表示機能を使用する場合は、カーネルパラメータの見直しをすること。

2.4 ストレージ設計

バッチジョブ実行システムのストレージ設計については、リリースノートを参照のこと。

3

機能

この章では、バッチジョブ実行システムの機能について説明する。

3.1 バッチジョブ実行システムの機能

3.1.1 バッチジョブ実行システムの機能の特長と処理手順

(1) 機能の特長

バッチジョブ実行システムの機能は、メインフレームと類似のジョブ管理システムを提供することを目的とし、システム資源を効果的に活用するために、次に示す特長を備えている。

- バッチジョブは、システム資源の定義をジョブ定義 XML ファイルで行う。
- JPI/AJS とバッチジョブ実行システムが連携してバッチジョブを一括管理し、システムへのジョブの入力および結果の出力を効率良く処理する。
- バッチジョブの実行結果はスプールに保存され、あとから印刷ができる。
- システム動作情報のジョブログ機能および実行ログ機能を提供する。

(2) ジョブ実行の処理手順

ジョブ実行の処理手順を次に示す。

1. ジョブ定義 XML ファイルの解析

ジョブ定義 XML ファイルを読み込み後、ジョブ実行前にシンタクスチェックをする。シンタクス上のエラーがある場合は、ジョブを実行することなく終了する。

2. システム資源の確保

ジョブステップ実行に必要なシステム資源を、それぞれジョブコントローラが確保する。このとき、ジョブステップ実行に必要なシステム資源が使用できない場合、異常と判断してジョブを終了する。

3. ジョブステップの実行

ジョブステップはジョブステッププロセスとして実行される。ジョブコントローラはシステム資源が割り当てられると、ジョブステッププロセスの生成を行い、処理プログラムの実行を開始する。

4. ジョブステップの終了

処理プログラムが終了すると、ジョブステッププロセスは消滅し、ジョブコントローラで不要となったファイルを削除する。このとき、ファイルの削除に失敗しても次のステップを実行する。

3.1.2 ジョブ定義 XML ファイル

バッチジョブの制御を行うために、ジョブ定義 XML ファイルに XML 言語でその要求を記述する。

ジョブ定義 XML ファイルの主な制御要素を次に示す。

- **JOB 要素**
ジョブの始まりを示し、ジョブの名称とそのジョブの属性などを定義する。
- **STEP 要素**
ジョブステップの始まりを示し、ジョブステップ名称を定義する。
- **EXEC 要素**
実行するプログラム名を定義する。
- **DD 要素**
ジョブまたはジョブステップで利用するファイルを定義する。
- **CALL 要素**
実行するプロシジャ名を定義する。
- **PROC 要素**
CALL 要素で呼び出すプロシジャを定義する。

(1) ジョブの構成

ジョブは、その処理内容に応じて（処理を行うプログラム対応に）幾つかの段階に分かれるのが一般的である。この各段階を"ジョブステップ"と呼ぶ。ジョブステップは、処理を行うプログラムの定義と、そのプログラムが使用するファイルなどの資源の定義から成る。

ジョブ内での複数個のジョブステップの間には前後関係が存在していて、並べられた順序に従って処理していくことを原則とするが、先行するジョブステップの終了状態によって後続するジョブステップの実行をやめることもできる。

(a) ジョブを構成するためのジョブ定義 XML ファイル

ジョブの実行をバッチジョブ実行システムに依頼するためには、ジョブ定義 XML ファイルを準備する必要がある。

ジョブを構成するためのジョブ定義 XML ファイル例を次の図に示す。これはジョブ制御、プログラム検索パスの定義などを使用する場合を含めてジョブ定義 XML ファイルの規則を示したものである。

図 3-1 ジョブを構成するためのジョブ定義 XML ファイル例

<HitachiBatchJobExec>	: ジョブ定義XMLである宣言
<JOB>	: ジョブの宣言
<DD NAME="JOBLIB" />	: ジョブ全体のプログラム検索パスの宣言
<PROC>	: 入力ストリーム中のプロシジャ宣言
<STEP>	: ジョブステップの宣言
<EXEC />	: ジョブステップで必要とするDD, EXEC
<DD NAME="STEPLIB" />	: ジョブステップ中のプログラム検索パスの宣言
<DD NAME="SYSIN">	: SYSINデータのDD宣言
:	: SYSINデータ本体
</DD>	: SYSINデータの終了宣言
</STEP>	: ジョブステップ終了宣言
</PROC>	: 入力ストリーム中のプロシジャ終了宣言
<STEP>	: 次のジョブステップの宣言
<CALL />	: プロシジャを呼び出す場合の宣言
</STEP>	: ジョブステップ終了宣言
<STEP>	: 次のジョブステップの宣言
:	:
</STEP>	: ジョブステップ終了宣言
</JOB>	: ジョブの終了宣言
</HitachiBatchJobExec>	: ジョブ定義XMLの終了宣言

ジョブ実行の手順はジョブ定義 XML ファイルによって記述され、定義に従いファイルの準備と環境変数を設定し、EXEC 要素で指定されたプログラムのプロセスを作成することによって実行される。

3.1.3 ジョブコントローラ

バッチジョブ実行システムでのジョブの実行を制御する。

ジョブの入力は JP1/AJS で行う。JP1/AJS から起動されたジョブコントローラは、ジョブに対して入出力装置や各種システム資源の割り当ておよび解放処理、すなわちジョブの実行準備および終了処理をする。また、ジョブの実行結果を管理し、ジョブ実行に付随するジョブログを出力する。

機能の特長を次に示す。

- ジョブの実行準備および終了処理、ファイルの割り当ておよび解放処理をする。
- ジョブを実行する前にジョブのシンタクスチェックができる。
- 処理プログラムの実行結果をスプールで管理できる。
- ジョブログは JP1/AJS - View から参照できる。JP1/AJS - View から参照できるデータ量については、注意事項がある。注意事項については、「3.1.3(2) ジョブログ出力機能」を参照のこと。

(1) ジョブのシンタクスチェック機能

シンタクスチェック機能は、bjexec コマンドで -c オプションを指定することによって実行できる。この機能は、ジョブ定義 XML ファイルのシンタクスチェックだけを行い、ジョブの実行はしない。

シンタクスチェック機能の特長を次に示す。

- ジョブを実行することなく、ジョブ定義 XML ファイルのエラーを検出できる。

- ジョブ定義 XML ファイルが正しい場合もジョブは実行されないため、ジョブの実行に必要なファイルなどの準備をする必要がない。

(2) ジョブログ出力機能

ジョブコントローラは、ジョブ実行の出力情報として、処理プログラムの実行結果のほかに、ジョブ実行に付随したログ情報を出力する。このログ情報をジョブログという。

(a) ジョブログに出力する内容

ジョブログには、JOBLOG、XML および SYSMMSG を出力する。ジョブログの内容は、ジョブ終了時にジョブコントローラの標準エラー出力に出力するので、JP1/AJS - View から参照できる。

注意事項

JP1/AJS ではジョブネットの UNIX ジョブとしてジョブを実行した場合、デフォルト設定ではジョブの標準エラー出力データを取得する。

このため、ジョブログは JP1/AJS - View から参照できる。

ただし、ジョブが出力する標準エラー出力データ量が大きい場合、次に示す事象が発生する要因となることがある。

- JP1/AJS - View での参照時のメモリ不足の発生
- JP1/AJS システムの負荷の増加
- 高負荷によるジョブの実行遅延
- ジョブ異常終了

このため、ジョブログの出力データは、多くても数 MB 以内になるように設計する必要がある。

また、エラーケースなど意図しないでジョブログ、XML、および SYSMMSG の出力データが多くなる場合も考えられるため、JP1/AJS の UNIX ジョブとして定義する前に、ジョブコントローラ単体で十分テストを実施する必要がある。

並列実行するジョブステップの場合、ジョブログと SYSMMSG は時系列に出力されるため、並列実行するジョブステップの情報はジョブ定義 XML ファイルに指定した順とはならない。

(b) JOBLOG の出力形式

JOBLOG には主に次の表に示す内容をメッセージ出力する。

表 3-1 ジョブログに出力する内容

出力契機	出力内容
ジョブログヘッダ	<ul style="list-style-type: none"> • ジョブ識別子 • スプールジョブディレクトリ • 実行ホスト名 • JP1 情報
ジョブ開始時	<ul style="list-style-type: none"> • ジョブの開始時刻

出力契機	出力内容
ジョブ開始時	<ul style="list-style-type: none"> • ジョブ識別子 • ジョブ名
ジョブステップ開始時	<ul style="list-style-type: none"> • ジョブステップの開始時刻 • ジョブ識別子 • ジョブ名 • ジョブステップ名
ジョブステップ終了時	<ul style="list-style-type: none"> • ジョブステップの終了時刻 • ジョブ識別子 • ジョブ名 • ジョブステップ名 • ジョブステッププロセスの終了コード • C-TIME • E-TIME
ジョブ終了時	<ul style="list-style-type: none"> • ジョブの終了時刻 • ジョブ識別子 • ジョブ名 • ジョブの終了コード • ジョブステップの C-TIME の合計 • ジョブステップの E-TIME の合計 • ジョブが利用した最大実メモリサイズ

ジョブログヘッダはJOBLOG ファイルの先頭に出力する。

JOBLOG の出力形式を次に示す（項番は項目の説明の番号と対応している）。

```
-----
uCosminexus Batch Job Execution Server 09-10

[Information]                                     ... (1)
Jobid       : 001338                             ... (2)
Spool Directory : /var/opt/bjex/spool/001338/     ... (3)
Date        : 2023/10/01                         ... (4)
Configure File : /opt/hitachi/bjex/conf/bjex.conf ... (5)
HostName    : nvm014442                          ... (6)
[JPI Parameter]                                 ... (7)
JPIJobName  : bjexec                             ... (8)
JPIJobID    : 1506                              ... (9)
JPI_USERNAME : bjex                             ... (10)
JPIUNCName  : nvm014442                          ... (11)
JPIINBQQueueName: ¥¥nvm014442¥¥SYSTEM          ... (12)
JPIPriority  : 3                                 ... (13)
AJSEXCID    : @A200                             ... (14)
AJSJOBNAME  : /BJEX-TEST/TEST-NET/JOB0001      ... (15)
-----

***** BJEX MESSAGE *****
14:30:21 001338 KAKC0091-I SAMPLE JCL Job started.
14:30:21 001338 KAKC0092-I SAMPLE JCL STEP1 Step started.
14:30:21 001338 KAKC0097-I SAMPLE JCL STEP1 Step ended. rc=0 E-Time=0.008s C-Time=0.010s
14:30:21 001338 KAKC0092-I SAMPLE JCL STEP2 Step started.
14:30:21 001338 KAKC0092-I SAMPLE JCL STEP3 Step started.
14:30:21 001338 KAKC0097-I SAMPLE JCL STEP3 Step ended. rc=0 E-Time=0.012s C-Time=0.010s
14:30:21 001338 KAKC0097-I SAMPLE JCL STEP2 Step ended. rc=0 E-Time=0.015s C-Time=0.000s
14:30:21 001338 KAKC0092-I SAMPLE JCL STEP4 Step started.
14:30:21 001338 KAKC0097-I SAMPLE JCL STEP4 Step ended. rc=0 E-Time=0.007s C-Time=0.010s
14:30:21 001338 KAKC0092-I SAMPLE JCL STEP5 SAMPLE PROC STEP1 Step started.
14:30:21 001338 KAKC0097-I SAMPLE JCL STEP5 SAMPLE PROC STEP1 Step ended. rc=0 E-Time=0.010s C-Time=0.000s
14:30:21 001338 KAKC0098-I SAMPLE JCL Job ended. rc=0 E-Time=0.040s C-Time=0.030s
```

(1) バッチジョブ実行システムに関する情報

(2) ジョブ識別子

バッチジョブ実行システムが割り当てたジョブ通し番号。

(3) スプールジョブディレクトリ

ジョブの出力情報が格納されるスプールジョブディレクトリ名。

(4) 日付

ヘッダ出力時の日付。yyyy/mm/dd 形式で出力する。

(5) 設定ファイルのパス

使用したバッチジョブ実行システムの設定ファイルのパス。

(6) ホスト名

設定ファイルの JPI_HOSTNAME パラメータで指定したホスト名。

ただし、bjexec コマンドで -h オプションを指定した場合、設定ファイルの JPI_HOSTNAME パラメータは無視され、論理ホスト名が有効になる。詳細については、「7. コマンド」を参照すること。

(7) JPI の環境変数の始まり

(8) JPI ジョブ名

JPI で定義した実行ファイル名、またはジョブ名。

(9) JPI ジョブ番号

JPI のジョブ番号。

(10)JP1 ユーザ名

JP1 のユーザ名。

(11)JP1 エージェントホスト名

JP1 の実行ホスト名。

(12)マネージャーホスト名とキュー名

ジョブの登録先である JP1/AJS のマネージャーホスト名 (255 バイト以内の文字列) とキュー名 (63 バイト以内の文字列) が「**¥¥**マネージャーホスト名**¥¥**キュー名」の形式で設定される。

(13)JP1 実行優先順位

JP1 のジョブ実行優先順位。

(14)JP1 実行 ID

JP1 のジョブ実行 ID。

(15)AJSJOBNAME

JP1 のジョブ名。

(c) XML の出力形式

ジョブ定義 XML ファイルを解析した XML のイメージを出力する。

- >>>>> ファイル名

カタログドプロシジャとしてファイル名で示すファイル解析の開始を示す。

- <<<<<< ファイル名

カタログドプロシジャとしてファイル名で示すファイル解析の終了を示す。

- 要素番号 記号 <要素 ~>

ファイル内に含む要素を表示する。

- 要素番号 ((000001~999999))

ファイル内の要素を 6 けたの 10 進数で表示する。

999999 を超える要素数の XML を入力した場合、7 けた以上の要素番号表示となる。

- 記号

++ : 入力ストリームプロシジャ

EXEC 要素, DD 要素, SETENV 要素, IF 要素, ELSE 要素, ENDIF 要素および SETENVFILE 要素を表示する。

+/: 修正された可能性のある入力ストリームプロシジャ

REPLACE 要素または REPLACESTEP 要素で修正された可能性のある要素を表示する。

XX : カタログドプロシジャ

EXEC 要素, DD 要素, SETENV 要素, IF 要素, ELSE 要素, ENDIF 要素および SETENVFILE 要素を表示する。

X/: 修正された可能性のあるカタログドプロシジャ

REPLACE 要素または REPLACESTEP 要素で修正された可能性のある要素を表示する。

**：読み込んだカタログドプロシジャ

空白：入力したジョブ定義 XML ファイルの内容

- **要素**

解析後の要素を示す。内容を含む場合、改行後の要素番号がない状態で表示する。

XML の表示例を次に示す。

```

***** XML IMAGE *****
000001 <?xml version="1.0" encoding="Shift-JIS" ?>
<!-- Sample job for uCosminexus Batch Job Execution Server -->

000002 <HitachiBatchJobExec version="1.3" os="unix" >
000003 <JOB NAME="SAMPLE_JCL">
000004 <STEP NAME="STEP1" >
000005 <EXEC PGM="UAP1" PARM="%%SAMPLE_JCL_STEP1%%" />
000006 <DD NAME="DD1" TYPE="SYSOUT" SYSOUT="*" HOLD="YES" OVERLAY="FOPA" OVERC="1" />
000007 </STEP>
000008 <STEP NAME="STEP2" >
000009 <EXEC PGM="UAP2" PARM="%%SAMPLE_JCL_STEP2%%" LANG="COBOL" JSPEC="YES" />
000010 <DD NAME="DD2_WRITE" TYPE="TEMPPREST" DSN="BJEXFIFO" />
000011 </STEP>
000012 <STEP NAME="STEP3" >
000013 <EXEC PGM="UAP3" PARM="%%SAMPLE_JCL_STEP3%%" LANG="COBOL" />
000014 <DD NAME="DD2_READ" TYPE="TEMPPREST" DSN="BJEXFIFO" />
000015 </STEP>
000016 <STEP NAME="STEP4" >
000017 <EXEC PGM="UAP4" PARM="%%SAMPLE_JCL_STEP4%%" LANG="COBOL" />
000018 <DD NAME="DD4" TYPE="DATA" SAMF="YES" >
<![CDATA[
!
INPUT DATA 1
INPUT DATA 2
!
]]>

000019 </DD>
000020 </STEP>
000021 <STEP NAME="STEP5" >
000022 <CALL NAME="SAMPLE_PROC" >

>>>>> /var/opt/bjex/proc/SAMPLE_PROC.xml
000001 ** <?xml version="1.0" encoding="Shift-JIS" ?>
<!-- Sample cataloged procedure for uCosminexus Batch Job Execution Server -->

000002 ** <HitachiBatchJobExec version="1.3" os="unix" >
000003 ** <PROC NAME="SAMPLE_PROC2">
000004 ** <STEP NAME="SAMPLE_PROC_STEP1" >
000005 ** <EXEC PGM="UAP5" PARM="%%SAMPLE_PROC_STEP1%%" />
000006 ** <DD NAME="PROC_DD1" TYPE="DATA">
<![CDATA[
!
PROC INPUT DATA 1
PROC INPUT DATA 2
!
]]>

000007 ** </DD>
000008 ** </STEP>
000009 ** </PROC>
000010 ** </HitachiBatchJobExec>
<<<<< /var/opt/bjex/proc/SAMPLE_PROC.xml

000023 <REPLACE>
<!-- Replace PARM attribute of procedure step -->

000024 <REPLACESTEP NAME="SAMPLE_PROC_STEP1" >
000025 <EXEC PARM="%%replace_SAMPLE_PROC_STEP3%%" />
000026 </REPLACESTEP>
000027 </REPLACE>
000028 </CALL>
000029 XX <STEP NAME="SAMPLE_PROC_STEP1" >
000030 X/ <EXEC PGM="UAP5" PARM="%%replace_Sample_PROC_STEP3%%" />
000031 XX <DD NAME="PROC_DD1" TYPE="DATA" >
PROC INPUT DATA 1
PROC INPUT DATA 2

000032 XX </DD>
000033 XX </STEP>
000034 </STEP>
000035 </JOB>
000036 </HitachiBatchJobExec>

```

(d) SYSMMSG の出力形式

ファイルの前処理、後処理の結果を出力する。

SYSMMSG に出力する内容を次の表に示す。

表 3-2 SYSMSG に出力する内容

出力契機	出力内容
ジョブ定義 XML ファイル変換時	<ul style="list-style-type: none"> ジョブ定義 XML ファイルの誤り ジョブのメッセージクラス
ファイル割り当て時	<ul style="list-style-type: none"> 割り当てたファイルの DD 名 パス名
ファイル割り当て解除時	<ul style="list-style-type: none"> 割り当て解除したファイルの DD 名 パス名

SYSMSG の表示例を次に示す。

```

***** SYSTEM MESSAGE *****
14:30:21 001338 KAKC0900-I Parsing of xml file "/home/bjex/jcl/SAMPLE_JCL.xml" started.
14:30:21 001338 KAKC1100-I Parsing of proc file "/var/opt/bjex/proc/SAMPLE_PROC.xml" started.
14:30:21 001338 KAKC1109-I Parsing of proc file "/var/opt/bjex/proc/SAMPLE_PROC.xml" ended. code=0
14:30:21 001338 KAKC0909-I Parsing of xml file "/home/bjex/jcl/SAMPLE_JCL.xml" ended. code=0
14:30:21 001338 KAKC0707-I MSGCLASS is "A".
14:30:21 001338 KAKC1600-I SAMPLE_JCL Job allocated.
14:30:21 001338 KAKC1601-I SAMPLE_JCL.STEP1 Step allocated.
14:30:21 001338 KAKC1602-I DD1, . /var/opt/bjex/spool/001338/001_STEP1_001_DD1
14:30:21 001338 KAKC1603-I DD1, . /var/opt/bjex/spool/001338/001_STEP1_001_DD1, size=621
14:30:21 001338 KAKC1601-I SAMPLE_JCL.STEP2 Step allocated.
14:30:21 001338 KAKC1602-I DD2_WRITE, . /tmp/TEMPPREST_001338.SAMPLE_JCL_BJEXFIFO
14:30:21 001338 KAKC1601-I SAMPLE_JCL.STEP3 Step allocated.
14:30:21 001338 KAKC1602-I DD2_READ, . /tmp/TEMPPREST_001338.SAMPLE_JCL_BJEXFIFO
14:30:21 001338 KAKC1603-I DD2_READ, . /tmp/TEMPPREST_001338.SAMPLE_JCL_BJEXFIFO
14:30:21 001338 KAKC1603-I DD2_WRITE, . /tmp/TEMPPREST_001338.SAMPLE_JCL_BJEXFIFO
14:30:21 001338 KAKC1601-I SAMPLE_JCL.STEP4 Step allocated.
14:30:21 001338 KAKC1602-I DD4, . /tmp/SYSIN_001338.SAMPLE_JCL_b3svLP
14:30:21 001338 KAKC1603-I DD4, . /tmp/SYSIN_001338.SAMPLE_JCL_b3svLP
14:30:21 001338 KAKC1604-I /tmp/SYSIN_001338.SAMPLE_JCL_b3svLP is deleted.
14:30:21 001338 KAKC1601-I SAMPLE_JCL.STEP5, SAMPLE_PROC.STEP1 Step allocated.
14:30:21 001338 KAKC1602-I PROC DD1, . /tmp/SYSIN_001338.SAMPLE_JCL_xnB3Rt
14:30:21 001338 KAKC1603-I PROC DD1, . /tmp/SYSIN_001338.SAMPLE_JCL_xnB3Rt
14:30:21 001338 KAKC1604-I /tmp/SYSIN_001338.SAMPLE_JCL_xnB3Rt is deleted.

```

(e) ジョブステップの標準エラー出力

各ジョブステップが出力した標準エラー出力を KAKC0719-I メッセージの下部に表示する。出力例を次に示す。

```

***** JOBSTEP OUTPUT *****
KAKC0719-I STEP 001, STEP1, STDERR
%%SAMPLE_JCL_STEP1%%
%%SAMPLE_JCL_STEP1%%

KAKC0719-I STEP 002, STEP2, STDERR
%%SAMPLE_JCL_STEP2%%
%%SAMPLE_JCL_STEP2%%

KAKC0719-I STEP 003, STEP3, STDERR
%%SAMPLE_JCL_STEP3%%
%%SAMPLE_JCL_STEP3%%

KAKC0719-I STEP 004, STEP4, STDERR
%%SAMPLE_JCL_STEP4%%
%%SAMPLE_JCL_STEP4%%

KAKC0719-I STEP 005, STEP5, SAMPLE_PROC_STEP1, STDERR
%%replace SAMPLE_PROC_STEP3%%
%%replace SAMPLE_PROC_STEP3%%

```

(f) ジョブログの出力例

JP1/AJS - View から参照した場合のジョブログの出力例を次の図に示す。

図 3-2 ジョブログの出力例

```
KAKC0090-I HitachiBatchJobExec started.
KAKC0400-I Parsing of config file "/opt/hitachi/bjex/conf/bjex.conf" started.
KAKC0409-I Parsing of config file "/opt/hitachi/bjex/conf/bjex.conf" ended. code=0
KAKC0724-I Jobid was assigned. Jobid=001338
KAKC0097-I SAMPLE_JCL.STEP1 Step ended. rc=0 E-Time=0.008s C-Time=0.010s
KAKC0097-I SAMPLE_JCL.STEP3 Step ended. rc=0 E-Time=0.012s C-Time=0.010s
KAKC0097-I SAMPLE_JCL.STEP2 Step ended. rc=0 E-Time=0.015s C-Time=0.000s
KAKC0097-I SAMPLE_JCL.STEP4 Step ended. rc=0 E-Time=0.007s C-Time=0.010s
KAKC0097-I SAMPLE_JCL.STEP5.SAMPLE_PROC.STEP1 Step ended. rc=0 E-Time=0.010s C-Time=0.000s
KAKC0098-I SAMPLE_JCL Job ended. rc=0 E-Time=0.040s C-Time=0.030s
-----
uCosminexus Batch Job Execution Server 09-10

[Information]
Jobid          : 001338
Spool Directory : /var/opt/bjex/spool/001338/
Date           : 2023/10/01
Configure File  : /opt/hitachi/bjex/conf/bjex.conf
HostName       : nvm014442
[JP1 Parameter]
JP1JobName     : bjexec
JP1JobID      : 1506
JP1_USERNAME   : bjex
JP1UNCGName    : nvm014442
JP1NBQSQueueName: ¥¥nvm014442¥¥SYSTEM
JP1Priority    : 3
AJSEXCID      : @A200
AJSJOBNAME    : /BJEX-TEST/TEST-NET/JOB0001
-----
***** BJEX MESSAGE *****
14:30:21 001338 KAKC0091-I SAMPLE_JCL Job started.
14:30:21 001338 KAKC0092-I SAMPLE_JCL.STEP2 Step started.
14:30:21 001338 KAKC0097-I SAMPLE_JCL.STEP1 Step ended. rc=0 E-Time=0.008s C-Time=0.010s
14:30:21 001338 KAKC0092-I SAMPLE_JCL.STEP2 Step started.
14:30:21 001338 KAKC0092-I SAMPLE_JCL.STEP3 Step started.
14:30:21 001338 KAKC0097-I SAMPLE_JCL.STEP3 Step ended. rc=0 E-Time=0.012s C-Time=0.010s
14:30:21 001338 KAKC0097-I SAMPLE_JCL.STEP2 Step ended. rc=0 E-Time=0.015s C-Time=0.000s
14:30:21 001338 KAKC0092-I SAMPLE_JCL.STEP4 Step started.
14:30:21 001338 KAKC0097-I SAMPLE_JCL.STEP4 Step ended. rc=0 E-Time=0.007s C-Time=0.010s
14:30:21 001338 KAKC0092-I SAMPLE_JCL.STEP5.SAMPLE_PROC.STEP1 Step started.
14:30:21 001338 KAKC0097-I SAMPLE_JCL.STEP5.SAMPLE_PROC.STEP1 Step ended. rc=0 E-Time=0.010s C-Time=0.000s
14:30:21 001338 KAKC0098-I SAMPLE_JCL Job ended. rc=0 E-Time=0.040s C-Time=0.030s
```

```

***** XML IMAGE *****
000001 <?xml version="1.0" encoding="Shift-JIS" ?>
<!-- Sample job for uCosminexus Batch Job Execution Server -->

000002 <HitachiBatchJobExec version="1.3" os="unix" >
000003 <JOB NAME="SAMPLE_JCL">
000004 <STEP NAME="STEP1" >
000005 <EXEC PGM="UAP1" PARM="%%SAMPLE_JCL_STEP1%%" />
000006 <DD NAME="DD1" TYPE="SYSOUT" SYSOUT="*" HOLD="YES" OVERLAY="FOPA" OVERC="1" />
000007 </STEP>
000008 <STEP NAME="STEP2" >
000009 <EXEC PGM="UAP2" PARM="%%SAMPLE_JCL_STEP2%%" LANG="COBOL" JSPEC="YES" />
000010 <DD NAME="DD2_WRITE" TYPE="TEMPPREST" DSN="BJEXFIFO" />
000011 </STEP>
000012 <STEP NAME="STEP3" >
000013 <EXEC PGM="UAP3" PARM="%%SAMPLE_JCL_STEP3%%" LANG="COBOL" />
000014 <DD NAME="DD2_READ" TYPE="TEMPPREST" DSN="BJEXFIFO" />
000015 </STEP>
000016 <STEP NAME="STEP4" >
000017 <EXEC PGM="UAP4" PARM="%%SAMPLE_JCL_STEP4%%" LANG="COBOL" />
000018 <DD NAME="DD4" TYPE="DATA" SAMF="YES" >
<![CDATA[
!
INPUT DATA 1
INPUT DATA 2
!
]]>

000019 </DD>
000020 </STEP>
000021 <STEP NAME="STEP5" >
000022 <CALL NAME="SAMPLE_PROC" >

>>>>> /var/opt/bjex/proc/SAMPLE_PROC.xml
000001 ** <?xml version="1.0" encoding="Shift-JIS" ?>
<!-- Sample cataloged procedure for uCosminexus Batch Job Execution Server -->

000002 ** <HitachiBatchJobExec version="1.3" os="unix" >
000003 ** <PROC NAME="SAMPLE_PROC2">
000004 ** <STEP NAME="SAMPLE_PROC_STEP1" >
000005 ** <EXEC PGM="UAP5" PARM="%%SAMPLE_PROC_STEP1%%" />
000006 ** <DD NAME="PROC_DD1" TYPE="DATA">
<![CDATA[
!
PROC INPUT DATA 1
PROC INPUT DATA 2
!
]]>

000007 ** </DD>
000008 ** </STEP>
000009 ** </PROC>
000010 ** </HitachiBatchJobExec>
<<<<< /var/opt/bjex/proc/SAMPLE_PROC.xml

000023 <REPLACE>
<!-- Replace PARM attribute of procedure step -->

000024 <REPLACESTEP NAME="SAMPLE_PROC_STEP1" >
000025 <EXEC PARM="%%replace_SAMPLE_PROC_STEP3%%" />
000026 </REPLACESTEP>
000027 </REPLACE>
000028 </CALL>
000029 XX <STEP NAME="SAMPLE_PROC_STEP1" >
000030 X/ <EXEC PGM="UAP5" PARM="%%replace_Sample_PROC_STEP3%%" />
000031 XX <DD NAME="PROC_DD1" TYPE="DATA" >
PROC INPUT DATA 1
PROC INPUT DATA 2

000032 XX </DD>
000033 XX </STEP>
000034 </STEP>
000035 </JOB>
000036 </HitachiBatchJobExec>

```

```

***** SYSTEM MESSAGE *****
14:30:21 001338 KAKC0900-I Parsing of xml file "/home/bjex/jcl/SAMPLE_JCL.xml" started.
14:30:21 001338 KAKC1100-I Parsing of proc file "/var/opt/bjex/proc/SAMPLE_PROC.xml" started.
14:30:21 001338 KAKC1109-I Parsing of proc file "/var/opt/bjex/proc/SAMPLE_PROC.xml" ended. code=0
14:30:21 001338 KAKC0909-I Parsing of xml file "/home/bjex/jcl/SAMPLE_JCL.xml" ended. code=0
14:30:21 001338 KAKC0707-I MSGCLASS is "A".
14:30:21 001338 KAKC1600-I SAMPLE_JCL Job allocated.
14:30:21 001338 KAKC1601-I SAMPLE_JCL.STEP1 Step allocated.
14:30:21 001338 KAKC1602-I DD1, . /var/opt/bjex/spool/001338/001_STEP1_001_DD1
14:30:21 001338 KAKC1603-I DD1, . /var/opt/bjex/spool/001338/001_STEP1_001_DD1, size=621
14:30:21 001338 KAKC1601-I SAMPLE_JCL.STEP2 Step allocated.
14:30:21 001338 KAKC1602-I DD2_WRITE, . /tmp/TEMPPREST_001338_SAMPLE_JCL_BJEXFIFO
14:30:21 001338 KAKC1601-I SAMPLE_JCL.STEP3 Step allocated.
14:30:21 001338 KAKC1602-I DD2_READ, . /tmp/TEMPPREST_001338_SAMPLE_JCL_BJEXFIFO
14:30:21 001338 KAKC1603-I DD2_READ, . /tmp/TEMPPREST_001338_SAMPLE_JCL_BJEXFIFO
14:30:21 001338 KAKC1603-I DD2_WRITE, . /tmp/TEMPPREST_001338_SAMPLE_JCL_BJEXFIFO
14:30:21 001338 KAKC1601-I SAMPLE_JCL.STEP4 Step allocated.
14:30:21 001338 KAKC1602-I DD4, . /tmp/SYSIN_001338_SAMPLE_JCL_b3svLP
14:30:21 001338 KAKC1603-I DD4, . /tmp/SYSIN_001338_SAMPLE_JCL_b3svLP
14:30:21 001338 KAKC1604-I /tmp/SYSIN_001338_SAMPLE_JCL_b3svLP is deleted.
14:30:21 001338 KAKC1601-I SAMPLE_JCL.STEP5, SAMPLE_PROC.STEP1 Step allocated.
14:30:21 001338 KAKC1602-I PROC.DD1, . /tmp/SYSIN_001338_SAMPLE_JCL_xnB3Rt
14:30:21 001338 KAKC1603-I PROC.DD1, . /tmp/SYSIN_001338_SAMPLE_JCL_xnB3Rt
14:30:21 001338 KAKC1604-I /tmp/SYSIN_001338_SAMPLE_JCL_xnB3Rt is deleted.

***** JOBSTEP OUTPUT *****
KAKC0719-I STEP 001, STEP1, STDERR
%%SAMPLE_JCL_STEP1%%
%%SAMPLE_JCL_STEP1%%

KAKC0719-I STEP 002, STEP2, STDERR
%%SAMPLE_JCL_STEP2%%
%%SAMPLE_JCL_STEP2%%

KAKC0719-I STEP 003, STEP3, STDERR
%%SAMPLE_JCL_STEP3%%
%%SAMPLE_JCL_STEP3%%

KAKC0719-I STEP 004, STEP4, STDERR
%%SAMPLE_JCL_STEP4%%
%%SAMPLE_JCL_STEP4%%

KAKC0719-I STEP 005, STEP5, SAMPLE_PROC.STEP1, STDERR
%%replace SAMPLE_PROC_STEP3%%
%%replace SAMPLE_PROC_STEP3%%

KAKC0099-I HitachiBatchJobExec ended, rc=0

```

3.1.4 スプール

バッチジョブ実行システムでのジョブの実行結果を管理する。

バッチジョブ実行システムでは、ジョブの実行結果やジョブログをスプールに残すことができる。

機能の特長を次に示す。

- ジョブの実行結果ファイルをプリンタに出力するための書式属性を外部から与えることができる。
- ジョブの実行ごとにジョブ識別子が付けられ、ジョブごとの操作が可能となる。

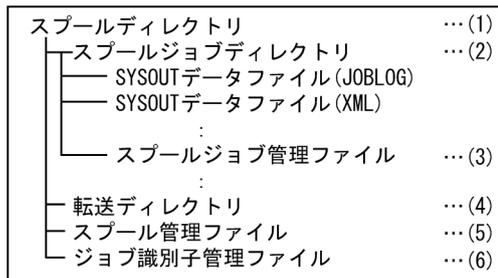
(1) スプールの制御

(a) スプールの構造

バッチジョブ実行システムでは、設定ファイルに指定したスプールディレクトリにジョブごとにディレクトリを作成し、ジョブ実行結果などのジョブごとの情報を出力する。

スプールのディレクトリの構造を次の図に示す。

図 3-3 スプールのディレクトリの構造



(1) スプールディレクトリ

スプールのルートディレクトリ（ディレクトリ名は、設定ファイルのスプールディレクトリ指定パラメータ SPOOL_DIR で指定）である。

(2) スプールジョブディレクトリ

ジョブ単位のディレクトリ（ディレクトリ名は、ジョブ識別子を使用する）である。

ジョブ実行結果および DD 要素の TYPE 属性で SYSOUT を指定した出力情報は、いったんスプール内に SYSOUT データファイルとして格納され、印刷指示に従い処理される。スプールジョブディレクトリは、SYSOUT データファイルとスプールジョブ管理ファイルから構成される。詳細については、「3.1.4(1)(b) スプールジョブの構造」を参照のこと。

(3) スプールジョブ管理ファイル

スプールジョブの構成や属性を管理するファイルである。

ファイル名：.sysout

(4) 転送ディレクトリ

スプールジョブ転送サービスによって、SYSOUT データファイルをプリントサーバに転送する場合に使用する作業用ディレクトリである。

ディレクトリ名：transfer

(5) スプール管理ファイル

スプールジョブ全体を管理するファイルであり、バッチジョブ実行システムが使用する。

このファイルが存在しない場合は、スプールジョブ転送デーモン起動時に自動生成する。

このファイルが破壊された場合は運用を停止し、このファイルを削除してスプールジョブ転送デーモンを起動後に bjexchjob コマンドの -S オプションで必要なスプールジョブごとに回復する。

ファイル名：.hold

(6) ジョブ識別子管理ファイル

このスプールのジョブ識別子を管理するファイルである。

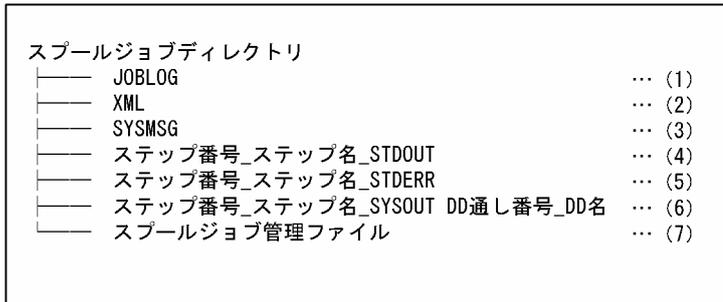
バッチジョブ実行システムが使用するため、参照や操作をしないこと。

ファイル名：.jobid

(b) スプールジョブの構造

スプールジョブのディレクトリの構造を次の図に示す。

図 3-4 スプールジョブのディレクトリの構造



(1) JOBLOG

ジョブ実行メッセージ。ジョブの動作状況を示すメッセージを格納する。

(2) XML

実行時のジョブ定義 XML。入力となるジョブ定義 XML ファイルを解析した結果である。

(3) SYSMSG

システムメッセージ。ジョブ定義 XML ファイルの解析結果やファイルの割り当てに関する情報をメッセージとして格納する。

(4) ステップ番号_ステップ名_STDOUT

このステップで実行したプログラムの標準出力。

(5) ステップ番号_ステップ名_STDERR

このステップで実行したプログラムの標準エラー出力。

(6) ステップ番号_ステップ名_SYSOUT DD 通し番号_DD 名

DD 要素の TYPE 属性に SYSOUT を指定したファイル。

スプールジョブディレクトリの項目の形式を次の表に示す。

表 3-3 スプールジョブディレクトリの項目の形式

項目	形式
ステップ番号	ジョブステップの通し番号。 プロシジャで展開したジョブステップも含めた通し番号。 001~255 の値とする。ステップ番号は 3 けたの 10 進数とする。
ステップ名	ジョブステップ名。 STEP 要素の NAME 属性に指定した文字列が 8 文字以上の場合、最初の 8 文字を SYSOUT データファイル名のステップ名として使用する。
SYSOUT DD 通し番号	ジョブステップ内に指定された SYSOUT DD の連番。 ジョブステップごとに 001~255 の値とする。DD 通し番号は 3 けたの 10 進数とする。

項目	形式
DD 名	ジョブ定義 XML ファイルで指定した SYSOUT データファイルの DD 名。

(7) スプールジョブ管理ファイル (.sysout)

このスプールジョブの、ジョブに関する情報、スプールジョブに含まれるファイルの情報、および印刷のための属性を管理するためにジョブコントローラが作成するファイルである。印刷する場合は、スプールジョブ管理ファイルの転送が完了したときに、Prinfina MANAGER SP および PDE が印刷ジョブの受信を完了したと判断し、印刷処理を開始する。

(c) スプールジョブのデータの流れ

ジョブ実行終了によってスプールジョブディレクトリに生成されたスプールジョブのデータは、bjexchjob コマンドなどの入力によって転送ディレクトリを経由してプリントサーバに転送される。転送ディレクトリのデータを印刷ジョブのデータ、プリントサーバへ転送されたデータを転送済みジョブのデータという。

スプールジョブのデータの流れを次の図と表に示す。

図 3-5 スプールジョブのデータの流れ



注※ ジョブ定義XMLファイルでのHOLD属性値でNOを指定、またはbjexchjobコマンドの-Rオプションの指定で印刷指示することを示す。

表 3-4 スプールジョブのデータの流れ

データ種別 名称	生成契機	削除契機	格納場所	意味
スプールジョブ	ジョブ実行終了	bjexrmjob コマンドによるスプールジョブ削除指示または転送処理後に保存が不要な場合	スプールジョブディレクトリ	実行が終了したジョブの結果をジョブごとにまとめたデータ。
印刷ジョブ	印刷指示	印刷指示 (転送完了後削除)	転送ディレクトリ	転送ディレクトリに格納された、転送待ちのジョブのデータ。
転送済みジョブ		プリントサーバ設定依存	プリントサーバ	プリントサーバに転送された、印刷待ちのジョブのデータ。

(2) 注意事項

スプールディレクトリはバッチジョブ実行システムで操作するため、ユーザは直接参照してはならない。スプールジョブ保管機能で保管したスプールジョブはユーザが自由に参照できる。

3.2 カタログドプロシジャと入力ストリームプロシジャ

3.2.1 プロシジャの概要

システムにジョブを入力する場合、ジョブ定義 XML ファイルを準備する必要がある。しかし、そのジョブがシステム資源をより多岐に利用する場合は、ジョブ定義 XML ファイルもより多く準備する必要がある。そこで、定型的ジョブについては、カタログドプロシジャまたは入力ストリームプロシジャの機能を利用することによって、ジョブ定義 XML ファイルの準備作業を簡素化できる。

(1) カタログドプロシジャ

あらかじめジョブ定義 XML の集まりをプロシジャライブラリ（ディレクトリ）に登録しておき、必要時にその集まりを CALL 要素で取り出すことができる。

(2) 入力ストリームプロシジャ

ジョブ定義 XML の集まりをファイル中に置き、そのジョブ内で必要時にジョブ定義 XML の集まりを取り出すことができる。

カタログドプロシジャと入力ストリームプロシジャを総称してプロシジャという。また、プロシジャ内には次に示すものを含むことはできない。

- ほかのプロシジャ名を指定した CALL 要素（プロシジャのネスト）
- JOB 要素, JOBPARM 要素, PERMISSION 要素, SETSYMBOL 要素, REPLACE 要素, および REPLACESTEP 要素
- NAME 属性に"JOBLIB"を指定した DD 要素

一度作成したプロシジャを変更して使用する場合、プロシジャそのものの変更をしないで一時的な修正ができる。これによって、プロシジャ使用時の変更、追加が容易となる。

プロシジャの修正については「[3.2.5 プロシジャの修正方法](#)」を参照のこと。

3.2.2 入力ストリームプロシジャの作成方法

入力ストリームプロシジャを使用する場合、事前に同一ジョブ内でプロシジャの定義をしておく必要がある。入力ストリームプロシジャの作成は、ジョブ定義 XML ファイルの PROC 要素および SYMBOL 要素で行う。

(1) PROC 要素

入力ストリームプロシジャの先頭に置き、プロシジャ名を定義する。

(2) SYMBOL 要素

プロシジャ内の記号パラメータに標準値を定義する。

3.2.3 カタログドプロシジャの使用方法

カタログドプロシジャを使用する場合、事前にプロシジャライブラリに登録しておく必要がある。プロシジャライブラリは、設定ファイルの PROCEDURE_DIR パラメータを定義して登録する。

カタログドプロシジャを作成する場合、カタログドプロシジャのファイル名はプロシジャ名の後部に ".xml" の拡張子を付加したファイルとして作成する。

カタログドプロシジャを呼び出す場合、CALL 要素の NAME 属性に、呼び出したいカタログドプロシジャのファイル名から.xml を除いた文字列を指定する。

カタログドプロシジャの使用例を次に示す。

カタログドプロシジャの例 (TEST.xml)

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <PROC NAME="TEST">
    <SYMBOL NAME="PAR1" VALUE="456" />
    <STEP>
      <EXEC PGM="PGMNAME" PARM="123#{PAR1}" />
      <DD NAME="DD1" DSN="#{PAR1}ABC" />
    </STEP>
  </PROC>
</HitachiBatchJobExec>
```

カタログドプロシジャ TEST を呼び出すジョブ定義 XML

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="CALLTEST" >
    <STEP NAME="STEP">
      <CALL NAME="TEST" >
        <SETSYMBOL NAME="PAR1" VALUE="789" />
      </CALL>
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

この例では、カタログドプロシジャ内の指定は次に示すとおりである。

- EXEC 要素の PARM 属性は"123456"
- DD 要素の DSN 属性は"456ABC"

しかし、プロシジャを呼び出す場合に SETSYMBOL 要素で PAR1 を利用して置き換えをしているため、次に示す値で実行される。詳細については「[3.2.5 プロシジャの修正方法](#)」を参照のこと。

- EXEC 要素の PARM 属性は"123789"
- DD 要素の DSN 属性は"789ABC"

3.2.4 入力ストリームプロシジャの使用法

入力ストリームプロシジャを使用する場合、事前に同一ジョブ定義 XML ファイル内で、PROC 要素を定義しておく必要がある。

入力ストリームプロシジャの使用例を次に示す（項番は項目の説明の番号と対応している）。

入力ストリームプロシジャの使用例

```

<?xml version="1.0" encoding="Shift_JIS" ?> ... (1)
<HitachiBatchJobExec version="1.3" os="unix"> ... (2)
  <JOB NAME="SAMPLE3"> ... (3)
    <PROC NAME="PROC3"> ... (4)
      <SYMBOL NAME="PAR1" VALUE="456" /> ... (5)
      <STEP NAME="STEP1"> ... (6)
        <EXEC PGM="PGM1" PARM="123#[PAR1]" /> ... (7)
      </STEP> ... (8)
    </PROC> ... (9)
    <STEP NAME="STEP2"> ... (10)
      <CALL NAME="PROC3"> ... (11)
        <SETSYMBOL NAME="PAR1" VALUE="789" /> ... (12)
      </CALL> ... (13)
    </STEP> ... (14)
  </JOB> ... (15)
</HitachiBatchJobExec> ... (16)

```

- (1)XML 宣言を示す。
- (2)ジョブ定義 XML ファイルのルート要素の開始を示す。
- (3)ジョブ SAMPLE3 の JOB 要素の開始を示す。
- (4)PROC 要素の開始を示す。
- (5)PROC 要素で定義された記号パラメータを SYMBOL 要素として定義する。
- (6)STEP 要素の開始を示す。
- (7)実行するプログラムを EXEC 要素として示す。
- (8)STEP 要素の終了を示す。
- (9)PROC 要素の終了を示す。
- (10)STEP 要素の開始を示す。
- (11)カタログドプロシジャ PROC3 の呼び出しのための CALL 要素の開始を示す。

(12)カタログドプロシジャ PROC3 の記号パラメータへの値の設定を示す。

(13)CALL 要素の終了を示す。

(14)STEP 要素の終了を示す。

(15)JOB 要素の終了を示す。

(16)ルート要素の終了を示す。

3.2.5 プロシジャの修正方法

プロシジャの修正方法を次に示す。この修正はプロシジャの使用時に一時的に修正するだけで、プロシジャそのものは変更しない。

- 記号パラメータを使用しない修正
 - プロシジャ内の EXEC 要素の修正
 - プロシジャ内の DD 要素の修正
 - プロシジャへの DD 要素の追加
- 記号パラメータによる修正
 - SETSymbOL 要素によるプロシジャ内の DD, EXEC, SETENV 要素の修正

3.2.6 プロシジャ内の EXEC 要素の修正方法

プロシジャ内の EXEC 要素の属性を修正したい場合、プロシジャを呼び出すための CALL 要素内に REPLACE 要素、REPLACESTEP 要素と修正する EXEC 要素の属性および値を記述する。REPLACESTEP 要素の NAME 属性には、修正したいプロシジャ内のジョブステップ名を記述する。属性の追加の場合も同じように記述する。ただし、PGM 属性は修正・追加できない。

この修正は、プロシジャ内の複数の EXEC 要素および各 EXEC 要素に対して複数の属性の修正を指定できる。

EXEC 要素の属性の置き換えまたは削除の指定方法を次に示す。

- プロシジャ内の EXEC 要素の属性をある値に置き換える場合

```
<REPLACE >  
  [<REPLACESTEP NAME="修正するプロシジャステップ名">  
    <EXEC 置き換えたい属性="値" />  
  [</REPLACESTEP>]  
</REPLACE>
```

- プロシジャ内の EXEC 要素の属性を無効にしたい場合

```
<REPLACE >
  [<REPLACESTEP NAME="修正するプロシジャステップ名">]
    <EXEC 無効にしたい属性="" />
  [</REPLACESTEP>]
</REPLACE>
```

EXEC 要素内のパラメータを REPLACE 要素で修正する場合、プロシジャ内のジョブステップ名を指定する REPLACESTEP 要素を省略すると、プロシジャ内の先頭のジョブステップを修正する。

プロシジャを REPLACE 要素で修正する方法を次に示す。

(1) 先頭ステップだけを修正する方法

カタログドプロシジャの例 (TEST2.xml)

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <PROC NAME="TEST2">
    <STEP>
      <EXEC PGM="PGMNAME" PARM="123" COND="0,NE" />
    </STEP>
  </PROC>
</HitachiBatchJobExec>
```

カタログドプロシジャ TEST2 を呼び出すジョブ定義 XML

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="CALLTEST2" >
    <STEP>
      <CALL NAME="TEST2" >
        <REPLACE>
          <EXEC PARM="456"/>
        </REPLACE>
      </CALL>
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

この例では、カタログドプロシジャ内の指定は次に示すとおりである。

- EXEC 要素の PGM 属性は"PGMNAME"
- EXEC 要素の PARM 属性は"123"
- EXEC 要素の COND 属性は"0,NE"

しかし、プロシジャを呼び出すときに REPLACE 要素で PARM 属性だけを置き換える。プロシジャ中の先頭ステップに対する修正であるため、REPLACESTEP 要素を記述する必要はない。COND 属性は指定をしていないため、最初に指定した"0,NE"が有効となり、次に示す値で実行する。

- EXEC 要素の PGM 属性は"PGMNAME"

- EXEC 要素の PARM 属性は"456"
- EXEC 要素の COND 属性は"0,NE"

(2) 先頭ステップ以外のステップを修正する方法

カタログドプロシジャの例 (TEST3.xml)

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <PROC NAME="TEST3">
    <STEP NAME="STEP1">
      <EXEC PGM="PGMNAME" PARM="123" COND="0,NE" />
    </STEP>
    <STEP NAME="STEP2">
      <EXEC PGM="PGMNAME" PARM="456" COND="8,LE" />
    </STEP>
  </PROC>
</HitachiBatchJobExec>
```

カタログドプロシジャ TEST3 を呼び出すジョブ定義 XML

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="CALLTEST3" >
    <STEP>
      <CALL NAME="TEST3" >
        <REPLACE>
          <REPLACESTEP NAME="STEP2">
            <EXEC PARM="789"/>
          </REPLACESTEP>
        </REPLACE>
      </CALL>
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

この例では、カタログドプロシジャ内の指定は次に示すとおりである。

- STEP1 の EXEC 要素の PGM 属性は"PGMNAME", PARM 属性は"123", COND 属性は"0,NE"
- STEP2 の EXEC 要素の PGM 属性は"PGMNAME", PARM 属性は"456", COND 属性は"8,LE"

しかし、プロシジャを呼び出すときに REPLACE 要素と REPLACESTEP 要素で STEP2 の PARM 属性だけを置き換える。COND 属性は指定をしていないため、最初に指定した"8,LE"が有効となり、次に示す値で実行する。

- STEP1 の EXEC 要素の PGM 属性は"PGMNAME", PARM 属性は"123", COND 属性は"0,NE"
- STEP2 の EXEC 要素の PGM 属性は"PGMNAME", PARM 属性は"789", COND 属性は"8,LE"

3.2.7 プロシジャ内の DD 要素の修正方法

プロシジャ内の DD 要素の修正では次に示すことができる。

- 属性値の変更
- 属性の追加
- 属性の削除
- DD 要素の無効化

プロシジャ内の DD 要素の属性を修正する場合は、プロシジャを呼び出すための CALL 要素に続けて、修正情報を持つ REPLACE 要素内に DD 要素を定義する。この REPLACE 要素には、修正したいプロシジャ内のジョブステップ名（プロシジャステップ名）を REPLACESTEP として定義し、実際に修正の対象となる DD 要素の名前（NAME 属性）、修正する属性、および属性値を記述する。また、属性値を無効にする修正の場合は、値を""で指定する。この修正は、EXEC 要素の修正と同じく、プロシジャ内の複数の DD 要素および各 DD 要素に対して複数の属性の修正が指定できる。

プロシジャ中の DD 要素は、そのプロシジャが呼び出されたとき、DD 要素の修正がすべて完了した時点で排他的関係のチェックをする。

なお、NAME 属性に対して属性値の変更を行うことはできない。

(1) DD 要素を修正するための指定方法

(a) 属性値の変更

DD 要素の属性値を変更するための指定方法を次に示す。

プロシジャを呼び出した CALL 要素に続いて、修正情報を持つ REPLACE 要素内に DD 要素を指定する。

```
<REPLACE >  
  [<REPLACESTEP NAME="修正するプロシジャステップ名">]  
    <DD NAME="修正したいDD要素のDD名" 修正する属性="修正する属性値" />  
  [</REPLACESTEP>]  
</REPLACE>
```

プロシジャステップを省略した場合、先頭のプロシジャステップが仮定される。

(b) 属性の追加

DD 要素の属性を追加するための指定方法を次に示す。

指定方法は「属性値の変更」と同じだが、プロシジャ内の DD 要素に指定していない属性を、修正する DD 要素に定義することによって追加となる。

```
<REPLACE >  
  [<REPLACESTEP NAME="修正するプロシジャステップ名">]  
    <DD NAME="修正したいDD要素のDD名" 追加する属性="追加する属性値" />
```

```
[</REPLACESTEP>]
</REPLACE>
```

この場合、追加した属性と排他的な関係の属性をプロシジャ内の DD 要素に指定していると、追加処理後に背反のチェックをする。

(c) 属性の削除

DD 要素の属性を削除するための指定方法を次に示す。

修正する属性の値を空文字列 (= "") にすることによって、プロシジャ内の DD 要素に指定している属性を削除できる。

```
<REPLACE >
  [<REPLACESTEP NAME="削除するプロシジャステップ名">]
    <DD NAME="削除したいDD要素のDD名" 削除する属性="" />
  [</REPLACESTEP>]
</REPLACE>
```

(2) プロシジャ内の DD 要素を無効化にする場合

DD 要素を無効化する場合は、TYPE 属性に DUMMY を指定する。ダミーファイルに変更することで、該当する DD 要素を無効化できる。なお、無効化される DD 要素に TYPE=DUMMY と背反する属性値が指定されている場合も、TYPE 属性値以外の属性値はすべて無視される。

(3) 連結した DD 要素の修正

プロシジャ内で連結した DD 要素が定義されている場合、その連結を 1 つの DD 要素で置き換えようとすると最初の DD 要素だけ置き換わる。連結している残りの DD 要素を置き換える場合には、各 DD 要素に対して、それぞれ置き換え用の DD 要素が必要である。置き換えを指定する DD 要素はプロシジャ中の連結 DD 要素と同じ順序とすること。

また、連結 DD 要素のどれかを変更したくない場合は、置き換えを指定する対応 DD 要素は、NAME 属性だけを指定する。

カタログドプロシジャの例 (TEST4.xml)

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <PROC NAME="TEST4">
    <STEP NAME="STEP00">
      <EXEC PGM="PGMNAME" PARM="123" />
      <DD NAME="DD00" DSN="A" />
      <DD NAME="DD00" DSN="B" />
      <DD NAME="DD00" DSN="C" />
      <DD NAME="DD00" DSN="D" />
    </STEP>
  </PROC>
</HitachiBatchJobExec>
```

カタログドプロシジャ TEST4 を呼び出すジョブ定義 XML

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="CALLTEST4" >
    <STEP NAME="PROCCALL">
      <CALL NAME="TEST4" >
        <REPLACE>
          <DD NAME="DD00" />
          <DD NAME="DD00" DSN="X" />
          <DD NAME="DD00" />
          <DD NAME="DD00" DSN="Y" />
        </REPLACE>
      </CALL>
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

修正後の DD 要素

```
<DD NAME="DD00" DSN="A" />
<DD NAME="DD00" DSN="X" /> ; 修正された
<DD NAME="DD00" DSN="C" />
<DD NAME="DD00" DSN="Y" /> ; 修正された
```

プロシジャ内に定義されている連結された DD 要素のうち、2 番目と 4 番目の DSN 属性をそれぞれ "X", "Y" に修正して実行することを示している。1 番目と 3 番目の DSN 属性は元の "A", "C" である。

3.2.8 プロシジャ内への DD 要素の追加方法

プロシジャ内に存在する DD 要素の属性の修正だけでなく、新たにプロシジャに DD 要素を追加できる。

DD 要素を追加するための指定方法を次に示す。

追加する DD 要素は、プロシジャを呼び出すための CALL 要素内の REPLACE 要素、REPLACESTEP 要素内に指定する。REPLACESTEP 要素の NAME 属性には、追加したいプロシジャ内のジョブステップ名を記述する。この追加 DD 要素には、そのジョブステップ内で指定していない NAME 要素を指定する。

```
<REPLACE >
  [<REPLACESTEP NAME="追加するプロシジャステップ名">]
  <DD NAME="追加したいDD要素のDD名" ~ />
  [</REPLACESTEP>]
</REPLACE>
```

プロシジャ内のジョブステップ名を指定する REPLACESTEP 要素を省略した場合、プロシジャ内の先頭ジョブステップに DD 要素を追加する。

3.2.9 プロシジャの記号パラメータによる修正方法

プロシジャを使用するたびに、プロシジャ内の属性値を修正する場合の簡易方法として記号パラメータによる方法がある。

(1) 記号パラメータの定義

記号パラメータは、プロシジャ内の属性値内に"#{"文字と"}"文字で囲むことによって定義する。

- 記号パラメータは、"#{"と"}"を前置して、それに1~31文字の英数字< (#, @, __, も含む) >から成る文字列であること。
- プロシジャ内で、同じ記号パラメータを与えることができる。すなわち、プロシジャ内の同一要素は、1種類の記号パラメータによって定義できる。

(2) 記号パラメータに対する値の割り当て

記号パラメータを含むプロシジャを使用する場合、それぞれの記号パラメータに値を割り当てるかまたは無効にするための手続きを次に示すようにして行う。

プロシジャを呼び出すためのCALL要素で、記号パラメータの値を割り当てるかまたは無効にする。

プロシジャ内の記号パラメータを定義する場合に、SYMBOL要素によって標準値を割り当てる。なお、SYMBOL要素は、記号パラメータに標準値を割り当てるときにだけプロシジャの先頭に置く。

記号パラメータを使用して、1つのパラメータ中の一部を修正できる。

記号パラメータでの値の割り当ての記述順序は、プロシジャ内の記号パラメータの順序と一致している必要はない。

- 記号パラメータを無効にするための指定方法を次に示す。

```
<SETSYMBOL NAME="記号パラメータ" VALUE="" />
```

- 記号パラメータ値を割り当てるための指定方法を次に示す。

```
<SETSYMBOL NAME="記号パラメータ" VALUE="値" />
```

割り当て時の注意事項を次に示す。

- SYMBOL要素を複数記述して、同一記号パラメータに複数の標準値を割り当てた場合は、最初に指定された値を割り当てる。この機能を利用して、同じ記号パラメータに対する値を複数指定しておき、使用するとき必要な値を先頭に持って来るようにするとよい。
- SETSYMBOL要素を複数記述して、同一記号パラメータに複数の値を割り当てた場合は、最後に指定された値を割り当てる。

- 置き換えができるのは、EXEC 要素、DD 要素および SETENV 要素の中の属性値（"で囲まれている中）である。ただし、EXEC 要素の PGM 属性と DD 要素および SETENV 要素の NAME 属性値は置き換えできない。
- ジョブ定義 XML ファイル中に記号パラメータの記述目的以外で「#{…}」または「#{ }」は記述できない。

3.2.10 プロシジャライブラリの定義方法

プロシジャライブラリは、カタログドプロシジャファイルを格納したディレクトリパスを設定ファイルで定義する。プロシジャライブラリは、複数のディレクトリを定義して連結されたライブラリとして処理できる。また、プロシジャライブラリにグループ名を付加してプロシジャグループとして定義できる。連結定義した場合、定義した順番にカタログドプロシジャファイルを検索する。

同一パスを複数のグループに定義することができ、グループで共通のライブラリを定義できる。また、同一グループの定義は連続している必要はない。

グループ名なしのプロシジャグループを定義しなくても、設定ファイルはエラーにならない。しかし、デフォルトグループのプロシジャライブラリを使用するジョブは、ジョブ実行時にカタログドプロシジャが見つからないためエラーとなる。

複数のプロシジャライブラリの定義例を次に示す。

例	
PROCEDURE_DIR /bjex/procdir	グループなしのプロシジャグループ
PROCEDURE_DIR /bjex/procdirA001 GROUPA	} GROUPAのプロシジャグループ
PROCEDURE_DIR /bjex/procdirA002 GROUPA	
PROCEDURE_DIR /bjex/procdirA003 GROUPA	
PROCEDURE_DIR /bjex/procdir GROUPB	} GROUPBのプロシジャグループ
PROCEDURE_DIR /bjex/procdirB001 GROUPB	
PROCEDURE_DIR /bjex/procdirB002 GROUPB	
PROCEDURE_DIR /bjex/procdirComm GROUPA...	GROUPAのプロシジャグループ
PROCEDURE_DIR /bjex/procdirComm GROUPB...	GROUPBのプロシジャグループ

- **グループ名なしのライブラリ**

/bjex/procdir はグループ名なしのプロシジャライブラリとして登録しており、プロシジャグループの指定がないジョブでは、グループ名なしのライブラリのカタログドプロシジャを使用する。

- **GROUPA のライブラリ**

/bjex/procdirA001, /bjex/procdirA002, /bjex/procdirA003, /bjex/procdirComm が連結されており、ジョブでプロシジャグループとして GROUPA を指定した場合、GROUPA のライブラリのカタログドプロシジャを使用する。検索順は定義順である。

- **GROUPB のライブラリ**

/bjex/procdir, /bjex/procdirB001, /bjex/procdirB002, /bjex/procdirComm が連結されており、ジョブでプロシジャグループとして GROUPB を指定した場合、GROUPB のライブラリのカタログドプロシジャを使用する。検索順は定義順である。

- 共通ライブラリ

/bjex/procdir はグループなしと GROUPB で共通に使用されている。また、/bjex/procdirComm は GROUPA と GROUPB で共通に使用される。

ジョブ定義 XML ファイルでプロシジャグループ名を指定することで使用するプロシジャライブラリを切り替えて使用できる。

ジョブ定義 XML ファイルでプロシジャグループ名を指定した場合の定義例を次に示す。

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
<JOB NAME="SAMPLE" >
<JOBPARM PROCLIB="GROUPA" />      -->GROUPAのプロシジャグループを使用する指定
:
:
```

また、ジョブを業務種類ごとに JP1/AJS のキューで分けている場合、そのキュー名をプロシジャグループ名として使用し、プロシジャライブラリを業務種類ごとに切り替えることができる。JP1/AJS のキュー名をプロシジャグループ名として利用する場合は、PROCGROUP_QUEUENAME パラメータで YES を指定する必要がある。

3.3 プログラムのシステムへの登録と使用

プログラムを実行するためには、プログラムを格納したファイルのパスを特定する必要がある。

PATH 環境変数にプログラムを格納したディレクトリをあらかじめ登録すれば、プログラムのファイル名だけを指定して、プログラムを実行できる。

バッチジョブ実行システムでは、JOB LIB DD 要素、STEPLIB DD 要素で PATH 環境変数に登録するディレクトリを追加できる。

3.3.1 システムのプログラム使用方法

システムのプログラムを使用するために、通常、DD 要素を用意する必要はない。システムのプログラムが格納されたディレクトリは、通常、ログインプロファイルで PATH 環境変数に登録される。

PATH 環境変数に登録されていないディレクトリに存在するシステムのプログラムを実行する場合は、「(2) ユーザプログラムの使用方法」で、システムのプログラムが存在するディレクトリを PATH 環境変数に登録する。

システムのプログラムは、EXEC 要素の PGM 属性に指定して実行する。

3.3.2 ユーザプログラムの使用方法

特定のジョブまたはジョブステップで使用するユーザプログラムは、通常、システムのプログラムとは別のディレクトリに格納する。ユーザプログラムを格納したディレクトリのパスを、JOB LIB DD 要素、または STEPLIB DD 要素で指定すると、バッチジョブ実行システムは、ジョブステップの実行開始時に、PATH 環境変数の先頭に登録し、ジョブステップの実行終了時に PATH 環境変数への登録を削除する。

STEPLIB DD 要素を指定した場合、STEPLIB DD 要素で指定されたディレクトリを、PATH 環境変数の先頭に登録する。

JOB LIB DD 要素を指定した場合、JOB LIB DD 要素で指定されたディレクトリを、PATH 環境変数の先頭に登録する。

STEPLIB DD 要素が指定された場合、PATH 環境変数に JOB LIB DD 要素で指定されたディレクトリは登録されない。

- **JOB LIB DD 要素**

1 つのジョブの中で有効なプログラムのディレクトリであり、DD 要素の名前は JOB LIB である。JOB LIB DD 要素の指定位置は、JOB 要素の中で最初の STEP 要素の前にする必要がある。

- **STEPLIB DD 要素**

1つのジョブステップの中で有効なプログラムのディレクトリであり、DD要素の名前はSTEPLIBである。STEP要素内でEXEC要素の後ろで定義する。

PATH環境変数にプログラムのディレクトリを複数登録する場合、DSN属性に登録するプログラムのディレクトリを記述したJOBLIB DD要素（またはSTEPLIB DD要素）を連続して記述する。バッチジョブ実行システムは、JOBLIB DD要素（またはSTEPLIB DD要素）を記述した順序で、プログラムのディレクトリをPATH環境変数の先頭に登録する。次に示す例の場合、PATH環境変数の値がDIR0であると、ジョブステップ実行時のPATH環境変数の値は、DIR1:DIR2:DIR0となる。

```
<DD NAME="JOBLIB" TYPE="LIB" DSN="DIR1" DISP="SHR"/>
<DD NAME="JOBLIB" TYPE="LIB" DSN="DIR2" DISP="SHR"/>
```

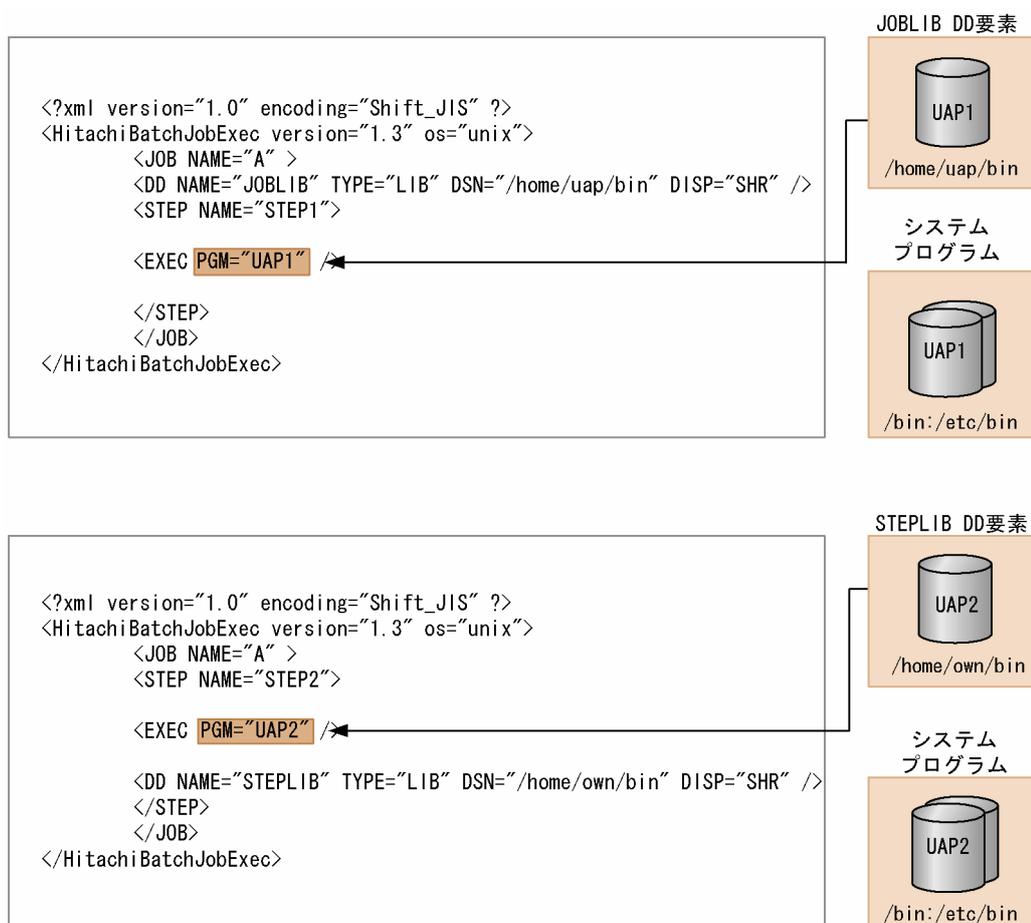
(1) JOBLIB, STEPLIB, および PATH 環境変数の使用方法

JOBLIB, STEPLIB, および PATH 環境変数の使用方法を次に示す。

- 使用するプログラムパスに対するDD要素を準備する。JOBLIBの場合はJOB要素の直後に、STEPLIBの場合はジョブステップ内のほかのDD要素群と一緒に置く。ジョブ内の複数のジョブステップが同一ディレクトリ内のプログラムを利用する場合は、JOBLIBを指定した方が簡単である。

JOBLIB および STEPLIB の使用例を次の図に示す。

図 3-6 JOBLIB および STEPLIB の使用例



- ジョブ内で JOBLIB および STEPLIB の両方を指定した場合、そのジョブステップでは STEPLIB を優先して、JOBLIB の指定を無視する。
- DD 要素の後処理 (DISP 属性) は省略するか、または SHR パラメータを指定する。

3.4 ファイルの割り当て

3.4.1 ファイルの準備

バッチジョブ実行システムでは、ファイル使用時のファイルのファイル名、パス名、および使用後のファイルの処置などをあらかじめジョブ定義 XML ファイルに記述する必要がある。

なお、この章以降の一時ファイルの記述は、特に断りのないかぎり一時 ISAM ファイルにも該当する。

ジョブコントローラは、次の表に示すようにこれらの情報を基にして、ジョブステッププロセスの実行に先立ちファイルを割り当てる。ジョブステッププロセス終了後は、不要となった資源を解放し、ほかのジョブで使用できるようにするための後処理をする。

表 3-5 DD 要素の DISP 属性値の指定とファイルの前処理との関係

DISP 属性値	ファイルの確認	ファイルの前処理	ファイルの準備
NEW パラメータ	確認する。	既存ファイルまたはディレクトリがあれば割り当てエラーとなる。	0 バイトファイルまたは新規ディレクトリを作成する。
RNW パラメータ※1		既存ファイルまたはディレクトリがあれば削除する。	
SHR パラメータ	「表 3-6 DISP 属性が SHR または OLD の場合の、ファイルの確認と前処理」を参照のこと。※2		なし。
OLD パラメータ			
MOD パラメータ※3 ※4	確認する。※5	既存ファイルがなく、かつ、DISPMOD_NOFILE パラメータに NEW を指定していればファイルを準備する。	既存ファイルがなければ 0 バイトファイルを作成する。
		既存ファイルがなく、かつ、DISPMOD_NOFILE パラメータに ERROR を指定していればエラーとなる。	なし。

注※1 ディレクトリに対してこの属性値を指定した場合は、ジョブの実行時にエラーとなり、ジョブが終了する。

注※2 世代ファイルの場合、登録先世代群名が存在していない場合や、世代管理ファイルに未登録の相対世代番号を指定した場合は、割り当てエラーとなる。

注※3 プログラム検索ディレクトリ、PREST ファイルに対してこの属性値を指定した場合は、SHR 指定として動作する。ディレクトリに対してこの属性値を指定した場合は、ジョブ定義 XML ファイル解析エラーとなる。

注※4 MOD 属性値の指定は、既存の世代ファイル（既存の相対世代番号）指定の場合、世代管理ファイルに登録されていないと割り当てエラーとなる。

注※5 一時ファイルに対してこの属性値を指定した場合は、先行ステップで PASS 指定された一時ファイルの確認をする。

表 3-6 DISP 属性が SHR または OLD の場合の、ファイルの確認と前処理

存在確認対象ファイル	ファイルの確認	ファイルの前処理
一時ファイル	先行ジョブステップで PASS 指定された一時ファイルについて存在確認を行う。	先行ジョブステップで PASS 指定された一時ファイルが存在しない場合、KAKC1614-E メッセージを出力して割り当てエラーとなる。
恒久ファイル、 世代ファイル	次の 2 点のどちらかの条件を満たす場合に確認する。 1. EXEC 要素の LANG 属性に "COBOL" を指定したジョブステップで次の条件をすべて満たす場合 ・ DD 要素でファイル連結として指定している。 ・ DD_EXISTCHECK_COBOL パラメータに CONCAT を指定している。 2. EXEC 要素の LANG 属性に "COBOL" を指定していないジョブステップで次の条件をすべて満たす場合 ・ DD 要素でファイル連結として指定している。 ・ DD_EXISTCHECK パラメータに CONCAT を指定している。	左記の条件を満たしてファイルの確認をした場合、ファイルが存在しなければ KAKC1660-E メッセージを出力して割り当てエラーとなる。
ディレクトリ	なし。	なし。

3.4.2 ファイルの状態と後処理

ジョブステップで必要とするファイルは、DD 要素の DISP 属性の指定に基づいて準備する。実際に処理されたファイルは、ジョブステップの終了時に DD 要素の DISP 属性の指定に基づいて後処理をする。

ジョブステップで必要とするファイルの準備および終了時のファイルの後処理をするために指定する DD 要素の DISP 属性の内容を次に示す。

- ファイルの状態
NEW, RNW, OLD, SHR, または MOD パラメータ
- ジョブステップ終了時のファイルの後処理
DELETE, KEEP, または PASS パラメータ

(1) ファイルの状態の指定

準備するファイルの状態を指定する。

排他/共用制御の詳細については、「3.13 ジョブ単位のファイルおよびディレクトリに対する排他/共用制御機能」を参照のこと。

(a) NEW パラメータ

該当するジョブステップで新しく作成するファイルの場合に指定する。ファイルがすでにある場合はエラーとなる。一時ファイルの場合は、同一ファイル名を指定するとユニークな名称を付けて別のファイルを作成する。

(b) RNW パラメータ

同一ファイルがある場合、該当するファイルを再使用する（すでにあるファイルをいったん削除し、再度ファイルを割り当てる）。同一 DD 要素から成るジョブの再実行を DD 要素の変更なしで可能とする。一時ファイルで RNW パラメータを指定した場合、NEW パラメータが仮定される。

また、指定したファイルシステム上に同一ファイル名を持つファイルがない場合は、NEW パラメータを指定した場合と同様に新規に領域を確保する。

RNW パラメータを指定する場合の注意事項を次に示す。

- PASS による情報を RNW 指定して参照する（受け取る）ことはできない。
- RNW パラメータで指定した同一ファイルがすでにある場合、該当するファイルを削除する。このため、ファイルを保護する目的として RNW パラメータの指定は同一ジョブの再実行を前提とする必要がある。
- ディレクトリに対して、RNW パラメータを指定すると、ジョブ実行時にエラーとなり、ジョブが終了する。

(c) OLD パラメータ

該当するジョブステップの開始以前にすでにあるファイルを指定する。このパラメータは、ファイルに対しての書き込み、またはファイル自身を削除する場合に指定する。

ジョブステップ開始前にファイルパスの存在の有無を確認するかどうかは、DD_EXISTCHECK_COBOL パラメータ、または DD_EXISTCHECK パラメータの指定に従う。詳細については、「8. 設定ファイル」の「DD_EXISTCHECK_COBOL (DD 要素に指定したファイルパスの存在確認要否を実施するための定義)」、または「DD_EXISTCHECK (DD 要素に指定したファイルパスの存在確認要否を実施するための定義)」を参照のこと。

(d) SHR パラメータ

該当するジョブステップの開始以前にすでにあるファイルを指定する。このパラメータは、ファイルに対して参照だけする場合に指定する。

ジョブステップ開始前にファイルパスの存在の有無を確認するかどうかは、DD_EXISTCHECK_COBOL パラメータ、または DD_EXISTCHECK パラメータの指定に従う。詳細については、「8. 設定ファイル」の「DD_EXISTCHECK_COBOL (DD 要素に指定したファイルパスの存在確認要否を実施するための定

義)], または「DD_EXISTCHECK (DD 要素に指定したファイルパスの存在確認要否を実施するための定義)」を参照のこと。

(e) MOD パラメータ

同一ファイルがある場合、該当するファイルに対して追加書きをするときに指定する。実行するプログラムがファイル操作ユーティリティ※, または EXEC 要素の LANG 属性の値が COBOL の場合に追加書きをする。それ以外の場合の書き込み動作は、ユーザプログラムに依存する。

また、指定したファイルシステム上に同一ファイル名を持つファイルがない場合に、新規ファイルを割り当てるかエラーにするかを DISPMOD_NOFILE パラメータで指定することができる。詳細については、「8. 設定ファイル」の「DISPMOD_NOFILE (DD 要素の DISP 属性に MOD を指定したファイルが存在しない場合の処理の定義)」を参照のこと。

DD 要素の TYPE 属性の値が FILE, TEMP, または GDG の場合だけ追加書きが有効である。

DD 要素の TYPE 属性の値が TEMPISAM である場合は、書き込み動作はユーザプログラムに依存する。

DD 要素の TYPE 属性の値が LIB, PREST, または TEMPPREST の場合は、SHR パラメータが指定されたものとして動作する。

DD 要素の TYPE 属性の値が DIR の場合は、ジョブ定義 XML ファイル解析エラーとなる。

注※ ファイル操作ユーティリティでは、bjexu_scopy, bjexu_ismut, および bjexu_ismut64 の場合に追加書きできる。このうち bjexu_scopy では、DD 要素に DISP="MOD"を指定したとき、MOD として処理するか、または OLD として処理するかを選択できる。詳細については、「3.18.7(1)(a)ファイルの複写」を参照のこと。

(2) ファイルの後処理の指定

ジョブステップ終了時のファイルの後処理を指定する。エラーが発生した場合は、原則ジョブ終了になるが、ファイル削除エラーの場合は、警告メッセージを出力してジョブを続行する。

なお、ファイル削除エラーが発生しても、ジョブステップのリターンコードの内容は変更されない。

(a) DELETE パラメータ

ファイルの存在を示す情報をファイルシステムから削除し、そのファイルが占有していたスペースを、ほかのファイルへ割り当て可能とする。

(b) KEEP パラメータ

後続のジョブステップまたはジョブで、そのファイルの削除を要求 (DELETE パラメータ指定) するまで、そのままの状態を保存する。

(c) PASS パラメータ

PASS パラメータは一時ファイルでだけ指定できる。

ジョブ内のジョブステップ間で同一の一時ファイル名を使用する場合、PASS パラメータを指定することによって順次、後続ジョブステップへそのファイルを受け渡すことができる。ただし、ジョブステップが異常終了した場合、ファイルの後処理で PASS パラメータは指定できない。後続ジョブステップへパスしたファイルは、ジョブ終了時に削除する。

一時ファイルに対して DISP="SHR"または DISP="OLD"で割り当てをする場合、先行ジョブステップで一時ファイルの作成を行い、PASS パラメータを指定した DD のパス情報を受け取れた場合だけ割り当てが可能となる。

一時ファイルに対して DISP="MOD"で割り当てをする場合、先行ジョブステップで一時ファイルを作成し、PASS パラメータを指定した DD のパス情報を受け取れたときは受け取ったパスの一時ファイルを割り当て、パス情報を受け取れなかったときは新規の一時ファイルを作成する。

一時ファイルの受け渡しをするには、DSN 属性値を同一にする必要がある。ファイル名には DSN 属性の最後の"/"より後ろの文字列だけ使用するが、一時ファイルの受け渡しに使用する識別名には最後の"/"より前も考慮されるので、パス名全体が同一になるようにすること。一時ファイルの受け渡しをする場合、誤りの原因となるため DSN 属性値にはファイル名だけを指定すること。

• 例 1

- ステップ STEP01 の DD 名 DD1 で作成した一時ファイルは、ステップ STEP02 の DD 名 DD2 で使用できる。
- ステップ STEP01 の DD 名 DD1 とステップ STEP02 の DD 名 DD2 で、TYPE=TEMP, DSN 属性に同一のファイル名を指定する必要がある。
- 先行ステップであるステップ STEP01 の DD 名 DD1 では、一時ファイルを作成する必要がある。作成した一時ファイルを後続ステップで使用できるようにするためには、DISP 属性の第 2 パラメータで PASS を指定する必要がある。PASS を指定しなかった場合、後続ステップで一時ファイルを使用できない。
- 後続ステップであるステップ STEP02 の DD 名 DD2 では、DISP の第 1 パラメータで、OLD, SHR, または MOD を指定する必要がある。NEW, RNW を指定した場合、先行ステップであるステップ STEP01 の DD 名 DD1 で作成した一時ファイルを使用できない。別の一時ファイルの新規作成となる。

```

<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version = "1.3" os="unix">
  <JOB NAME="TESTJOB">
    <STEP NAME="STEP01">
      <EXEC PGM="PROGRAM1" />
      <DD NAME="DD1" TYPE="TEMP" DSN="temp01" DISP="NEW,PASS" />
    </STEP>
    <STEP NAME="STEP02">
      <EXEC PGM="PROGRAM2" />
      <DD NAME="DD2" TYPE="TEMP" DSN="temp01" DISP="OLD" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

• 例 2

同一の DSN 属性で、PASS 指定した一時ファイルが複数ある場合

- 同一の DSN 属性で、複数の一時ファイルを作成し、PASS 指定することは、誤りの原因となるため、使用してはならない。
- 例では、ジョブステップ STEP01 の DD 名 DD1 とジョブステップ STEP02 の DD 名 DD2 で、同一の DSN 属性 "temp01" で一時ファイルが作成され、DISP 属性の第 2 パラメータで PASS が指定されている。
- 同一の DSN 属性 "temp01" であっても、ジョブステップ STEP01 の DD 名 DD1 とジョブステップ STEP02 の DD 名 DD2 の一時ファイルは、異なる一時ファイルとなる。
- ジョブステップ STEP03 の DD 名 DD3 は、ジョブステップ STEP01 の DD 名 DD1 で作成した一時ファイルを使用する。
- ジョブステップ STEP04 の DD 名 DD4 は、ジョブステップ STEP03 の DD 名 DD3 の DISP 属性で DELETE の指定が行われ削除されているため、ジョブステップ STEP02 の DD 名 DD2 で作成した一時ファイルを使用する。

```

<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version = "1.3" os="unix">
  <JOB NAME="TESTJOB">
    <STEP NAME="STEP01">
      <EXEC PGM="PROGRAM1" />
      <DD NAME="DD1" TYPE="TEMP" DSN="temp01" DISP="NEW,PASS" />
    </STEP>
    <STEP NAME="STEP02">
      <EXEC PGM="PROGRAM2" />
      <DD NAME="DD2" TYPE="TEMP" DSN="temp01" DISP="NEW,PASS" />
    </STEP>
    <STEP NAME="STEP03">
      <EXEC PGM="PROGRAM3" />
      <DD NAME="DD3" TYPE="TEMP" DSN="temp01" DISP="OLD,DELETE" />
    </STEP>
    <STEP NAME="STEP04">
      <EXEC PGM="PROGRAM4" />
      <DD NAME="DD4" TYPE="TEMP" DSN="temp01" DISP="OLD" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

• 例 3

例 2 でジョブステップ STEP03 の DD 名 DD3 で、DISP 属性で PASS を指定した場合

- ジョブステップ STEP04 の DD 名 DD4 は、ジョブステップ STEP01 の DD 名 DD1 で作成した一時ファイルを使用する。
- 同一の DSN 属性で、PASS 指定された一時ファイルが複数ある場合、先に作成した一時ファイルから順に使用する。
- 例 2, 例 3 で示すように、同一の DSN 属性で、PASS 指定された一時ファイルが複数ある状況は、PASS 指定の有無で、後続のジョブステップが使用する一時ファイルが異なる。誤りの原因となるので、同一の DSN 属性である複数の一時ファイルを使用しないこと。

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version = "1.3" os="unix">
  <JOB NAME="TESTJOB">
    <STEP NAME="STEP01">
      <EXEC PGM="PROGRAM1" />
      <DD NAME="DD1" TYPE="TEMP" DSN="temp01" DISP="NEW.PASS" />
    </STEP>
    <STEP NAME="STEP02">
      <EXEC PGM="PROGRAM2" />
      <DD NAME="DD2" TYPE="TEMP" DSN="temp01" DISP="NEW.PASS" />
    </STEP>
    <STEP NAME="STEP03">
      <EXEC PGM="PROGRAM3" />
      <DD NAME="DD3" TYPE="TEMP" DSN="temp01" DISP="OLD.PASS" />
    </STEP>
    <STEP NAME="STEP04">
      <EXEC PGM="PROGRAM4" />
      <DD NAME="DD4" TYPE="TEMP" DSN="temp01" DISP="OLD" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

(d) ファイル割り当て処理中にエラーが発生した場合の後処理の指定

バッチジョブ実行システムでは、ファイルを DD 要素に指定して割り当て、または割り当てたファイルの割り当てを解除する場合、DISP 属性値の指定に従って処理する。

ジョブステップの割り当て処理中にエラーが発生し、割り当てたファイルの後処理をする場合、デフォルト状態（または設定ファイルのパラメータ DD_DEALLOC_ON_ALLOCERR に OPT1 を指定した場合は、表 3-7 に従って処理する。

ジョブステップ開始以前に存在するファイルを DISP 属性値の指定に関係なく KEEP で保存する運用をする場合、DD_DEALLOC_ON_ALLOCERR に OPT2 を指定することで表 3-8 に示す動作にできる。

ファイル割り当て処理中にエラーが発生した場合の後処理方法を、デフォルト状態または DD_DEALLOC_ON_ALLOCERR パラメータに OPT1 を指定している場合の動作を表 3-7 に、OPT2 を指定している場合の動作を表 3-8 に示す。なお、表 3-7 と表 3-8 で異なる部分については下線で示す。

表 3-7 ファイル割り当て処理中にエラーが発生した場合の後処理方法（デフォルト状態または DD_DEALLOC_ON_ALLOCERR パラメータに OPT1 を指定している場合の動作）

対象ファイル	エラーが発生した時点でそのジョブステップで割り当て済みの DD 要素の場合	エラーが発生した時点ではそのジョブステップで未割り当ての DD 要素の場合
一時ファイル	DISP 属性の第 2 パラメータが PASS の場合はジョブ終了時に削除し、PASS 以外の場合はジョブステップ終了時に削除する。	そのジョブステップ開始前の状態のままとする。
恒久ファイル	DISP 属性の第 1 パラメータが NEW の場合 ジョブステップ終了時に削除する。 DISP 属性の第 1 パラメータが NEW 以外の場合 <u>DISP 属性の第 2 パラメータの指定に従って処理する。</u>	
ディレクトリ	恒久ファイルの処理と同様。*	
世代ファイル	GDG_REGISTER_JOBRC_THRESHOLD 非適用 恒久ファイルの処理と同様（世代管理情報を更新しない）。 GDG_REGISTER_JOBRC_THRESHOLD 適用 (1)DISP 属性の第 2 パラメータが KEEP の場合 ジョブ終了処理のしきい値判定処理に従う。 (2)DISP 属性の第 2 パラメータが DELETE の場合 <u>恒久ファイルの処理と同様（世代管理情報を更新しない）。</u>	

注※ DISP 属性値に"DELETE"が指定されている場合、ディレクトリ内にファイルやディレクトリが存在していると削除されないことがある。

表 3-8 ファイル割り当て処理中にエラーが発生した場合の後処理方法（DD_DEALLOC_ON_ALLOCERR パラメータに OPT2 を指定している場合の動作）

対象ファイル	エラーが発生した時点でそのジョブステップで割り当て済みの DD 要素の場合	エラーが発生した時点ではそのジョブステップで未割り当ての DD 要素の場合
一時ファイル	DISP 属性の第 2 パラメータが PASS の場合はジョブ終了時に削除し、PASS 以外の場合はジョブステップ終了時に削除する。	そのジョブステップ開始前の状態のままとする。
恒久ファイル	DISP 属性の第 1 パラメータが NEW の場合 ジョブステップ終了時に削除する。 DISP 属性の第 1 パラメータが NEW 以外の場合 <u>KEEP を仮定しファイルを保存する（DISP 属性の第 1 パラメータが RNW の場合は、0 バイトのファイルを保存する）。</u>	
ディレクトリ	恒久ファイルの処理と同様。*	
世代ファイル	GDG_REGISTER_JOBRC_THRESHOLD 非適用 恒久ファイルの処理と同様（世代管理情報を更新しない）。	

対象ファイル	エラーが発生した時点でそのジョブステップで割り当て済みの DD 要素の場合	エラーが発生した時点ではそのジョブステップで未割り当ての DD 要素の場合
世代ファイル	GDG_REGISTER_JOBRC_THRESHOLD 適用 (1)DISP 属性の第 2 パラメータが KEEP の場合 ジョブ終了処理のしきい値判定処理に従う。 (2)DISP 属性の第 2 パラメータが DELETE の場合 (a)DISP 属性の第 1 パラメータが NEW の場合 <u>ジョブステップ終了時に削除する。</u> (b)DISP 属性の第 1 パラメータが NEW 以外の場合 • <u>ジョブ実行中に作成された世代ファイルである場合</u> <u>ジョブステップ終了時に削除し、世代管理情報は更新しない。</u> • <u>上記以外の場合</u> 最新世代のファイルをそのまま保存し、世代管理情報は更新しない。	そのジョブステップ開始前の状態のままとする。

注※ DISP 属性値に"DELETE"が指定されている場合、ディレクトリ内にファイルやディレクトリが存在していると削除されないことがある。

(3) ファイルの状態と後処理の指定省略

ファイルの状態と後処理を指定する DD 要素の DISP 属性または DISP 属性内のパラメータの一部を省略した場合、システムは次に示す処理をする。

(a) ファイルの状態のパラメータを省略した場合

NEW パラメータを仮定した処理をする。

(b) ファイルの後処理の第 2 パラメータを省略した場合

ファイルの状態によって、後処理を次に示すように仮定する。

- SHR, OLD, MOD パラメータ指定の場合
KEEP パラメータを仮定した後処理をする。
一時ファイルおよび一時 ISAM ファイルの場合は、PASS パラメータを仮定した後処理をする。
- NEW, RNW パラメータ指定の場合
DELETE パラメータを仮定した後処理をする。

(c) ファイルの後処理の第 3 パラメータを省略した場合

第 2 パラメータと同じ値を仮定する。

(d) DISP 属性を省略した場合

TYPE 属性値が FILE, TEMP, TEMPISAM, GDG および DIR の場合, DISP="NEW,DELETE,DELETE"が仮定され, TYPE 属性値が LIB, PREST, および TEMPPREST の場合, DISP="SHR,KEEP,KEEP"が仮定される。

3.4.3 ファイルの連結処理

COBOL2002 で作成したユーザアプリケーションおよび SORT EE と連携して, 複数のファイルを論理的に連結した 1 つのファイルとして扱うことができる。

NAME 属性が同一である DD 要素を連続して記述することで, 該当する一連のファイルを連結する指定になる。この場合, バッチジョブ実行システムは, 連結対象である一連の DD 要素の DSN 属性に指定されたパス情報を環境変数として設定してから, ジョブステップのプログラムを実行する。環境変数に設定されたパス情報に該当するファイルは, COBOL アプリケーションまたは SORT EE によって, 自動的に論理的な 1 つのファイルとして解釈されて処理される。連結されたファイル 1 つにつき, 最大 255 の DD 要素を連続して記述できる。

ファイルの連結処理のためにバッチジョブ実行システムが設定する環境変数の名称は, 先頭の DD 要素に RENAME 属性が指定されていなければ, 先頭に"DDN_"または"CBL_"を付加した DD 要素名となる。先頭の DD 要素に RENAME 属性が指定されていれば, その RENAME 属性の値となる。また, この環境変数の値には, 先頭の DD 要素に指定された名称の環境変数に, DD 要素を記述した順序で, 割り当てた各ファイルのパス名を区切り文字コロンの ":"で連結した文字列が設定される。例を次に示す。

- ジョブ定義 XML ファイルの指定例 (COBOL アプリケーションと連携する場合)

```
<STEP NAME="MYSTEP">
  <EXEC PGM="MYUAP" LANG="COBOL" />
  <DD NAME="MYDATA" TYPE="FILE" DSN="/bjexuser/BJEX/IN/input1" DISP="SHR" />
  <DD NAME="MYDATA" TYPE="FILE" DSN="/bjexuser/BJEX/IN/input2" DISP="SHR" />
</STEP>
```

- 上記の指定に対してバッチジョブ実行システムが設定する環境変数の内容

```
CBL_MYDATA=/bjexuser/BJEX/IN/input1:/bjexuser/BJEX/IN/input2
```

この例の場合, COBOL アプリケーションは, input1 および input2 ファイルをまとめて, 論理的な 1 つのファイル「MYDATA」と解釈する。

- ジョブ定義 XML ファイルの指定例 (SORT EE と連携する場合)

```
<STEP NAME="STEP01">
  <EXEC PGM="*">
  <![CDATA[!
exec 1)&2
/opt/HIEXSORT/bin/exsort -p ${DDN_SYSIN}
!]]>
  </EXEC>
  <DD NAME="SYSIN" TYPE="DATA">
```

```

<![CDATA[!
-function sort
:
-inpfile
file=%DDN_INFILE%
:
!]]>
</DD>
<DD NAME="INFILE" TYPE="FILE" DSN="/bjexuser/BJEX/IN/inpsamf1" DISP="SHR" />
<DD NAME="INFILE" TYPE="FILE" DSN="/bjexuser/BJEX/IN/inpsamf2" DISP="SHR" />
<DD NAME="OUTFILE" TYPE="FILE" DSN="/bjexuser/BJEX/OUT/outsamf" DISP="RNW,KEEP" />
</STEP>

```

- 上記の指定に対してバッチジョブ実行システムが設定する環境変数の内容

```
DDN_INFILE="/bjexuser/BJEX/IN/inpsamf1:/bjexuser/BJEX/IN/inpsamf2"
```

この例の場合、SORT EE の-inpfile パラメータの file オペランドで指定した%DDN_INFILE%には、例で示した、2つのファイルパスをコロンで区切って連結した文字列が設定される。SORT EE は、それらファイルを論理的に連結された1つのファイルと解釈して入力データとする。

NAME 属性が同じ DD 要素が STEP 要素内で連続していない場合は、エラーとなる。

ファイルの連結は、COBOL アプリケーションおよび SORT EE から利用できる。また、JOB LIB, STEPLIB DD 要素に対して使用した場合、複数のパス情報をコロンで連結して PATH 環境変数に設定できる。

連結したファイルを COBOL アプリケーションから使用方法の詳細については、COBOL2002 のドキュメントもあわせて参照のこと。

また、JOB LIB, STEPLIB DD 要素の詳細については、「[3.3 プログラムのシステムへの登録と使用](#)」を参照のこと。

DD 要素の TYPE 属性によって、ファイル連結可能であるかどうかが決められている。TYPE 属性の値によるファイル連結可否の一覧を次の表に示す。ファイル連結できない TYPE 属性を指定した DD 要素を連結した場合は、警告メッセージを出力してジョブを続行する。

表 3-9 TYPE 属性の値によるファイル連結可否の一覧

TYPE 属性	ファイル種別	ファイル連結可否
DATA	SYSIN データ	○
DUMMY	ダミーファイル	○
FILE	恒久ファイル	○
TEMP	一時ファイル	○
TEMPISAM	一時 ISAM ファイル	×
GDG	世代ファイル	○

TYPE 属性	ファイル種別	ファイル連結可否
LIB	プログラム検索ディレクトリ	○
SYSOUT	SYSOUT データファイル	×
DIR	ディレクトリ	×
PREST	ジョブ間 PREST ファイル	×
TEMPPREST	ジョブステップ間 PREST ファイル	×

(凡例)

- ：ファイル連結できる。
- ×：ファイル連結できない。

また、連結できるファイルの形式は COBOL2002 や SORT EE によって決まっているため、各プログラムのドキュメントを参照して、連結できるファイルの形式を確認する必要がある。

注意事項

- ダミーファイルをファイルの連結処理の途中に指定した場合、VOS3 ではその指定をファイルの連結指定の終了と見なし、後続のファイルを連結しない。しかし、バッチジョブ実行システムでは、その指定をファイルの連結指定の終了と見なさないで、後続のファイルを連結する。VOS3 から移行したジョブ定義 XML ファイルに上記の指定があり、VOS3 と同じ動作としたい場合は、ダミーファイルより後ろの DD 要素を削除すること。
- ファイルの連結処理によって連結したファイルは、COBOL プログラムまたは SORT EE の出力先として指定できない。指定した場合、連結したファイルであっても先頭のファイルにだけ出力される。

指定例

(例 1) 正しい指定方法の例

```
<STEP NAME="MYSTEP">
  <EXEC PGM="MYPGM" LANG="COBOL" />
  <DD NAME="MYDATA" TYPE="FILE" DSN="/bjexuser/BJEX/INPUT/input1" DISP="SHR" />
  <DD NAME="MYDATA" TYPE="FILE" DSN="/bjexuser/BJEX/INPUT/input2" DISP="SHR" />
  <DD NAME="MYDATA" TYPE="FILE" DSN="/bjexuser/BJEX/INPUT/input3" DISP="SHR" />
  <DD NAME="MYDATA" TYPE="FILE" DSN="/bjexuser/BJEX/INPUT/input4" DISP="SHR" />
</STEP>
```

この指定例では、input1、input2、input3、および input4 の 4 ファイルが、論理的に結合された単一のファイルとして、ファイル名「MYDATA」で COBOL プログラムからアクセスできる。

(例 2) 誤った指定方法の例

```
<STEP NAME="MYSTEP">
  <EXEC PGM="MYPGM" LANG="COBOL" />
  <DD NAME="MYDATA" TYPE="FILE" DSN="/bjexuser/BJEX/INPUT/input1" DISP="SHR" />
  <DD NAME="MYDATA" TYPE="FILE" DSN="/bjexuser/BJEX/INPUT/input2" DISP="SHR" />
  <DD NAME="MYDATA" TYPE="FILE" DSN="/bjexuser/BJEX/INPUT/input3" DISP="SHR" />
  <DD NAME="NEWDATA" TYPE="FILE" DSN="/bjexuser/BJEX/INPUT/inputx" DISP="SHR" />
```

```
<DD NAME="MYDATA" TYPE="FILE" DSN="/bjexuser/BJEX/INPUT/input4" DISP="SHR" />
</STEP>
```

この指定例では、NAME 属性値「MYDATA」を持つ DD 要素のうち、input3 と input4 との間に、別の NAME 属性値を持つ DD 要素が記述されている。そのため、NAME 属性値「MYDATA」を持つ DD 要素の二重定義として扱われ、ジョブ定義 XML ファイルの構文解析エラーとなる。

3.4.4 一時ファイル

一時ファイルとはジョブ内で作成され、削除される暫定的なファイルである。ほかのジョブとの間で共有されることはない。このため、システム内で唯一になるように、システムによってユニークな名称が付けられる。なお、この項で説明する一時ファイルは、DD 要素の TYPE 属性で TEMP を指定した場合のファイルである。

(1) 一時ファイルの指定方法

一時ファイルを作成する場合、DD 要素の TYPE 属性で TEMP を指定する。DSN 属性の指定は任意である。また、一時ファイルは DISP 属性値に関係なくジョブ終了時に削除される。

- DD 名が SYSUT2 のファイルを、一時ファイルとして使用する例

```
<DD NAME="SYSUT2" TYPE="TEMP" DISP="NEW,DELETE" />
```

(2) 一時ファイルを作成するディレクトリ

一時ファイルを作成するディレクトリは、設定ファイルの TEMP_FILE_DIR_XML_USE パラメータの指定によって、次のとおりとなる。

(a) TEMP_FILE_DIR_XML_USE パラメータに YES を指定した場合

ジョブ定義 XML ファイルの DD 要素の DSN 属性値に、絶対パス形式でディレクトリ名を指定していた場合、指定値の最後の"/"までを一時ファイル作成ディレクトリとする。また、DSN 属性値の最後の"/"より後ろを一時ファイル識別名とする。

- ジョブ定義 XML ファイルの DD 要素の指定例

```
<DD NAME="DD1" TYPE="TEMP" DSN="/home/user/tmp/worktmp" DISP="NEW" />
```

- 作成する一時ファイル作成ディレクトリと一時ファイル名

```
/home/user/tmp/TEMP_ジョブ識別子_ジョブ名_worktmp_ユニークな名称
```

なお、DSN 属性に相対パスを指定した場合および DSN 属性の指定がない場合は、設定ファイルの TEMP_FILE_DIR パラメータで指定したディレクトリに作成する。

(b) TEMP_FILE_DIR_XML_USE パラメータを省略または NO を指定した場合

ジョブ定義 XML ファイルの DD 要素の DSN 属性で指定したディレクトリ名は無視され、設定ファイルの TEMP_FILE_DIR パラメータで指定したディレクトリに作成する。

(3) 一時ファイルを利用する場合の注意事項

- ファイル名は、システム内で唯一になるようにユニークな名称が付けられる。DD 要素での DSN 属性の指定の有無によって、次に示すファイル名が作成される。

- DSN 属性で一時ファイル識別名を指定した場合

```
TEMP_ジョブ識別子_ジョブ名_一時ファイル識別名_ユニークな名称
```

- DSN 属性を指定しない場合

```
TEMP_ジョブ識別子_ジョブ名_ユニークな名称
```

- 設定ファイルの TEMP_FILE_DIR_XML_USE パラメータで YES を指定した場合、DSN 属性値はディレクトリ名として扱うため、ディレクトリ名として指定できない文字や、存在しないディレクトリ名を指定しないこと。指定したときは、ジョブステップ実行開始時のファイル割り当て処理でエラーとなる。
- DSN 属性値には、"/"以外のファイル名として指定できない文字は含めないこと。"/"は記述できるが、最後の"/"より後ろの文字列を一時ファイル識別名と見なす。最後の"/"より後ろの文字列にファイル名として使用できない文字が含まれている場合、ジョブステップ実行開始時のファイル割り当て処理でエラーとなる。
- 生成されるファイル名の長さが、プラットフォームで決められた最大長を超えないこと。超えた場合、ジョブ定義 XML ファイルの解析時にはエラーとならないで、ジョブステップ実行開始時のファイル割り当て処理でエラーとなる。生成されるファイル名の長さは、DSN 属性値の長さ、JOB 要素の NAME 属性値の長さによって変わる。
- DD 要素の DISP 属性値を指定しない場合、一時ファイルを定義したジョブステップで新たにファイルを作成し、削除する (DISP 属性の第 1 パラメータは NEW、第 2・第 3 パラメータは DELETE) と見なされる。
- 同じ一時ファイルを複数のジョブステップで使用する場合は、DISP 属性の第 2 パラメータに PASS を指定する。
- DD 要素の DISP 属性の第 1 パラメータで RNW を指定した場合、NEW が仮定される。
- DD 要素の DISP 属性の第 1 パラメータで NEW または RNW を指定した場合、同一ファイル名を指定したときは、ユニークな名称を付けて別ファイルを作成する。
- DD 要素の DISP 属性の第 1 パラメータで OLD または SHR を指定して割り当てできる一時ファイルは、先行ジョブステップで DD 要素の DISP 属性の第 2 パラメータに PASS を指定した一時ファイルだけである。
- DD 要素の DISP 属性の第 1 パラメータで MOD を指定した場合、先行ジョブステップで DD 要素の DISP 属性の第 2 パラメータに PASS を指定した一時ファイルがあればその一時ファイルを割り当てて、一時ファイルがない場合は設定ファイルの DISPMOD_NOFILE パラメータの指定に従う。

- DD 要素の DISP 属性の第 2 パラメータで KEEP を指定した場合、PASS が仮定される。
- DISP 属性の第 2 パラメータに PASS を指定した場合のジョブステップ間での一時ファイルの受け渡しについては、「3.4.2(2)(c) PASS パラメータ」の説明および例も参照すること。
- DD 要素の DISP 属性の第 3 パラメータで KEEP を指定した場合、DELETE が仮定される。

3.4.5 一時 ISAM ファイル

一時 ISAM ファイルとは、索引順編成ファイル管理 ISAM による索引順編成の一時ファイルである。キーを持つデータを処理する場合に、データの間接結果を一時的に格納するために利用する。

一時 ISAM ファイルはジョブ内で作成され、削除される暫定的なファイルである。ほかのジョブとの間で共用されることはない。このため、システム内で唯一になるように、システムによってユニークな名称が付けられる。

ジョブコントローラで作成する一時 ISAM ファイルは、主キーによるアクセスだけができる。

DD 要素の TYPE 属性に TEMP を指定した場合に作成される一時ファイルと同様に、ジョブ内のジョブステップ間で同一の一時 ISAM ファイル名を使用する場合、DISP 属性に PASS パラメータを指定することによって、後続ジョブステップへその一時 ISAM ファイルを受け渡すことができる。

一時 ISAM ファイルを使用する場合、設定ファイルの ISAMLIB_MODE パラメータに使用する ISAM ライブラリを指定する。

(1) 一時 ISAM ファイルの指定方法

一時 ISAM ファイルを使用する場合、DD 要素の TYPE 属性で TEMPISAM を指定する。DSN 属性の指定は任意である。

一時 ISAM ファイルの作成時には一時 ISAM ファイルの属性を定義するために、次に示す属性を指定する必要がある。

- ISKEY
キー長、キー位置、およびキー属性
- ISRECFM
レコード種別（順編成固定長または可変長）
- ISRECL
レコード長（レコード種別が可変長の場合は、最大レコード長および最小レコード長）

なお、ISKEY、ISRECFM、および ISRECL 属性の指定は DISP 属性の値に NEW パラメータを指定した場合に有効である。DISP 属性の値が SHR または OLD パラメータの場合に指定したときは、ISKEY、ISRECFM、および ISRECL 属性の指定は無効となる。ただし、属性値のチェックは行われる。

DISP 属性の値に MOD パラメータを指定した場合は、先行ジョブステップで DD 要素の DISP 属性の第 2 パラメータに PASS を指定した一時 ISAM ファイルがあれば、ISKEY、ISRECFM、および ISRECL 属性の指定が有効となる。一時 ISAM ファイルがなければ、ISKEY、ISRECFM、および ISRECL 属性の指定は無効となる。ただし、属性値のチェックは行われる。

- DD 名が SYSUT2 のファイルを、一時 ISAM ファイルとして使用する例

```
<DD NAME="SYSUT2" TYPE="TEMPISAM" DSN="WKDS" DISP="NEW,DELETE" ISRECFM="F" ISKEY="4,0,C" ISRECL="80" />
```

(2) 一時 ISAM ファイルを作成するディレクトリ

一時 ISAM ファイルを作成するディレクトリは、設定ファイルの TEMP_FILE_DIR_XML_USE パラメータの指定によって、次のとおりとなる。

(a) TEMP_FILE_DIR_XML_USE パラメータに YES を指定した場合

ジョブ定義 XML ファイルの DD 要素の DSN 属性値に、絶対パス形式でディレクトリ名を指定していた場合、指定値の最後の "/" までを一時 ISAM ファイル作成ディレクトリとする。また DSN 属性値の最後の "/" より後ろを一時 ISAM ファイル識別名とする。

- ジョブ定義 XML ファイルの DD 要素の指定例

```
<DD NAME="DD1" TYPE="TEMPISAM" DSN="/home/user/tmp/workisam" DISP="NEW" ISRECFM="F" ISKEY="4,0,C" ISRECL="80" />
```

- 作成する一時 ISAM ファイル作成ディレクトリと一時 ISAM ファイル名

```
/home/user/tmp/ISAM_ジョブ識別子_ジョブ名_workisam_ユニークな名称
```

なお、DSN 属性に相対パスを指定した場合および DSN 属性の指定がない場合は、設定ファイルの TEMP_FILE_DIR パラメータで指定したディレクトリに作成する。

(b) TEMP_FILE_DIR_XML_USE パラメータを省略または NO を指定した場合

ジョブ定義 XML ファイルの DD 要素の DSN 属性で指定したディレクトリ名は無視され、設定ファイルの TEMP_FILE_DIR パラメータで指定したディレクトリに作成する。

(3) 一時 ISAM ファイルを構成するファイル

ジョブコントローラで作成する一時 ISAM ファイルは、次に示すファイルから構成される。

- キー定義ファイル (拡張子".DEF")
- 主キーファイル (拡張子".K01")
- データファイル (拡張子".DAT")

(4) 一時 ISAM ファイルのファイル名

一時 ISAM ファイルのファイル名称は、システム内で唯一になるようにユニークな名称が付けられる。DD 要素での DSN 属性の指定の有無によって、次に示すファイル名称が生成される。

- DSN 属性で一時 ISAM ファイル識別子の指定があった場合

```
ISAM_ジョブ識別子_ジョブ名_一時ISAMファイル識別子_ユニークな名称
```

- DSN 属性の指定がなかった場合

```
ISAM_ジョブ識別子_ジョブ名_ユニークな名称
```

一時 ISAM ファイルを構成するファイル名は、上記の生成したファイル名称に拡張子が付加される。一時 ISAM ファイルを構成するファイル名を次の表に示す。

表 3-10 一時 ISAM ファイルを構成するファイル名

構成ファイルの種類	構成ファイル名
キー定義ファイル	生成したファイル名称.DEF
主キーファイル	生成したファイル名称.K01
データファイル	生成したファイル名称.DAT

なお、ジョブコントローラが設定する環境変数 ["CBL_"+DD 要素名], ["DDN_"+DD 要素名] に格納されるパス名には、キー定義ファイル、主キーファイル、およびデータファイルの名称は設定されないで、拡張子を除いたファイル名が 1 つだけ設定される。

また、DSN 属性を指定する場合は次に示す注意が必要である。

- 設定ファイルの TEMP_FILE_DIR_XML_USE パラメータで YES を指定した場合、DSN 属性値はディレクトリ名として扱うため、ディレクトリ名として指定できない文字や、存在しないディレクトリ名を指定しないこと。指定したときは、ジョブステップ実行開始時のファイル割り当て処理でエラーとなる。
- DSN 属性値には、"/"以外のファイル名として指定できない文字は含めないこと。"/"は記述できるが、最後の"/"より後ろの文字列を一時 ISAM ファイル識別子と見なす。最後の"/"より後ろの文字列にファイル名として使用できない文字が含まれている場合、ジョブステップ実行開始時のファイル割り当て処理でエラーとなる。
- 生成されるファイル名の長さが、プラットフォームで決められた最大長を超えないこと。超えた場合、ジョブ定義 XML ファイルの解析時にはエラーとならないで、ジョブステップ実行開始時のファイル割り当て処理でエラーとなる。生成されるファイル名の長さは、DSN 属性値の長さ、JOB 要素の NAME 属性値の長さによって変わる。なお、生成されるファイル名の長さには拡張子の文字列 (".DEF",".K01",".DAT") の長さ 4 文字分も考慮すること。

(5) 作成する一時 ISAM ファイルの定義内容

ジョブコントローラで作成する一時 ISAM ファイルでは、ISKEY、ISRECFM、および ISRECL 属性で定義する情報以外は、次に示す表の内容で定義される。

表 3-11 一時 ISAM ファイルの定義内容

一時 ISAM ファイルの定義情報	内容
障害発生時のデータファイル内容保証	保証しない。
主キーのキー順序	昇順となる。
主キーの重複キー許可	許可しない。
主キーの重複キー順序保証	保証する。
主キーの圧縮レベル	圧縮なし。
主キーのスペース文字許可	許可しない。

(6) 一時 ISAM ファイルを利用する場合の注意事項

- DD 要素の DISP 属性値を指定しない場合、一時 ISAM ファイルを定義したジョブステップで新たにファイルを作成し、削除する (DISP 属性の第 1 パラメータは NEW、第 2、第 3 パラメータは DELETE) と見なされる。複数のジョブステップで使用する場合は、DISP 属性の第 2 パラメータに PASS を指定する。
- DD 要素の DISP 属性の第 2 パラメータで KEEP を指定した場合、PASS が仮定される。
- DD 要素の DISP 属性の第 3 パラメータで KEEP を指定した場合、DELETE が仮定される。
- DD 要素の DISP 属性の第 1 パラメータで RNW を指定した場合、NEW が仮定される。
- DD 要素の DISP 属性の第 1 パラメータで NEW または RNW を指定した場合、同一ファイル名を指定したときは、ユニークな名称を付けて別ファイルを作成する。
- DD 要素の DISP 属性の第 1 パラメータで OLD または SHR を指定して割り当てできる一時 ISAM ファイルは、先行ジョブステップで DD 要素の DISP 属性の第 2 パラメータに PASS を指定した一時 ISAM ファイルだけである。
- DD 要素の DISP 属性の第 1 パラメータで MOD を指定した場合、先行ジョブステップで DD 要素の DISP 属性の第 2 パラメータに PASS を指定した一時 ISAM ファイルがあればその一時 ISAM ファイルを割り当てが、一時 ISAM ファイルがない場合は設定ファイルの DISPMOD_NOFILE パラメータの指定に従う。
- 一時 ISAM ファイルに対してファイルの連結指定 (NAME 属性が同一である DD 要素を連続して記述) をしないこと。
- 一時 ISAM ファイルに対する COBOL プログラムの「OPEN OUTPUT」文の動作は、DD 要素の DISP 属性の指定内容に関係なく、追加書ききとなる。レコードが格納されている一時 ISAM ファイルを再使用 (レコードがない状態) したい場合は、次に示すどれかの対処をする必要がある。

- COBOL プログラムで全レコードを削除する処理を追加する。
- COBOL プログラムの実行時に次に示す環境変数のどちらかを指定する。
 - (i)環境変数 CBLD_ファイル名に ISAMDL を設定する。環境変数「CBLD_ファイル名」の「ファイル名」は、COBOL プログラムの SELECT 句に指定したファイル名である。
 - (ii)環境変数 CBLISAMDL に YES を指定する。
 なお、上記のどちらかの環境変数を指定した場合、「OPEN OUTPUT」文の動作時に一時 ISAM ファイルが再作成（一時 ISAM ファイルを削除してから作成）される。
- バッチジョブ実行システムでは、ノーマルファイル形式の ISAM ファイルを扱い、ラージファイル形式の ISAM ファイルは作成できない。COBOL2002 のユーザアプリケーションに対して渡せる一時 ISAM ファイルもノーマルファイル形式となる。

3.4.6 ダミーファイル

ダミーファイルとは、不要なデータの出力先とするファイルである。出力したデータは破棄される。また、ダミーファイルは、入力データがないファイルである。入力すると、実際のファイルの終わり（EOF）が報告される。

(1) ダミーファイルの主な用途

ダミーファイルは、次に示す用途で利用する。

- プログラムが出力するデータが、不要である場合、DD 要素にダミーファイルを指定する。
- プロシジャ中の DD 要素を毎回呼び出すごとに変更するようなプロシジャで、デフォルト値としてその DD 要素にダミーファイルを指定する。
- プロシジャ中の DD 要素を無効にしたい場合、プロシジャ内の DD 要素の修正機能でダミーファイルに修正する。

(2) ダミーファイルの指定方法

ダミーファイルを作成する場合、DD 要素の TYPE 属性値に DUMMY を指定する。

(3) ダミーファイル入出力要求

ダミーファイルのオープン、クローズは一般のファイルと同様に行われるが、入出力要求は次に示すようになる。

(a) 設定ファイルで DUMMY_FILE_TYPE パラメータに DEVNULL を指定している場合

ダミーファイルとして、/dev/null を割り当てる。

- 入力要求

ダミーファイルへの入力要求があると、システムは直ちにファイルの終わりを報告する。

- **出力要求**

ファイルへの出力要求を受け付けるが、出力データを破棄する。

(b) 設定ファイルで DUMMY_FILE_TYPE パラメータに TEMP を指定している場合

ダミーファイルとして、一時ファイルを割り当てる。

- **入力要求**

入力要求の前にデータを出力していなければ、ダミーファイルへの入力要求に、システムは直ちにファイルの終わりを報告する。

入力要求の前にデータを出力していれば、ダミーファイルへの入力要求に、システムは入力要求の前に出力したデータを渡すことがある。

- **出力要求**

ファイルへの出力要求を受け付け、実際に出力する。

(4) 注意事項

- ダミーファイルを COBOL プログラムまたは SORT EE に入力する場合、索引順ファイルおよび可変長順ファイルは指定できない。指定した場合は COBOL プログラムおよび SORT EE でエラーとなる。
- ダミーファイルに対して COBOL プログラムまたは SORT EE からデータを出力する場合、索引順ファイルは指定できない。指定した場合は COBOL プログラムおよび SORT EE でエラーとなる。

3.4.7 プログラムへのファイル割り当て機能

(1) DD 要素の書き方

ジョブステップで実行するプログラムが利用するファイルを、あとから自由に変更できるように、ファイル名をジョブ定義 XML ファイルで指定できる。

ジョブ定義 XML ファイルでは通常のファイルの場合は、次に示すように指定する。

```
<DD NAME="DD名" TYPE="FILE" DSN="ファイルのパス名"/>
```

DD 要素には、TYPE 属性値に通常ファイルであることを示す"FILE"を指定し、DSN 属性値にファイルのパス名を指定する。

同じ STEP 要素内で実行する COBOL2002 で作成したプログラムからは、NAME 属性に対応した名称を指定することで、このファイルのパス名をアクセスできる。これによって、プログラムを修正することなく、ジョブ定義 XML ファイルを変更するだけで、プログラムからアクセスするファイルを変更して割り当てることができる。

(a) DSN 属性の指定方法

DSN 属性の指定値および設定ファイルのパラメータの指定によってアクセスするパス名の扱いを次の表に示す。

表 3-12 DSN 属性の指定値および設定ファイルのパラメータの指定によってアクセスするパス名の扱い

DSN 属性の値	パス名
/[パス名/]ファイル名	絶対パスとして指定した絶対パスのファイルにアクセスする。
./[パス名/]ファイル名	プログラムが実行しているカレントディレクトリ（ワーキングディレクトリ）からの相対パスのファイルにアクセスする。
[パス名/]ファイル名	設定ファイルの PERMANENT_FILE_DIR で指定された特定のディレクトリからの相対パスのファイルにアクセスする。
\$(BJEX_DIR_任意名) [パス名/]ファイル名	設定ファイルの BJEX_DIR_任意名で指定されたパス名置換識別子を相対パスとしたファイルにアクセスする。

(b) PERMANENT_FILE_DIR の指定方法

DSN 属性に指定された相対パスは、設定ファイルの PERMANENT_FILE_DIR に指定されたパスからの相対パスとなる。

- 設定ファイルの指定

```
PERMANENT_FILE_DIR /home/test/
```

- DD 要素の指定

```
<DD NAME="DD1" TYPE="FILE" DSN="dir1/file1" />
```

- 設定される環境変数

```
DDN_DD1=/home/test/dir1/file1
```

(c) BJEX_DIR_任意名の指定方法

設定ファイルに識別子とディレクトリを定義しておき、DSN 属性に \${ } で囲まれた識別子 BJEX_DIR_xxxx を指定した場合、識別子 BJEX_DIR_xxxx に指定されたパスからの相対パスとなる。

- 設定ファイルの指定

```
BJEX_DIR_xxxx /tmp/
```

- DD 要素の指定

```
<DD NAME="DD1" TYPE="FILE" DSN="${BJEX_DIR_xxxx}dir1/file1" />
```

- 設定される環境変数

```
DDN_DD1=/tmp/dir1/file1
```

(2) DD 要素をプログラムから使用する方法

ジョブ実行時にプログラムが入出力を行うファイルは、ジョブ定義 XML ファイルの DD 要素と対応づけることができる。定義されたファイルのパス名は環境変数に設定されるため、プログラムから環境変数を参照してファイルを使用できる。

ジョブコントローラが設定する環境変数名は、DD 要素の NAME 属性に指定された NAME 値に文字列を付加した名称とする。付加する文字列は、EXEC 要素に指定するプログラム言語によって異なる。環境変数の値を次の表に示す。

表 3-13 EXEC 要素に指定するプログラム言語による環境変数

項目	プログラム言語	
	COBOL	指定なし
EXEC 要素の指定	<EXEC PGM="UAPI" LANG="COBOL" />	<EXEC PGM="UAPI" />
環境変数名	先頭に"CBL_"を付加した DD 要素名 ("CBL_" + DD 要素名) 例 <DD NAME="ABC" /> 環境変数名: CBL_ABC	先頭に"DDN_"を付加した DD 要素名 ("DDN_" + DD 要素名) 例 <DD NAME="ABC" /> 環境変数名: DDN_ABC
環境変数の値	DSN 属性値	DSN 属性値

DD 要素の記述例を次に示す (恒久ファイル用ディレクトリなしの場合)。

```
<EXEC PGM="UAPI" />  
<DD NAME="DD1" DSN="/home/file1" />
```

上記例の場合に設定される環境変数を次に示す。この場合、環境変数 DDN_DD1 を参照して、/home/file1 のファイルを使用できる。

```
DDN_DD1=/home/file1
```

(3) 環境変数に設定される内容

環境変数に設定される内容を次の表に示す。

表 3-14 環境変数に設定される内容

TYPE 属性値	ファイル種別	環境変数に設定される内容
FILE	通常のファイル	DSN 属性で指定されたファイルのファイル名
DIR	ディレクトリ	DSN 属性で指定されたディレクトリのディレクトリ名

TYPE 属性値	ファイル種別	環境変数に設定される内容
GDG	世代ファイル	相対世代番号に対応するファイルのファイル名
TEMP	一時ファイル	ジョブコントローラプロセスが作成した一時ファイルのファイル名
TEMPISAM	一時 ISAM ファイル	ジョブコントローラプロセスが作成した一時ファイルのファイル名
DATA	SYSIN データ	ジョブコントローラプロセスが作成した一時ファイルのファイル名
DUMMY	ダミーファイル	<ul style="list-style-type: none"> • DUMMY_FILE_TYPE パラメータに TEMP を指定 ジョブコントローラプロセスが作成した一時ファイルのファイル名 • DUMMY_FILE_TYPE パラメータに DEVNULL を指定 /dev/null
SYSOUT	SYSOUT データファイル	作成したスプールファイルのファイル名
LIB	プログラム検索ディレクトリ	<ul style="list-style-type: none"> • NAME 属性で指定された環境変数 DSN 属性で指定されたディレクトリのディレクトリ名を、DD 要素の出現順に":"(コロン)で区切った文字列である。 • PATH 環境変数 NAME 属性で指定された環境変数の文字列を、PATH 環境変数でのプログラム検索パスの並びの先頭に付加する。
PREST	ジョブ間 PREST ファイル	ジョブコントローラプロセスが作成したジョブ間 PREST ファイル名
TEMPPREST	ジョブステップ間 PREST ファイル	ジョブコントローラプロセスが作成したジョブステップ間 PREST ファイル名

(4) 環境変数名を任意の名称に設定する方法

ジョブコントローラが設定する環境変数名を、先頭に"DDN_"または"CBL_"を付加しない任意の名称とする場合、DD 要素に RENAME 属性を指定する。

COBOL2002 の特殊な環境変数など、"DDN_"または"CBL_"で始まらない環境変数名を使用する場合は、RENAME 属性を指定する。

RENAME 属性を用いて、任意の環境変数名を指定する場合の記述例を次に示す。

```
<EXEC PGM="UAP1" />
<DD NAME="DD1" DSN="/home/file1" RENAME="ANYNAME" />
```

上記例の場合に設定される環境変数を次に示す。この場合、/home/file1 のファイルを使用するための環境変数名は、"DDN_DD1"ではなく"ANYNAME"になる。

```
ANYNAME=/home/file1
```

RENAME 属性を使用する場合の注意事項を次に示す。

- 同一の STEP 要素内では、RENAME 属性の値が同じ DD 要素を複数指定しないこと。指定した場合、環境変数には後ろに指定した DD 要素のファイルパスが設定される。

- ファイル連結の場合、2 番目以降の DD 要素に指定した RENAME 属性は無効になる。RENAME 属性で環境変数名を変更する場合は、先頭の DD 要素に RENAME 属性を指定すること。
- RENAME 属性で指定した値を名称とする環境変数は、ジョブステップ終了時にジョブコントローラが削除する。したがって、RENAME 属性に次に示す値を指定しないこと。
 - 「3.21 環境変数」で示したバッチジョブ実行システムが使用する環境変数
 - ログインスクリプトで定義した環境変数など、ジョブコントローラ起動時にすでに設定されている環境変数
 - ジョブ定義 XML ファイルの SETENV 要素で指定した環境変数
 - ジョブステップで実行するユーザプログラムやシェルスクリプトの内部で使用する環境変数
- JOBLIB DD 要素および STEPLIB DD 要素には、RENAME 属性を指定しないこと。指定した場合、ジョブ定義 XML ファイルの解析時にエラーとなる。
- RENAME 属性を使用した場合、その DD 要素で指定した FTYPE 属性、RECFM 属性、および TRUNC 属性は無効になる。また、DISP 属性の第 1 パラメータに"MOD"（追加書き）を指定していても、ユーザプログラムからその DD 要素のファイルに対する書き込み方法はユーザプログラムの処理に依存する。

(5) ファイルの割り当て時に付与されるアクセス権限

バッチジョブ実行システムによって実行されるジョブの中でファイルが割り当てられる場合、次の原則でファイルへのアクセス権限が設定される。

(a) DD 要素で割り当てるファイルとディレクトリ

ファイルのアクセス権限は、次のどれかに指定された権限値を持つファイルが割り当てられる。

- ジョブ定義 XML ファイル内の DD 要素の FPERM 属性または PERMISSION 要素
- 設定ファイルの PERMISSION_DIR パラメータおよび PERMISSION_FILE パラメータ

これらの情報を設定していない場合、バッチジョブ実行システムはファイルに対して権限 0600 を、ディレクトリに対して権限 0700 を仮定してファイルまたはディレクトリを割り当てる。

DD 要素によってファイルまたはディレクトリが新規に割り当てられる場合、ユーザ環境の umask 設定は無効であり、ジョブ定義 XML ファイルまたは設定ファイルで指定された権限値がそのままアクセス権限として適用される。

(b) EXEC 要素の PGM 属性で指定したユーザアプリケーションが作成したファイルとディレクトリ

ユーザ環境の umask に基づいてファイルが作成される。

(c) CDATA 内に記述したシェルスクリプト内で作成したファイルとディレクトリ

ユーザ環境の umask に基づいてファイルが作成される。

3.4.8 逆方向参照

ジョブ定義 XML には逆方向参照機能が備わっている。これを利用すると、ジョブ内の先行 DD 要素にある情報を取得するようシステムに指示できる。

逆方向参照は先行 DD 要素の TYPE 属性が、FILE、DATA、または TEMP であることが必要である。なお、JSPEC 機能適用ジョブステップの場合、先行 DD 要素の TYPE 属性が FILE だけ参照できる。

逆方向参照する場合、参照先の TYPE 属性を参照元に引き継ぐ。このため、参照する側の TYPE 属性の指定は不要であり、指定しても無視される。ただし、参照する側の TYPE 属性のシンタクスチェックは行われる。

なお、参照元の TYPE 属性で PREST を指定した場合、逆方向参照は行われなくて、指定されたファイル名を基に PREST ファイルとして使用される。

TYPE 属性と逆方向参照可否についての関係を次の表に示す。

表 3-15 TYPE 属性と逆方向参照可否についての関係

参照される側の TYPE 属性	FILE	DIR	GDG	TEMP	DATA	DUM MY	SYSO UT	LIB	TEMP ISAM	PREST	TEMP PREST
逆方向参照可否	○	×	×	○*	○*	×	×	×	×	×	×

(凡例)

○：参照先のファイル名を参照できる。

×：参照先のファイル名を参照できない。エラーとなる。

注※ JSPEC 時は参照できない。

一時ファイルを逆方向参照する場合、参照元の DISPS 属性値によって、正しく参照できないときがある。

参照元の DISP 属性の第 1 パラメータで "NEW"、または "RNW" を指定した場合には、逆方向参照とならないで新たにファイルを割り当てる。

参照元の DISP 属性指定値と実際に参照されるファイルの関係を次の表に示す。

表 3-16 参照元の DISP 属性の第 1 パラメータ指定値と参照されるファイルの関係

DISP 属性指定値	NEW	RNW	OLD	SHR	MOD
参照されるファイル	新規に作成したファイル		参照先のファイル		

逆方向参照は DD 要素の DSN 属性に指定する方法が次に示す 3 とおりある。

- <DD NAME="DD 名" DSN="*.参照する DD 名" />
同一ジョブステップ中で定義されている DD 要素を参照する場合、この形式を使用する。
- <DD NAME="DD 名" DSN="*.ステップ名.参照する DD 名" />

ステップ名で指定された先行ジョブステップ中で定義されている DD 要素を参照する場合、この形式を使用する。

プロシジャ内でこの形式を指定した場合、ジョブコントローラは次の順序で DD 要素を検索する。

AIX の場合

- ・設定ファイルの EXEC_COND_STEPSEARCH パラメータの値に COMPATIV3 を指定した場合
該当するプロシジャ内で DD 要素を参照するジョブステップ名を検索する。見つからない場合は、ジョブ定義 XML ファイルの先頭から DD 要素を参照するプロシジャ外のジョブステップ名を検索する。
- ・設定ファイルの EXEC_COND_STEPSEARCH パラメータの値に BJEX を指定した場合
ジョブ定義 XML ファイルの先頭から DD 要素を参照するプロシジャ外のジョブステップ名を検索する。

Linux の場合

該当するプロシジャ内で DD 要素を参照するジョブステップ名を検索し、見つからない場合はジョブ定義 XML ファイルの先頭から、DD 要素を参照するプロシジャ外のジョブステップ名を検索する。

・ <DD NAME="DD 名" DSN="*.ステップ名.プロシジャステップ名.参照する DD 名" />

ステップ名で指定された先行ジョブステップで呼び出されるカタログドプロシジャ、または入力ストリームプロシジャのプロシジャステップ中で定義された DD 要素を参照する場合、この形式を使用する。

使用例

同一ジョブステップおよび先行するジョブステップで指定された DSN 属性値を、逆方向参照で複写する例を次に示す。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE" >
    <STEP NAME="STEP1">
      <EXEC PGM="UAP1" />
      <DD NAME="DD1" TYPE="FILE" DSN="/home/file1" DISP="NEW, KEEP, KEEP" />
      <DD NAME="DD2" TYPE="FILE" DSN="*. DD1" DISP="OLD, KEEP, KEEP" />*1
    </STEP>
    <STEP NAME="STEP2">
      <EXEC PGM="UAP2" />
      <DD NAME="DD3" TYPE="FILE" DSN="*. STEP1. DD1" DISP="OLD, KEEP, KEEP" />*2
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

注※1 逆方向参照を使用したDD要素を示す（環境変数DDN_DD2に同一ジョブステップのDD1に指定されたファイルを割り当てる）。

注※2 逆方向参照を使用したDD要素を示す（環境変数DDN_DD3にSTEP1のDD1に指定されたファイルを割り当てる）。

注意事項

- ・「DSN="*.参照する DD 名"」によって同一ジョブステップ内の DD 要素を検索する場合は、必ず逆方向参照を行う DD 要素より前に定義されている DD 要素を指定すること。

- 参照先の DD 要素を持つジョブステップが、COND 属性や IF 要素により実行されなかった場合、参照したファイルが作成されていないことが考えられる。このようなとき、参照元のジョブステップ時実行時にエラーとなることがある。例を次に示す。



注※ ジョブ定義XMLファイルでのHOLD属性値でNOを指定、またはbjexchjobコマンドの-Rオプションの指定で印刷指示することを示す。

- 一時ファイルを後続ステップから逆方向参照する場合、ステップ終了時に一時ファイルが削除されないよう、参照先の DD 要素では、DISP=(,PASS)を指定する必要がある。
- TYPE=DATA 指定のファイルは、一時ファイルのためジョブステップ終了時に削除しているが、後続 DD 要素で逆方向参照されている場合には、ジョブ終了時に削除される。逆方向参照されていないものについては、従来どおりジョブステップ終了時に削除される。

3.5 ジョブステップ

ジョブステップを実行する場合、EXEC 要素に記述したプログラムが実行される。プログラムの実行には、プログラムコマンドを実行する方法と、シェルスクリプト内でコマンドを実行する方法がある。

3.5.1 プログラムコマンドを実行

EXEC 要素内の PGM 属性値にコマンド名を記述し、渡される引数を PARM 属性値に指定することで、シェルを経由しないで実行する。

- 指定方法

```
<EXEC PGM="program名" PARM="引数1△引数2" />
```

- 実行するコマンド

```
program名△引数1△引数2
```

(凡例)

△：空白を示す。PARM 属性値の空白の扱いを説明するために必要な空白だけを「△」と明示的に表現する。

3.5.2 シェルスクリプトを実行

EXEC 要素内の PGM 属性値に"*"の指定と EXEC 要素の内容にシェルの指定をした場合、シェルスクリプトとして実行する。内容の形式は「[3.10.2 SYSIN データとなる文字列](#)」と同じ指定方法を利用して先頭の"!"から終端の"!"にシェルを記述する。

シェルスクリプトとして実行するため、SHELL 環境変数には、ksh (AIX の場合) または bash (Linux の場合) を設定しておく必要がある。

- 指定方法

```
<EXEC PGM="*" >  
<![CDATA[  
!  
program名△引数1△引数2  
!  
]]>  
</EXEC>
```

- 実行するコマンド

```
program名△引数1△引数2
```

(凡例)

△：空白を示す。PARM 属性値の空白の扱いを説明するために必要な空白だけを「△」と明示的に表現する。

3.5.3 PARM 属性値が空白を含む場合

PARM 属性の値が空白を含む場合、その EXEC 要素の PARMSPLIT 属性の指定によって、次に示すように引数が決定する。

(1) EXEC 要素に PARMSPLIT = "YES" の指定がある場合

UNIX での一般的な引数インタフェースである C 言語インタフェースに従った形式で、空白で分割した複数の引数としてプログラムに渡される。

- 指定方法

```
<EXEC PGM="program" PARM="AAA△BBB△△△CCC△△" PARMSPLIT="YES" />
```

(凡例)

△：空白を示す。PARM 属性値の空白の扱いを説明するために必要な空白だけを「△」と明示的に表現する。

- 実行するコマンド

```
実行プログラム : program  
引数 1 : AAA  
引数 2 : BBB  
引数 3 : CCC
```

(2) EXEC 要素に PARMSPLIT="NO" の指定がある場合

VOS3 から移行した COBOL プログラムなどの VOS3 インタフェースに従った形式で、空白を含めて全体を 1 つの引数としてプログラムに渡される。

- 指定方法

```
<EXEC PGM="program" PARM="AAA△BBB△△△CCC△△" PARMSPLIT="NO" />
```

- 実行するコマンド

```
実行プログラム : program  
引数 1 : AAA△BBB△△△CCC△△
```

(凡例)

△：空白を示す。PARM 属性値の空白の扱いを説明するために必要な空白だけを「△」と明示的に表現する。

(3) PARMSPPLIT 属性を省略した場合

設定ファイルの EXEC_PARMSPPLIT パラメータの指定に従う。EXEC_PARMSPPLIT パラメータを省略した場合は、C 言語インタフェースに従った形式で、空白で分割した複数の引数としてプログラムに渡される。

- 指定方法

```
<EXEC PGM="program" PARM="AAA△BBB△△△CCC△△" />
```

- 実行するコマンド

EXEC_PARMSPPLITパラメータのPARM属性値の空白分割指定="YES"の場合

実行プログラム : program

引数 1 : AAA

引数 2 : BBB

引数 3 : CCC

EXEC_PARMSPPLITパラメータのPARM属性値の空白分割指定="NO"の場合

実行プログラム : program

引数 1 : AAA△BBB△△△CCC△△

EXEC_PARMSPPLITパラメータを省略した場合

実行プログラム : program

引数 1 : AAA

引数 2 : BBB

引数 3 : CCC

(凡例)

△ : 空白を示す。PARM 属性値の空白の扱いを説明するために必要な空白だけを「△」と明示的に表現する。

3.5.4 ジョブステップの正常終了／異常終了

ジョブステップの実行結果と、設定ファイルの PGM_RCABEND パラメータの指定によって、ジョブステップが正常終了するか、異常終了するかが決定する。

ジョブステップが正常終了する場合、ジョブステップ完了メッセージとして KAKC0097-I が出力される。ジョブステップが異常終了する場合、ジョブステップ完了メッセージとして KAKC0096-E が出力される。

ジョブステップが正常終了するか、異常終了するかの条件を次の表に示す。

表 3-17 ジョブステップの正常終了／異常終了の条件

ジョブステップの実行結果	PGM_RCABEND の指定値	正常終了／異常終了
ユーザプログラムがシグナルを受信しないで終了	ユーザプログラムの終了コードと一致しない	ジョブステップ正常終了 (RC=ユーザプログラムの終了コード)
	ユーザプログラムの終了コードと一致する	ジョブステップ異常終了 (RC=ユーザプログラムの終了コード)

ジョブステップの実行結果	PGM_RCABEND の指定値	正常終了／異常終了
ユーザプログラムがシグナルを受信して異常終了 (RC=128+シグナル番号)	—	ジョブステップ異常終了 (RC=128+シグナル番号)
ジョブステップ実行時にジョブコントローラでエラーが発生	—	ジョブステップ異常終了 (RC=バッチジョブ実行システムのエラー終了コード)

(凡例)

—：なし。

ジョブステップ実行時に、ジョブコントローラでエラーが発生した場合を除くジョブステップ異常終了の場合、ファイルの後処理は DISP 属性の第 3 パラメータに従う。また、後続ジョブステップは、COND 属性に EVEN または ONLY が指定されているステップだけ実行する。

JSPEC 機能を使用している場合、ジョブステップが異常終了であれば JSPECAN 機能によるジョブステップ打ち切りが行われる。詳細については、「[3.12.1\(3\)\(b\) JOBPARM_JSPECAN パラメータを指定する方法 \(JSPECAN 機能\)](#)」を参照のこと。

3.5.5 注意事項

- PARM 属性値に\${変数名}と指定しても環境変数を参照できない。環境変数を引数とする場合は、シェルスクリプトとして実行すること。
- ジョブステッププロセスの umask は、ジョブコントローラ起動ユーザの umask が設定される。

3.6 ジョブステップの条件付き実行 (COND 属性)

ジョブは、幾つかのジョブステップから構成されている。ジョブ内のジョブステップは順次実行されていくが、先行するジョブステップの実行結果によって、後続するジョブステップを実行しないで、そのジョブを終了させたり、または特定のジョブステップの実行へ移したりすることが、JOB 要素または EXEC 要素の COND 属性を指定することで実現できる。これをジョブステップの条件付き実行という。条件付き実行のための条件は、各ジョブステップの実行結果としてコマンドが返す終了コードを基本とする。

3.6.1 条件付き実行の指定方法

(1) JOB 要素

JOB 要素の COND 属性の指定方法を次に示す。

```
[COND="([条件コード, 演算符号[ ]) [, (条件コード, 演算符号)]+"]
```

ジョブステップの実行結果によって、後続のジョブステップの実行をしないで、そのジョブを終了させたい場合は、JOB 要素に COND 属性で終了コードの判定条件を指定する。判定条件は 1~8 個指定できるが、それ以上指定した場合、シンタクスエラーとなりジョブは異常終了する。

終了コードの判定は、終了コード値と COND 属性で指定した 0~255 のしきい値と演算符号を用いる。

演算符号を次の表に示す。

表 3-18 演算符号

演算符号	意味
EQ (Equal to)	しきい値は終了コードと等しい。
NE (Not Equal to)	しきい値は終了コードと等しくない。
GT (Greater Than)	しきい値は終了コードより大きい。
GE (Greater than or Equal to)	しきい値は終了コードより大きいか等しい。
LT (Less Than)	しきい値は終了コードより小さい。
LE (Less than or Equal to)	しきい値は終了コードより小さいか等しい。

(2) EXEC 要素

EXEC 要素の COND 属性の指定方法を次に示す。

```
[COND="([しきい値, 演算符号 [, ステップ名[, プロシジャステップ名]] [ ])
```

```
[, (しきい値, 演算符号 [, ステップ名[, プロシジャステップ名]])]+  
[, {EVEN|ONLY}] ”]
```

- 先行するジョブステップの実行結果によって、後続するジョブステップの1個をスキップさせたい場合は、EXEC要素のCOND属性で終了コードの判定条件を指定する。判定条件は1~8個指定できるが、それ以上指定した場合、シンタクスエラーとなりジョブは異常終了する。
終了コードの判定は、終了コード値とCOND属性で指定した0~255のしきい値と演算符号を用いる。指定する演算符号の内容は、JOB要素のCOND属性の演算符号と同じである。演算符号については、「表3-18 演算符号」を参照のこと。
- 通常、先行するジョブステップが異常終了すると、後続するジョブステップはスキップされるが、該当するジョブステップを実行させたい場合、EXEC要素のCOND属性でEVENまたはONLYを指定する。

(3) 注意事項

- 実行するコマンドによっては終了コードが設定されないコマンドがある。
- COND属性の指定方法を次に示す。
 - 条件が1つの場合は、括弧で囲まないで、間に空白を開けないで指定する。
指定例を次に示す。

```
COND="10, EQ"
```

- 条件が複数の場合は、それぞれを一对の括弧で囲み、コンマで区切る。間に空白を開けないで指定する。
指定例を次に示す。

```
COND="(10, EQ), (20, EQ)"
```

3.6.2 条件付き実行

(1) JOB要素でCOND属性を指定した場合

ジョブ内の各ジョブステップの終了時点で、各ジョブステップが生成する終了コードとJOB要素のCOND属性で指定した演算符号を基に論理演算をする。

- 論理演算結果が真の場合
後続のジョブステップを実行しないで、そのジョブは終了する。
- 論理演算結果が偽の場合
そのまま後続のジョブステップを実行する。

JOB要素のCOND属性で終了コードの判定条件を指定した場合の流れの例を次の図に示す。

図 3-7 JOB 要素の COND 属性で終了コードの判定条件を指定した場合の流れの例

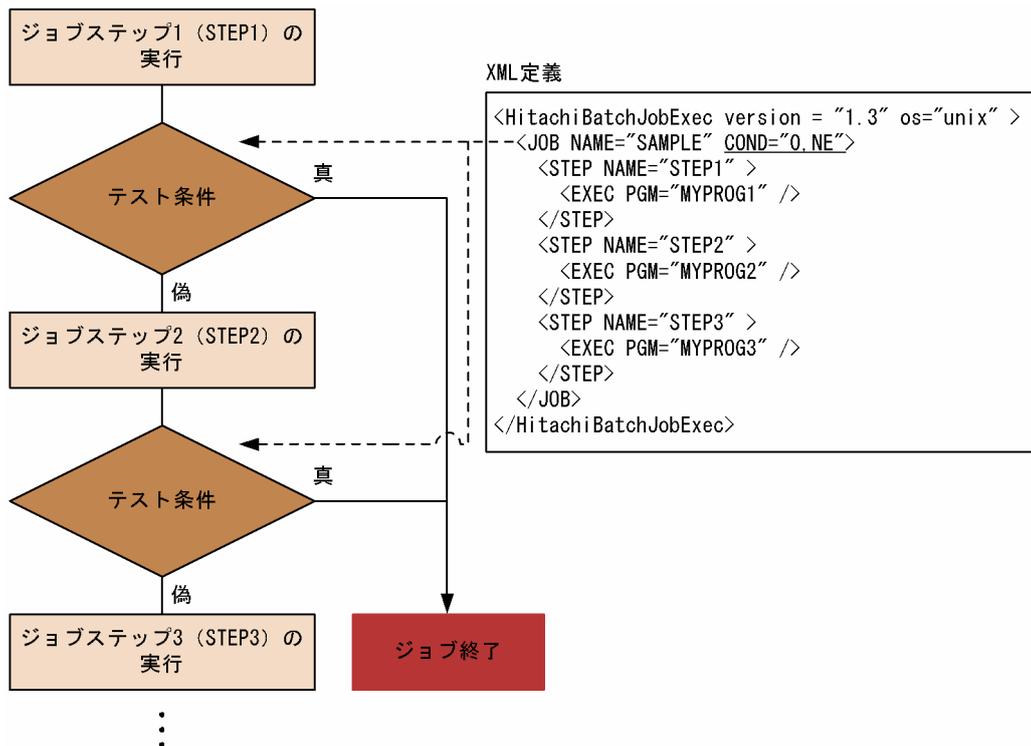


図 3-7 は、「ジョブ中のどれかのステップの終了コードが 0 以外である」という条件（下線部の「0,NE」指定）を満たした時点で、後続のジョブステップをすべて実行しないでジョブを終了する例である。

(2) EXEC 要素の COND 属性で終了コードの判定条件を指定した場合

EXEC 要素の COND 属性で指定したジョブステップの実行開始時点で、次に示す処理をする。

(a) 先行するジョブステップ名を指定しない場合

設定ファイルの EXEC_COND_NOSTEPNAME パラメータに STEPALL を指定するか、または EXEC_COND_NOSTEPNAME パラメータを指定していない場合、先行するジョブステップが生成するすべての終了コードと EXEC 要素の COND 属性で指定した演算符号を基に論理演算をする。また、設定ファイルの EXEC_COND_NOSTEPNAME パラメータに STEPLAST を指定した場合、最後に実行されたジョブステップが生成する終了コードだけと論理演算をする。

- 論理演算結果が真の場合
そのジョブステップの実行をスキップして、後続するジョブステップの処理へ移る。
- 論理演算結果が偽の場合
そのジョブステップを実行する。

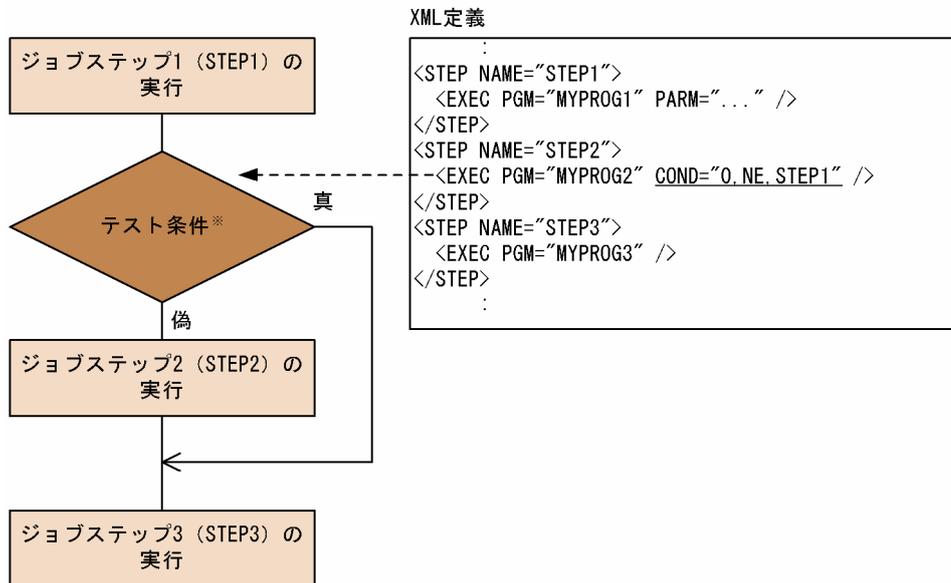
(b) 先行するジョブステップ名を指定した場合

指定した先行するジョブステップが生成する終了コードと EXEC 要素の COND 属性で指定した演算符号を基に論理演算をする。論理演算結果の後処理は、(a)と同じである。

また、指定した先行するジョブステップが条件付き実行によって実行がスキップされている場合は、EXEC要素のCOND属性は無視される（COND属性が指定されなかったものとして処理される）。ジョブ内で重複しているジョブステップ名を指定した場合、最初に一致したジョブステップに対して論理演算をする。

EXEC要素のCOND属性で終了コードの判定条件を指定した場合の流れの例を次の図に示す。

図 3-8 EXEC要素のCOND属性で終了コードの判定条件を指定した場合の流れの例



注※ ジョブステップ2 (STEP2) のEXEC要素のCOND属性で指定終了コードのテスト

図 3-8 は、STEP2 を実行する前に「STEP1 の終了コードが 0 以外」という条件（下線部の「0,NE,STEP1」指定）をチェックし、条件を満たした場合には STEP2 を実行しないで、その次のステップ STEP3 を実行する例である。

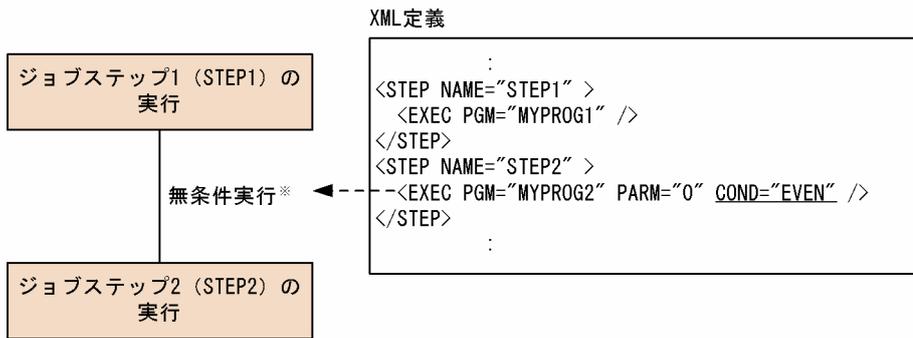
(3) EXEC要素のCOND属性でEVENまたはONLYを指定した場合

EXEC要素のCOND属性を指定したジョブステップの実行開始時点で、次に示す処理をする。

- EVENを指定した場合、無条件に該当するジョブステップを実行する。
- ONLYを指定した場合、1つ以上の先行するジョブステップのどれかが異常終了している場合に、該当するジョブステップを実行する。
- EVENまたはONLYがジョブステップの終了コードの判定条件とともに指定している場合、終了コードの判定条件を満たすと、EVENまたはONLYを無視し、そのジョブステップを実行しない。
- 先行するジョブステップが異常終了した場合、後続のジョブステップにEVENまたはONLYの指定がないかぎり、該当するジョブステップをスキップする。

EXEC要素のCOND属性でEVENを指定した場合の流れの例を次の図に示す。

図 3-9 EXEC 要素の COND 属性で EVEN を指定した場合の流れの例

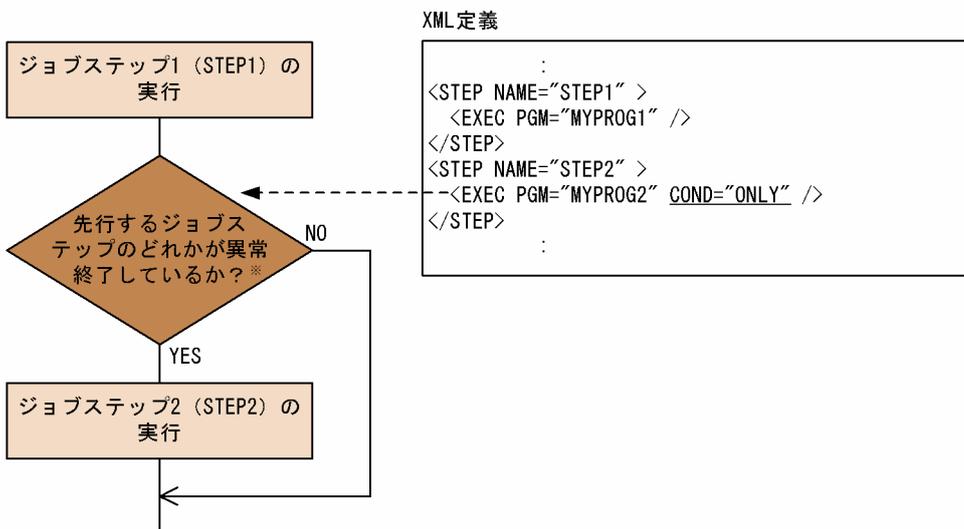


注※ ジョブステップ2 (STEP2) のEXEC要素のCOND属性でEVENを指定

図 3-9 は、仮に STEP1 のプログラム MYPROG1 が異常終了した場合でも、COND="EVEN"指定（下線部）によって STEP2 は必ず実行される例である。

EXEC 要素の COND 属性で ONLY を指定した場合の流れの例を次の図に示す。

図 3-10 EXEC 要素の COND 属性で ONLY を指定した場合の流れの例

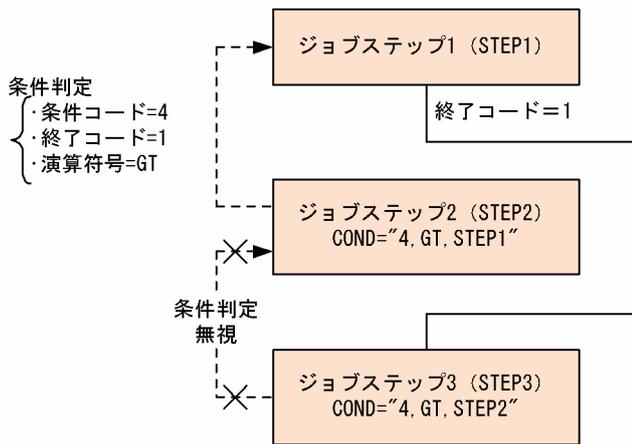


注※ ジョブステップ2 (STEP2) のEXEC要素のCOND属性でONLYを指定

図 3-10 は、STEP1 のプログラム MYPROG1 が異常終了した場合にだけ、COND="ONLY"指定（下線部）によって STEP2 が実行される例である。

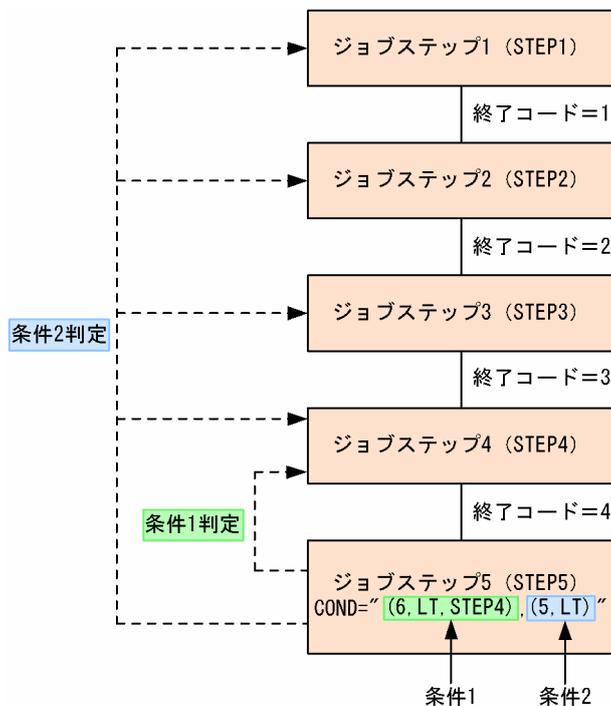
ジョブステップの条件付きの実行を次の図に示す。

図 3-11 スキップされたジョブステップの判定例



ジョブステップ2がジョブステップ1の終了コード判定でスキップされるケースがある。この場合、ジョブステップ3でジョブステップ2の終了コードの条件判定をしているが、ジョブステップ2をスキップしているとき、EXEC要素のCOND属性の指定は無視して、ジョブステップ3を実行する。

図 3-12 先行するすべてのジョブステップの終了コードとの判定例



条件1 (COND属性の(6,LT,STEP4))を判定したが、この条件を満たさないため条件2 (COND属性の(5,LT))の判定をする。条件2の場合、ジョブステップ名がないためすべての先行するジョブステップの終了コードとの判定をするが、この場合も条件を満たさないためジョブステップ5を実行する。

3.6.3 ジョブステップを条件付きで実行するときのステップ名の検索範囲

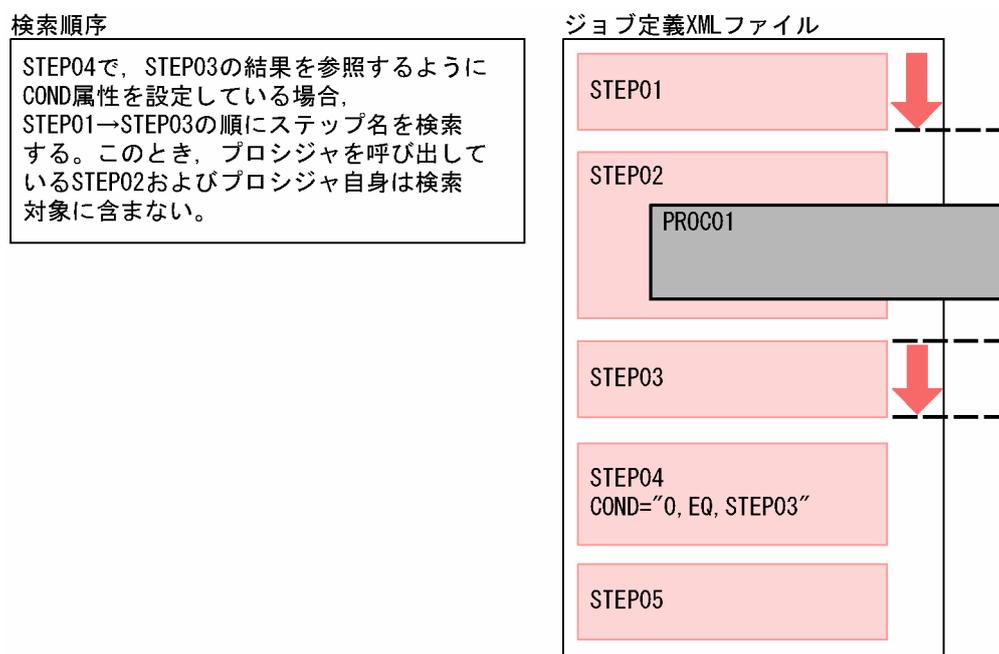
ジョブステップを条件付きで実行する場合は、COND 属性の指定内容によってステップ名の検索範囲が異なる。COND 属性の指定形式ごとに、ステップ名が検索される順序を次に示す。

(1) COND 属性がプロシジャ外に指定されている場合

(a) ステップ名だけ指定されている場合

バッチジョブ実行システムは、ジョブ定義 XML ファイルの先頭から COND 属性が指定されているジョブステップの直前のジョブステップまで、プロシジャを呼び出すジョブステップおよびプロシジャ内を除いた範囲でステップ名を検索する。検索例を次の図に示す。

図 3-13 ステップ名だけで検索する例 (COND 属性がプロシジャ外に指定されている場合)



(凡例)

-  : COND属性で指定されたステップ名の検索範囲を示す。
-  : ジョブステップを示す。
-  : カタログドプロシジャを示す。

(b) ステップ名とプロシジャステップ名が指定されている場合

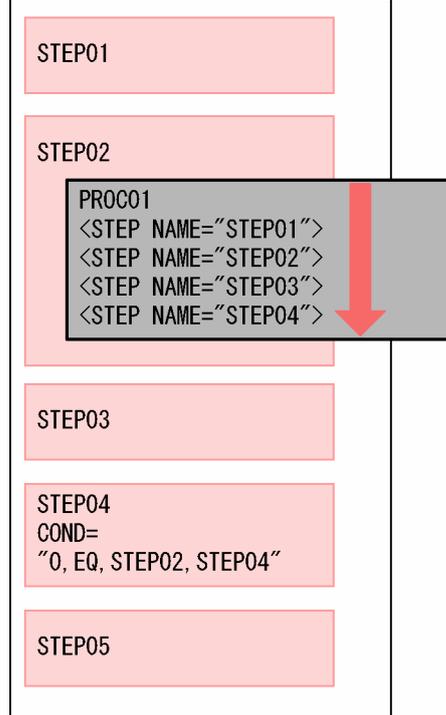
バッチジョブ実行システムは、指定されたステップ名が一致するプロシジャ内のプロシジャステップ名を先頭から末尾まで検索する。検索例を次の図に示す。

図 3-14 ステップ名とプロシジャステップ名で検索する例 (COND 属性がプロシジャ外に指定されている場合)

検索順序

STEP04で、STEP02で呼び出したプロシジャ内のプロシジャステップSTEP04の結果を参照するようにCOND属性を設定している場合、プロシジャステップ名STEP04をプロシジャ0C01の先頭から検索する。

ジョブ定義XMLファイル



(凡例)

-  : COND属性で指定されたステップ名の検索範囲を示す。
-  : ジョブステップを示す。
-  : カタログドプロシジャを示す。

(2) COND 属性がプロシジャ内に指定されている場合

(a) ステップ名だけ指定されている場合

バッチジョブ実行システムは、該当するプロシジャの先頭から、COND 属性が指定されているプロシジャステップの直前のプロシジャステップまでの範囲でプロシジャステップ名を検索する。この時点で一致するステップ名が見つかれば、その戻り値を COND 判定の対象とする。もしこの時点でステップ名が見つからない場合は、プロシジャ外のジョブ定義 XML ファイルの先頭から、該当するプロシジャを呼び出したステップの直前まで検索する。なお、このとき検索途中に別のプロシジャを呼び出す処理があっても、それらプロシジャ内は検索範囲には含まない。検索例を次の図に示す。

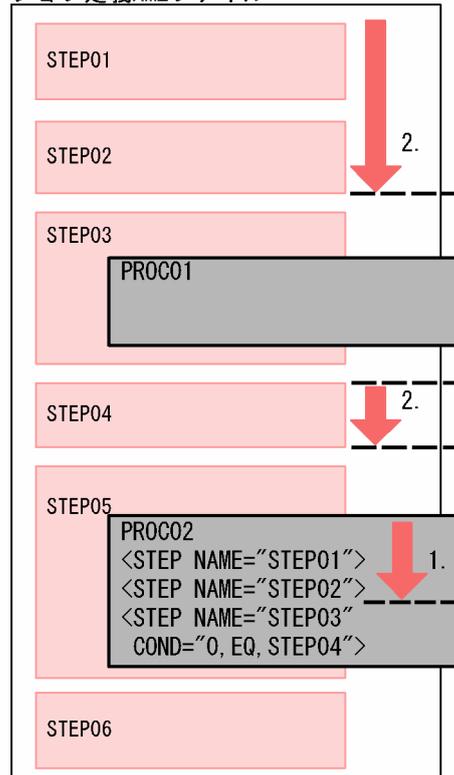
図 3-15 ステップ名だけで検索する例 (COND 属性がプロシジャ内に指定されている場合)

検索順序

プロシジャPROC02内で、ステップ名 (STEP04) だけ記述したCOND属性を指定した場合、次の順にステップ名を検索する。

1. はじめに同プロシジャの先頭から、COND属性が指定されたステップの直前のステップの範囲まで、STEP04を検索する。
2. 1でステップ名が見つからなかった場合、ジョブ定義XMLファイルの先頭から、プロシジャを呼び出したステップの直前のステップ (STEP04) の範囲まで、ステップ名を検索する。このとき、途中に現れるプロシジャ呼び出し (STEP03) は検索範囲には含めない。

ジョブ定義XMLファイル



(凡例)

- : COND属性で指定されたステップ名の検索範囲を示す。
- : ジョブステップを示す。
- : カタログドプロシジャを示す。

【AIX 限定】

「図 3-15 ステップ名だけで検索する例 (COND 属性がプロシジャ内に指定されている場合)」は、設定ファイルの EXEC_COND_STEPSEARCH パラメータの値に COMPATIV3 を指定した場合の動作である。値に BJEX を指定した場合、またはパラメータを省略した場合は、「(1)(a) ステップ名だけ指定されている場合」で示した検索範囲となる。

(b) ステップ名とプロシジャステップ名が指定されている場合

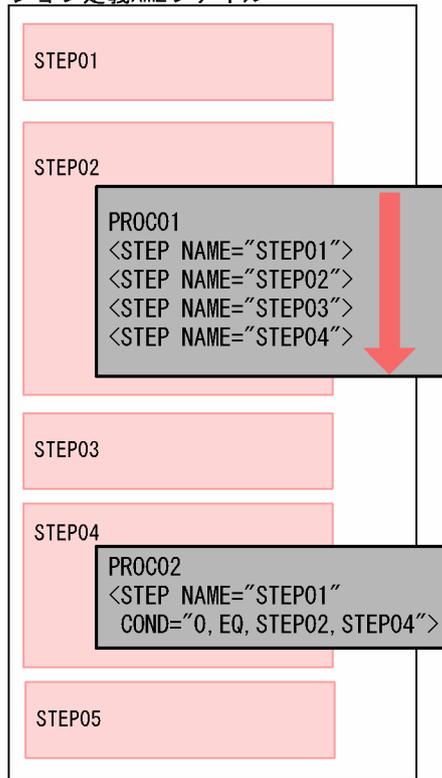
バッチジョブ実行システムは、COND 属性で指定したステップから呼び出されているプロシジャに限定して、その先頭からプロシジャステップ名を検索する。検索例を次の図に示す。

図 3-16 ステップ名とプロシジャステップ名で検索する例 (COND 属性がプロシジャ内に指定されている場合)

検索順序

STEP04で呼び出したプロシジャPROC02で、STEP02で呼び出したプロシジャ内のプロシジャステップSTEP04の結果を参照するようにCOND属性を設定している場合、プロシジャステップ名STEP04をプロシジャPROC01の先頭から検索する。

ジョブ定義XMLファイル



(凡例)

-  : COND属性で指定されたステップ名の検索範囲を示す。
-  : ジョブステップを示す。
-  : カタログドプロシジャを示す。

3.7 ジョブステップの実行制御 (IF 要素)

ジョブ定義 XML ファイルの EXEC 要素の COND 属性を使用することで、ジョブステップを条件付きで実行できる。しかし、この機能はそのジョブステップの実行条件を記述するだけであるため、複数ステップの実行をスキップしたい場合は、それらのすべてのジョブステップに COND 属性を記述する必要がある。

IF 要素、ELSE 要素および ENDIF 要素を使用することでジョブステップのリターンコードを判定して実行するジョブステップの範囲を制御できる。ただし、COND 属性によるジョブステップの条件付き実行ではジョブに対して制御できるが、IF 要素によるジョブステップの実行制御では、ジョブに対しては制御できない。

なお、IF 要素、ELSE 要素または ENDIF 要素と COND 属性の機能は背反ではなく、併用できる。

ジョブステップの実行制御 (IF 要素) とジョブステップの条件付き実行 (COND 属性) はジョブ実行時に次の順序で処理される。

1. IF 要素の条件判定によるジョブステップの実行抑止
2. EXEC 要素の COND 条件判定
3. ジョブステップ実行
4. JOB 要素の COND 条件判定

3.7.1 ジョブステップの実行制御の各要素の使用規則

IF 要素は ELSE 要素と ENDIF 要素を組み合わせで使用する。

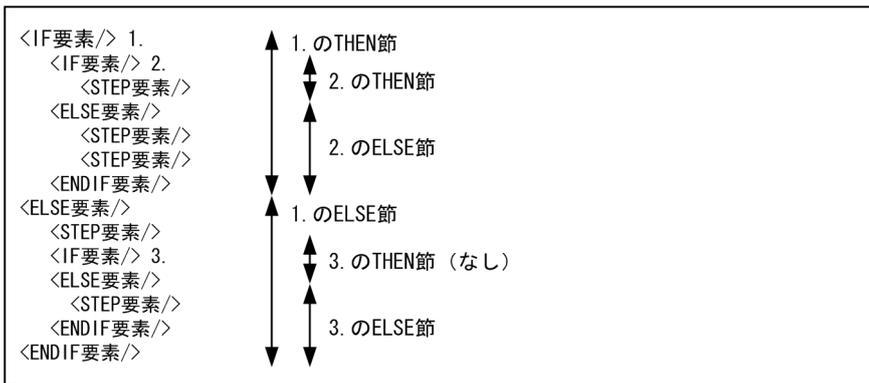
IF 要素から ELSE 要素までを THEN 節と呼ぶ。また、ELSE 要素から ENDIF 要素までを ELSE 節と呼ぶ。

ELSE 要素は省略できる。ELSE 要素を省略した場合、ELSE 節はなく、IF 要素から ENDIF 要素までが THEN 節となる。

IF 要素の OP 属性で指定した条件が真の場合 THEN 節のステップ群が実行され、偽の場合 ELSE 節のステップ群が実行される。

IF 要素、ELSE 要素または ENDIF 要素は階層構造で指定できる。また、STEP 要素を含まない THEN 節や ELSE 節があってもよい。IF 要素、ELSE 要素または ENDIF 要素の階層構造を次の図に示す。

図 3-17 IF 要素, ELSE 要素または ENDIF 要素の階層構造



JSPEC 機能を使用する場合、並列実行するジョブステップ内で使用できない。

IF 要素, ELSE 要素または ENDIF 要素で実行しないと判定されたジョブステップの EXEC 要素または COND 属性に EVEN や ONLY があっても処理の対象にならない。また、実行しないジョブステップの直後に指定した SETMAXRC 属性も処理されない。

3.7.2 ジョブステップの実行制御の使用例

ジョブステップの実行制御を使用した例を、次のジョブ定義 XML に示す。

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.4" os="unix" >
<JOB NAME="TT02_MNBM22" MSGCLASS="A" >
  <STEP NAME="MNBM2280" >
    <EXEC PGM="UAP001" COND="1,NE" />
    <DD NAME="SYSPPRINT" TYPE="SYSOUT" SYSOUT="A" />
  </STEP>
  <SETMAXRC VALUE="0" />
  <IF OP="(MNBM2280, LE, 8)"/>
    ==>ジョブステップMNBM2280のRCが9以上ならMNBM229Bから実行する。
    RCが8以下なら次を処理する。
  <IF OP="(MNBM2280, LE, 4)"/>
    =>ジョブステップMNBM2280のRCが4以下ならジョブステップMNBM2285を実行する。
    ジョブステップMNBM2280のRCが5から8ならジョブステップMNBM2290から実行する。
  <STEP NAME="MNBM2285">
    <EXEC PGM="UAP2" COND="1,NE" />
    <DD NAME="SORTIN" TYPE="FILE" DSN="/home/user01/t01" DISP="OLD" />
    <DD NAME="SORTOUT" TYPE="FILE" DSN="/home/user01/t01a" DISP="RNW,KEEP" />
  </STEP>
  <SETMAXRC VALUE="2"/>
</IF>
  <STEP NAME="MNBM2290" >
    <EXEC PGM="UAP3" COND="1,NE" />
    <DD NAME="SYSPPRINT" TYPE="SYSOUT" SYSOUT="A" />
  </STEP>
  <SETMAXRC VALUE="3"/>
</ENDIF>
</ENDIF>
  
```

```
<STEP NAME="MNBM229B" >
  <EXEC PGM="UAP3" COND="1,NE" />
  <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
</STEP>
<SETMAXRC VALUE="3"/>
<IF OP="( (MNBM2280, LE, 4) OR (NOT (MNBM2290, EQ, 5) ) ) AND (MNBM229B, AB, )"/>
  ==>ジョブステップMNBM2280のRCが4以下またはジョブステップMNBM2290のRCが5でない
  かつジョブステップMNBM229Bが異常終了ならジョブステップMNBM229Cを実行する。
  <STEP NAME="MNBM229C" >
    <EXEC PGM="UAP3"/>
    <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
  </STEP>
</ENDIF/>
</JOB>
</HitachiBatchJobExec>
```

3.8 ステップリスタート

ジョブが異常終了した場合、障害要因を取り除いてからジョブを再実行する場合、正常に終了したジョブステップを再度実行しないで、途中のジョブステップから実行を再開したいときがある。このとき、bjexec コマンドで再実行するジョブステップ名を指定することで、途中のジョブステップから実行を開始できる。この機能を、ステップリスタートと呼ぶ。

3.8.1 ステップリスタートを実行するための条件

ステップリスタートを実行するためには、次の条件を満たす必要がある。

- bjexec コマンドの-r オプションでリスタートを開始するジョブステップのステップ名を指定する。
- DD 要素の変更が必要な場合、DD 要素を変更する。
- リスタートするステップが JSPEC 機能で並列実行するステップでない。
- リスタートするステップが IF 要素の THEN 節や ELSE 節のステップでない。

ユーザはステップリスタート機能を適用したい場合、ジョブが途中で中断した場合に任意のステップから再実行しても問題がないようにジョブを作成しておく必要がある。

3.8.2 JP1/AJS - View でのステップリスタートの実行

ジョブが異常終了した場合に JP1/AJS - View で特定のジョブステップ (STEP02) から実行するときの例を次に示す。

(1) ジョブ定義 XML ファイルの例

次の例でジョブステップ (STEP01) は正常終了したが、STEP02 が異常終了し、実行できなかった。

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version="1.4" os="unix" >
  <JOB NAME="TESTPROC">
    <STEP NAME="STEP01" >
      <EXEC PGM="*" >
        <![CDATA[
!
exit 0
!
          ]]>
        </EXEC>
      </STEP>
      <STEP NAME="STEP02" >
        <EXEC PGM="*" >
          <![CDATA[
!
```

```

exit 1
!
    ]]>
    </EXEC>
</STEP>
<STEP NAME="STEP03" >
    <EXEC PGM="*" >
        <![CDATA[
!
exit 0
!
    ]]>
    </EXEC>
</STEP>
</JOB>
</HitachiBatchJobExec>

```

(2) JP1/AJS - View によるジョブの再実行

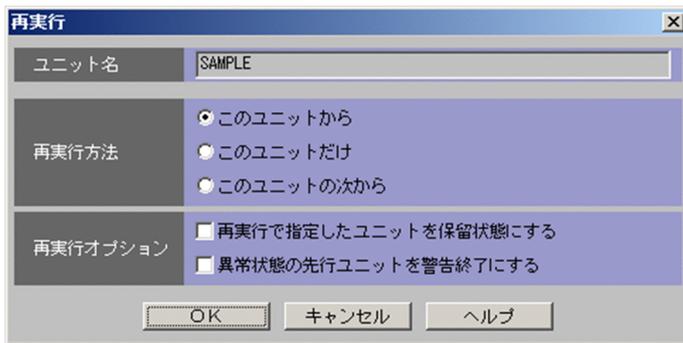
STEP02 からジョブを再開する。まず、JP1/AJS - View の詳細定義でジョブステップ (STEP02) を追加する。

図 3-18 JP1/AJS - View の詳細定義でのジョブステップの追加

詳細定義-[UNIX Job]	
ユニット名	SAMPLE
コメント	
実行エージェント	
定義 転送ファイル 属性	
コマンド文	
スクリプトファイル名	/opt/hitachi/bjex/bin/bjexec
パラメーター	STEP02 /home/user01/PROC_JCL_PRSX.xml
環境変数	AJS_BJEX_STOP=TERM
環境変数ファイル名	
ワークパス	
実行優先順位	なし
標準入力ファイル名	
標準出力ファイル名	<input type="checkbox"/> 追加書き
標準エラー出力ファイル名	<input type="checkbox"/> 追加書き
終了判定	判定結果: しきい値による判定
	警告しきい値: <input type="text"/> 異常しきい値: 0
実行時のユーザー	
OK キャンセル ヘルプ	

次に STEP02 以降のジョブステップを実行する。

図 3-19 JP1/AJS - View でのジョブの再実行



(3) 実行結果

STEP01 を実行しないで、STEP02 から実行した実行結果を次に示す。

```

***** BJEX MESSAGE *****
10:14:16 000047 KAKC0091-I TESTPROC Job started.
10:14:16 000047 KAKC0112-I Job is restarted from specified step. stepname=STEP02
10:14:16 000047 KAKC0092-I TESTPROC.STEP02 Step started.
10:14:16 000047 KAKC0097-I TESTPROC.STEP02 Step ended. rc=1 E-Time=0.246s C-Time=0.000s
10:14:16 000047 KAKC0092-I TESTPROC.STEP03 Step started.
10:14:17 000047 KAKC0097-I TESTPROC.STEP03 Step ended. rc=0 E-Time=0.255s C-Time=0.010s
10:14:17 000047 KAKC0098-I TESTPROC Job ended. rc=1 E-Time=0.501s C-Time=0.010s MaxRSS=2856K
B

***** XML IMAGE *****
000001 <?xml version="1.0" encoding="Shift-JIS" ?>
000002 <HitachiBatchJobExec version="1.4" os="unix" >
:
(中略)
:
000014 </EXEC>
000015 </STEP>
000016 </JOB>
000017 </HitachiBatchJobExec>

***** SYSTEM MESSAGE *****
10:14:16 000047 KAKC0900-I Parsing of xml file "/home/user01/PROC_JCL_PRSX.xml" started.
10:14:16 000047 KAKC0909-I Parsing of xml file "/home/user01/PROC_JCL_PRSX.xml" ended. code=
0
10:14:16 000047 KAKC0707-I MSGCLASS is "A".
10:14:16 000047 KAKC1600-I TESTPROC Job allocated.
10:14:16 000047 KAKC1601-I TESTPROC.STEP02 Step allocated.
10:14:16 000047 KAKC1601-I TESTPROC.STEP03 Step allocated.

***** JOBSTEP OUTPUT *****
KAKC0719-I STEP 002,STEP02,STDERR

KAKC0719-I STEP 003,STEP03,STDERR

KAKC0099-I HitachiBatchJobExec ended. rc=1

```

3.8.3 bjexec コマンドでのステップリスタートの実行

ステップリスタートを実行するためには、bjexec コマンドでリスタートを開始するジョブステップのステップ名を指定する。

```
bjexec -r リスタートステップ名 ジョブ定義XMLファイル名
```

- -r リスタートステップ名

リスタートするジョブステップ名を指定する。指定したジョブステップから実行を開始する。
リスタートステップ名は次の形式で指定する。

- ステップ名[.プロシジャステップ名]

ステップ名：リスタートするジョブステップ名を記述する。

プロシジャステップ名：リスタートするジョブステップがプロシジャの場合、ピリオド (.) に続けてプロシジャ内のステップ名を記述する。

指定したジョブステップが存在しないときは、ジョブを実行しないで終了する。

プロシジャを呼び出しているジョブステップは指定できない。プロシジャの先頭からリスタートする場合、プロシジャの先頭ジョブステップを指定すること。

「-r ""」などで空白だけの文字列や¥0 だけの空文字列を指定すると通常ジョブとして先頭から実行する。同一ステップ名のジョブステップが複数存在する場合、先頭から探して最初に見つかったジョブステップから実行する。ただし、JSPEC 機能で並列実行するジョブステップ、または IF 要素の THEN 節や ELSE 節内のジョブステップは検索の対象にならない。

3.8.4 ステップリスタートを実行する上でのファイルの扱い

ステップリスタートを実行するジョブでファイルを扱う場合、次の点に注意する必要がある。

- ステップリスタートを実行する可能性がある場合、ジョブで使用するファイルおよびディレクトリは、次のように指定する必要がある。
 - リスタートするジョブステップで新たに作成されるファイルおよびディレクトリでは、DISP="{NEW/RNW/MOD},x,DELETE"を指定して、使用後は削除しておく。
 - リスタートの前のジョブステップで作成したファイルを参照する場合は、DISP="{OLD/SHR},x,KEEP"を指定して、リスタート時に参照できるようにしておく。
- 前のステップで作成した一時ファイルを参照できないため、恒久ファイルに変更する必要がある。
- 次の理由で PREST ファイルの利用は適さない。
 - サブジョブのリスタートができない。
 - 相手ジョブも起動している必要がある。

- 相対世代番号で世代ファイルを参照している場合、リスタート時に別のファイルを参照しないようにするため、リスタート時には相対世代番号をリスタート前に戻す必要がある。このため、世代ファイルの使用は推奨されない。
- リスタートするジョブステップより前のジョブステップからパスされたファイルを、リスタートするジョブステップ以降で受け取る場合、未実行のジョブステップからファイルを受け取ることはできないので、ジョブ定義 XML ファイルの変更が必要となる。
- リスタートするジョブステップより前の DD 要素を参照している場合、ジョブ定義 XML ファイルを修正して逆方向に参照しないように変更する必要がある。
- ファイルはリスタートするステップの開始前の状態には戻ることはない。このため、リスタート時に扱うファイルが問題なく使用できるようにジョブを作成しておく必要がある。

3.8.5 ステップリスタートでの注意事項

ステップリスタートを実行する上でのほかの注意事項を次に示す。

- JSPEC 機能を使用して並列実行するジョブステップからリスタートできない。並列実行するジョブステップを指定しても検索対象にならないため、ほかに同一名のジョブステップがある場合は、そのジョブステップがリスタートの対象になる。ほかに同一名のジョブステップが存在しない場合は、KAKC0110-E メッセージを出力してリスタートに失敗する。
- ファイル、ディレクトリの排他/共用制御やジョブ定義 XML ファイルはリスタートで指定したジョブステップより前のジョブステップも処理の対象になる。このため、リスタートステップより前のジョブステップだけで使用するファイルがほかのジョブと排他して使用する定義の場合、実際にはそのファイルを使用しなくても、後から投入されたジョブが同一ファイルを使用していると待ち状態になる。また、リスタート時にほかのジョブで排他要求しているとリスタートしたジョブの実行が待ち状態になることがある。
- 指定したリスタートステップより前のジョブ定義 XML の STEP 要素、SETMAXRC 要素は処理されない。ただし、ファイルの排他/共用制御の対象になる。
- リスタートステップ以降でリスタートステップより前のジョブステップの終了コードを判定している場合、正しく判定できないためジョブ定義 XML ファイルの修正が必要となる。
- IF 要素の THEN 節や ELSE 節内のジョブステップからはリスタートできない。そのジョブステップを指定しても検索対象にならないため、ほかに同一名のジョブステップがある場合は、そのジョブステップがリスタートの対象になる。ほかに同一名のジョブステップが存在しない場合は、KAKC0110-E メッセージを出力してリスタートに失敗する。
- JP1/AJS のジョブ定義のパラメータで指定したリスタートステップ名の指定は、ジョブ終了後に、次回の実行に備えて削除しておく必要がある。

3.9 世代ファイル

3.9.1 世代ファイルの概要

給与ファイルや在庫ファイルなどの月ごとに更新されるファイルの旧ファイルと新ファイルや、旧バージョンと新バージョンでのプログラムの関係のように、互いに時間的または歴史的な関連を持った一連のファイルグループがある。これらのファイルグループは、1つのグループ名とその中の番号で参照ができ、このグループを世代データ群といい、その中の各ファイルを世代ファイルという。

世代データ群は、そのグループを識別する世代データ群名の基に登録され管理される。各世代ファイルは世代データ群名と世代番号によって識別される。

世代ファイル機能は、次の2種類がある。

1. 相対世代番号指定機能
2. 日付世代番号指定機能

(1) 相対世代番号指定機能

相対世代番号指定機能の世代番号は、絶対世代番号と相対世代番号で表す。

世代番号の指定は、相対世代番号で指定する。

(a) 絶対世代番号

各世代ファイルに対して、そのグループ内で一意的な絶対世代番号が与えられる。

絶対世代番号の形式は"Gnnnn"である。nnnnは世代番号を表し0000~9999の数値の範囲となる。

各世代ファイルは、世代データ群名のあとに絶対世代番号を"."（ピリオド）で付加したファイル名で識別される。絶対世代番号でDD要素に記述はできない。

<例>

世代データ群名が"ACCOUNT.X"、世代番号0003のファイル名は、次に示すようになる。

```
ACCOUNT.X.G0003
```

(b) 相対世代番号

相対世代番号とは最新の世代ファイルの世代番号を0とし、これよりも古い世代ファイルの世代番号を負の整数、新しく登録する世代ファイルの世代番号を正の整数で表した相対的な世代番号である。

各世代ファイルは、世代データ群名のあとに相対世代番号を括弧でくくって付加したファイル名で識別される。

<例>

ACCOUNT.X を世代データ群名として次に示す順番で3つの世代ファイルを登録する。

1. ACCOUNT.X.G0001
2. ACCOUNT.X.G0002
3. ACCOUNT.X.G0003

次のジョブで各世代ファイルを相対世代番号を用いて参照する場合、ファイル名は次に示すようになる。

```
ACCOUNT.X(-2)
ACCOUNT.X(-1)
ACCOUNT.X(0)
```

また、正の相対世代番号は、現在登録している最新の世代ファイルの世代番号にこの数値を加えた世代番号を持つ新しい世代ファイルを登録したい場合、または該当するジョブ内の以前のステップで世代データ群に登録した世代ファイルを参照する場合に用いる。

例えば、上記の例で ACCOUNT.X(+2)を指定して新しい世代ファイルを登録する場合、システムが自動的に ACCOUNT.X.G0005 の絶対世代番号を作成して登録する。

(2) 日付世代番号指定機能

世代番号の指定は、日付世代番号で指定する。

(a) 日付世代番号

日付世代番号指定機能を使用して世代ファイルを日付で管理するための世代番号である。日付世代番号の形式は、"EXD"*¹ または"Dyyyyymmdd"*² である。各世代ファイルは世代データ群.Dyyyyymmdd の形式で登録される。

<例>

ACCOUNT.X を世代データ群名とする世代ファイル

1. ACCOUNT.X.D20130611
2. ACCOUNT.X.D20130613
3. ACCOUNT.X.D20130615

各世代ファイルを、日付世代番号を用いて参照する場合、ファイル名は次に示すようになる。

- EXD を指定する場合

次のようにルートジョブネットの実行開始予定日 (EXD) を基準とし、これより古い世代ファイルを EXD に負の整数を付けて指定する。

```
ACCOUNT.X(EXD-2)
ACCOUNT.X(EXD-1)
ACCOUNT.X(EXD)
```

ただし、EXD に付けた-1 や-2 は、EXD に格納された日付の 1 日前、2 日前という意味ではなく、1 世代前、2 世代前という意味である。例えば、上記の例で実行開始予定日が 2013 年 6 月 15 日の場合、ACCOUNT.X(EXD-1)を指定すると、ACCOUNT.X.D20130613 を指定することとなる。

注※1 世代ファイルを日付で指定するために使用する環境変数「AJSEXDATE」に設定される日付に対応した世代ファイルを指定する際に使用する。AJSEXDATE は、JP1/AJS から実行されるルートジョブネット実行時に設定される環境変数であり、JP1/AJS からバッチジョブを起動するとジョブネットの実行開始日が設定される。ジョブネットの実行が遅延したり、翌日に再実行したりした場合でも、当初の実行予定日が設定される。この環境変数が設定されていない場合、EXD によって世代ファイルを指定することはできない。詳細については、JP1 のマニュアルを参照のこと。

注※2 yyyy：西暦，mm：月，dd：日

本機能は、通常は EXD を指定することによって使用する機能だが、障害時など例外的に特定の日付を指定したい場合、Dyyyymmdd を指定することで直接日付を指定することができる。

- Dyyyymmdd を指定する場合
ACCOUNT.X(D20130611)
ACCOUNT.X(D20130613)
ACCOUNT.X(D20130615)

(3) 世代データ群を構成するファイル

世代データ群を使用する場合、世代ファイルの割り当てに先立ち、世代管理ファイルを用意する必要がある。

- 世代管理ファイルの作成

世代管理ファイルは bjexgdg コマンドによって生成し、次に示すファイルによって構成される。

- "世代データ群名.gdg"
- ".世代データ群名.gdg"

世代で管理する世代ファイルは、相対世代番号指定機能と日付世代番号指定機能で次のようになる。

- 相対世代番号指定機能
世代データ群名に".Gnnnn"を付けたファイル名 (nnnn は 0000~9999 の数値)
- 日付世代番号指定機能
世代データ群名に".Dyyyymmdd"を付けたファイル名 (yyyy：西暦，mm：月，dd：日)

また、各世代ファイルは世代管理ファイルと同一ディレクトリに作成される。世代管理ファイルと世代ファイルが別々のディレクトリに点在している場合、これらのファイルでは世代管理ができなくなるため、必ず同一ディレクトリに配置する。

- 世代管理ファイルの削除

不要となった世代管理ファイルは、bjexgdg コマンドで削除できる。

3.9.2 世代データ群の作成

世代ファイルを作成する前に、世代ファイルを管理する世代データ群名を登録する必要がある。

(1) 世代データ群の作成

世代データ群の作成は、bjexgdg コマンドで行う。

bjexgdg コマンドの書き方を次に示す。

(a) 相対世代番号指定機能

```
bjexgdg -m -l エントリ数 世代ディレクトリ名/世代データ群名
```

- エントリ数

この世代データ群に属する世代ファイルの最大数（1 から 255 の数字）を指定する。

- 世代ディレクトリ名

世代データ群を作成するディレクトリ名を指定する。指定しない場合はカレントディレクトリとなる。

- 世代データ群名

世代データ群の名称を指定する。

bjexgdg コマンドの詳細については、「7. コマンド」の「bjexgdg（世代データ群の生成および更新（一般ユーザ用）」を参照のこと。

<例>

世代データ群名 ACCOUNT.X を/home/gdg ディレクトリに登録する。また、この世代データ群の最大世代ファイル数は 20 とし、世代ファイル数がこの最大数を超えた場合、最も古い世代ファイルから順に削除する場合（自動消去機能）の指定例を次に示す。

```
bjexgdg -m -l 20 /home/gdg/ACCOUNT.X
```

(b) 日付世代番号指定機能

```
[bjexgdg -m -d -l エントリ数 世代ディレクトリ名/世代データ群名]
```

- エントリ数

この世代データ群に属する世代ファイルの最大数（1 から 511 の数字）を指定する。

- 世代ディレクトリ名

世代データ群を作成するディレクトリ名を指定する。指定しない場合はカレントディレクトリとなる。

- 世代データ群名

世代データ群の名称を指定する。

<例>

世代データ群名 ACCOUNT.X を/home/gdg ディレクトリに登録する。また、この世代データ群の最大世代ファイル数は 20 とし、自動消去機能を指定する場合の指定例を次に示す。

```
[bjexgdg -m -d -l 20 /home/gdg/ACCOUNT.X]
```

(2) 世代データ群の配置

世代データ群は、1つの世代ディレクトリ下に1つを配置することが望ましい。

3.9.3 世代ファイルの作成と参照

(1) 世代ファイルの作成

(a) 相対世代番号指定機能

世代ファイルを作成する場合、世代データ群名のあとに相対世代番号を括弧で囲って世代ファイルを指定する。

作成例を次に示す。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM" />
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(+1)" TYPE="GDG" DISP="NEW, KEEP" /> • ACCOUNT.X.G0001の作成
      <DD NAME="DD2" DSN="/home/gdg/ACCOUNT.X(+2)" TYPE="GDG" DISP="NEW, KEEP" /> • ACCOUNT.X.G0002の作成
      <DD NAME="DD3" DSN="/home/gdg/ACCOUNT.X(+3)" TYPE="GDG" DISP="NEW, KEEP" /> • ACCOUNT.X.G0003の作成
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

上記の作成例を実行した場合、次に示す世代ファイルが作成される。

- ACCOUNT.X.G0001
- ACCOUNT.X.G0002
- ACCOUNT.X.G0003

(b) 日付世代番号指定機能

世代ファイルを作成する場合、世代データ群名のあとに EXD を括弧で囲って世代ファイルを指定する。

作成例を次に示す。このとき、+、-を指定することはできない。

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM"/>
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(EXD)" TYPE="GDG" DISP="NEW,KEEP" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

上記の作成例を実行した場合、次に示す世代ファイルが作成される。次の例は実行開始予定日が 2013 年 6 月 15 日の場合である。

- ACCOUNT.X.D20130615

障害時など例外的に特定の日付を指定したい場合、例外的に Dyyyymmdd^{*}と指定することによって、直接日付を指定することもできる。作成例を次に示す。

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM"/>
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(D20130615)" TYPE="GDG" DISP="NEW,KEEP" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

注※ yyyy：西暦，mm：月，dd：日

ただし，yyyy に指定できる西暦は，DISP 属性の第 1 パラメータが NEW の場合，バッチジョブ実行日の西暦の±2 年以内となる。例えば，バッチジョブ実行日が 2013 年 6 月 15 日の場合，2011 年 1 月 1 日～2015 年 12 月 31 日まで指定可能となる。また，バッチジョブ実行日が 2014 年 4 月 10 日の場合，2012 年 1 月 1 日～2016 年 12 月 31 日まで指定可能となる。

(2) 世代ファイルの参照

(a) 相対世代番号指定機能

作成した世代ファイルを参照する場合，世代データ群名のあとに相対世代番号を括弧で囲って世代ファイルを指定する。「(1) 世代ファイルの作成」で作成した世代ファイルの参照例を次に示す。

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM" />
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(-2)" TYPE="GDG" DISP="SHR" /> • ACCOUNT.X.G0001の参照
      <DD NAME="DD2" DSN="/home/gdg/ACCOUNT.X(-1)" TYPE="GDG" DISP="SHR" /> • ACCOUNT.X.G0002の参照
      <DD NAME="DD3" DSN="/home/gdg/ACCOUNT.X(0)" TYPE="GDG" DISP="SHR" /> • ACCOUNT.X.G0003の参照
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

(b) 日付世代番号指定機能

作成した世代ファイルを削除する場合、世代データ群名のあとに EXD, EXD-n, または Dyyyymmdd を括弧で囲って世代ファイルを指定する。参照例を次に示す。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM"/>
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(EXD-1)" TYPE="GDG" DISP="SHR,KEEP" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

障害時など例外的に特定の日付を指定したい場合、Dyyyymmdd と指定することによって、直接日付を指定することもできる。参照例を次に示す。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM"/>
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(D20130615)" TYPE="GDG" DISP="SHR,KEEP" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

(3) 世代ファイルの参照 (世代データ群名指定)

世代データ群名だけを指定して世代ファイルの参照をする場合、指定した世代データ群に登録されているすべての世代ファイルを参照する。

(a) 相対世代番号指定機能

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM"/>
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X" TYPE="GDG" DISP="SHR"/>
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

ACCOUNT.X に登録されている
世代ファイルの参照

上記指定で世代データ群 ACCOUNT.X に世代ファイルが 3 世代登録されている場合、次に示す DD 要素で指定した内容と同じ世代ファイルを参照する。

```

<?xml version="1.0" encoding="Shift_JIS"?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM"/>
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(0)" TYPE="GDG" DISP="SHR"/>
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(-1)" TYPE="GDG" DISP="SHR"/>
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(-2)" TYPE="GDG" DISP="SHR"/>
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

(b) 日付世代番号指定機能

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM"/>
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X" TYPE="GDG" DISP="SHR" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

上記の指定で世代データ群 ACCOUNT.X に世代ファイルが 3 世代登録されている場合、次に示す DD 要素で指定した内容と同じ世代ファイルを参照する。

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM"/>
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(EXD)" TYPE="GDG" DISP="SHR" />
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(EXD-1)" TYPE="GDG" DISP="SHR" />
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(EXD-2)" TYPE="GDG" DISP="SHR" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

3.9.4 世代ファイルおよび世代データ群の削除

世代データ群を削除する場合、次に示す順序で行う。

1. 世代ファイルの削除をする。
2. 世代データ群の削除をする。

(1) 世代ファイルの削除

- 相対世代番号指定機能

作成した世代ファイルを削除する場合、世代データ群名のあとに相対世代番号を括弧で囲って世代ファイルを指定する。

作成した世代ファイルの削除例を次に示す。

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM" />
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(-2)" TYPE="GDG" DISP="OLD,DELETE" />・ACCOUNT.X.G0001の削除
      <DD NAME="DD2" DSN="/home/gdg/ACCOUNT.X(-1)" TYPE="GDG" DISP="OLD,DELETE" />・ACCOUNT.X.G0002の削除
      <DD NAME="DD3" DSN="/home/gdg/ACCOUNT.X(0)" TYPE="GDG" DISP="OLD,DELETE" />・ACCOUNT.X.G0003の削除
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

- 日付世代番号指定機能

作成した世代ファイルを削除する場合、世代データ群名のあとに EXD または EXD-n を括弧で囲って世代ファイルを指定する。削除例を次に示す。

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM" />
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(EXD-1)" TYPE="GDG" DISP="OLD,DELETE" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

障害時など例外的に特定の日付を指定したい場合、Dyyyyymmdd と指定することによって、直接日付を指定することもできる。削除例を次に示す。

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM" />
      <DD NAME="DD1" DSN="/home/gdg/ACCOUNT.X(D20130615)" TYPE="GDG" DISP="OLD,DELETE" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

(2) 世代データ群の削除

世代データ群の削除は、bjexgdg コマンドで行う。

bjexgdg コマンドの書き方を次に示す。

```
bjexgdg -q [世代ディレクトリ名/]世代データ群名
```

- 世代ディレクトリ名

世代データ群を作成するディレクトリ名を指定する。指定しない場合は、カレントディレクトリとなる。

- 世代データ群名

世代データ群の名称を指定する。

bjexgdg コマンドの詳細については、「7. コマンド」の「bjexgdg (世代データ群の生成および更新 (一般ユーザ用))」を参照のこと。

例

世代データ群名 ACCOUNT.X を削除する。

世代ディレクトリ /home/gdg

```
bjexgdg -q /home/gdg/ACCOUNT.X
```

3.9.5 世代ファイルの保存日数超過による一括削除機能

- 相対世代番号指定機能

世代データ群は、世代管理ファイルの定義によって管理する世代ファイル数を制限 (1~255) できる。世代データ群の定義で、最も古い世代ファイル情報が世代管理対象から外れた場合、SCRATCH (自動消去機能) と NOSCRATCH (データ保存機能) のどちらかを選択し、世代ファイルの後処理の方法を定義できる。この場合、最も古い世代ファイル情報が最新の世代ファイル登録時に世代管理対象から外される。また、世代管理する世代ファイル数の制限いっぱいに登録されている場合に、新たな世代ファイルの登録要求が行われたときに動作する。

この機能では、世代管理ファイルの登録状態に関係なく、ユーザが指定する保存日数の範囲で、保存日数を超過した世代ファイルの削除と世代管理ファイルに登録されている情報を削除する。なお、この機能を使用する場合は、あらかじめ世代ファイルをバックアップすることを推奨する。世代ファイルをバックアップすることで、誤って世代ファイルを一括削除してしまった場合にリストアできる。

- 日付世代番号指定機能

世代データ群は、世代管理ファイルの定義によって管理する世代ファイル数を制限 (1~511) できる。日付世代番号指定機能では、世代データ群の定義で、SCRATCH しか指定することができない。世代管理する世代ファイル数の制限いっぱいに登録されている場合に、新たな世代ファイルの登録要求が行われると、最も古い世代ファイル情報が最新の世代ファイル登録時に世代管理対象から外される。ただし、「最も古い」の意味は、実行開始日 (EXD に格納された日付または Dyyyymmdd) が最も古いという意味であり、世代ファイルを作成した日付が古いというわけではない。例えば、次の3つの世代ファイルが登録されており、登録世代数が上限に達している状態で、2013/04/20 の世代ファイルを作成した場合、ACCOUNT.X.20130417 の世代ファイルおよび世代情報が削除される。

- 2013/04/20 の世代ファイル作成前

```
ACCOUNT.X. 20120419  
ACCOUNT.X. 20120418  
ACCOUNT.X. 20120417
```

また、同じく次の3つの世代ファイルが登録されており、登録世代数が上限に達している状態で、2013/04/16 のジョブを実行した場合、生成される世代ファイルは、最も古い世代 (EXD に格納された日付または Dyyyymmdd が最も古い) となる。このような場合、世代管理ファイルは更新せず、世代ファイルは一時的に作成し、すぐに削除する。

- 2013/04/16 の世代ファイル作成前

```
ACCOUNT.X. 20120419
ACCOUNT.X. 20120418
ACCOUNT.X. 20120417
```

- 2013/04/16 の世代ファイル作成後

ACCOUNT.X. 20120416 が作成されたが、すぐに削除される。

```
ACCOUNT.X. 20120419
ACCOUNT.X. 20120418
ACCOUNT.X. 20120417
```

(1) 世代ファイルの保存日数指定による一括削除の方法

世代ファイルの保存日数指定による一括削除機能は、相対世代番号指定機能だけでサポートしている機能であり、日付世代番号指定機能ではサポートしていない。

世代ファイルの保存日数指定による一括削除の例を次に示す。

世代群"bjgdg"は次のような登録状態となっている。

- bjexgdg -e bjgdg コマンドの表示結果 (8月7日の 15:40 時点)

```
GDG BASE ----- bjgdg
ATTRIBUTES
LIMIT ----- 10 SCRATCH
INFILE ----- 7
ASSOCIATIONS
( 0) -- bjgdg.G0007 0020
(-1) -- bjgdg.G0006 0021
(-2) -- bjgdg.G0005 0022
(-3) -- bjgdg.G0004 0023
(-4) -- bjgdg.G0003 0024
(-5) -- bjgdg.G0002 0025
(-6) -- bjgdg.G0001 0026
```

- ls コマンドの表示結果

世代ファイルの状態を ls コマンドで表示 (表示されている日付は最終更新日付)。

```
-rw-r--r-- 1 user job 584125988 Jul 12 15:32 bjgdg.G0001
-rw-r--r-- 1 user job 584126200 Jul 13 15:32 bjgdg.G0002
-rw-r--r-- 1 user job 584126488 Jul 14 15:32 bjgdg.G0003
-rw-r--r-- 1 user job 584126680 Jul 15 15:32 bjgdg.G0004
-rw-r--r-- 1 user job 584126750 Jul 16 15:32 bjgdg.G0005
-rw-r--r-- 1 user job 584126998 Jul 17 15:32 bjgdg.G0006
-rw-r--r-- 1 user job 584127210 Jul 18 15:32 bjgdg.G0007
```

8月7日の 15:40 時点で保存日数が 20 日を超えている (21 日以上経過しているもの) 世代ファイルと世代管理ファイルの情報を一括して削除したい場合、次のコマンドを入力する。

```
bjexhkgdg -d 20 bjgdg
```

コマンド実行によって、最終更新日付がコマンド実行時点から 21 日以上経過したものは削除され、世代管理情報からも削除される。

- bjexgdg -e bjgdg コマンドの表示結果

```
GDG BASE ----- bjgdg
ATTRIBUTES
LIMIT ----- 10  SCRATCH
INFILE ----- 1
ASSOCIATIONS
( 0) -- bjgdg.G0007 0020
```

- ls コマンドの表示結果

```
-rw-r--r-- 1 user job 584127210 Jul 18 15:32 bjgdg.G0007
```

(2) bjexhkgdg コマンド実行に関する注意事項

- このコマンドで指定する保存日数を間違った場合、指定した保存日数を超過した世代ファイルは無条件に削除する。そのため、指定する保存日数は間違わないようにすること。なお、このコマンドを実行する前に該当する世代管理ファイルおよび世代データセットのバックアップを取得しておくこと。
- 世代管理ファイルおよび世代ファイルの更新は、bjexhkgdg コマンドの実行ユーザに対して更新権限が必要である。
- 世代管理ファイルの属性（自動消去機能またはデータ保存機能）に関係なく、bjexhkgdg コマンドを実行した場合、保存日数超過した世代ファイルと世代管理情報は削除される。
- bjexhkgdg コマンド実行時、日付世代番号指定機能の世代データ群を指定するとエラーとなる。

3.9.6 世代ファイルの世代管理ファイルへの登録タイミング変更機能

世代ファイルを割り当てるジョブで、世代ファイルを作成するジョブステップは正常に実行できたが、後続のジョブステップで異常終了した場合でも、世代ファイルを新規に生成した世代データ群の世代管理情報は更新される（ユーザ指定に従った処理動作を行う）。この機能では、世代管理情報の更新タイミングを世代ファイルを生成したジョブステップ単位ではなくジョブ終了時に変更し、ジョブの終了状態（ジョブ終了時に設定されている戻り値）によって、世代管理情報の更新を行うかどうかを判定し、登録の要否を決定する機能である。

(1) 設定方法

この機能を使用する場合は、バッチジョブ実行システムの設定ファイルに GDG_REGISTER_JOBRC_THRESHOLD パラメータを設定すること。

(2) 世代ファイルの世代管理ファイルへの登録タイミング

世代ファイルの世代管理ファイルへの登録タイミングについて説明する。

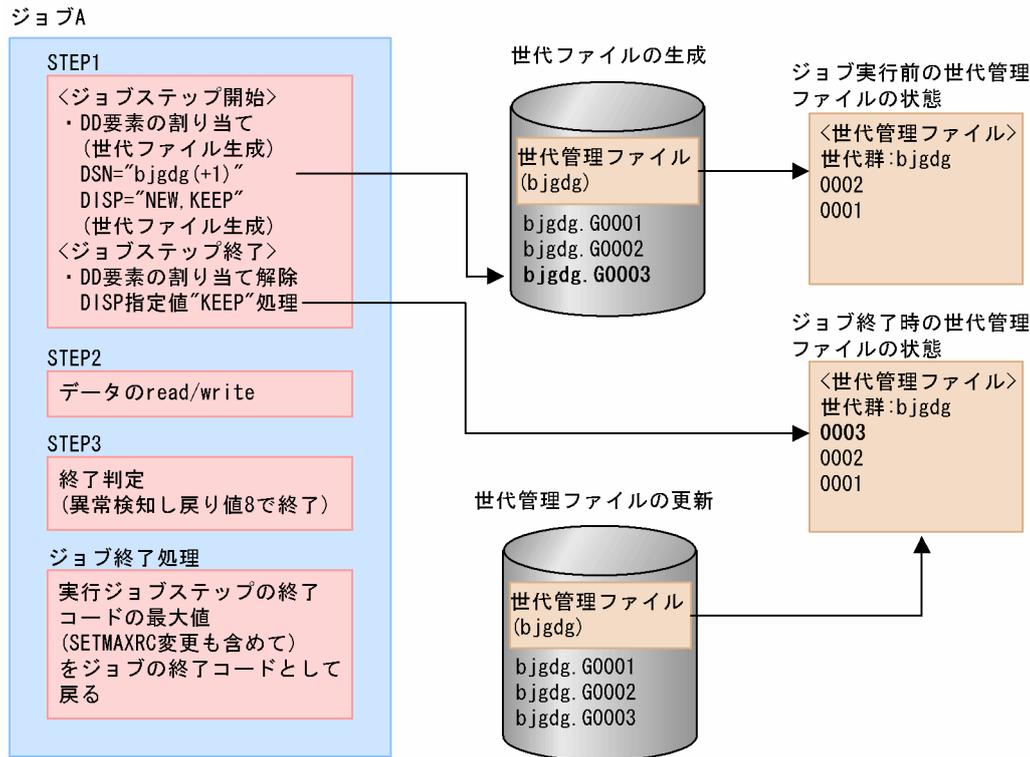
(a) 通常の世代更新タイミングの場合

世代ファイルを生成するジョブステップが正常に実行された場合、生成した世代ファイルの登録は DD 要素の DISP 属性の指定値に従い行われる。そのため、後続のジョブステップでの異常有無に関係なく、更新された世代管理ファイルの状態はジョブ実行前の状態には戻らない。

世代ファイルの世代管理ファイルへの登録タイミング変更機能を使用しない場合の世代ファイルの世代管理ファイルへの登録タイミング（通常の世代更新タイミング）を次の図に示す。

図 3-20 世代ファイルの世代管理ファイルへの登録タイミング（通常の世代更新タイミング）

通常の世代更新のタイミング



(b) ジョブ終了コードが異常終了と認識された場合

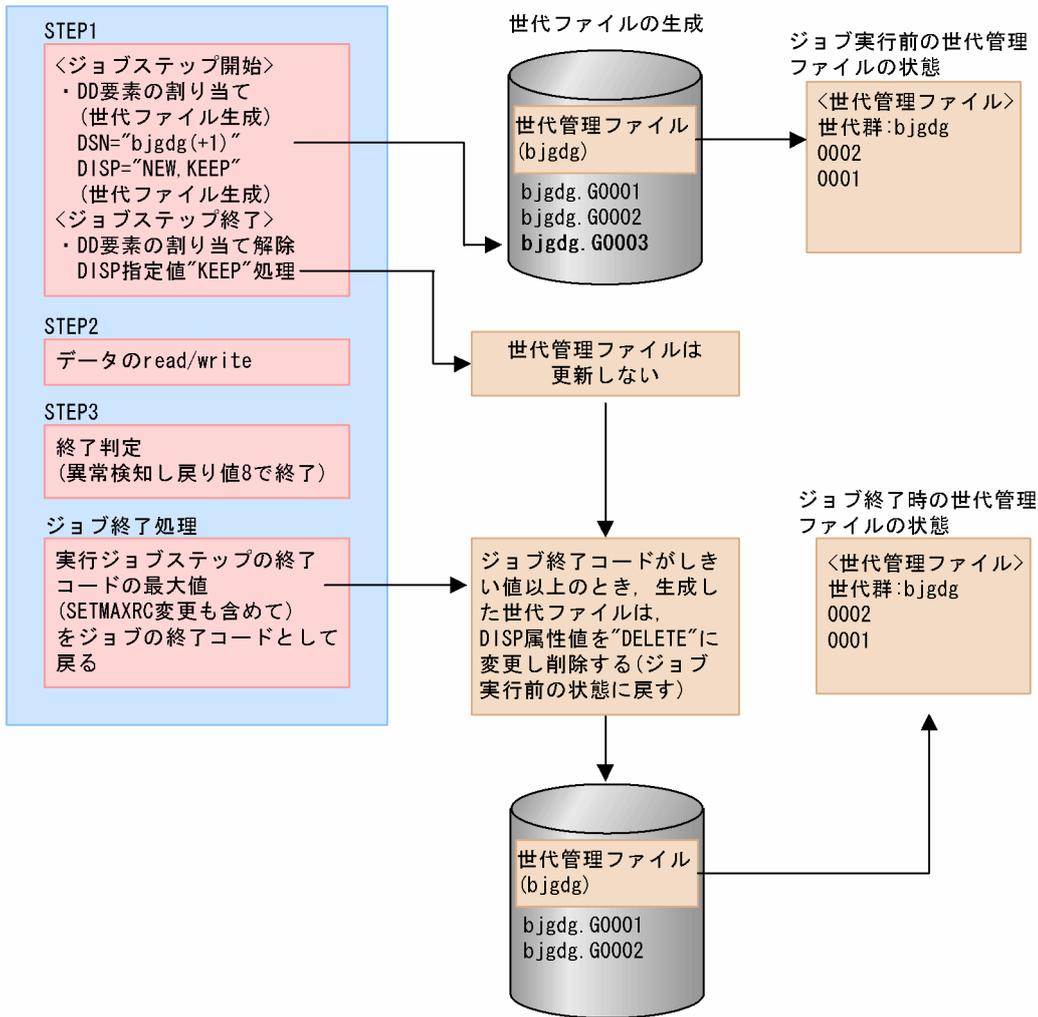
世代ファイルを生成するジョブステップでは、ジョブステップ終了時に世代管理ファイルの更新は行わない。ジョブ終了時に更新の可否を判定し、更新不可（ジョブは異常終了と認識された場合、世代管理ファイルの状態はジョブ開始時点の状態を保持する）と判定した場合、世代管理ファイルの更新は行わないで、生成した世代ファイルを削除（DELETE）する。

世代ファイルの世代管理ファイルへの登録タイミング（ジョブ終了コードが異常終了と認識された場合）を次の図に示す。

図 3-21 世代ファイルの世代管理ファイルへの登録タイミング（ジョブ終了コードが異常終了と認識された場合）

世代ファイルの世代管理ファイルへの登録タイミング変更機能を適用した場合
 （ジョブ終了コードが異常終了と認識された場合）

ジョブB



(c) ジョブ終了コードが正常終了と認識された場合

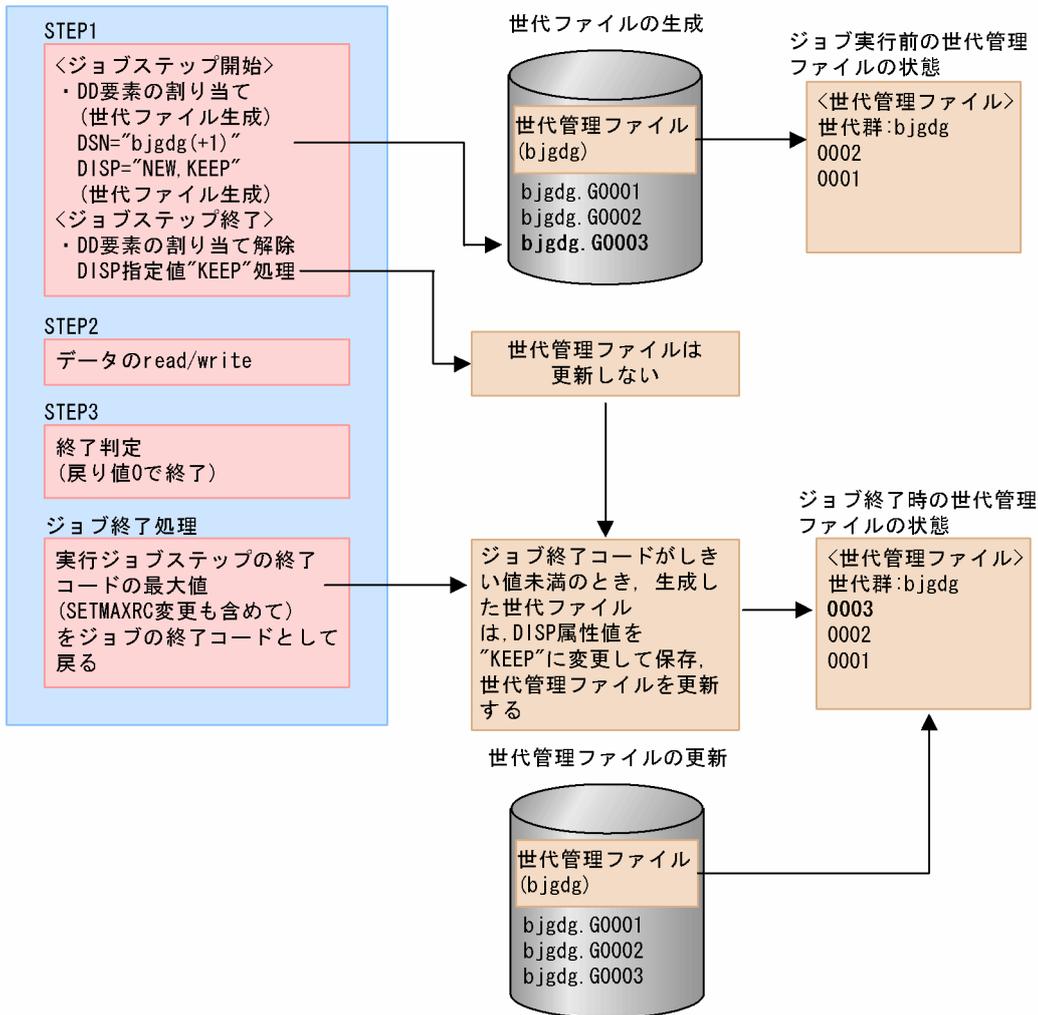
世代ファイルを生成するジョブステップでは、ジョブステップ終了時に世代管理ファイルの更新は行わない。ジョブ終了時に更新の可否を判定し、更新可能（ジョブは正常終了と認識された場合、世代管理ファイルの状態は最新世代ファイルの情報を更新した状態となる）と判定した場合、世代管理ファイルの更新を行い、生成した世代ファイルは保持（KEEP）される。

世代ファイルの世代管理ファイルへの登録タイミング（ジョブ終了コードが正常終了と認識された場合）を次の図に示す。

図 3-22 世代ファイルの世代管理ファイルへの登録タイミング（ジョブ終了コードが正常終了と認識された場合）

世代ファイルの世代管理ファイルへの登録タイミング変更機能を適用した場合
 （ジョブ終了コードが正常終了と認識された場合）

ジョブC



(3) 注意事項

- 新規に生成する世代ファイルを DISP="NEW,KEEP" で保持する指定の場合、世代管理ファイルの更新をジョブ終了時に延期する機能であるため、DISP="NEW,DELETE" のように削除指定を行っているときは、この機能の適用の有無に関係なく、ジョブステップ終了時に世代ファイルは削除される。
- ジョブの正常/異常終了は GDG_REGISTER_JOBRC_THRESHOLD パラメータに指定したしきい値によって判定する。そのため、ジョブの終了コードが 0 以外であっても正常終了と見なされる場合がある。
- この機能を使用した場合、ジョブの実行が完了するまでは、世代管理情報が確定しない。そのため、後続のジョブステップでは、先行ジョブステップで新規に作成した世代ファイルを含めた世代データ群名指定による世代ファイルの参照はしないこと。世代データ群名指定については、「3.9.3(3) 世代ファイルの参照（世代データ群名指定）」を参照のこと。

- この機能を使用した場合、ジョブの実行が完了して世代管理情報が確定するまでは、世代ファイルは一時ファイルと同じ扱いとなる。そのため、ジョブステップの UAP の異常終了などによってジョブが終了した場合、該当するジョブでそれまでに作成された世代ファイルはすべて削除される。

3.9.7 世代ファイルの注意事項

(1) 世代登録可能数以上の世代ファイル番号の登録が発生した場合の注意事項

世代データ群に登録できる世代ファイルの上限は、次のようになっている。

世代番号指定機能：1～255

日付世代番号指定機能：1～511

登録可能数は、bjexgdg コマンドの-l オプションに指定する値によって変更可能となっている。

なお、世代データ群の属性によって、登録可能数分の世代ファイルが登録済みの状態で、新たに世代ファイルの登録が行われると、世代ファイルが次に示す扱いとなる。

(a) 自動消去機能 (SCRATCH) が設定された世代データ群の場合

- 相対世代番号指定機能
新たな世代ファイルの登録によって、世代登録可能数を超えた場合、最古の世代ファイルを世代データ群の管理から削除する。世代データ群の管理から削除された世代ファイルは、自動的に消去される。
- 日付世代番号指定機能
新たな世代ファイルの登録によって、世代登録可能数を超えた場合、実行開始日 (EXD に格納された日付または Dyyyymmdd) が最も古い世代ファイルを世代データ群の管理から削除する。世代データ群の管理から削除された世代ファイルは、自動的に消去される。

(b) データ保存機能 (NOSCRATCH) が設定された世代データ群の場合

- 相対世代番号指定機能
新たな世代ファイルの登録によって、世代データ群の管理数を超えた場合、最も古い世代ファイルは世代データ群の管理から削除する。ただし、世代ファイルの消去は行わないため、世代ファイルが増加し続ける。世代ファイルに与える 4 けたの通し番号 (0000～9999) を使い切った時点で世代ファイル機能自体が利用できなくなるため、世代データ群を作る場合は必ずエントリ数を指定して作成すること。なお、世代データ群の管理数を超えた世代ファイルは、必要に応じて rm コマンドで削除すること。
- 日付世代番号指定機能
日付世代番号指定機能の世代データ群に対して NOSCRATCH を指定することはできない。

(2) 世代ファイルを作成する場合の DD 要素の注意事項

- TYPE 属性は GDG を指定する。
- 同一ジョブ内で作成した世代ファイルを後方のジョブステップで参照する場合、作成時と同じ相対世代番号を指定する。
- DISP 属性の第 1 パラメータは NEW、第 2 パラメータは KEEP を指定する。

(3) 世代ファイルを参照する場合の DD 要素の注意事項

- DSN 属性と DISP 属性は必ず指定すること。

<例>

```
<DD NAME="DD1" DSN="ACCOUNT.X(0)" TYPE="GDG" DISP="SHR" />
```

- 同一ジョブ内で作成した世代ファイルを後方のジョブステップで参照する場合、作成時と同じ相対世代番号を指定する。
- DISP 属性の第 1 パラメータに MOD を指定して世代ファイルを参照する場合、指定した世代ファイルが存在しないと割り当てエラーになる。

(4) 自動消去機能 (SCRATCH) が設定された世代データ群の注意事項

相対世代番号指定機能では、世代データ群で管理される世代ファイルは、次に示す 3 つの絶対世代番号群に分類される。

- I 群：G0000～G0999 の範囲に含まれる番号
- II 群：G1000～G8999 の範囲に含まれる番号
- III 群：G9000～G9999 の範囲に含まれる番号

通常の大関係では、"I 群<II 群"または"II 群<III 群"となる。

また、世代データ群に絶対世代番号群が III 群登録されている場合、I 群の絶対世代番号群の世代ファイルを登録すると絶対世代番号がラップアラウンドする。

ラップアラウンド後、III 群の絶対世代番号群が存在する場合、I 群の世代ファイルを登録すると、G10000～G10999 と解釈するため、大関係は、"III 群<I 群"となる。

絶対世代番号 G0000～G9999 の世代ファイル登録時の扱いを次の表に示す。

表 3-19 絶対世代番号 G0000～G9999 の世代ファイル登録時の扱い

絶対世代番号	同一世代データ群中に G9000～G9999 のファイルを含まない場合	同一世代データ群中に G9000～G9999 のファイルを含む場合
Gnnnn が G0000～G0999 のファイル	G0000～G0999 と見なす。	G10000～G10999 と見なす。

絶対世代番号	同一世代データ群中に G9000～G9999 のファイルを含まない場合	同一世代データ群中に G9000～G9999 のファイルを含む場合
Gnnnn が G1000～G9999 のファイル	G1000～G9999 と見なす。	

しかし、世代データ群にすべての絶対世代番号群が登録されると、大小関係が"II 群<III 群<I 群"となるため、常に最新に登録した世代ファイルが、世代データ群の管理上、最新の世代ファイルとならなくなるため、KAKC4064-E メッセージを出力し、世代ファイルの登録防止をする。

世代ファイルが登録防止となる条件を次の表に示す。

表 3-20 世代ファイルが登録防止となる条件

登録する世代ファイルの絶対世代番号群	世代データ群の管理上、存在する絶対世代番号群	システムでの処置
I 群：G0000～G0999 の範囲	II 群および III 群	KAKC4064-E メッセージを出力し、世代ファイルの登録防止をする。
II 群：G1000～G8999 の範囲	III 群および I 群	
III 群：G9000～G9999 の範囲	I 群および II 群	

KAKC4064-E メッセージを出力し、世代ファイルの登録防止となる例

世代データ群作成時、エントリ数 255 を指定し、常に世代ファイルを相対(+255)で登録する。

世代ファイルが登録防止となる例を次の表に示す。

表 3-21 世代ファイルが登録防止となる例

登録される世代ファイルの絶対世代番号群	登録される世代ファイル	システムでの処置
I 群：G0000～G0999 の範囲	G0255, G0510, G0765	世代ファイルは登録される。
II 群：G1000～G8999 の範囲	G1020, G1275～G8925	
III 群：G9000～G9999 の範囲	G9180 (登録防止となる)	KAKC4064-E メッセージを出力し、世代ファイルの登録防止をする。

(5) 最新世代として登録した世代ファイル番号が登録されない場合

世代ファイル登録時に KAKC4064-E メッセージを出力し、登録不可となった場合は、メッセージの処置に従い、最古の絶対世代番号群に含まれるすべての世代ファイルを、削除し、再度ジョブを実行する必要がある。

(6) 世代ファイルを削除する場合の注意事項

世代ファイルは DISP 属性で DELETE パラメータを指定することで削除できる。

ただし、世代データ群名指定の場合、DISP 属性で DELETE パラメータを指定しても削除はできない。世代データ群名指定で DISP 属性に DELETE パラメータを指定した場合、バッチジョブ実行システムは DELETE パラメータを指定しても KEEP パラメータを仮定する。

すべての世代ファイルを削除する場合は、世代ファイルの削除を行い世代データ群に登録されている世代ファイルが1つもなくなった時点で、bjexgdg コマンドの-q オプション指定によって世代データ群を削除する。上記の指定で削除ができない場合は、rm コマンドによって次に示すファイルをすべて削除する。

- "世代データ群名.Gnnnn" (nnnn は 0000～9999 の数字) または、"世代データ群名.Dyyyyymmdd" (yyyy：西暦, mm：月, dd：日)
- "世代データ群名.gdg"
- ".世代データ群名.gdg"

(7) 世代ファイルのバックアップとリストア

世代ファイルのバックアップおよびリストアは、バッチジョブ実行システムが停止している状態で行う必要がある。

バックアップおよびリストアを行う対象ファイルを次に示す。

- "世代データ群名.Gnnnn" (nnnn は 0000～9999 の数字) または、"世代データ群名.Dyyyyymmdd" (yyyy：西暦, mm：月, dd：日)
- "世代データ群名.gdg"
- ".世代データ群名.gdg"

上記のファイルをバックアップおよびリストアしない場合、世代管理の整合性を保つことができなくなる。

(8) 世代管理ファイル作成時の注意事項

世代管理ファイルは bjexgdg コマンドで作成する。このとき、bjexgdg コマンドの起動時の umask の指定に従い、ファイルパーミッションが設定される。世代管理ファイルには、世代ファイルを使用するすべてのユーザに対して書き込み権限・読み込み権限を設定する必要がある。世代ファイルを参照するだけでも世代管理ファイルには書き込み権限が必要である。

(9) 日付世代番号指定機能の注意事項

(a) 排他制御機能について

日付世代番号指定機能では、LOCK_OPTION パラメータで ENABLE を指定し、排他制御機能を有効にしなければ使用することはできない。

(10) バッチジョブの強制終了によって生じる世代管理ファイルの矛盾についての注意事項

世代ファイルを作成し、世代管理ファイル更新前にバッチジョブが強制終了すると、作成された世代ファイルは世代管理されていない状態になることがある。この場合この世代ファイルは、世代ファイルとしての機能を使用することができないため、rm コマンドなどによって削除してから、ジョブを再実行する必要がある。

この世代ファイルを削除しないでジョブを再実行すると、相対世代番号指定機能の場合は、世代管理ファイルに管理されていない世代ファイルが残り、その次の絶対世代番号の世代ファイルを作成する。日付世代番号指定機能の場合は、同一名称の世代ファイルが存在するためエラーとなる。

3.10 SYSIN データの読み込み

ジョブ定義 XML ファイル中に置かれた、ジョブ実行時に使用するデータを SYSIN データという。

3.10.1 SYSIN データの書き方

DD 要素の TYPE 属性値が DATA である場合を SYSIN データとする。SYSIN データの内容は、DD 要素の内容として指定する。

指定例を次に示す。

```
<DD NAME="DD名" TYPE="DATA">
  <![CDATA[
  !
  SYSINデータの内容
  !
  ]]>
</DD>
```

- CDATA セクションの終了を示す"]]>"以降に"!"を記述しないこと。
- SYSIN データの内容に文字列"]]>"を含めないこと。含めた場合、CDATA セクションの終了と解釈することがある。

3.10.2 SYSIN データとなる文字列

SYSIN データは複数行にわたり記述できる。

SYSIN データは、"!" (感嘆符) で囲む。"!" (感嘆符) で SYSIN データの範囲を示す。

XML では、特定の文字は直接文字として記述できないため、SYSIN データは CDATA セクションとして記述し、ジョブ定義 XML と解釈されないようにする。CDATA セクションは、<![CDATA[と]]>に囲まれた文字列であり、]>以外の文字列を指定できる。

SYSIN データの開始は、場合によって異なる。

SYSIN データの先頭を 1 カラム目から記述できるように、先頭の"!"のある行が空白（存在しなくてもよい）と改行だけであれば、その次の行から SYSIN データとして扱う。SYSIN データの終了は、最後の"!"の直前の文字である。

- 例 1

```
<![CDATA[
!
△△ABC△△
EFG
```

```
!  
]]>
```

- 例 2

```
<![CDATA[  
!△△  
△△ABC△△  
EFG  
!  
]]>
```

先頭の"!"のある行に空白以外の文字がある場合、先頭の"!"の次の文字から SYSIN データとして扱う。例 2 の書き方は誤りの原因となりやすい。通常は、例 1 のように記述すること。

- 例 3

```
<![CDATA[  
!△△ABC△△  
EFG  
!  
]]>
```

先頭の"!"と最後の"!"が同一行に存在する場合（先頭の"!"と最後の"!"の間に改行がない場合）、その間の文字列をすべて、SYSIN データとして扱う。

例 3 の書き方は誤りの原因となりやすい。通常は、例 1 のように記述すること。

- 例 4

```
<![CDATA[ _ !△△ABC△△! _ ]]>
```

- 例 5

```
<![CDATA[  
!△△ABC△△!  
]]>
```

- 例 6

```
<![CDATA[ !! ]]>
```

0 バイトの SYSIN データは、次に示すように記述する。先頭と最後の"!"を連続させる。

- 例 7

```
<![CDATA[  
!  
△△ABC△△  
EFG!  
]]>
```

最終行を改行しない場合、次に示すようにデータの直後で改行しないで、"!"を記述する。

(凡例)

△：空白を示す。

ゴシック：SYSIN データを示す。

3.10.3 SYSIN データを順編成固定長ファイルとしてプログラムに渡す方法

SYSIN データは、改行で区切られたテキスト編成ファイルとしてプログラムに渡される。しかし、VOS3 から移行した COBOL プログラムなど、プログラムが SYSIN データを順編成固定長ファイルとして扱っている場合、プログラムが正しく動作しない。

SYSIN データを順編成固定長ファイルとしてプログラムに渡すには、ジョブ定義 XML ファイルの SYSIN データを指定する DD 要素に、SAMF="YES"を指定する。この場合、バッチジョブ実行システムは SYSIN データをレコード長 80 バイトの順編成ファイルに変換し、プログラムに渡す。DD 要素に SAMF="YES"を指定した場合、レコード中に改行を含めることはできない。

SYSIN データの変換規則を次に示す。

- 「1 行のバイト数が 80 バイト+改行コード」の場合、改行を削除する。
- 「1 行のバイト数が 80 バイト未満+改行コード」の場合、改行を削除し、1 行が 80 バイトになるよう末尾に半角空白を挿入する。
- 「1 行のバイト数が 80 バイトより大きい+改行コード」の場合、81 バイト目以降に空白以外の文字があれば、ジョブ定義 XML ファイルの解析時にエラーとなり、ジョブを実行しないで終了する。「81 バイト目以降が 1 つ以上の空白+改行」の場合、それらを削除する。
- 最後の行が改行で終わっていない場合、半角空白だけを挿入する。

指定例を次に示す。

- プログラムが SYSIN データをテキスト編成ファイルとして扱う場合

ジョブ定義 XML ファイル

```
<DD TYPE="DATA" SAMF="NO">
<![CDATA[
!
AAAAAAAAAA...AAAAAAAAAAAAAAAAAAAA¥n ←80バイト+改行
BBBBBBBBBB...BBBBBBBBBB¥n ←70バイト+改行
!
]]>
</DD>
```

(凡例)

¥n：改行を示す。

レコード

```
レコード 1 AAAAAAAAAA...AAAAAAAAAAAAAAAAAAAA¥n ←80バイト+改行
レコード 2 BBBBBBBBBB...BBBBBBBBBB¥n ←70バイト+改行
```

(凡例)

¥n：改行を示す。

COBOL プログラムに渡される SYSIN データの内容

```
AAAAAAAAAA...AAAAAAAAAAAAAAAAAAAA¥n  
BBBBBBBBBB...BBBBBBBBBB¥n  
[EOF]
```

(凡例)

¥n：改行を示す。

[EOF]：ファイルの終わりを示す。

- プログラムが SYSIN データを順編成固定長ファイルとして扱う場合

ジョブ定義 XML ファイル

```
<DD TYPE="DATA" SAMF="YES">  
<![CDATA[  
!  
AAAAAAAAAA...AAAAAAAAAAAAAAAAAAAA¥n ←80バイト+改行  
BBBBBBBBBB...BBBBBBBBBB¥n ←70バイト+改行  
!  
]]>  
</DD>
```

(凡例)

¥n：改行を示す。

レコード

```
レコード 1  AAAAAAAAAA...AAAAAAAAAAAAAAAAAAAA ←80バイト※1  
レコード 2  BBBBBBBBBB...BBBBBBBBBB^^^^^^^^^^^^ ←80バイト※2
```

(凡例)

^：半角空白を示す。

注※1 改行を削除。

注※2 半角空白 10 バイト挿入，改行を削除。

COBOL プログラムに渡される SYSIN データの内容

```
AAAAAAAAAA...AAAAAAAAAAAAAAAAAABBBBBBBBBB...BBBBBBBBBB^^^^^^^^^^^^[EOF]
```

(凡例)

^：半角空白を示す。

[EOF]：ファイルの終わりを示す。

この機能を使用する場合の注意事項を次に示す。

- DD 要素に SAMF 属性を指定する場合は，TYPE 属性値に"DATA"が指定されていて，かつ同一ジョブステップの EXEC 要素の LANG 属性値に"COBOL"が指定されている必要がある。それ以外の場合には，ジョブ定義 XML ファイルの解析時にエラーとなり，ジョブを実行しないで終了する。

- DD 要素の SAMF 属性に "YES" を指定した場合は、改行を含む SYSIN データを COBOL プログラムに渡すことはできない。

3.11 SYSOUT の機能

SYSOUT データファイルの定義は、DD 要素の TYPE 属性で SYSOUT を指定して行う。システムはこの指定に従って、いったんスプール内に置かれた出力情報を、出力装置に出力する。この SYSOUT データファイルを定義する DD 要素の属性では、次に示す指定ができる。

- (a)文字セットイメージ（文字配列テーブル名または文字セットイメージ名の指定）
- (b)注釈（任意の文字列の指定）
- (c)出力情報の複写（複写回数の指定）
- (d)グループ複写回数（グループ複写回数の指定）
- (e)出力先（出力先識別子の指定）
- (f)出力の形式（fcb 識別子の指定）
- (g)デフォルト機能キャラクタイメージ（デフォルト機能キャラクタイメージ名の指定）
- (h)出力の書式（書式番号の指定）
- (i)出力保留属性（HOLD 属性での YES の指定）
- (j)印字モード（プリンタの印字モードの指定）
- (k)行単位での文字列配列テーブルの切り替え（行単位に使用する文字列配列テーブルの変更有無の指定）
- (l)書式オーバーレイ適用枚数（書式オーバーレイ適用枚数の指定）
- (m)書式オーバーレイモジュール（書式オーバーレイモジュール名の指定）
- (n)印刷制御文字の種別（印刷制御文字種別の指定）
- (o)出力クラス（クラス名の指定）
- (p)出力後保存，出力後非保存（KEEP 属性の指定）
- (q)レベル 1 外字名の指定（GAJII1 属性の指定）
- (r)PDE ファイル形式 2 の指定（FTYPE 属性の指定）
- (s)SYSOUT データファイルのエンコード形式の指定（ENCODE 属性の指定）

JOBPARM 要素では、特別な指定のない ((f), (h)の指定がない) SYSOUT 全体の書式番号, FCB, および ROOM (仕分け番号) の指定ができる。

3.11.1 出力クラス

出力クラスの定義は DD 要素の SYSOUT 属性を指定して行う。この出力クラスの目的は、出力情報をグループ分けすることである。

- 出力クラスは、1～31 文字の記号名称で指定する。ただし、* を指定した場合、JOB 要素の MSGCLASS 属性で指定したものと同一の出力クラスを指定したものと見なす。
- 出力クラスのグループ分けは、設定ファイルで定義する。SYSOUT 属性では、設定ファイルで定義した出力クラスを指定する。
- SYSOUT 情報をジョブログと同じ出力装置に出力したい場合には、SYSOUT 属性で指定する出力クラスを、JOB 要素の MSGCLASS 属性で指定したものと同じにする。特殊な指定をした場合、同じ出力装置への出力が保証されない。
- 出力クラスと出力装置の対応づけは、設定ファイルで転送先のクラスを定義する。
- 出力待ちの出力ジョブの出力クラスは変更できない。

3.11.2 出力情報の複写

ジョブ実行の結果として出力される出力情報は、DD 要素の COPIES 属性を指定することで複写できる。

3.11.3 出力形式

プリンタを使用してジョブを印刷する場合、DD 要素の FCB 属性または JOBPARM 要素の FCB 属性で FCB 識別子を指定することによって、lpi を決めることができる。PDE 側に定義されている属性を指定する。

3.11.4 出力保留

SYSOUT データファイルはいったんスプール内に格納されたあと、スプールジョブ転送サービスによって取り出されプリントサーバに転送されるが、DD 要素の HOLD 属性で YES を指定することによって出力を保留できる。これを出力保留という。

また、出力保留の要求は設定ファイルの出力クラスの定義に、センタで出力保留属性が指定されているクラス名を指定することによってもできる。

3.11.5 出力後保存

スプールジョブ転送サービスによって出力装置にいったん出力された SYSOUT データファイル（ジョブログを含む）は、出力後保存属性の指定がある場合、出力が完了するとスプール内に保存される。

出力後に保存された SYSOUT データファイルは、必要となった時点で bjexchjob コマンドの指定で属性を変更することで、スプールジョブ転送サービスの取り出し対象となる。この SYSOUT データファイルは、出力後は再び保存の対象となる。このため、bjexrmjob コマンドで定期的に削除すること。

出力後保存属性がない SYSOUT データファイルは、転送サービスの取り出しの対象となった時点でスプールから削除する。このため、転送処理で失敗した場合に再度転送したいときは、転送ディレクトリから bjextrjob コマンドで再転送する。

出力後保存属性は、SysoutClass パラメータの KEEP オペランドで YES を指定する。または DD 要素の KEEP 属性で YES を指定する。

3.11.6 プリンタへの出力要求

SYSOUT データファイルをプリンタに出力する場合、次に示す機能がある。

(1) 文字配列テーブル名または文字セットイメージ名

SYSOUT データファイル出力時、標準以外の文字を印刷する場合、SYSOUT データファイルを定義する DD 要素の CHARS 属性で文字配列テーブル名または文字セットイメージ名を指定する。詳細については、マニュアル「PRINT DATA EXCHANGE for Open」または PDE のドキュメントを参照のこと。

(2) 一定書式、枠組の印刷

一定の書式、枠組を SYSOUT データファイルのレコードにかぶせて印刷（オーバーレイ印刷）するには、PDE と連携して書式オーバーレイを使用する。書式オーバーレイを使用する場合、SYSOUT データファイルを定義する DD 要素の OVERLAY 属性で書式オーバーレイモジュール名を指定する。

(3) その他の情報

SYSOUT データファイル出力時に、漢字モード・ノンインパクトモードの切り替えや、文字セットイメージの変更の有無を指定できる。

- 漢字モードとして出力
MODE=K
- ノンインパクトモードとして出力
MODE=AN
- 行単位での文字セットイメージ（CHARS 指定）の変更の有無
OPTCDJ={YES | NO}
- 印刷データに指定されている印刷制御文字の種別
RECFM={A | M | N}

(4) 注意事項

各設定が実際の帳票に有効となるかどうかについては、PDE 側の設定内容に依存する。詳細については、マニュアル「PRINT DATA EXCHANGE for Open」または PDE のドキュメントを参照のこと。

(5) SYSOUT 出力の書式属性指定機能

スプールに出力するファイルに対する書式属性は設定ファイル (SysoutClass パラメータ) およびジョブ定義 XML ファイル (DD 要素および JOBPARM 要素) で指定できる。

スプールに出力するファイルに対する書式属性の指定を次の表に示す。

表 3-22 スプールに出力するファイルに対する書式属性

オペランドまたは属性	書式属性	SysoutClass パラメータ	DD 要素※ ¹	JOBPARM 要素※ ²
CHARS	文字セットイメージ名	○	○	—
CMT	注釈	—	○	—
COPIES	複写回数	○	○	—
COPYG	グループ複写回数	○	○	—
DEST	出力先識別子	○	○	—
FCB	イメージ名	○	○	○
FCDFLT	デフォルト機能キャラクタイメージ名	○	○	—
FORMS	書式番号	○	○	○
HOLD	出力保留	○	○	—
MODE	プリンタモード	○	○	—
OPTCDJ	行単位で文字配列テーブルを切り替える指定	—	○	—
OVERC	適用枚数	○	○	—
OVERLAY	イメージ名	○	○	—
RECFM	制御文字の指定	—	○	—
ROOM	仕分け番号	—	—	○
SYSOUT	クラス名	○	○	—

(凡例)

- ：指定できる。
- ：指定できない。

注※1 ファイル別に書式属性を指定する場合に指定する。

注※2 DD 要素に指定がない場合の仮定値を指定する。

印刷属性の優先順位を次に示す。

1. DD 要素

2. JOBPARM 要素

3. SysoutClass パラメータ

SysoutClass パラメータにも指定がない場合、PDE の仮定値に従う。

3.11.7 レベル 1 外字名指定

PDE での SYSOUT データファイル出力時に外字のフォントを選択したい場合、SYSOUT データファイルを定義する DD 要素の GAIJ11 属性でレベル 1 外字名を指定する。詳細については、マニュアル「PRINT DATA EXCHANGE for Open」を参照のこと。

レベル 1 外字名は、設定ファイルの出力クラス、またはジョブ定義 XML ファイルの DD 要素の GAIJ11 属性で指定する。

設定ファイルの指定はジョブ定義 XML ファイルで指定がないときのデフォルト値の定義であり、ジョブ定義 XML ファイルでの指定が設定ファイルの指定よりも優先する。

3.11.8 0 バイトの SYSOUT ファイル転送機能

転送する SYSOUT ファイルの件数を管理し、必ず全件のファイルを転送する運用では、件数に合うようにファイルを送る必要がある。実際に転送するファイルがない場合は、0 バイトのファイルを送って件数を調整する。この運用に対応して、バッチジョブ実行システムでも 0 バイトの SYSOUT ファイルを転送できるように設定ファイルで指定できる。

SYSOUT ファイル転送機能の設定ファイルの詳細については、「8. 設定ファイル」の「SysoutClass (出力クラスの属性値定義)」の TRANSFER_EMPTY_FILE オペランドを参照のこと。

3.12 ジョブステップを並列実行する機能

3.12.1 JSPEC 機能

ジョブステップを並列実行する機能を JSPEC 機能と呼ぶ。

通常ジョブステップは、ジョブ定義 XML ファイルで定義した順に逐次実行される。

JSPEC 機能を使用することで、逐次実行する必要のない独立したジョブステップを並列に実行できる。これによって、ジョブの実行時間を短縮できる。

また、PREST 機能を使用して並列に実行するジョブステップ間で、データを受け渡すことができる。これによって、ジョブの実行時間を短縮できる。

PREST 機能については、「[3.12.2 PREST 機能](#)」を参照のこと。

(1) 並列実行するジョブステップの指定

ジョブ定義 XML ファイルの EXEC 要素に JSPEC = "YES" を指定することで、ジョブステップを並列実行できる。バッチジョブ実行システムは、EXEC 要素に JSPEC = "YES" を指定したジョブステップを起動したあとに、そのジョブステップの終了を待つことなく、後続のジョブステップを起動する。後続のジョブステップにも並列実行の指定がある場合には、さらに後続のジョブステップを起動する。並列実行の指定がないジョブステップを起動したあとに、並列実行しているすべてのジョブステップが終了するのを待つ。

JSPEC 機能を適用したジョブで、EXEC 要素に JSPEC = "YES" の指定があるジョブステップをサブステップと呼び、指定がないジョブステップをメインステップと呼ぶ。メインステップまたはサブステップを区別する必要がない場合には、単にジョブステップと呼ぶ。

メインステップおよびサブステップは、それぞれ別のプロセスとして起動する。

JSPEC 機能の適用の有無によるジョブステップの実行を次の図に示す。

図 3-23 JSPEC 機能の適用の有無によるジョブステップの実行

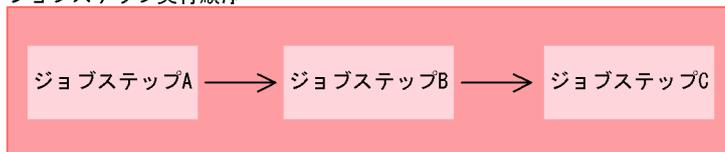
●JSPEC機能を適用しない場合

ジョブ定義XMLファイル

```
<STEP NAME="A" >
  <EXEC PGM="PRGA" />
</STEP>
<STEP NAME="B" >
  <EXEC PGM="PRGB" />
</STEP>
<STEP NAME="C" >
  <EXEC PGM="PRGC" />
</STEP>
```



ジョブステップ実行順序



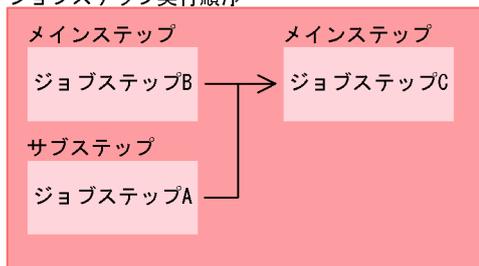
●JSPEC機能を適用する場合

ジョブ定義XMLファイル

```
<STEP NAME="A" >
  <EXEC PGM="PRGA" JSPEC="YES" />
</STEP>
<STEP NAME="B" >
  <EXEC PGM="PRGB" />
</STEP>
<STEP NAME="C" >
  <EXEC PGM="PRGC" />
</STEP>
```



ジョブステップ実行順序



(2) 並列実行するジョブステップの制限事項

並列実行できるジョブステップは、PREST ファイル以外には関連がないようにする必要がある。このため、次に示す制限がある。

PREST ファイルについては、「3.12.2 PREST 機能」を参照のこと。

並列実行するジョブステップの制限事項を次の表に示す。

表 3-23 並列実行するジョブステップの制限事項

項目	内容
<ul style="list-style-type: none"> • DISP=",DELETE" • DISP="NEW" • DISP="RNW" • DISP="OLD" • DISP="MOD" 	<ul style="list-style-type: none"> • PREST ファイルを除いて並列実行するジョブステップ間で使用する DSN の値が同一のファイル、ライブラリ、一時ファイル、一時 ISAM ファイル、および一致するパスを含むディレクトリに対しては、左記内容を指定しないこと。明示指定で左記内容が指定された場合、ジョブ定義 XML ファイルの解析時にエラーとなる。また、省略時に左記内容が仮定された場合は、ジョブ実行時にエラーとなる。 • DSN を省略した一時ファイルまたは一時 ISAM ファイルは、並列実行するステップ間では DISP 属性に PASS を指定してもファイルの受け渡しをしない。そのため、並列実行ステップ間で同じファイルにアクセスすることはない。したがって、並列実行ステップ間に DSN を省略した一時ファイルまたは一時 ISAM ファイルを 2 つ以上指定しても、エラーにならない。 • DSN の指定に同一ファイルの相対パス/絶対パスによる指定が混在していたり、シンボリックリンクを指定した場合、それらは同一ファイルと見なさないため、ジョブ定義 XML ファイルの作成時に注意する必要がある。
世代ファイル	サブステップには指定できない。指定した場合は、ジョブ定義 XML ファイルの解析時にエラーとなる。
一時ファイル	PREST ファイルを除いて並列実行するジョブステップ間で、一時ファイル識別名が同一の一時ファイルまたは一時 ISAM ファイルを指定できない。指定した場合は、ジョブ定義 XML ファイルの解析時にエラーとなる。
EXEC 要素の COND 属性	並列実行するジョブステップの終了コードを参照しても無効となる。
プログラムおよびスクリプト内で独自に使用するファイル	同一ファイルに対して入力以外の操作、または排他をするプログラムおよびシェルは並列実行できない。

(3) ジョブステップ異常終了時の処理

PREST ファイルを使用するジョブステップが異常終了した場合、相手のジョブステップが PREST ファイルのオープン待ち状態となり、終了しないことがある。この場合、JP1/AJS から手作業で待ち状態のジョブを強制終了する必要がある。

長時間待ち状態になることを回避するには、JP1/AJS の打ち切り時間を設定する方法と、設定ファイルの JOBPARM_JSPECAN パラメータを指定する方法とがある。

(a) JP1/AJS の打ち切り時間を設定する方法

JP1/AJS のジョブ定義で、PREST ファイルを使用するジョブに対して適切な打ち切り時間を設定する。

(b) JOBPARM_JSPECAN パラメータを指定する方法 (JSPECAN 機能)

設定ファイルの JOBPARM_JSPECAN パラメータに YES を指定する。並列実行しているジョブステップの 1 つのジョブステップが、次の表に示す原因で異常終了した場合、バッチジョブ実行システムがそのジョブステップの実行が終了していないジョブステップに SIGKILL を送り強制終了させる。

表 3-24 JSPECAN 機能がジョブステップを打ち切る条件とジョブステップの終了コード

JSPECAN 機能がジョブステップを打ち切る条件	ジョブステップの終了コード※1
アロケートエラーやプロセス生成エラーなど、UAP プロセス生成前のエラー	rc=バッチジョブ実行システムのエラー終了コード
シグナルによる異常終了	rc=128+N※2
JOB 要素の COND 属性による打ち切り	rc=UAP の終了コード

注※1 ジョブ間 PREST ファイルはこの方法を使用しても長時間待ち状態になることを回避できないため、JP1/AJS の打ち切り時間を設定すること。

注※2 シグナルで打ち切られた場合のシグナルコード。

(4) 並列実行するジョブステップのジョブログ

並列実行するジョブステップのジョブログと SYSMSG は時系列に出力されるため、並列実行するジョブステップの情報はジョブ定義 XML ファイルに指定した順とはならない。

出力例 (STEP2 から STEP5 を並列実行) を次に示す。

```

KAKC0090-I HitachiBatchJobExec started.
KAKC0400-I Parsing of config file "/home/bjex/bjex.conf" started.
KAKC0409-I Parsing of config file "/home/bjex/bjex.conf" ended. code=0
KAKC0724-I Jobid was assigned. Jobid=000001
KAKC0097-I SAMPLE1.STEP01 Step ended. rc=0 E-Time=0.316s C-Time=0.010s
KAKC0097-I SAMPLE1.STEP04 Step ended. rc=0 E-Time=0.423s C-Time=0.010s
KAKC0097-I SAMPLE1.STEP05 Step ended. rc=0 E-Time=0.388s C-Time=0.015s
KAKC0097-I SAMPLE1.STEP03 Step ended. rc=0 E-Time=0.998s C-Time=0.020s
KAKC0097-I SAMPLE1.STEP02 Step ended. rc=0 E-Time=1.428s C-Time=0.010s
KAKC0098-I SAMPLE1 Job ended. rc=0 E-Time=1.744s C-Time=0.040s MaxRSS=2456KB

```

ジョブステップの
終了順に出力

uCosminexus Batch Job Execution Server 08-00

[Information]

```

Jobid          : 000001
Spool Directory : /home/bjex/gyoumu/spool/000001/
Date           : 2010/02/16
Configure File  : /opt/hitachi/bjex/conf/bjex.conf
HostName       : io10288.3

```

[JP1 Parameter]

```

JP1JobName     : bjexec
JP1JobID       : 10038
JP1_USERNAME   : bjex
JP1UNCName     : io10288.3
JP1NBQQueueName : ¥¥io10288¥io10288.3
JP1Priority     : 1
AJSEXCID       : @A1999

```

***** BJEX MESSAGE *****

```

01:23:09 000001 KAKC0091-I SAMPLE1 Job started.
01:23:09 000001 KAKC0092-I SAMPLE1.STEP01 Step started.
01:23:10 000001 KAKC0097-I SAMPLE1.STEP01 Step ended. rc=0 E-Time=0.316s C-Time=0.010s
01:23:10 000001 KAKC0092-I SAMPLE1.STEP02 Step started.
01:23:10 000001 KAKC0092-I SAMPLE1.STEP03 Step started.
01:23:10 000001 KAKC0092-I SAMPLE1.STEP04 Step started.
01:23:10 000001 KAKC0092-I SAMPLE1.STEP05 Step started.
01:23:11 000001 KAKC0097-I SAMPLE1.STEP04 Step ended. rc=0 E-Time=0.423s C-Time=0.010s
01:23:11 000001 KAKC0097-I SAMPLE1.STEP05 Step ended. rc=0 E-Time=0.388s C-Time=0.015s
01:23:11 000001 KAKC0097-I SAMPLE1.STEP03 Step ended. rc=0 E-Time=0.998s C-Time=0.020s
01:23:11 000001 KAKC0097-I SAMPLE1.STEP02 Step ended. rc=0 E-Time=1.428s C-Time=0.010s
01:23:11 000001 KAKC0098-I SAMPLE1 Job ended. rc=0 E-Time=1.744s C-Time=0.040s MaxRSS=2456KB

```

ジョブステップの
終了順に出力

***** XML IMAGE *****

```

000001 <?xml version="1.0" encoding="Shift_JIS" ?>
000002 <HitachiBatchJobExec version="1.3" os="unix">
000003 <JOB NAME="SAMPLE1" MSGCLASS="A">
<!-- ステップの開始 -->

000004 <STEP NAME="STEP01">
000005 <EXEC PGM="/home/bjex/cobolpg/Write3" LANG="COBOL">
000006 </EXEC>
000007 <DD NAME="SYS009" TYPE="SYSOUT" SYSOUT="A" HOLD="NO" OVERLAY="TEMP" FCB="A4L6"
COPIES="5" />
000008 <DD NAME="SYS011" TYPE="FILE" DSN="FILE001" DISP="NEW,DELETE" />
000009 <DD NAME="SYS012" TYPE="FILE" DSN="FILE002" DISP="NEW,KEEP" />
000010 <DD NAME="SYS013" TYPE="FILE" DSN="FILE003" DISP="NEW,KEEP" />
000011 <DD NAME="SYS014" TYPE="FILE" DSN="FILE004" DISP="NEW,KEEP" />
000012 <DD NAME="SYS015" TYPE="FILE" DSN="FILE005" DISP="NEW,KEEP" />
000013 <DD NAME="SYS020" TYPE="FILE" DSN="FILE00A" DISP="NEW,KEEP" />

```

```

000014 </STEP>
000015 <STEP NAME="STEP02">
000016 <EXEC PGM="/home/bjex/cobolpg/Write3" LANG="COBOL" JSPEC="YES">
000017 </EXEC>
000018 <DD NAME="SYS009" TYPE="SYSOUT" SYSOUT="A" HOLD="NO" OVERLAY="TEMP" />
000019 <DD NAME="SYS010" TYPE="FILE" DSN="FILE002" DISP="SHR,KEEP" />
000020 <DD NAME="SYS020" TYPE="FILE" DSN="FILE00B" DISP="NEW,KEEP" />
000021 </STEP>
000022 <STEP NAME="STEP03">
000023 <EXEC PGM="/home/bjex/cobolpg/Write3" LANG="COBOL" JSPEC="YES">
000024 </EXEC>
000025 <DD NAME="SYS009" TYPE="SYSOUT" SYSOUT="A" HOLD="NO" OVERLAY="TEMP" />
000026 <DD NAME="SYS010" TYPE="FILE" DSN="FILE003" DISP="SHR,KEEP" />
000027 <DD NAME="SYS020" TYPE="FILE" DSN="FILE00C" DISP="NEW,KEEP" />
000028 </STEP>
000029 <STEP NAME="STEP04">
000030 <EXEC PGM="/home/bjex/cobolpg/Write3" LANG="COBOL" JSPEC="YES">
000031 </EXEC>
000032 <DD NAME="SYS009" TYPE="SYSOUT" SYSOUT="A" HOLD="NO" OVERLAY="TEMP" />
000033 <DD NAME="SYS010" TYPE="FILE" DSN="FILE004" DISP="SHR,KEEP" />
000034 <DD NAME="SYS020" TYPE="FILE" DSN="FILE00D" DISP="NEW,KEEP" />
000035 </STEP>
000036 <STEP NAME="STEP05">
000037 <EXEC PGM="/home/bjex/cobolpg/Write3" LANG="COBOL" JSPEC="YES">
000038 </EXEC>
000039 <DD NAME="SYS009" TYPE="SYSOUT" SYSOUT="A" HOLD="NO" OVERLAY="TEMP" />
000040 <DD NAME="SYS010" TYPE="FILE" DSN="FILE005" DISP="SHR,KEEP" />
000041 <DD NAME="SYS020" TYPE="FILE" DSN="FILE00E" DISP="NEW,KEEP" />
000042 </STEP>
000043 </JOB>
000044 </HitachiBatchJobExec>

***** SYSTEM MESSAGE *****
01:23:09 000001 KAKC0900-I Parsing of xml file "/home/bjex/jcl/3_15_4_samplejcl.txt" started.
01:23:09 000001 KAKC0909-I Parsing of xml file "/home/bjex/jcl/3_15_4_samplejcl.txt" ended. code=0
01:23:09 000001 KAKC0707-I MSGCLASS is "A".
01:23:09 000001 KAKC1600-I SAMPLE1 Job allocated.
01:23:10 000001 KAKC1601-I SAMPLE1.STEP01 Step allocated.
01:23:10 000001 KAKC1602-I SYS009, , /home/bjex/gyoumu/spool/000001/001_STEP01_001_SYS009
01:23:10 000001 KAKC1602-I SYS011, NEW, /home/bjex/perm/FILE001
01:23:10 000001 KAKC1602-I SYS012, NEW, /home/bjex/perm/FILE002
01:23:10 000001 KAKC1602-I SYS013, NEW, /home/bjex/perm/FILE003
01:23:10 000001 KAKC1602-I SYS014, NEW, /home/bjex/perm/FILE004
01:23:10 000001 KAKC1602-I SYS015, NEW, /home/bjex/perm/FILE005
01:23:10 000001 KAKC1602-I SYS020, NEW, /home/bjex/perm/FILE00A
01:23:10 000001 KAKC1603-I SYS009, , /home/bjex/gyoumu/spool/000001/001_STEP01_001_SYS009
01:23:10 000001 KAKC1603-I SYS011, DELETE, /home/bjex/perm/FILE001
01:23:10 000001 KAKC1604-I /home/bjex/perm/FILE001 is deleted.
01:23:10 000001 KAKC1603-I SYS012, KEEP, /home/bjex/perm/FILE002
01:23:10 000001 KAKC1603-I SYS013, KEEP, /home/bjex/perm/FILE003
01:23:10 000001 KAKC1603-I SYS014, KEEP, /home/bjex/perm/FILE004
01:23:10 000001 KAKC1603-I SYS015, KEEP, /home/bjex/perm/FILE005
01:23:10 000001 KAKC1603-I SYS020, KEEP, /home/bjex/perm/FILE00A
01:23:10 000001 KAKC1601-I SAMPLE1.STEP02 Step allocated.
01:23:10 000001 KAKC1602-I SYS009, , /home/bjex/gyoumu/spool/000001/002_STEP02_001_SYS009
01:23:10 000001 KAKC1602-I SYS010, SHR, /home/bjex/perm/FILE002
01:23:10 000001 KAKC1602-I SYS020, NEW, /home/bjex/perm/FILE00B
01:23:10 000001 KAKC1601-I SAMPLE1.STEP03 Step allocated.
01:23:10 000001 KAKC1602-I SYS009, , /home/bjex/gyoumu/spool/000001/003_STEP03_001_SYS009
01:23:10 000001 KAKC1602-I SYS010, SHR, /home/bjex/perm/FILE003

```

```

01:23:10 000001 KAKC1602-I SYS020, NEW, /home/bjex/perm/FILE00C
01:23:10 000001 KAKC1601-I SAMPLE1, STEP04 Step allocated.
01:23:10 000001 KAKC1602-I SYS009, . /home/bjex/gyoumu/spool/000001/004_STEP04_001_SYS009
01:23:10 000001 KAKC1602-I SYS010, SHR, /home/bjex/perm/FILE004
01:23:10 000001 KAKC1602-I SYS020, NEW, /home/bjex/perm/FILE00D
01:23:11 000001 KAKC1601-I SAMPLE1, STEP05 Step allocated.
01:23:11 000001 KAKC1602-I SYS009, . /home/bjex/gyoumu/spool/000001/005_STEP05_001_SYS009
01:23:11 000001 KAKC1602-I SYS010, SHR, /home/bjex/perm/FILE005
01:23:11 000001 KAKC1602-I SYS020, NEW, /home/bjex/perm/FILE00E
01:23:11 000001 KAKC1603-I SYS009, . /home/bjex/gyoumu/spool/000001/004_STEP04_001_SYS009
01:23:11 000001 KAKC1603-I SYS010, KEEP, /home/bjex/perm/FILE004
01:23:11 000001 KAKC1603-I SYS020, KEEP, /home/bjex/perm/FILE00D
01:23:11 000001 KAKC1603-I SYS009, . /home/bjex/gyoumu/spool/000001/005_STEP05_001_SYS009
01:23:11 000001 KAKC1603-I SYS010, KEEP, /home/bjex/perm/FILE005
01:23:11 000001 KAKC1603-I SYS020, KEEP, /home/bjex/perm/FILE00E
01:23:11 000001 KAKC1603-I SYS009, . /home/bjex/gyoumu/spool/000001/003_STEP03_001_SYS009
01:23:11 000001 KAKC1603-I SYS010, KEEP, /home/bjex/perm/FILE003
01:23:11 000001 KAKC1603-I SYS020, KEEP, /home/bjex/perm/FILE00C
01:23:11 000001 KAKC1603-I SYS009, . /home/bjex/gyoumu/spool/000001/002_STEP02_001_SYS009
01:23:11 000001 KAKC1603-I SYS010, KEEP, /home/bjex/perm/FILE002
01:23:11 000001 KAKC1603-I SYS020, KEEP, /home/bjex/perm/FILE00B
-----
***** JOBSTEP OUTPUT *****
KAKC0719-I STEP 001, STEP01, STDERR

KAKC0719-I STEP 002, STEP02, STDERR

KAKC0719-I STEP 003, STEP03, STDERR

KAKC0719-I STEP 004, STEP04, STDERR

KAKC0719-I STEP 005, STEP05, STDERR

KAKC0099-I HitachiBatchJobExec ended. rc=0

```

↑
ジョブステップの終了順に出力

(5) bjexlsjob コマンドでの表示

並列実行するジョブステップの SYSOUT データファイルを bjexlsjob コマンドで表示する場合、ジョブ定義 XML ファイルの指定順ではなく、ジョブステップの終了順に表示する。また、ステップ番号も実行が完了した順番に付与する。

(6) SETMAXRC 要素に関する注意事項

- JSPEC 機能を用いてジョブステップを並列実行する場合、SETMAXRC 要素はメインステップの直後に指定すること。複数のサブステップの直後に SETMAXRC 要素を指定した場合、最終的にどの SETMAXRC 要素の指定が有効になるかはサブステップの終了順序に依存し、最後に完了したサブステップの直後に指定した SETMAXRC 要素が有効になる。
- リターンコードの最大値がどのように更新されたのかを確認する場合、JOBLOG に出力される KAKC1909-I メッセージを調べること。

(7) DISP 属性に PASS を指定する場合の注意事項

- サブステップの DD 要素の DISP 属性に PASS を指定した場合、そのファイルを後続ステップへ引き継ぐことはできない。サブステップで指定したファイルを受け取ろうとした場合、ほかのメインステップから引き継がれる。引き継ぐことのできるメインステップが存在しなければ、ジョブ実行時にエラーとなる。

- DSN を省略した一時ファイル、一時 ISAM ファイルの DISP 属性に PASS を指定して、並列実行するステップ間で受け渡すことはできない。

3.12.2 PREST 機能

ジョブおよびジョブステップ間で使用するファイルに、名前付きパイプ (FIFO) が指定できる。この機能を PREST 機能と呼び、名前付きパイプを PREST ファイルと呼ぶ。

PREST ファイルを使用することで、逐次実行していたジョブおよびジョブステップを並列に実行することが可能となり、処理時間を短縮できる。

PREST ファイルはジョブ定義 XML ファイルの DD 要素で指定することで、ジョブ実行時にジョブコントローラによって一時ファイルディレクトリに自動生成され、プログラムで利用可能となる。PREST ファイルの名称は、一般のファイルと同様に環境変数に設定する。ジョブコントローラが生成した PREST ファイルは、ジョブ終了時に PREST ファイルを利用するジョブがなくなった時点で、ジョブコントローラが自動的に消去する。

PREST ファイルには、ジョブ間 PREST ファイルおよびジョブステップ間 PREST ファイルの 2 種類がある。ジョブ間 PREST ファイルを使用する場合は、設定ファイルの LOCK_OPTION パラメータで ENABLE を指定し、バッチジョブ実行システムの排他制御機能を有効にする必要がある。

(1) ジョブ間 PREST ファイル

ジョブ間 PREST ファイルは、ジョブ間で PREST ファイルを使用する場合に使用する。

ジョブ間 PREST ファイルは、同一ジョブ内で並列実行するジョブステップ間での使用もできるが、ジョブ間での使用を推奨する。

ジョブ間 PREST ファイルは、ジョブ定義 XML ファイルの DD 要素で TYPE = "PREST" を指定した場合に作成される。この場合、DSN 属性にジョブ間 PREST ファイルを識別するための PREST 識別名を指定する。また、DISP 属性は省略する。DISP 属性を指定した場合、DISP = "SHR,KEEP,KEEP" を仮定する。

設定ファイルの LOCK_OPTION パラメータに DISABLE を指定している場合、TYPE = "PREST" を指定すると、ジョブ定義 XML ファイルの解析時にエラーとなる。

ジョブ定義 XML ファイルの DD 要素での指定例を次に示す。

```
<DD NAME="A" DSN="X" TYPE="PREST"/>
```

- ジョブ間 PREST ファイルの名称

ジョブ間 PREST ファイルは、ジョブコントローラによってバッチジョブ実行システムの一時ファイルディレクトリに、次に示す名称で生成される。

```
PREST_ PREST識別名
```

この名称がプラットフォームの上限値を超えない範囲で運用する必要がある。上限を超えた場合、ジョブ定義 XML ファイルの解析時にはエラーとならないで、実行時の PREST ファイル割り当て時にエラーとなる。また、新規割り当て時に同一名のファイルがある場合、ファイルにアクセス権がなければプログラムはエラーで終了する（この場合、名前付きパイプを削除するコマンドなどで削除するまでは、この PREST ファイルを使用するジョブは実行できない）。

- ジョブ間 PREST ファイルのパーミッション

ジョブ間 PREST ファイルのパーミッションは、読み込みを行うジョブを実行するユーザに対して読み込み権限、書き込みを行うジョブを実行するユーザに対して書き込み権限がそれぞれ必要である。パーミッションは、次に示す優先順位に従って決定する（優先順位は 1 が最も高い）。

1. ジョブ定義 XML ファイルの DD 要素の FPERM 属性
2. ジョブ定義 XML ファイルの PERMISSION 要素の FILE 属性
3. 設定ファイルの PERMISSION_FILE パラメータ（省略時は 0600）

ただし、上記の要素・属性に 0600 を指定し、同一ユーザのジョブで使用することを推奨する。

(2) ジョブステップ間 PREST ファイル

ジョブステップ間 PREST ファイルは、並列実行するジョブステップ間で PREST ファイルを使用する場合に使用する。

ジョブステップ間 PREST ファイルは、ジョブ定義 XML ファイルの DD 要素で TYPE="TEMPPREST" を指定した場合に生成される。この場合、DSN 属性として一時ファイル識別名を指定する。また、DISP 属性は省略する。DISP 属性を指定した場合、DISP="SHR,KEEP,KEEP" を仮定する。

ジョブ定義 XML ファイルの DD 要素での指定例を次に示す。

```
<DD NAME="A" DSN="X" TYPE="TEMPPREST"/>
```

- ジョブステップ間 PREST ファイルの名称

ジョブステップ間 PREST ファイルは、ジョブコントローラによってバッチジョブ実行システムの一時ファイルディレクトリに、次に示す名称で生成される。

```
TEMPPREST_ジョブ識別子_ジョブ名_一時ファイル識別名
```

この名称がプラットフォームの上限値を超えない範囲で運用する必要がある。上限を超えた場合、ジョブ定義 XML ファイルの解析時にはエラーとならないで、実行時の PREST ファイル割り当て時にエラーとなる。また、新規割り当て時に同一名のファイルがある場合、そのファイルをそのまま使用するが、ファイルにアクセス権がなければプログラムはエラーで終了する（この場合、名前付きパイプを削除するコマンドなどで削除するまでは、この PREST ファイルを使用するジョブは実行ができない）。

- ジョブステップ間 PREST ファイルのパーミッション

ジョブステップ間 PREST ファイルのパーミッションは 0600 で作成する。

ジョブステップ間 PREST ファイルのパーミッションは変更できない。

(3) ジョブ間 PREST ファイルの使用状況の表示

ジョブ間 PREST ファイルの使用状況は、bjexlslock コマンドで表示できる。

ジョブ間 PREST ファイルを使用するジョブは、PREST ファイルを利用してデータを受け渡しする相手のジョブがない場合に待ち状態になる。このような状態のジョブを発見するために bjexlslock コマンドを使用する。

次に示す内容で bjexlslock コマンドを実行すると、待ち状態になっているジョブ間 PREST ファイルの表示ができる。

```
bjexlslock -l -t PREST
```

表示されたジョブ間 PREST ファイルの名称とジョブ識別子を基に、ジョブやジョブスケジュールの定義を見直し、待ち状態の原因を調査する。

(4) PREST ファイルを使用できるプログラムとファイル

PREST ファイルは順次入力または順次出力だけを実行するプログラム間に適用できる。入力側のプログラムは、出力側のプログラムが PREST ファイルに出力したデータを、すべて入力する必要がある。また、複数の PREST ファイルを使用する場合、ファイルのオープン順序などがプログラム間でデッドロックまたは長時間の待ち状態とならないようにする必要がある。PREST ファイルはすべてのプログラムで適用できるわけではないため、PREST ファイルを使用する場合には、プログラムの処理が名前付きパイプに対応している必要がある。

バッチジョブ実行システムでは、PREST 機能に対応した SORT EE、COBOL またはファイル操作ユーティリティ以外で PREST ファイルを使用した場合の動作は保証しない。ファイルのオープン順序が特定できないプログラムでは、PREST ファイルの使用は 1 個にとどめることを推奨する。

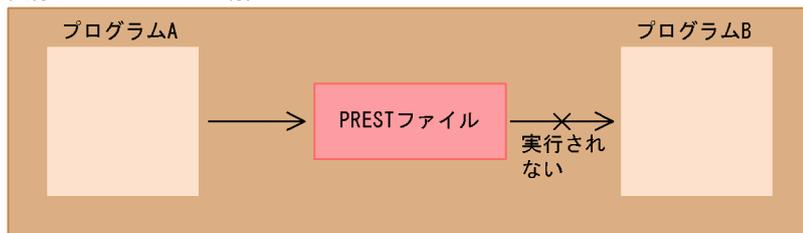
また、SORT EE、COBOL またはファイル操作ユーティリティで利用できる PREST ファイルの形式は、順編成だけである。

(a) プログラムが長時間待ち状態となる例

プログラムが長時間待ち状態となる例を次に示す。

- 例 1

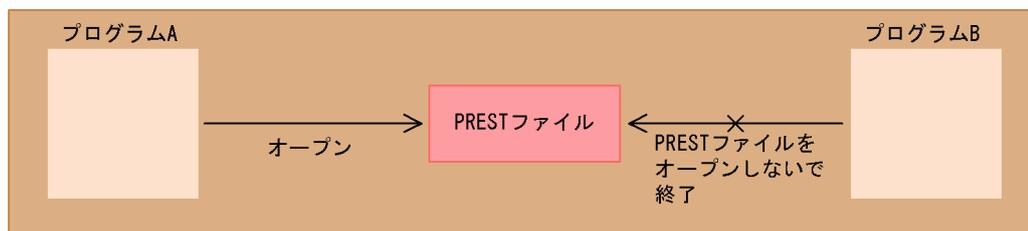
PRESTファイルで連携する相手プログラム(プログラムB)が実行されない場合



プログラムAは、プログラムBの起動を待ち続ける。

• 例 2

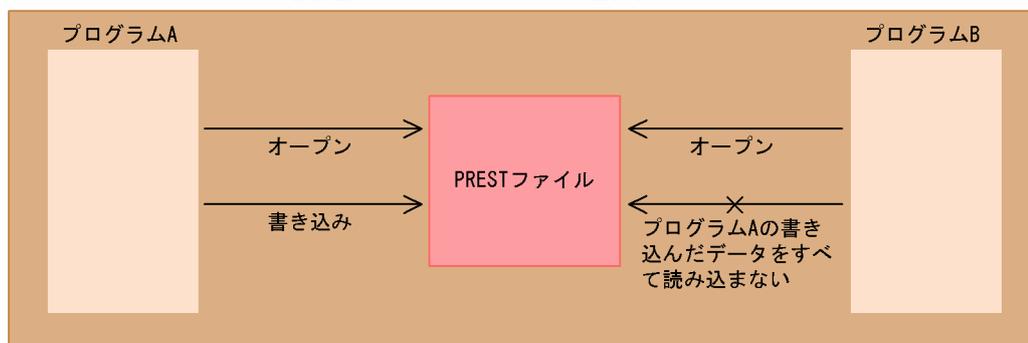
PRESTファイルをオープンしないで、プログラムBが終了した場合



プログラムAは、プログラムBからPRESTファイルがオープンされることを待ち続ける。

• 例 3

プログラムBがプログラムAの書き込んだデータをすべて読み込まない場合



プログラムAはバッファが満杯になった時点で、プログラムBがPRESTファイルからデータを読み込むのを待ち続ける。

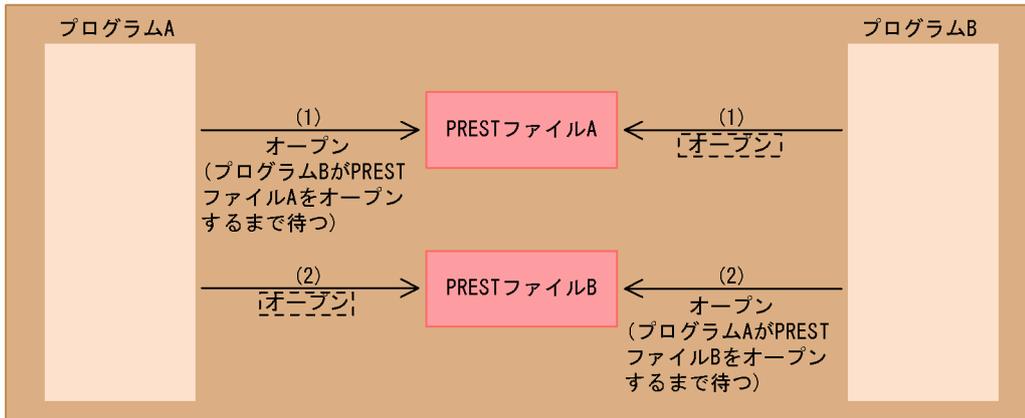
(b) プログラムがデッドロックになる例

プログラムがデッドロックになる例を次に示す。

• 例 1

次に示す場合

- (1) プログラムAはPRESTファイルAをオープンした場合に、プログラムBがPRESTファイルAをオープンするまで待たされる
- (2) プログラムBはPRESTファイルBをオープンした場合に、プログラムAがPRESTファイルBをオープンするまで待たされる。



結果として互いに待ち状態になりデッドロックとなる。

(凡例)

[]: 実行前の内容を示す。

• 例 2

プログラムA、Bともに読み込みモードでPRESTファイルを開く場合



書き込みモードでオープンするプログラムが存在しないため、結果として互いに待ち状態になりデッドロックとなる。

(5) PREST 機能に関する注意事項

- ジョブ間 PREST ファイルを使用する場合、ジョブ間 PREST ファイルを使用するジョブが同一システム内で同時に実行する運用となるように配慮する必要がある。片方だけしか実行しない場合には、ジョブは待ち状態になる。待ち状態になった場合には、JP1/AJS から手作業でジョブを強制終了する必要がある。また、JP1/AJS でジョブ実行時間の打ち切り時間を設定して、長時間待ち状態になることを回避する必要がある。
- PREST ファイルを使用する場合、入力側のプログラムは、出力側が出力したデータをすべて読み込む必要がある。データの入力が途中で中止されると、PREST ファイルのバッファが出力データで満杯と

なった時点で、出力側のプログラムは待ち状態になる。また、入力側が PREST ファイルをクローズしてから出力側のプログラムがデータを出力すると、出力側のプログラムが異常終了する。

- PREST ファイルは入力側と出力側の関係が 1 対 1 となっている必要があり、1 対多または多対 1 での使用はできない。
1 対 1 でない場合、ジョブ定義 XML ファイルの解析時または PREST ファイル割り当て時にエラーとなる。
- ジョブステップ間 PREST ファイルは、並列実行するジョブステップ間でだけ使用できる。並列実行しないジョブステップ間で、ジョブステップ間 PREST ファイルの指定が 1 対 1 でない場合、ジョブ定義 XML ファイルの解析時にエラーとなる。また、並列実行しないジョブステップにジョブステップ間 PREST ファイルを指定した場合、ジョブ定義 XML ファイルの解析時にエラーとなる。
- バッチジョブ実行システムに障害があった場合、使用されない PREST ファイルが残る可能性がある。この場合、一時ファイルディレクトリを調査して使用されていない PREST ファイル（名前付きパイプ）を手作業で削除する必要がある。AIX では OS の unlink コマンドで削除する。
- PREST 機能を使用するジョブステップが JOB 要素または EXEC 要素の COND 属性で実行されない場合、相手のジョブまたはジョブステップは、待ち状態になる。このため、PREST 機能を使用するジョブの JOB 要素または EXEC 要素に COND 属性を指定する場合は、注意が必要である。
- PREST ファイルはファイル連結しないこと。

3.13 ジョブ単位のファイルおよびディレクトリに対する排他/共用制御機能

排他/共用制御機能は、設定ファイルの LOCK_OPTION パラメータで ENABLE を指定することで使用可能となる。LOCK_OPTION パラメータで指定を省略、または DISABLE を指定した場合は、排他/共用制御機能は使用できない。

ジョブで指定した DD 要素の DISP 属性値に従った排他/共用制御を、ジョブ実行単位に行う。

排他/共用制御は、ジョブ開始時にジョブで指定したすべての DD 要素の指定を確認し、一括して確保する。

排他/共用制御の解除は、設定ファイルの LOCK_RELEASE パラメータの指定によって次の契機で行われる。

- LOCK_RELEASE パラメータを省略または JOB を指定した場合
ジョブ終了時に一括して解除する。
- LOCK_RELEASE パラメータで STEP を指定した場合
ジョブステップ終了時に以降のジョブステップで指定されていないファイルおよびディレクトリの排他/共用制御を解除する。

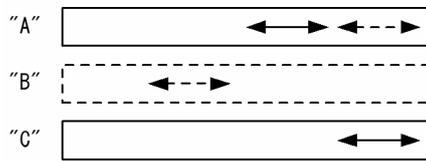
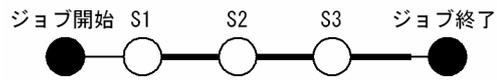
ジョブ中の DD 要素で指定したファイルおよびディレクトリに、1 つでも排他/共用制御によるリソースに対するロック確保待ちが発生した場合、すべてのリソースに対するロック確保を行わないで、すべてのリソースに対するロック確保が可能となった時点でロック確保をする。

- **共用制御**
ファイルの内容が参照されるだけで変更されることがない場合、バッチジョブ実行システムはこのファイルを複数のジョブ間で共用できる状態とする。
- **排他制御**
ファイル中のデータを変更、追加、または削除する場合、ジョブが該当ファイルを使い終わるまで、ほかのジョブの実行を待たせる。また、ほかのジョブが先に該当ファイルを使用している場合には、ほかのジョブが該当ファイルを使い終わるまで、ジョブの実行を待たせる。

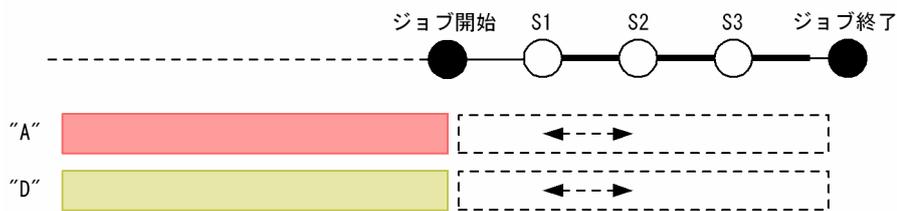
ジョブ単位のファイルおよびディレクトリに対する排他/共用制御を次の図に示す。

図 3-24 ジョブ単位のファイルおよびディレクトリに対する排他/共用制御 (LOCK_RELEASE JOB の場合)

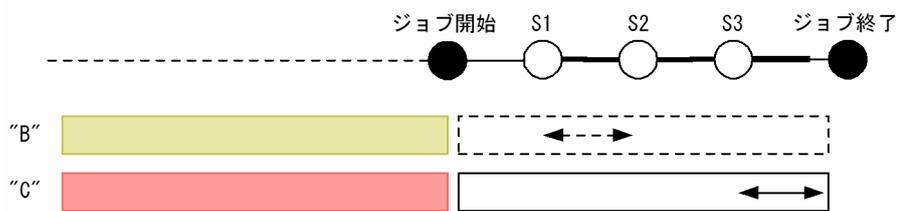
<JOB: JOB0001>



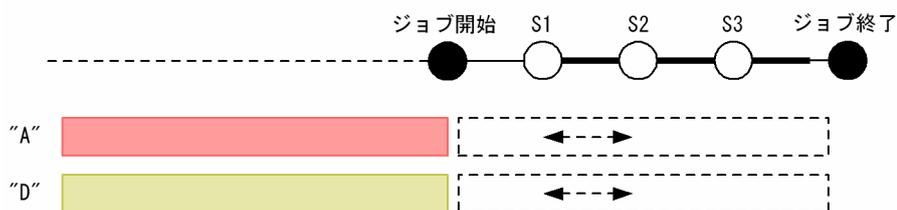
<JOB: JOB0002>



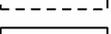
<JOB: JOB0003>



<JOB: JOB0004>



(凡例)

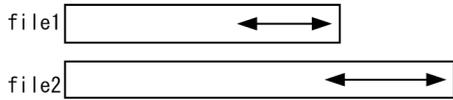
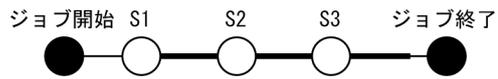
-  : ジョブの実行区間を示す。
-  : ジョブステップの実行区間を示す。
- "n" : 排他/共用対象ファイル・ディレクトリ名を示す。
-  : 共用要求のDD要素のスコープを示す。
-  : 排他要求のDD要素のスコープを示す。
- : 排他制御による実行待ち期間を示す。
-  : 排他/共用ロック確保待ち中を示す。
-  : ロック未確保 (ほかの排他/共用ロック確保待ちのため確保を延期している状態) を示す。
-  : 共用ロック確保区間を示す。
-  : 排他ロック確保区間を示す。

ジョブステップ終了時にファイルおよびディレクトリに対する排他/共用制御を解除する場合を次の図に示す。

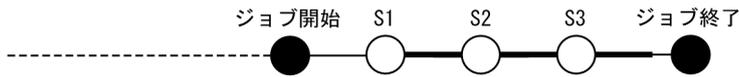
図 3-25 ジョブステップ終了時に排他/共用制御を解除する場合 (LOCK_RELEASE STEP の場合)

【通常ジョブ】

<JOB:JOB0001>

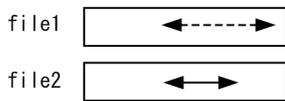
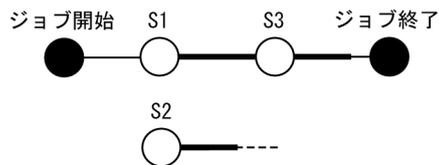


<JOB:JOB0002>

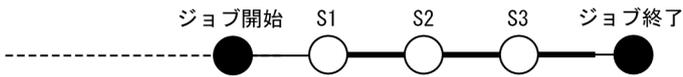


【JSPECジョブ】

<JOB:JOB0001>



<JOB:JOB0002>



(凡例)

- : ジョブの実行区間を示す。
- : ジョブステップの実行区間を示す。
- ◄-----► : 共用要求のDD要素の範囲を示す。
- ◄=====► : 排他要求のDD要素の範囲を示す。
- : 排他制御による実行待ち期間を示す。
- : 排他/共用ロック確保待ち中を示す。
- : ロック未確保(ほかの排他/共用ロック確保待ちのため確保を延期している状態)を示す。
- : 排他ロック確保区間を示す。

同一ファイルおよび同一ディレクトリに対して排他/共用要求をジョブ間で行った場合、次に示す表の状態となる。

表 3-25 排他/共用要求によるジョブの状態

先行ジョブの要求	後続ジョブの要求	
	共用要求 (SHR)	排他要求 (EXC)
共用要求 (SHR)	後続ジョブは共用要求のため資源確保ができ、実行できる。	後続ジョブはロック確保待ち。
排他要求 (EXC)	後続ジョブはロック確保待ち。	後続ジョブはロック確保待ち。

3.13.1 排他/共用制御の指定方法と適用範囲

ジョブで割り当てるファイルおよびディレクトリに対して排他/共用制御を指定する場合、DD 要素の DISP 属性で次の表に示す指定をすることで排他/共用要求指示となる。

表 3-26 DISP 属性値ごとの排他/共用要求

DISP 属性値	排他/共用※1, ※2
MOD	排他要求
NEW	排他要求
OLD	排他要求
RNW	排他要求
SHR	共用要求

注※1 DISP 属性値の指定で、第 2 または第 3 パラメータに DELETE を指定している場合 (DISP="SHR,DELETE", DISP="SHR,,DELETE" など) は、排他要求と見なす。

注※2 TYPE="PREST" を指定した場合、DISP 属性値に "SHR" 以外を指定しても "SHR" を指定 (共用要求) したと見なす。

指定例を次に示す。

- DISP 属性値に "OLD" を指定

```
<DD NAME="DD0001" TYPE="FILE" DSN="FILE1" DISP="OLD"/>
```

DISP 属性値に "OLD" を指定したファイル割り当て要求のため、排他要求となる。

- DISP 属性値に "SHR" を指定

```
<DD NAME="DD0002" TYPE="FILE" DSN="FILE2" DISP="SHR"/>
```

DISP 属性値に "SHR" を指定したファイル割り当て要求のため、共用要求となる。

- DISP 属性値に"MOD"を指定

```
<DD NAME="DD0003" TYPE="FILE" DSN="FILE3" DISP="MOD"/>
```

DISP 属性値に"MOD"を指定したファイル割り当て要求のため、排他要求となる。

- DISP 属性値を省略

```
<DD NAME="DD0004" TYPE="FILE" DSN="FILE3" />
```

DISP 属性値を省略したファイル割り当て要求のため、DISP 属性値のデフォルト値"NEW"を仮定し排他要求となる。

- TYPE 属性値に"PREST"を指定

```
<DD NAME="DD0005" TYPE="PREST" DSN="FILE4" DISP="OLD"/>
```

TYPE 属性値に"PREST"を指定した場合、DISP 属性値の指定に関係なく、すべて"SHR"を指定したと見なし、共用要求となる。

- DISP 属性値に"SHR"を指定し、第 2 パラメータに"DELETE"を指定

```
<DD NAME="DD0006" TYPE="FILE" DSN="FILE5" DISP="SHR,DELETE"/>
```

DISP 属性値で"SHR"を指定しているが、第 2 パラメータに"DELETE"を指定しているため、排他要求となる。

同一ジョブ内に、同一ファイルおよび同一ディレクトリを指定し、それぞれの DISP 属性値で排他または共用要求をしている場合、排他要求を優先する。

指定例を次に示す。

```
<?xml version="1.0" encode="SHIFT-JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
<JOB NAME="BJEX_JOB001">
<STEP NAME="STEP0001">
<EXEC PGM="PGM001"/>
<DD NAME="DD0001" TYPE="FILE" DSN="FILE1" DISP="OLD"/> ... (1)
</STEP>
<STEP NAME="STEP0002">
<EXEC PGM="PGM002"/>
<DD NAME="DD0002" TYPE="FILE" DSN="FILE1" DISP="SHR"/> ... (2)
</STEP>
</JOB?
</HitachiBatchJobExec>
```

(1)と(2)では同一ファイル名を指定しているが、DISP 属性値の指定が"OLD"と"SHR"で異なる指定となっている。この場合の排他/共用制御は、"排他要求"を優先するためファイル"FILE1"は排他要求となる。

排他/共用制御は、同一サーバ内で実行されるバッチジョブ実行システムから起動したジョブごとに行われる。ただし、DD 要素に指定したものの以外は、バッチジョブ実行システムの排他/共用制御の対象とはならない。

排他/共用制御では DD 要素の TYPE 属性に指定したファイルおよびディレクトリが対象となるが、ジョブ実行単位にジョブ間での排他/共用制御が必要ないものについては対象としない。

TYPE 属性値ごとの排他/共用制御の適用可否を次の表に示す。

表 3-27 TYPE 属性値ごとの排他/共用制御の適用可否

TYPE 属性値	排他/共用制御の適用可否	理由
DIR	○	ジョブ間で使用されるため。
DUMMY	×	実体がないため。
FILE	○	ジョブ間で使用されるため。
GDG	○※1	
LIB	○	
PREST	○※2	
DATA	×	
SYSOUT	×	ジョブ内で生成・削除するもので、ジョブ間で使用することはないため。
TEMP	×	
TEMPISAM	×	
TEMPPREST	×	
	×	

(凡例)

- ：排他/共用制御に適用できることを示す。
- ×：排他/共用制御に適用できないことを示す。

注※1 世代ファイルは、世代データ群単位の排他/共用制御となる。

注※2 ジョブ間で同時に実行する必要がある指定のため、排他要求を行っても共用要求となる。

3.13.2 ディレクトリ割り当て時の排他/共用要求の範囲

ディレクトリ割り当て時、ディレクトリ配下の全ファイルおよびディレクトリに対し、再帰的に排他/共用処理を行うかどうかを設定ファイルの LOCK_DIR_RECURSIVE パラメータで指定できる。

LOCK_DIR_RECURSIVE パラメータの指定内容による排他/共用処理の違いを次に示す。

(1) LOCK_DIR_RECURSIVE パラメータを省略または YES を指定した場合

DD 要素で指定したディレクトリを割り当て時、配下のファイルおよびディレクトリに対して再帰的に排他/共用制御を行う。

(a) 先行ジョブで割り当てているファイルの上位階層のディレクトリを後続ジョブで割り当てる場合

【先行ジョブ】

```
<DD NAME="DD0010" TYPE="FILE" DSN="/home/user/data/output/outputData1" DISP="OLD"/>
```

【後続ジョブ】

```
<DD NAME="DD0011" TYPE="DIR" DSN="/home/user/data" DISP="SHR"/>
```

先行ジョブでファイル"/home/user/data/output/outputData1"を排他要求で割り当てており、後続ジョブでは上位階層のディレクトリ"/home/user/data"を共用要求で割り当てる場合、ファイル"/home/user/data/output/outputData1"に対する共用要求のロック待ちとなる。

図 3-26 排他/共用制御の範囲 (LOCK_DIR_RECURSIVE パラメータを省略または YES を指定した場合で先行ジョブで割り当てているファイルの上位階層のディレクトリを後続ジョブで割り当てるとき)

ファイル構成	排他/共用制御の範囲	
	先行ジョブ	後続ジョブ
/homeディレクトリ		
└ /userディレクトリ		
└└ /dataディレクトリ		○
└└└ /inputディレクトリ		○
└└└└ inputData1		○
└└└ /outputディレクトリ		○
└└└└ outputData1	◎	○
└└ summary		○
└└ total		○

(凡例)

- ◎ : 排他要求
- : 共用要求

(b) 先行ジョブで割り当てているファイルと同一階層のディレクトリを後続ジョブで割り当てる場合

【先行ジョブ】

```
<DD NAME="DD0010" TYPE="FILE" DSN="/home/user/data/output/outputData1" DISP="OLD"/>
```

【後続ジョブ】

```
<DD NAME="DD0011" TYPE="DIR" DSN="/home/user/data/output" DISP="SHR"/>
```

先行ジョブでファイル"/home/user/data/output/outputData1"を排他要求で割り当てており、後続ジョブでは同一階層のディレクトリ"/home/user/data/output"を共用要求で割り当てる場合、ファイル"/home/user/data/output/outputData1"に対する共用要求のロック待ちとなる。

図 3-27 排他/共用制御の範囲 (LOCK_DIR_RECURSIVE パラメータを省略または YES を指定した場合で先行ジョブで割り当てているファイルと同一階層のディレクトリを後続ジョブで割り当てるとき)

ファイル構成	排他/共用制御の範囲	
	先行ジョブ	後続ジョブ
<pre> /homeディレクトリ ├─ /userディレクトリ │ └─ /dataディレクトリ │ ├── /inputディレクトリ │ │ └─ inputData1 │ ├── /outputディレクトリ │ │ └─ outputData1 │ ├── summary │ └─ total </pre>	◎	○

(凡例)

- ◎ : 排他要求
- : 共用要求

(2) LOCK_DIR_RECURSIVE パラメータで NO を指定した場合

DD 要素で指定したディレクトリを割り当て時、ディレクトリおよびディレクトリ直下のファイルに対してだけ、排他/共用制御を行う。

(a) 先行ジョブで割り当てているファイルの上位階層のディレクトリを後続ジョブで割り当てる場合

【先行ジョブ】

```
<DD NAME="DD0010" TYPE="FILE" DSN="/home/user/data/output/outputData1" DISP="OLD"/>
```

【後続ジョブ】

```
<DD NAME="DD0011" TYPE="DIR" DSN="/home/user/data" DISP="SHR"/>
```

先行ジョブでファイル"/home/user/data/output/outputData1"を排他要求で割り当てており、後続ジョブでは上位階層のディレクトリ"/home/user/data"を共用要求で割り当てる場合、ディレクトリおよびディレクトリ直下のファイルに対してだけ排他/共用制御を行うため、先行ジョブで割り当てているファイルとのロック待ちは発生しない。

図 3-28 排他/共用制御の範囲 (LOCK_DIR_RECURSIVE パラメータで NO を指定した場合で
先行ジョブで割り当てているファイルの上位階層のディレクトリを後続ジョブで割り
当てるとき)

ファイル構成	排他/共用制御の範囲	
	先行ジョブ	後続ジョブ
/homeディレクトリ └─ /userディレクトリ └─ /dataディレクトリ │ └─ /inputディレクトリ │ │ └─ inputData1 │ └─ /outputディレクトリ │ │ └─ outputData1 └─ summary └─ total	◎	○ ○ ○

(凡例)

- ◎ : 排他要求
- : 共用要求

(b) 先行ジョブで割り当てているファイルと同一階層のディレクトリを後続ジョブで割 り当てる場合

【先行ジョブ】

```
<DD NAME="DD0010" TYPE="FILE" DSN="/home/user/data/output/outputData1" DISP="OLD"/>
```

【後続ジョブ】

```
<DD NAME="DD0011" TYPE="DIR" DSN="/home/user/data/output" DISP="SHR"/>
```

先行ジョブでファイル"/home/user/data/output/outputData1"を排他要求で割り当てており、後続ジョブでは同一階層のディレクトリ"/home/user/data/output"を共用要求で割り当てるとき、同一階層のファイル"/home/user/data/output/outputData1"に対する共用要求のロック待ちとなる。

図 3-29 排他/共用制御の範囲 (LOCK_DIR_RECURSIVE パラメータで NO を指定した場合で
先行ジョブで先行ジョブで割り当てているファイルと同一階層のディレクトリを後続
ジョブで割り当てるとき)

ファイル構成	排他/共用制御の範囲	
	先行ジョブ	後続ジョブ
<pre> /homeディレクトリ ├─ /userディレクトリ │ └─ /dataディレクトリ │ ├─ /inputディレクトリ │ └─ inputData1 │ ├─ /outputディレクトリ │ └─ outputData1 │ └─ summary │ └─ total </pre>	◎	○ ○

(凡例)

- ◎ : 排他要求
- : 共用要求

3.13.3 排他/共用制御による実行待ち状態の確認方法

(1) ファイルまたはディレクトリがロック確保待ちとなり、ジョブが実行待ち状態の場合

同一サーバ内で実行されるバッチジョブ実行システムから起動した複数のジョブが、DD 要素で指定したファイルおよびディレクトリの割り当て要求を行い、「表 3-25 排他/共用要求によるジョブの状態」で示す「後続ジョブはロック確保待ち」の状態となった場合、次に示す KAKC1650-W メッセージを出力する。

- stderr および JOBLOG に出力されるメッセージ

```
KAKC1650-W Waiting for locking a file.
```

- JP1 イベント発行によって JP1 イベントコンソールに出力されるメッセージ

```
KAKC1650-W ロック確保待ちが発生しました
```

また、ロック確保待ちが発生し、ロック確保待ちが解除された場合、次に示す KAKC1651-I メッセージを出力する。

- stderr および JOBLOG に出力されるメッセージ

```
KAKC1651-I Quit waiting for locking a file.
```

- JP1 イベント発行によって JP1 イベントコンソールに出力されるメッセージ

```
KAKC1651-I ロック確保待ちは解除されました
```

ロック確保待ちが長時間解除されない、またはロック確保待ちが多発する場合、ロックの競合を起こしているジョブおよびロックの情報を表示させることができる。

ロック確保状態ジョブとロック確保待ちジョブを表示するコマンドの指定例、および表示例を次に示す。

```
$ bjexlslock
JOBID STATUS ELAPSED PID JP1JobID JOBNAME
-----
000001 LOCK 00:10:16 21614 1001 JOB0001
000002 WAIT 00:05:09 42802 1002 JOB0002
```

上記で表示されたジョブ (JOB0001 および JOB0002) の資源確保状態を表示させる場合の表示コマンドの指定例、および表示例を次に示す。

```
$ bjexlslock -j 2
JOBID STATUS ELAPSED PID JP1JobID JOBNAME
-----
000002 WAIT 00:05:09 42802 1002 JOB0002
MODE STATUS LOCKJOB TYPE FILEPATH
-----
EXC WAIT 000001 FILE /opt/hitachi/bjex/bin/public/A
```

表示例のジョブ (JOB0001) が長時間ロック確保状態となっている場合、後続ジョブのロック確保待ちを解除し実行させる必要がある。この場合、ロック確保状態のジョブが処理を続行しロック解放を待つか、またはロック確保状態のジョブをキャンセルする必要がある。

同一資源に対するロック確保待ちが多発する場合、センタ管理者は同一資源を割り当てるジョブの実行計画を見直す必要がある。ロック確保待ち時間は、実行待ちが発生したジョブが出力した KAKC1650-W と KAKC1651-I メッセージの出力時間の差から求められる。

(2) ジョブ多重度によるジョブ実行待ち状態の場合

同一サーバ内で実行されるバッチジョブ実行システムから起動した複数のジョブで、ジョブ多重度制御によって実行待ち状態となった場合、次に示す KAKC1652-W メッセージを出力する。

- stderr および JOBLOG に出力されるメッセージ

```
KAKC1652-W Waiting for locking a semaphore.
```

- JP1 イベント発行によって JP1 イベントコンソールに出力されるメッセージ

```
KAKC1652-W ジョブ実行待ちが発生しました
```

実行待ちが発生し、実行待ちが解除された場合、次に示す KAKC1653-I メッセージを出力する。

- stderr および JOBLOG に出力されるメッセージ

```
KAKC1653-I Quit waiting for locking a semaphore.
```

- JP1 イベント発行によって JP1 イベントコンソールに出力されるメッセージ

```
KAKC1653-I ジョブ実行待ちは解除されました
```

(3) 排他資源管理テーブルの空き待ちとなり、ジョブが実行待ち状態の場合

ロック獲得中またはロック待ち中のファイル合計数が、設定ファイルの LOCK_MAXFILE で指定したファイル数を越えたことによって、排他資源管理テーブルの空き待ちが発生した場合、次に示す KAKC1654-W メッセージを出力する。

- stderr および JOBLOG に出力されるメッセージ

```
KAKC1654-W Waiting for lock resource control table.
```

- JP1 イベント発行によって JP1 イベントコンソールに出力されるメッセージ

```
KAKC1654-W 排他資源管理テーブルの空き待ちが発生しました
```

また、排他資源管理テーブルの空き待ちが発生した後、排他資源管理テーブルの空き待ちが解除された場合、次に示す KAKC1655-I メッセージを出力する。

- stderr および JOBLOG に出力されるメッセージ

```
KAKC1655-I Quit waiting for lock resource control table.
```

- JP1 イベント発行によって JP1 イベントコンソールに出力されるメッセージ

```
KAKC1655-I 排他資源管理テーブルの空き待ちが解除されました
```

排他資源管理テーブルの空き待ちが発生した場合、排他資源管理テーブルの空き待ちを起こしているジョブの情報を表示させることができる。

排他資源管理テーブルの空き待ちジョブを表示するコマンドの指定例、および表示例を次に示す。

```
$ bjexlslock
JOBID STATUS ELAPSED PID JP1JobID JOBNAME
-----
000084 LOCK 00:00:40 17575 JOB0003
000085 WAIT 00:00:32 17578 JOB0004
```

上記で表示されたジョブ (JOB0003 および JOB0004) の資源確保状態を表示させる場合の表示コマンドの指定例、および表示例を次に示す。

```
$ bjexlslock -j 000085
JOBID STATUS ELAPSED PID JP1JobID JOBNAME
-----
000085 WAIT 00:00:43 17578 JOB0004
MODE STATUS LOCKJOB TYPE FILEPATH
-----
EXC UNLK NONE FILE /opt/hitachi/bjex/bin/public/B
EXC UNLK NONE FILE /opt/hitachi/bjex/bin/public/C
```

表示例のジョブ (JOB0004) が長時間排他資源管理テーブルの空き待ち状態となっている場合、当該ジョブの実行を待つか、当該ジョブを強制終了する必要がある。

排他資源管理テーブルに対する空き待ちが多発する場合、センタ管理者は設定ファイル LOCK_MAXFILE で指定した排他資源管理テーブルのエントリ数を見直す必要がある。

3.13.4 排他/共用制御に関する注意事項

- 世代データ群に対する排他/共用制御は、世代データ群名にだけ行われ、世代ファイルに対しては行わない。
- 排他/共用制御では共用メモリおよびセマフォを使用する。バッチジョブ実行システムが使用する共用メモリおよびセマフォは、bjexpd の起動時に生成され、bjexpd の停止時に解放する。そのため、設定ファイルの設定値を変更した場合、変更後の値を有効にするためには bjexpd を停止し、その後 bjexpd を再起動する必要がある。
- 排他/共用制御では、DD 要素の DSN 属性値を絶対パスに変換し、排他/共用制御の対象を判別する。指定されたファイルまたはディレクトリの上位階層のディレクトリが存在している必要がある。
- DD 要素の DSN 属性値の指定で、ディレクトリ名またはファイル名にハードリンクが含まれている場合、排他/共用制御が正しく動作できない場合があるため指定しないこと。
- DD 要素で指定したファイルだけが排他/共用制御の対象となる。コマンドラインなどで DD 要素で与えられる環境変数を使用しないで指定したファイルは、該当する制御の対象とならないため、注意が必要である。
- 論理ホストを複数構築し、バッチジョブ実行システムを複数稼働している場合、排他/共用制御はバッチジョブ実行システムごとに行う。あるバッチジョブ実行システムから別のバッチジョブ実行システムの排他/共用制御の情報を参照することはできない。

3.14 実行中のジョブ情報表示機能

実行中のジョブ情報を表示する `bjexlsexjob` コマンドによって、コマンド投入時点で実行しているジョブの一覧や詳細を表示できる。

3.14.1 表示されるジョブ情報

表示されるジョブ情報は、全ジョブ一覧の簡易表示と特定ジョブの詳細表示に分けられ、表示コマンドのオプションで指定する。

- 簡易表示
実行中のジョブに対する一覧表示と CPU 使用時間表示があり、システム内で実行中のジョブ全体を表示・確認できる。
- 詳細表示
特定のジョブの実行中のプログラム名などの詳細情報を表示できる。

表示されるジョブ情報一覧を次の表に示す。

表 3-28 表示されるジョブ情報一覧

ジョブ情報	簡易表示		詳細表示
	実行中のジョブ一覧表示	実行中のジョブの CPU 使用時間表示	
JP1 のジョブ番号	○	○	○
JP1 の実行優先順位	—	—	○
ジョブ識別子	○	○	○
ジョブ名	○	○	○
ステップ名	○	○	○
プロシジャステップ名	○	—	○
ジョブ実行開始時刻	○	○	○
ステップ実行開始時刻	○	○	○
ジョブコントローラプロセスのプロセス番号	○	—	○
ジョブステッププロセスのプロセス番号	○	—	○
ジョブで使用した CPU 時間	—	○	○
ジョブステップで使用した CPU 時間	—	—	○
ジョブステッププロセスの優先度	—	○	○
ジョブステッププロセスの nice 値	—	○	○

ジョブ情報	簡易表示		詳細表示
	実行中のジョブ一覧表示	実行中のジョブの CPU 使用時間表示	
ジョブステッププロセスの実行プログラム名	—	—	○
ジョブ定義 XML ファイル名	—	—	○

(凡例)

- ：表示される情報
- ：表示されない情報

3.14.2 設定ファイルの指定

実行中のジョブ情報表示機能は、設定ファイルの JOB_STATUS_DISPLAY パラメータで ENABLE を指定することで使用できるようになる。また、実行中のジョブ情報は共有メモリで管理しており、同時に実行するジョブの多重度を LOCK_JOBMPLEX パラメータで設定する必要がある。

3.14.3 ジョブの多重度制御

実行中のジョブ情報表示機能使用時は、ジョブの多重度制御が行われる。

LOCK_JOBMPLEX パラメータで指定したジョブの多重度を超えた場合、新たなジョブの実行は待たされ、KAKC1652-W メッセージを出力する。

- stderr および JOBLOG に出力されるメッセージ

```
KAKC1652-W Waiting for locking a semaphore.
```

- JP1 イベント発行によって JP1 イベントコンソールに出力されるメッセージ

```
KAKC1652-W ジョブ実行待ちが発生しました
```

その後、実行中のジョブが終了し実行待ちが解除された場合、KAKC1653-I メッセージを出力し、ジョブ実行を開始する。

- stderr および JOBLOG に出力されるメッセージ

```
KAKC1653-I QUIT waiting for locking semaphore.
```

- JP1 イベント発行によって JP1 イベントコンソールに出力されるメッセージ

```
KAKC1653-I ジョブ実行待ちは解除されました
```

3.15 メッセージ制御機能

メッセージ制御機能は、JP1/IM - View の画面を使ってメッセージを出力し、応答を入力する機能である。

3.15.1 メッセージ制御機能の種類

メッセージの出力と応答の入力の方法には、次の3種類がある。

- COBOL2002 で作成した COBOL アプリケーションの使用
 - 32 ビットモード：AIX で使用できる。
 - 64 ビットモード：AIX, Linux で使用できる。
- bjexm_message コマンドの使用
- カタログドプロシジャの使用

(1) COBOL2002 で作成した COBOL アプリケーションの使用

従来 COBOL2002 で作成した COBOL アプリケーションから stdout/stdin を使用して入出力していたメッセージを、バッチジョブ実行システムでアプリケーションを実行した場合に、JP1/IM - View の画面を使って入出力できるように自動的に変更する方法である。

この方法を使用するためには、COBOL2002 で次に示す機能を使ったプログラムを作成する必要がある。

- DISPLAY UPON CONSOLE 文
- ACCEPT FROM CONSOLE 文
- STOP 文 (定数指定)
- CBLTYPE サービスルーチン

これらのプログラムの作成方法については、マニュアル「COBOL2002 使用の手引 手引編」を参照のこと。

この方法を使用することで、COBOL2002 で作成した COBOL アプリケーションからのメッセージは、バッチジョブ実行システムの設定ファイルで指定した JP1/IM - Manager に接続した JP1/IM - View から参照が可能となる。さらに、ACCEPT 文、STOP 文および CBLTYPE サービスルーチンについては、stdin から入力していた作業を、JP1/IM - View から入力できるようになり、従来のバッチ業務と同様の手順で運用ができるため、移行が容易になる。

(2) メッセージ出力コマンド (bjexm_message コマンド) の使用

メッセージ出力コマンド (bjexm_message コマンド) を使用することで JP1/IM - View の画面を利用してメッセージの出力と応答の入力ができる。bjexm_message コマンドは応答内容に従ってリターンコードを出力する。このリターンコードを判定することで応答に従った動作ができる。

(3) カタログドプロシジャの使用

次のカタログドプロシジャを使用することでJP1/IM - View の画面を利用してメッセージの出力と応答の入力ができる。カタログドプロシジャのジョブステップのリターンコードで応答内容を識別して、応答内容に従って実行するジョブステップを制御できる。

表 3-29 カタログドプロシジャの機能

カタログドプロシジャ名	機能
bjex_PAUSE	応答要求メッセージを出力することでジョブの実行を一時停止する。実行を再開するにはメッセージに任意の文字を応答する。
bjex_WRITE	メッセージを出力する。
bjex_WRITER	応答要求メッセージを出力し、応答入力待つ。応答メッセージに従ったリターンコードを出力してジョブステップは終了する。設定された応答以外を応答した場合はリターンコード 127 でジョブステップは終了する。
bjex_WRITERX	応答要求メッセージを出力し、応答入力待つ。応答メッセージに従ったリターンコードを出力してジョブステップは終了する。設定された応答以外を応答した場合は再度応答要求メッセージを出力する。

(a) bjex_PAUSE カタログドプロシジャ

注釈に指定したメッセージを出力して応答待ちになり、ジョブの実行を一時停止する。ジョブの実行を継続するには注釈のメッセージに応答する。

```
<CALL NAME="bjex_PAUSE" >  
  <SETSYMBOL NAME="MSG" VALUE="注釈" />  
</CALL>
```

注釈 ~<空白を除く任意文字列>((1~255 バイト))<<Job_is_paused.>>

出力するメッセージを指定する。

SETSYMBOL 要素の VALUE 属性で指定できる上限を超えるとジョブ定義 XML ファイルの解析エラーとなる。

注釈を省略する場合は、SETSYMBOL 要素全体を省略する。

空白を含む場合は、正しく動作しない。

ジョブステップのリターンコード

0：一時停止が解除された。

127：予期しない障害が発生した。

(b) bjex_WRITE カタログドプロシジャ

メッセージを出力し、ジョブを継続する。

```
<CALL NAME="bjex_WRITE" >
<SETSYMBOL NAME="MSG" VALUE="メッセージ" />
</CALL>
```

メッセージ ~<空白を除く任意文字列>((1~255 バイト))<<message>>

出力するメッセージを指定する。

SETSYMBOL 要素の VALUE 属性で指定できる上限を超えるとジョブ定義 XML ファイルの解析エラーとなる。

メッセージを省略する場合は、SETSYMBOL 要素全体を省略する。

空白を含む場合は、正しく動作しない。

ジョブステップのリターンコード

0: メッセージを出力した。

127: 予期しないエラーが発生した。

(c) bjex_WRITER カタログドプロシジャ

応答要求メッセージを出力して応答待ちになる。

ジョブの実行を継続するには応答 0 から応答 n のどれかのメッセージを応答する。ジョブステップは応答に対応したリターンコードを出力して終了する。例えば、応答 0 を応答するとリターンコードは 0 になり、応答 1 を応答するとリターンコードは 1 になり、応答 n を応答するとリターンコードは n となる。

定義した応答以外の応答をするとリターンコード 127 でジョブステップは終了する。

```
<CALL NAME="bjex_WRITER" >
  <SETSYMBOL NAME="MSG" VALUE="応答要求メッセージ" />
  <SETSYMBOL NAME="ANS" VALUE="応答0 応答1 … 応答n" />
</CALL>
```

応答要求メッセージ ~<空白を除く任意文字列>((1~255 バイト)) <<Key_in[yes,no,other-error]>>

出力するメッセージを指定する。

SETSYMBOL 要素の VALUE 属性で指定できる上限を超える場合、ジョブ定義 XML ファイルの解析エラーとなる。

応答要求メッセージを省略する場合は、SETSYMBOL 要素全体を省略する。

空白を含む場合は、空白以降は応答メッセージと解釈する。

応答 n ~<ASCII 文字列>((全体で 1~255 文字))<<yes no>>

入力する応答メッセージを指定する。応答はスペースで区切って最大 32 個指定できる。

応答を 33 個以上指定してもバッチジョブ実行システムはエラーにしないで 33 個以上の応答メッセージは無視する。

ASCII 文字以外を指定してもバッチジョブ実行システムはチェックしないが、ASCII 文字以外の応答を入力できない。すべての応答メッセージとそれを区切る空白文字の合計が SETSYMBOL 要素の

VALUE 属性で指定できる上限 (255 バイト) を超えた場合、ジョブ定義 XML ファイルの解析エラーとなる。

省略する場合は、SETSYMBOL 要素全体を省略する。

応答要求メッセージと応答 n は連結されて EXEC 要素の PARM 属性の値に置き換わる。ただし、全体の長さが PARM 属性の上限値-4 バイトを超えた場合、ジョブ定義 XML ファイルの解析エラーとなる。

ジョブステップのリターンコード

0：応答 0 を応答した。

n：応答 n を応答した。

127：予期しないエラーが発生した。または、定義された応答以外を応答した。

(d) bjex_WRITERX カタログドプロシジャ

応答要求メッセージを出力して応答待ちになる。

ジョブの実行を継続するには応答 0 から応答 n のどれかのメッセージを応答する。

ジョブステップは応答に対応したリターンコードを出力して終了する。例えば、応答 0 を応答するとリターンコードは 0 になり、応答 1 を応答するとリターンコードは 1 になり、応答 n を応答するとリターンコードは n となる。

定義した応答以外の応答をすると KAKC5208-E メッセージを出力して、再び応答要求メッセージを出力して応答待ちとなる。

```
<CALL NAME="bjex_WRITERX" >  
  <SETSYMBOL NAME="MSG" VALUE="応答要求メッセージ" />  
  <SETSYMBOL NAME="ANS" VALUE="応答0 応答1 … 応答n" />  
</CALL>
```

応答要求メッセージ ~<空白を除く任意文字列>((1~255 バイト))<<Key_in[yes,no]>>

出力するメッセージを指定する。

SETSYMBOL 要素の VALUE 属性で指定できる上限を超えるとジョブ定義 XML ファイルの解析エラーとなる。

応答要求メッセージを省略する場合は、SETSYMBOL 要素全体を省略する。

空白を含む場合は、空白以降は応答メッセージと解釈する。

応答 n ~<ASCII 文字列>((全体で 1~255 文字))<<yes no>>

入力する応答メッセージを指定する。応答はスペースで区切って最大 32 個指定できる。

応答を 33 個以上指定してもバッチジョブ実行システムはエラーにしないで 33 個以上の応答メッセージは無視する。

ASCII 文字以外を指定してもバッチジョブ実行システムはチェックしないが、ASCII 文字以外は応答を入力できない。すべての応答メッセージとそれを区切る空白文字の合計が SETSYMBOL 要素の VALUE 属性で指定できる上限 (255 バイト) を超えるた場合、ジョブ定義 XML ファイルの解析エラーとなる。

省略する場合は、SETSYMBOL 要素全体を省略する。

応答要求メッセージと応答 n は連結されて EXEC 要素の PARM 属性値に置き換わる。ただし、全体の長さが PARM 属性の上限-4 を超えた場合、ジョブ定義 XML ファイルの解析エラーとなる。

ジョブステップのリターンコード

0：応答 0 を応答した。

n：応答 n を応答した。

127：予期しないエラーが発生した。

(4) カタログドプロシジャの使用例

カタログドプロシジャを使用したジョブ定義 XML ファイルの例を次に示す。

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version = "1.4" os="unix" >
  <JOB NAME="TESTPROC">
    <STEP NAME="AASTEP01" >
      <CALL NAME="bjex_PAUSE" > <SETSYMBOL NAME="MSG" VALUE="This_step_paused._now!!" />
    </CALL>
  </STEP>
  <STEP NAME="AASTEP02" >
    <CALL NAME="bjex_WRITE" > <SETSYMBOL NAME="MSG" VALUE="Information_message" />
  </CALL>
  </STEP>
  <STEP NAME="AASTEP03" >
    <CALL NAME="bjex_WRITER" >
  <SETSYMBOL NAME="MSG" VALUE="ENTER:" />
  <SETSYMBOL NAME="ANS" VALUE="AA BB CC" />
  </CALL>
  </STEP>
  <STEP NAME="AASTEP04" >
    <CALL NAME="bjex_WRITERX" > <SETSYMBOL NAME="MSG" VALUE="XXENTER:" />
  <SETSYMBOL NAME="ANS" VALUE="AAX BBX CCX" /> </CALL>
  </STEP>
  <STEP NAME="AA" >
    <EXEC PGM="*" COND="(0, EQ, AASTEP03, STEP01)">
      <![CDATA[
!
echo "AASTEP03 rc not 0"
!
      ]]>
    </EXEC>
  </STEP>
  <STEP NAME="Ab" >
    <EXEC PGM="*" COND="(0, NE, AASTEP03, STEP01)">
      <![CDATA[
!
echo "AASTEP03 rc=0"
!
      ]]>
    </EXEC>
  </STEP>
```

(5) カタログドプロシジャの実行結果

カタログドプロシジャの使用例を実行した結果を次に示す。

```
***** BJEX MESSAGE *****
21:07:37 000041 KAKC0091-I TESTPROC Job started.
21:07:37 000041 KAKC0092-I TESTPROC.AASTEPO1.STEP01 Step started.
21:07:37 000041 KAKC5013-I This_step_paused._now!! (000041, 001, io10288.3, 1280923657, 0)
21:07:58 000041 KAKC5001-I (000041, 001, io10288.3, user01)
21:07:58 000041 KAKC0097-I TESTPROC.AASTEPO1.STEP01 Step ended. rc=0 E-Time=21.417s C-
Time=0.010s
21:07:58 000041 KAKC0092-I TESTPROC.AASTEPO2.STEP01 Step started.
21:07:59 000041 KAKC5012-I Information_message (000041, 002, io10288.3, 1280923679, 0)
21:07:59 000041 KAKC0097-I TESTPROC.AASTEPO2.STEP01 Step ended. rc=0 E-Time=0.407s C-
Time=0.020s
21:07:59 000041 KAKC0092-I TESTPROC.AASTEPO3.STEP01 Step started.
21:07:59 000041 KAKC5013-I ENTER: (000041, 003, io10288.3, 1280923679, 0)
21:08:38 000041 KAKC5001-I AA (000041, 003, io10288.3, user01)
21:08:38 000041 KAKC0097-I TESTPROC.AASTEPO3.STEP01 Step ended. rc=0 E-Time=39.425s C-
Time=0.030s
21:08:39 000041 KAKC0092-I TESTPROC.AASTEPO4.STEP01 Step started.
21:08:39 000041 KAKC5013-I XXENTER: (000041, 004, io10288.3, 1280923719, 0)
21:09:03 000041 KAKC5001-I CCX (000041, 004, io10288.3, user01)
21:09:03 000041 KAKC0097-I TESTPROC.AASTEPO4.STEP01 Step ended. rc=2 E-Time=24.415s C-
Time=0.030s
21:09:03 000041 KAKC1907-I TESTPROC.AA Step was skipped because the value of COND attribute of
EXEC element was met.
21:09:03 000041 KAKC0092-I TESTPROC.Ab Step started.
21:09:03 000041 KAKC0097-I TESTPROC.Ab Step ended. rc=0 E-Time=0.252s C-Time=0.020s
21:09:04 000041 KAKC0098-I TESTPROC Job ended. rc=2 E-Time=85.916s C-Time=0.110s MaxRSS=4084KB

***** XML IMAGE *****
000001 <?xml version="1.0" encoding="Shift-JIS" ?>
000002 <HitachiBatchJobExec version = "1.4" os="unix" >
000003 <JOB NAME="TESTPROC">
000004 <STEP NAME="AASTEPO1" >
000005 <CALL NAME="bjex_PAUSE" >
:
:
```

```
***** SYSTEM MESSAGE *****
:
(省略)
:
21:07:37 000041 KAKC0707-I MSGCLASS is "A".
21:07:37 000041 KAKC1600-I TESTPROC Job allocated.
21:07:37 000041 KAKC1601-I TESTPROC.AASTEPO1.STEP01 Step allocated.
21:07:37 000041 KAKC5013-I This_step_paused._now!! (000041, 001, io10288.3, 1280923657, 0)
21:07:59 000041 KAKC1601-I TESTPROC.AASTEPO2.STEP01 Step allocated.
21:07:59 000041 KAKC5012-I Information_message (000041, 002, io10288.3, 1280923679, 0)
21:07:59 000041 KAKC1601-I TESTPROC.AASTEPO3.STEP01 Step allocated.
21:07:59 000041 KAKC5013-I ENTER: (000041, 003, io10288.3, 1280923679, 0)
21:08:39 000041 KAKC1601-I TESTPROC.AASTEPO4.STEP01 Step allocated.
21:08:39 000041 KAKC5013-I XXENTER: (000041, 004, io10288.3, 1280923719, 0)
21:09:03 000041 KAKC1601-I TESTPROC.Ab Step allocated.
```

(6) カタログドプロシジャの内容

提供されるカタログドプロシジャの内容を次に示す。

(a) bjex_PAUSE.xml

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version = "1.4" os="unix" >
  <PROC NAME="bjex_PAUSE">
    <SYMBOL NAME="MSG" VALUE="Job_is_paused." />
    <STEP NAME="STEP01" >
      <EXEC PGM="/opt/hitachi/bjex/bin/bjexm_message" PARM="-p #{MSG}" PARSPLIT="YES" />
    </STEP>
  </PROC>
</HitachiBatchJobExec>
```

(b) bjex_WRITE.xml

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version = "1.4" os="unix" >
  <PROC NAME="bjex_WRITE">
    <SYMBOL NAME="MSG" VALUE="message" />
    <STEP NAME="STEP01" >
      <EXEC PGM="/opt/hitachi/bjex/bin/bjexm_message" PARM="-m #{MSG}" PARSPLIT="YES" />
    </STEP>
  </PROC>
</HitachiBatchJobExec>
```

(c) bjex_WRITER.xml

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version = "1.4" os="unix" >
  <PROC NAME="bjex_WRITER">
    <SYMBOL NAME="MSG" VALUE="Key_in[yes,no,other-error]" />
    <SYMBOL NAME="ANS" VALUE="yes no" />
    <STEP NAME="STEP01" >
      <EXEC PGM="/opt/hitachi/bjex/bin/bjexm_message" PARM="-r #{MSG} #{ANS}"
PARMSPLIT="YES" />
    </STEP>
  </PROC>
</HitachiBatchJobExec>
```

(d) bjex_WRITERX.xml

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version = "1.4" os="unix" >
  <PROC NAME="bjex_WRITERX">
    <SYMBOL NAME="MSG" VALUE="Key_in[yes,no]" />
    <SYMBOL NAME="ANS" VALUE="yes no" />
    <STEP NAME="STEP01" >
      <EXEC PGM="/opt/hitachi/bjex/bin/bjexm_message" PARM="-c #{MSG} #{ANS}"
PARMSPLIT="YES" />
    </STEP>
  </PROC>
</HitachiBatchJobExec>
```

3.15.2 応答要求メッセージ

バッチジョブ実行システムでは、ユーザからの応答入力が必要なメッセージを応答要求メッセージと呼ぶ。

例えば、COBOL2002 で作成した COBOL アプリケーションが ACCEPT 文、STOP 文または CBLTYPE サービスルーチンでメッセージを出力した場合に、COBOL アプリケーションが処理を中断してオペレータの応答を待つときのメッセージである。

また、応答要求メッセージは bjexm_message コマンドまたはメッセージ出力用カタログドプロシジャ (bjex_PAUSE, bjex_WRITER, bjex_WRITERX) を使用して出力できる。

応答要求メッセージには、次に示す 2 つの状態がある。

- 応答待ちメッセージ

バッチジョブ実行システムが、COBOL2002 で作成した COBOL アプリケーション、コマンドまたはカタログドプロシジャから応答要求メッセージを受け付けて、オペレータからの応答を待っている状態である。

- 受信待ちメッセージ

バッチジョブ実行システムが、応答要求メッセージに対するオペレータからの応答を受け付けて、COBOL2002 で作成した COBOL アプリケーション、コマンドまたはカタログドプロシジャが応答を受け取るのを待っている状態である。

3.15.3 JP1/IM - View との関係

バッチジョブ実行システムは、メッセージ制御機能のメッセージを JP1 イベントとして出力する。JP1 イベントとして出力したメッセージは、JP1/IM - View で参照できる。応答要求メッセージの JP1 イベントの場合は、JP1/IM - Manager で応答待ちイベントとして扱われるので、応答要求メッセージは JP1/IM - View で滞留して表示され、JP1/IM - View から応答が入力できる。

JP1/IM - View から応答入力する場合に、[応答入力] 画面に表示される状態と意味を次の表に示す。

表 3-30 JP1/IM - View の [応答入力] 画面に表示される状態と意味

状態	意味	JP1/IM - Manager からの 応答入力可否
READY TO RESPOND	バッチジョブ実行システムは、応答入力を待っている。	○
NO LONGER MANAGED BY BJEX	バッチジョブ実行システムは、応答待ちイベントの発行元を管理していない。	×
RESPONDED SUCCESSFULLY	バッチジョブ実行システムは、JP1/IM - View から応答を受け取った。	×
ALREADY RESPONDED	バッチジョブ実行システムは、すでに応答を受け取っていて、応答待ちイベントの発行元には応答が返されていない。	×

状態	意味	JP1/IM - Manager からの 応答入力可否
INTERNAL ERROR	バッチジョブ実行システムは、内部エラーのため処理を続行できない。	×

(凡例)

- ：応答できる。
- ×

3.15.4 JP1/IM - Manager との関係

メッセージ制御機能で出力するメッセージは、バッチジョブ実行システムによって JP1 イベント（イベント ID：00005C20 または 00005C21）として出力され、JP1/IM - View に表示される。JP1/IM - View に表示されるメッセージには、メッセージ制御機能が出力するメッセージのあとに情報が付与される。出力形式を次に示す。

メッセージ※（ジョブ識別子、ステップ番号、ホスト名、メッセージ識別子、保守情報）

注※ メッセージ制御機能が出力するメッセージ情報の意味を次に示す。

ジョブ識別子：バッチジョブ実行システムが与えたジョブ識別子を示す。

ステップ番号：ジョブ定義 XML ファイルの解析時に決定したステップ番号を示す。

ホスト名：ジョブを実行した場合に JP1/AJS が管理する論理ホスト名または物理ホスト名を示す。

メッセージ識別子：メッセージごとに割り当てる識別子を示す。

COBOL2002 で作成した COBOL アプリケーションによって、"処理を続行しますか?(Y/N)"というメッセージを出力する場合の表示例を次に示す。

処理を続行しますか?(Y/N)(000001, 001, host1, 1198043777, 0)

このメッセージの参照方法および JP1/IM - View から応答する方法については、JP1 のマニュアルを参照のこと。

3.15.5 JP1/IM - View からの応答入力

(1) 応答入力

応答要求メッセージへの応答入力は、JP1/IM - View から行う。応答手順については、JP1 のマニュアルを参照のこと。

(2) 応答内容表示

JP1/IM - View から入力された応答は、KAKC5001-I メッセージで JOBLOG および実行ログに出力される。

また、設定ファイルの DISPLAY_REPLY_MESSAGE パラメータに YES を指定することによって JP1 イベント（イベント ID：00005C25）として JP1/IM - View の画面に出力することができる。

(a) JP1/IM - View への出力方法

DISPLAY_REPLY_MESSAGE パラメータに YES を指定することによって応答要求メッセージに対する応答メッセージを JP1/IM - View の画面に出力する。

また、SETENV 要素に環境変数 BJEX_DISPLAY_REPLY を指定することで、特定のジョブやジョブステップごとに応答メッセージを JP1/IM-View の画面に出力するかどうかを設定することができる。環境変数 BJEX_DISPLAY_REPLY の指定値ごとの動作を次に示す。

表 3-31 環境変数 BJEX_DISPLAY_REPLY の設定ごとの JP1 イベント出力

環境変数 BJEX_DISPLAY_REPLY の設定	応答メッセージ出力
YES	JP1/IM - View の画面に出力する。
NO	JP1/IM - View の画面に出力しない。
YES または NO 以外	JP1/IM - View の画面に出力しない。
設定していない	設定ファイルの DISPLAY_REPLY_MESSAGE パラメータの指定が有効になる。

(b) 環境変数設定例

環境変数 BJEX_DISPLAY_REPLY は、SETENV 要素で指定する。

SETENV 要素を用いて環境変数 BJEX_DISPLAY_REPLY を指定する例を次に示す。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <SETENV NAME="BJEX_DISPLAY_REPLY" VALUE="YES"/> . . . (1)ジョブ全体で有効な環境変数
    <STEP NAME="STEP1">                               BJEX_DISPLAY_REPLYに
      <EXEC PGM="MYPROG1" LANG="COBOL"/>               YESを指定することを示す。
    </STEP>
    <STEP NAME="STEP2">
      <EXEC PGM="MYPROG2" LANG="COBOL"/>
    </STEP>
    <STEP NAME="STEP3">
      <EXEC PGM="MYPROG3" LANG="COBOL"/>
      <SETENV NAME="BJEX_DISPLAY_REPLY" VALUE="NO"/> . . . (2)ジョブステップで有効な環境変数
    </STEP>                                           BJEX_DISPLAY_REPLYに
  </JOB>                                              NOを指定することを示す。
</HitachiBatchJobExec>
```

(1)JOB 要素内で環境変数 BJEX_DISPLAY_REPLY に YES を指定して、ジョブ全体で応答内容を JP1/IM - View の画面に出力する。

(2)STEP3 の STEP 要素内で環境変数 BJEX_DISPLAY_REPLY に NO を指定する。(1)の指定よりも優先度が高いため、STEP3 は応答内容を JP1/IM - View の画面に出力しない。

3.15.6 メッセージ制御機能でのデータ置き換え機能

メッセージ制御機能で指定した出力メッセージ中に特定の 1 バイトコードが現れたときに特定の 1 バイトコードを半角空白に置き換える機能である。

この機能を使用するには、設定ファイルに MESSAGE_REPLACE_SPACE パラメータを定義すること。MESSAGE_REPLACE_SPACE パラメータのオペランドに、半角空白に置き換える 1 バイトコードを定義する。詳細については「MESSAGE_REPLACE_SPACE (半角空白に置き換える 1 バイトコードの定義)」を参照のこと。

COBOL の LOW-VALUE など出力メッセージ中に NULL 文字(X'00')文字が含まれているとメッセージ制御機能がエラーとなるが、「MESSAGE_REPLACE_SPACE 00」を定義することでエラーを防ぐことができる。

3.15.7 DISPLAY 文による JP1 イベント出力の抑止機能

COBOL2002 で作成した COBOL アプリケーションが DISPLAY 文を大量に発行すると、JP1/IM - View の画面に多くのメッセージが出力される。その結果、バッチジョブ実行時間が増加することがある。ここでは、DISPLAY 文を発行することによって JP1/IM - View の画面にメッセージが出力されることを抑止する機能について説明する。

(1) 抑止方法

環境変数 BJEX_SUPPRESS_CBLDISPLAY に YES を指定することによって JP1/IM - View の画面へのメッセージの出力を抑止する。実行ログ、ジョブログ、SYSMSG への出力は抑止しない。指定できる値は、YES または NO である。指定値ごとの動作を次に示す。

表 3-32 環境変数 BJEX_SUPPRESS_CBLDISPLAY の設定ごとの JP1 イベント出力

環境変数 BJEX_SUPPRESS_CBLDISPLAY の設定	JP1 イベント出力
YES	抑止する
NO	抑止しない
YES と NO 以外、または設定されていない	抑止しない

(2) 環境変数設定例

環境変数 BJEX_SUPPRESS_CBLDISPLAY は、SETENV 要素によって指定する。

SETENV 要素を用いて指定する例を次に示す。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="SAMPLE">
    <SETENV NAME="BJEX_SUPPRESS_CBLDISPLAY" VALUE="YES"/> . . . (1)ジョブ全体で有効な環境変数
    <STEP NAME="STEP1">                                       BJEX_SUPPRESS_CBLDISPLAYに
      <EXEC PGM="MYPROG1" LANG="COBOL"/>                         YESを指定することを示す。
    </STEP>
    <STEP NAME="STEP2">
      <EXEC PGM="MYPROG2" LANG="COBOL"/>
    </STEP>
    <STEP NAME="STEP3">
      <EXEC PGM="MYPROG3" LANG="COBOL"/>
      <SETENV NAME="BJEX_SUPPRESS_CBLDISPLAY" VALUE="NO"/> . . . (2)ジョブステップで有効な環境
変数                                                                BJEX_SUPPRESS_CBLDISPLAYに
    </STEP>                                                                 NOを指定することを示す。
  </JOB>
</HitachiBatchJobExec>
```

(1)JOB 要素内で環境変数 BJEX_SUPPRESS_CBLDISPLAY に YES を指定して、ジョブ全体で JP1/IM - View の画面へのメッセージの出力を抑止する。

(2)STEP3 の STEP 要素内で環境変数 BJEX_SUPPRESS_CBLDISPLAY に NO を指定する。(1)の指定よりも優先度が高いため、STEP3 は JP1/IM - View の画面へのメッセージの出力を抑止しない。

(3) 注意事項

環境変数 BJEX_SUPPRESS_CBLDISPLAY に YES を指定した場合、COBOL アプリケーションで発行した DISPLAY 文による JP1 イベント発行はすべて抑止される。

そのため、DISPLAY 文によって要求内容を出し、ACCEPT 文によって応答要求を行う場合、DISPLAY 文による要求内容が JP1/IM - View の画面に表示されなくなる。

要求内容を確認する方法の例を次に示す。

- COBOL プログラムの ACCEPT 文を CBLTYPE サービスルーチンに変更する。
- ジョブ定義 XML ファイルの SETENV 要素に環境変数 BJEX_SUPPRESS_CBLDISPLAY を NO に指定することにより対象のジョブステップを抑止対象外にする。

3.15.8 メッセージ制御機能でのイベント到達確認

メッセージ制御機能での JP1 イベントの送信処理では、JP1IM_MANAGER_HOST パラメータで指定したホスト (JP1/IM - Manager が稼働しているマネージャホスト) の JP1/Base に対してイベントがネッ

トワークを通して正しく到達することを確認している。これをイベント到達確認と呼び、JP1IM_MANAGER_HOST パラメータで指定したホストがバッチジョブ実行システムのホストと異なる場合に実行される。イベント到達確認には、イベント到達確認のタイミング変更機能とリトライ機能がある。

(1) イベント到達確認のタイミング変更機能

ネットワークや JP1/IM - Manager が稼働しているホストの状態や設定によっては、イベントが到達するまで時間が掛かり、到達確認のリトライが発生する場合がある。この場合、この機能によってイベント発行から到着確認実施までのタイミングを変更でき、リトライの発生を防止できる。

この機能を使用するには、設定ファイルに JP1IM_MANAGER_CHKDELAY パラメータを指定することで、最初のイベント到達確認を行うまでの待ち時間を指定できる。常に時間が掛かることが見込まれる場合は、JP1IM_MANAGER_CHKDELAY パラメータを設定すること。

(2) イベント到達確認のリトライ機能

相手ホストの JP1/Base にイベントが到達しない場合、3 秒ごとにイベント到達確認のリトライを行う。通常メッセージは、イベント到達確認を 3 回まで行う。3 回目でも到達が確認できない場合は、リトライをやめてジョブを継続させる。応答要求メッセージは MESSAGE_RETRY_TIMEOUT パラメータを指定することで、イベント到達確認のリトライを中断するまでの時間を指定できる。

また、メッセージ制御機能が通常メッセージおよび応答要求メッセージのリトライを中断しても、JP1 の設定内容によってはそのあとでイベント送信が成功する場合がある。詳細は、JP1 のマニュアルを参照のこと。

3.15.9 メッセージ制御機能に関する注意事項

(1) メッセージの出力条件

メッセージを JP1/IM - Manager へ送付するのにメッセージ制御機能を利用するには、バッチジョブ実行システムのジョブステップとして実行されている必要がある。ジョブステップとして実行されていない場合、COBOL2002 で作成したアプリケーションのメッセージの入出力では、従来どおり stdin/stdout を使用する。また、bjexm_message コマンドまたはメッセージ出力用カタログドプロシジャでは、メッセージの出力に失敗する。

- COBOL2002 で作成したアプリケーションプログラムの場合

日本語メッセージを表示したい場合は、シフト JIS で作成する必要がある。このため、EUC 言語環境では ASCII の範囲だけで利用する必要がある。ACCEPT 文および CBLTYPE サービスルーチンで受け取ることができる文字コードは、ASCII 文字コード（制御文字を除く）の範囲だけである。

DISPLAY 文および CBLTYPE サービスルーチンを使用してメッセージを出力する場合には、JP1/IM - View 側に出力されるメッセージに ID は付与されないため、JP1/IM - Manager の運用時にメッセージが識別できるように十分に検討して、プログラミングする必要がある。

- bjexm_message コマンドまたはメッセージ出力用カタログドプロシジャの場合
日本語メッセージを表示したい場合、LANG 環境変数に対応するエンコードでメッセージを指定する。LANG 環境変数に対応するエンコードでない場合、メッセージは正しく表示されない。応答として受け取れる文字コードは ASCII 文字列だけである。
また、出力するメッセージにはメッセージ ID が付与されないため、JP1/IM - Manager の運用時にメッセージが識別できるように十分に配慮する必要がある。

メッセージを出力する件数の推奨値を次に示す。

表 3-33 JP1/IM - Manager へ送付するメッセージの出力件数の推奨値

プログラム, コマンドまたはカタログドプロシジャ	出力件数	種別
DISPLAY UPON CONSOLE 文	2 件/秒より少ない。	通常メッセージ
bjexm_message -m コマンド		
bjex_WRITE カタログドプロシジャ		
ACCEPT FROM CONSOLE 文	1 件/分より少ない。	応答要求メッセージ
STOP 文 (定数指定)		
CBLTYPE サービスルーチン		
bjexm_message -p コマンド		
bjexm_message -r コマンド		
bjexm_message -c コマンド		
bjex_PAUSE カタログドプロシジャ bjex_WRITER カタログドプロシジャ bjex_WRITERX カタログドプロシジャ		

(2) エラー情報

メッセージ制御機能でエラーが発生した場合、COBOL2002 では KCCC2060R-S メッセージが出力される。また、bjexm_message コマンドまたはメッセージ出力用カタログドプロシジャでは KAKC5205-E メッセージが出力される。KCCC2060R-S メッセージまたは KAKC5205-E メッセージに表示される戻り値と意味を次の表に示す。戻り値とともに表示されるエラー番号の意味については、「付録 D メッセージ制御機能のエラー情報」を参照のこと。

表 3-34 KCCC2060R-S メッセージまたは KAKC5205-E メッセージに出力される戻り値と意味

戻り値	意味	考えられる主な原因	対処
1	環境が不正である。	JP1/AJS, またはバッチジョブ実行システムを利用して呼び出していない可能性がある。	該当するプログラムを JP1/AJS からバッチジョブ実行システムを通して呼び出すように変更して再実行する。

戻り値	意味	考えられる主な原因	対処
2	共有メモリまたは実行ログの操作でエラーが発生した。	<ul style="list-style-type: none"> スプールジョブ転送デーモンが起動していない。 bjexchmsg コマンドで応答状態を強制的に解除した。 	スプールジョブ転送デーモンが起動していることを確認する。
3	システムコールでエラーが発生した。	スプールディレクトリにアクセス権がないため、ジョブログに出力できない。	スプールディレクトリに、実行ユーザの書き込み権限を与える。
4	JP1 イベント送信時に通信エラーが発生した。	JP1/IM - Manager が稼働するホストに接続できない。	ネットワークの問題を解決する。
5	JP1 イベント送信時に通信以外エラーが発生した。	JP1/Base のライブラリが見つからない。	JP1/Base をインストールする。

(3) 運用について

- メッセージを作成する場合、メッセージテキストによって運用側で判断ができるように設計すること。
- JP1IM_MANAGER_HOST パラメータで指定した JP1/IM - Manager に接続した JP1/IM - View 以外から応答はできない (バッチジョブ実行システムが提供する bjexlsmg および bjexchmsg コマンドでの応答はできる)。
- JP1/IM - Manager が滞留できる応答待ちイベント数には上限があるため、これを超えない運用を設計すること。
- 応答入力ができる文字コードは、ASCII 文字コード (制御文字を除く) の範囲だけである。この範囲以外の文字コードを入力した場合は、エラーメッセージが表示されるので、範囲内の文字コードで再入力すること。
- JP1 イベントの送信処理で、一時的な通信エラーまたはリソース不足などの時間を空ければ回復の見込みのあるエラーが発生した場合は、3 秒間隔でリトライを繰り返す。そのため、回復の見込みのあるエラーが発生し続ける場合は、エラーを回復しないかぎり、メッセージ制御機能が終了しなくなる。
また、リトライする場合は、JP1IM_MANAGER_HOST パラメータで指定したホストに対して、JP1/Base が TCP/IP の接続の切断と新たなポートでの接続を繰り返す。出力件数が推奨値の場合、接続が使用していたポートは、OS の MSL (Maximum Segment Lifetime) × 2 (秒) の間使用できなくなるため、MSL の値が大きいかまたはポート数が少ない場合には、リトライによってポートが枯渇する可能性がある。そのため、MSL × 2 (秒) の間にメッセージ制御機能によって出力するメッセージ数、MSL、およびポート数は、次に示す条件を満たすようにすること。

$$n \times \text{MSL} \times 2/3 < \text{ポート数}$$

$$n : \text{MSL} \times 2 \text{ (秒) の間にメッセージ制御機能によって出力するメッセージ数}$$
- メッセージ制御機能を使用してメッセージを出力する COBOL アプリケーションは、SJIS または ASCII 文字を前提としており、EUC 環境では日本語メッセージが文字化けすることがある。
- 作成するメッセージ中に ¥0' など画面に表示できない文字が含まれていると、COBOL アプリケーションの実行時にエラーとなる場合があるため、表示不可能文字をメッセージに含めないこと。

bjexm_message コマンドまたはメッセージ出力用カタログドプロシジャの場合も同様にエラーとなるため、表示不可能文字を含めてはならない。

3.16 スプールジョブ保管機能

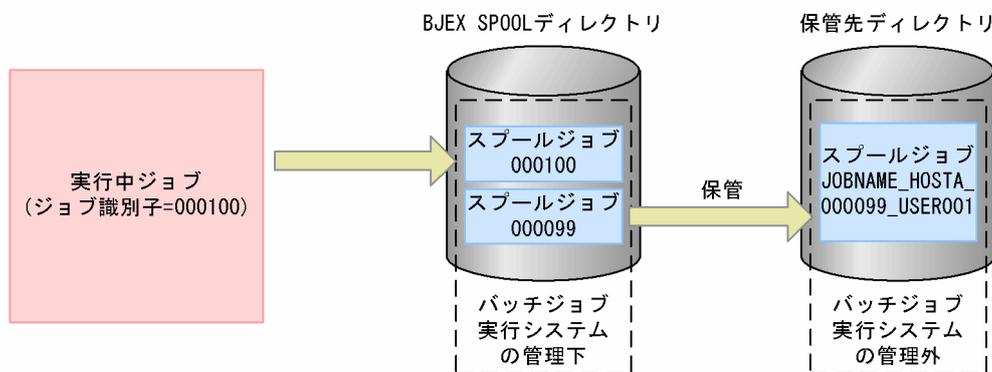
スプールジョブ保管機能は、ジョブ終了時、スプールジョブを転送する前に任意のディレクトリにスプールジョブを保管できる。また、bjexchjob コマンドを使用して一括してスプールジョブを保管できる。

バッチジョブ実行システムのスプールディレクトリにあるスプールジョブはバッチジョブ実行システムで操作できるが、ユーザは直接参照してはならない。この機能を使うことで、保管先ディレクトリにあるスプールジョブをユーザが自由に参照できる。

3.16.1 スプールジョブ保管機能の概要

スプールジョブ保管機能の概要を次の図に示す。

図 3-30 スプールジョブ保管機能の概要



スプールジョブの保管時には「ジョブ名_ホスト名_ジョブ識別子_ユーザ名」のディレクトリに複写して保管する。元のスプールジョブは従来どおり残り、転送処理が行われる。保管されたスプールジョブはバッチジョブ実行システムの管理から外す。

スプールジョブ保管機能を使用して、次のことができる。

- ジョブ終了時のジョブコントローラによる自動保管
ジョブ終了時にジョブコントローラが自動でスプールジョブを設定ファイルで指定した保管先に保管する。
- bjexchjob コマンドによるスプールジョブの保管
bjexchjob コマンドで-C オプションを指定し、スプールジョブを設定ファイルで指定した保管先に保管する。
- 保管するスプールジョブのパーミッションの変更
ジョブ終了時と bjexchjob コマンドでの保管では、保管先の SYSOUT データファイルのパーミッションや所有者などは、元の SYSOUT データファイルの情報を引き継ぐ。保管先のスプールジョブの権限を変更する場合には、設定ファイルの PERMISSION_SPOOL_STORE_JOB パラメータを指定する。
- 日付や時間ごとのサブディレクトリへの保管

保管先ディレクトリの下にジョブの実行開始日付ごとまたは実行開始日時ごとにサブディレクトリを作成し、スプールジョブをサブディレクトリの下に保管する。

- 保管先ディレクトリの名称変更

スプールジョブの保管時には、通常「ジョブ名_ホスト名_ジョブ識別子_ユーザ名」のディレクトリに保管するが、この名称を変更できる。

3.16.2 スプールジョブ保管機能の設定方法

スプールジョブ保管機能は、ジョブ終了時のジョブコントローラによる自動保管と bjexchjob コマンドによる保管の 2 種類に分けられる。それぞれの保管方法ごとに設定が必要なパラメータを次の表に示す。

表 3-35 スプールジョブ保管機能に関連する設定ファイルのパラメータ

設定ファイルのパラメータ	意味	保管の方法	
		ジョブ終了時のジョブコントローラによる自動保管	bjexchjob コマンドによるスプールジョブの保管
SPOOL_STORE_DIR	保管先ディレクトリのパスを定義して、スプールジョブ保管機能を有効にする。	◎	◎
SPOOL_STORE_SUBDIR	保管先ディレクトリのサブディレクトリを定義する場合に指定する。	○	○
SPOOL_JOB_STORE	ジョブ終了時にスプールジョブを保管する場合に指定する。	◎	—
PERMISSION_SPOOL_STORE_JOB	保管するスプールジョブに設定するパーミッションを指定する。	○	○
SPOOL_STORE_DIRNAME	スプールジョブを保管するときの保管先ディレクトリ名称を定義する場合に指定する。	○	○

(凡例)

- ◎：指定は必須。
- ：指定は任意。
- ：指定は不要。

- ジョブ終了時のジョブコントローラによる自動保管の場合、設定ファイルに SPOOL_STORE_DIR パラメータ、および SPOOL_JOB_STORE パラメータの両方を指定すること。どちらかの指定を省略した場合、この機能は有効とならない。

- bjexchjob コマンドで-C オプションを指定してスプールジョブ保管機能を使用する場合、設定ファイルに SPOOL_STORE_DIR パラメータを指定すること。SPOOL_STORE_DIR パラメータの指定を省略した場合、この機能は有効とならない。

3.16.3 日付や時間ごとのサブディレクトリへの保管

保管先ディレクトリパス名に加えて設定ファイルの SPOOL_STORE_SUBDIR によってサブディレクトリ定義を指定した場合、バッチジョブ実行システムは保管先ディレクトリの下にサブディレクトリを自動的に作成し、スプールジョブをサブディレクトリの下に保管する。

サブディレクトリはジョブの実行開始日付ごとまたは実行開始日時ごとに作成され、該当するサブディレクトリ下に保管される。

図 3-31 サブディレクトリ定義（ジョブの実行開始日をサブディレクトリ名とする）を指定した場合のスプールジョブ保管



図 3-32 サブディレクトリ定義（ジョブの実行開始日+開始時刻をサブディレクトリ名とする）を指定した場合のスプールジョブ保管



3.16.4 保管先ディレクトリの名称

設定ファイルの SPOOL_STORE_DIRNAME によって保管先のスプールジョブディレクトリの名称を変更できる。保管先のデフォルトのスプールジョブディレクトリ名は「ジョブ名_ホスト名_ジョブ識別子_ユーザ名」であるが、名称を変更する際には、ジョブ名、ホスト名、ジョブ識別子、ユーザ名、AJSJOBNAME の組み合わせを指定する。それぞれが使用する内容を次に示す。

表 3-36 スプールジョブ保管機能が使用する内容

項目	内容
ジョブ名	ジョブ実行時のジョブ名
ホスト名	設定ファイルの JP1_HOSTNAME パラメータで指定したホスト名を使用する。ただし、バッチジョブ実行システムのコマンドで -h オプションを指定した場合、設定ファイルの JP1_HOSTNAME パラメータは無視され、論理ホスト名が有効になる。詳細については、「7. コマンド」を参照すること。
ジョブ識別子	ジョブ実行時のジョブ識別子
ユーザ名	環境変数 JP1_USERNAME [*] 、または OS から求めたユーザ名
AJSJOBNAME	環境変数 AJSJOBNAME [*] のうち、ディレクトリ区切り文字の"/"（スラント）を含めたディレクトリの、最後のディレクトリ区切り文字"/"（スラント）より後ろの文字列を使用する。AJSJOBNAME の設定がない場合は、""（空文字）を使用する。

注※ 設定される内容については、JP1/AJS のマニュアルを参照のこと。

3.16.5 スプールジョブ保管機能使用時の注意事項

スプールジョブ保管機能使用時の注意事項を次に示す。

- 保管されたスプールジョブはユーザ責任で削除する必要がある。
- 保管されたスプールジョブはバッチジョブ実行システムのコマンドで印刷指示などの操作はできない。
- ジョブ終了を契機とする保管時に保管先にすでに同一のディレクトリが存在する場合、保管処理に失敗する。
- ジョブ終了時に保管する SYSOUT データファイルは、プリンタサーバへの転送有無とは無関係にすべて保管の対象となる。このため、転送処理で削除されるダミー出力クラスの SYSOUT データファイルやデータサイズが 0 の SYSOUT データファイルも削除されないで保管される。スプールジョブ管理ファイルも転送処理後の内容とは異なる。
- bjexchjob コマンドで -C オプションを指定してスプールジョブを保管する場合、ジョブを実行したユーザと同一ユーザまたは特権ユーザで実行する必要がある。
- 保管は bjexchjob コマンドで -C オプションを指定しても実行できる。この場合、保管先にすでに同一のディレクトリが存在する場合、上書きで保管する。また、コマンドで保管するスプールジョブは一度転送処理が行われており、ジョブ終了時に保管される内容とは一致しない。保管するときのディレクトリ名はジョブ終了時と同一であるが、ホスト名やユーザ名はスプールジョブに格納されている値を使用する。
- 保管したスプールジョブはバッチジョブ実行システムのスプールに戻すことはできない。このため、スプールのバックアップとしては利用できないため、再度印刷する予定があるスプールジョブはスプールから削除しないようにする必要がある。ただし、スプールから削除してしまった場合は、次の方法で SYSOUT データファイルごとに印刷できる。

- PDE で印刷する場合

印刷する SYSOUT データファイルを特定する。印刷する SYSOUT データファイルの印刷属性をジョブ定義 XML ファイルの DD 要素、設定ファイルの SysoutClass パラメータから調査する。これらの情報を基に PDE のバッチジョブ実行システムと連携しない方法で印刷する。詳細については、マニュアル「PRINT DATA EXCHANGE for Open」を参照のこと。

- EUR で印刷する場合

印刷する SYSOUT データファイルを特定する。印刷する SYSOUT データファイルで使用する EUR 定義ファイルから実行する EUR のコマンド情報を調査する。これらを基に直接 EUR のコマンドを使って印刷する。詳細については、EUR のマニュアルを参照のこと。

3.17 PDE ファイル形式 2

PDE ファイル形式 2 の SYSOUT データファイルを作成できる。

PDE ファイル形式 2 は、PDE と COBOL2002 が連携することで VOS3 の帳票印刷から高い移行性を実現するデータ形式であり、PDE による書式、行制御印刷を行うときに使用するファイル形式である。詳細については、マニュアル「PRINT DATA EXCHANGE for Open」、および「COBOL2002 マイグレーションライブラリ」の取扱説明書を参照のこと。

バッチジョブ実行システムで PDE ファイル形式 2 の SYSOUT データファイルを作成する場合、出力クラスを定義する SysoutClass パラメータの FTYPE オペランドおよび RECFM オペランド、または PDE ファイル形式 2 の SYSOUT データファイルを定義する DD 要素の FTYPE 属性および RECFM 属性で指定する。

3.18 ファイル操作ユーティリティ

3.18.1 ファイル操作ユーティリティの概要

ファイル操作ユーティリティには、次のプログラムがある。

(1) bjexu_dummy プログラム

bjexu_dummy は、VOS3 の JDJDUMMY プログラムと同じく、ダミーのプログラムである。実行すると終了コードが 0 で終了する。ダミーのプログラムなので、バッチジョブ実行システムのジョブ定義 XML ファイルの DD 要素で行うファイル、またはディレクトリの操作だけを実行したい場合などに使用する。

(2) bjexu_pcopy プログラム

bjexu_pcopy は、VOS3 のデータセットユーティリティの JSDPCPY プログラムに準じたファイルの複写を行う。複数の複写元ディレクトリから、1 つの複写先ディレクトリへファイル単位に複写する。複写するファイルは、全部またはファイル名によって選択できる。

(3) bjexu_scopy プログラム

bjexu_scopy は、VOS3 のデータセットユーティリティの JSDSCPYPY プログラムに準じたファイルの複写を行う。ファイルの内容をレコードの単位で操作して複写できる。

(4) bjexu_ismut および bjexu_ismut64 プログラム

bjexu_ismut および bjexu_ismut64 は、VOS3 の VSAM ユティリティの JSCVSUT プログラムに準じたファイルの作成、削除と複写を行う。索引順編成ファイルを作成、削除でき、索引順編成ファイルと順編成ファイルを複写、変換および併合できる。

3.18.2 ファイル操作ユーティリティのサポートするファイル

(1) ファイル編成

ファイル操作ユーティリティでは、次の表に示すファイル編成をサポートする。

表 3-37 ファイル操作ユーティリティのサポートするファイル編成

ファイル編成	説明	サポートの有無	
		入力ファイル	出力ファイル
固定長順編成ファイル	COBOL2002 で作成したファイル, またはほかのアプリケーションで作成したファイルサイズが, レコード長の整数倍の長さを持つバイナリファイル	○	○
可変長順編成ファイル	COBOL2002 で作成したファイル	○	○
固定長相対編成ファイル	COBOL2002 で作成したファイル	—	—
可変長相対編成ファイル	COBOL2002 で作成したファイル	—	—
固定長索引順編成ファイル	<ul style="list-style-type: none"> • COBOL2002 で作成したファイル • ISAM で作成したファイル 	○※1	○※1
可変長索引順編成ファイル	<ul style="list-style-type: none"> • COBOL2002 で作成したファイル • ISAM で作成したファイル 	○※1	○※1
テキスト編成ファイル	テキストファイルの各行 (テキストレコード) は, 改行文字 (X'0A') で区切られている次のファイル <ul style="list-style-type: none"> • COBOL2002 で作成したテキスト編成ファイル • vi エディタなどで作成したテキストファイル 	○※2	○※2
CSV 編成ファイル	<ul style="list-style-type: none"> • COBOL2002 で作成した CSV 編成ファイル • 表計算プログラムで作成した CSV ファイル 	—	—

(凡例)

- : サポートする。
- : サポートしない。

注※1 bjexu_ismut および bjexu_ismut64 でサポートする。

注※2 bjexu_scopy および bjexu_pcopy でサポートする。

(2) ラージファイル対応

ファイル操作ユーティリティでサポートするファイルのうち, ISAM で作成する索引順編成ファイルは, ラージファイル (ファイルサイズが 2GB 以上のファイル) をサポートしない。その他のラージファイル (ファイルサイズが 2GB 超えのファイル) への対応については, 「1.3.3 2GB 超えのファイル (ラージファイル) の扱い」を参照のこと。

(3) ディレクトリやファイル名指定時の注意事項

(a) ハードリンクとシンボリックリンクについて

ファイル名として, ハードリンクとシンボリックリンクを使用できる。また, ディレクトリにはシンボリックリンクを使用できる。

ただし、ファイル名やディレクトリ名の重複チェックは、ジョブ定義 XML ファイルや制御文に指定された内容で行うため、ハードリンクやシンボリックリンクが指定されていると、ファイル名の実体が同一かどうかのチェックが行えないことがある。

たとえば、入力ファイルがシンボリックファイル、出力ファイルが相対パス名または絶対パス名の場合、入出力ファイルの重複チェックはできない。

(b) 入力ファイルと出力ファイルの指定について

ファイル名には、パス名の区切りを示す文字 (/ : スラント) を指定できない。また、入力と出力に指定したファイルが同一の実体にならないようにすること。入力と出力のファイルが同一の実体になっていた場合、ファイルの内容が初期化されることがある。

(4) ファイルのパーミッション

(a) ジョブ定義 XML ファイルで指定

ファイル操作ユティリティで使用するファイルは、ジョブ定義 XML ファイルの DD 要素で指定する。出力ファイルを新規に作成する場合、ファイルに設定するパーミッションは、ジョブ定義 XML ファイルの DD 要素でファイルを新規に作成する場合の仕様に従う。バッチジョブ実行システムでは、ファイルを新規に作成時、DD 要素の FPERM 属性でファイルのパーミッションを指定できる。また、指定しない場合は、PERMISSION 要素の属性に従う。

次に DD 要素の FPERM 属性で指定する例を示す。

例1: DD要素のFPERM属性で指定する

```
<STEP NAME="STEP0001" >
  <EXEC PGM="bjexu_scopy" />
  <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
  <DD NAME="SYSUT1" TYPE="FILE" DSN="/test/infile" DISP="OLD" />
  <DD NAME="SYSUT2" TYPE="FILE" DSN="/test/outfile" DISP="NEW,KEEP" FPERM="0644" />
  <DD NAME="SYSIN" TYPE="DATA" >
    <![CDATA[
!
      INPFILE FILEORG=F, RECLLEN=80
!
    ]]>
  </DD>
</STEP>
```

(b) ファイル操作ユティリティで新規に作成するファイル

ジョブ定義 XML ファイルでの指定がなく、制御文で指定したファイルを新規に作成する場合、次のようにパーミッションを設定する。

- コピー元ファイルがある場合

コピー元ファイルのパーミッション情報 (所有者, グループ, その他に対するアクセス権限) にファイル操作ユティリティを実行しているプロセスの `umask` でマスクしたパーミッションを設定する。

その他の sticky ビット, set-user-ID ビット, set-group-ID ビットの情報は引き継がない。また、ファイルのタイムスタンプも引き継がない。

- コピー元ファイルがない場合

"0666"にファイル操作ユーティリティを実行しているプロセスの umask でマスクしたパーミッションを設定する。

次に制御文で指定したファイルを新規作成する例を示す。

例2: 制御文で指定したファイルを新規作成する

```
<STEP NAME="STEP0001" >
  <EXEC PGM="bjexu_pcopy" />
  <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
  <DD NAME="IN" TYPE="DIR" DSN="/test/infile" DISP="OLD"/>
  <DD NAME="OUT" TYPE="DIR" DSN="/test/outfile" DISP="OLD"/>
  <DD NAME="SYSIN" TYPE="DATA" >
    <![CDATA[
!
      COPY INDD=(IN),OUTDD=OUT
      SELECT MEMBER=(AAA,BBB,CCC)
!
    ]]>
  </DD>
</STEP>
```

SELECT 文で指定したファイル (AAA など) がコピー先に新しく作成される場合、所有者のパーミッションが設定される。

所有者のパーミッションは、コピー元のファイルに設定されているパーミッションに umask でマスクしたパーミッションを設定する。

(c) 既存ファイルの更新

既存ファイルを更新するためのパーミッションの設定は、システムの標準仕様に従う。既存ファイルを更新するには、ファイルを更新できるアクセス権限が必要である。

(5) PREST ファイル

bjexu_scopy および bjexu_ismut で順編成ファイル (固定長順編成ファイルと可変長順編成ファイル) の入出力をするとき、PREST ファイルを使用できる。PREST ファイルを使用することで、逐次実行していたジョブおよびジョブステップを並列に実行することが可能となり、処理時間を短縮できる。

3.18.3 ファイル操作ユーティリティの実行方法

ファイル操作ユーティリティを実行するためには、バッチジョブ実行システムのジョブ定義 XML ファイルの EXEC 要素の PGM 属性に、使用するユーティリティのプログラムを指定する。なお、ファイル操作ユーティリティで排他制御は行わないため、排他制御が必要な場合は、バッチジョブ実行システムの設定ファイルの LOCK_OPTION パラメータを使用すること。

ファイル操作ユーティリティが格納されているディレクトリを PATH 環境変数にあらかじめ登録すれば、EXEC 要素の PGM 属性にプログラム名だけを指定してファイル操作ユーティリティを実行できる。JOB LIB の DD 要素や STEPLIB の DD 要素で、PATH 環境変数に登録するディレクトリを追加することもできる。

ファイル操作ユーティリティのプログラムの格納場所を次の表に示す。

表 3-38 ファイル操作ユーティリティのプログラムの格納場所

プログラム名	機能概要	プログラムの格納場所
bjexu_dummy	ダミープログラム	/opt/hitachi/bjex/bin
bjexu_pcopy	ファイルの複写 (ディレクトリ単位)	
bjexu_scopy	ファイルのレコード単位の複写 (編集操作を含む)	
bjexu_ismut	索引順編成ファイルの作成, 削除, 索引順編成ファイルと順編成ファイルの複写, 変換および併合	
bjexu_ismut64		

3.18.4 ファイル操作ユーティリティの制御文の形式

ファイル操作ユーティリティの制御文は、次の形式で指定する。

[記号]△₁ コマンド名 △₁オペランド [△₁ 注釈]

(凡例)

△₁ : 1 バイト以上の空白文字を必ず指定する。

記号

制御文を識別するのに使用する。記号の指定は第 1 欄から始める。

コマンド名

制御文の種別を表す。コマンド名の指定は、記号のあとに 1 個以上の空白を入れる。記号を指定しない場合は第 2 欄以降から始める。

オペランド

コマンド名のあとに 1 個以上の空白を入れて始める。オペランドの内容については、各制御文の説明を参照のこと。

注釈

最終オペランドから 1 個以上の空白を入れて始める。注釈を書くのは任意である。

制御文が 1 行に入らない場合、その行の最終オペランドの直後にコンマ (,) を指定して制御文を次の行に続ける。2 行目以降では、第 1 欄以降から開始する。ある行の最終オペランドの直後にコンマ (,) を指定し、制御文を次行に続ける場合にも、コンマの次に 1 個以上の空白を入れてコメントを記述できる。

1 つの制御文は、継続行を含め最大 4,094 バイト指定できる。

3.18.5 bjexu_dummy (ファイルおよびディレクトリ操作ダミー処理の実行)

(1) 機能

ダミーの処理を行う。実行すると終了コードが 0 で終了する。

(2) 入出力構成

bjexu_dummy が必要とする入出力はない。

(3) ジョブ定義 XML ファイル

bjexu_dummy を実行する場合は、次に示す EXEC 要素の指定が必要である。

EXEC 要素の PGM に bjexu_dummy を指定する。

```
<EXEC PGM="bjexu_dummy" />
```

(4) bjexu_dummy 制御文

制御文はない。

(5) 使用例

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="DUMMYPROG" >
    <STEP NAME="STEP0001" >
      <EXEC PGM="bjexu_dummy" />           ... (1)
      <DD NAME="NEW1" TYPE="FILE" DSN="WORK1" DISP="NEW,KEEP" /> ... (2)
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

(1)bjexu_dummy を実行する。

(2)恒久ファイル WORK1 を生成し保存することを指定する。

3.18.6 bjexu_pcopy (ファイルの複写)

(1) 機能

1 個、または複数個のディレクトリに存在するファイルを、ほかのディレクトリへ複写する。複写するファイルは、ファイル名で特定したり、すべてのファイルを対象にしたりできる。なお、ディレクトリ名での複写や、ディレクトリの再帰コピーはできない。

複写元は 1 個、または複数個指定でき、入力ディレクトリと呼ぶ。複写先は 1 個だけで出力ディレクトリと呼ぶ。

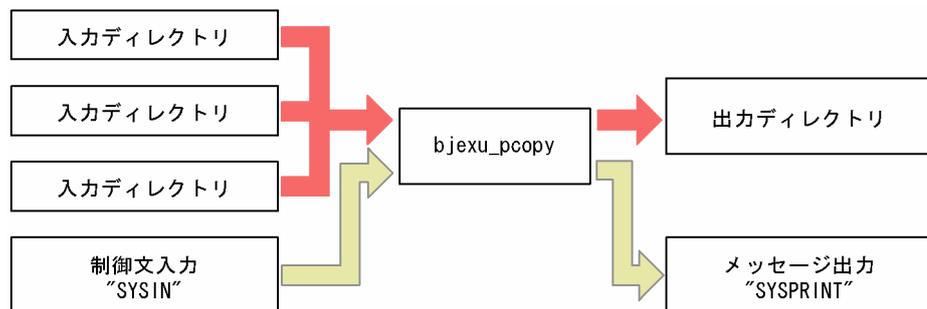
入力ディレクトリが 2 個以上指定されている場合、指定した順にすべての入力ディレクトリを順次処理する。ただし、bjexu_pcopy 制御文が SELECT 文の MEMBER 指定の場合だけは、先に指定された入力ディレクトリに対象のファイルが見つかったら、後続の入力ディレクトリに同じ名称のファイルがあっても無視する。詳細については、「(4)(b) SELECT 文」の注意事項を参照のこと。

複写する対象をすべてのファイルやファイルの総称名（ファイル名の頭文字）で指定した場合、複写するファイルの順序はシステムの状態によって異なる。

(2) 入出力構成

bjexu_pcopy の入出力構成を次の図に示す。

図 3-33 bjexu_pcopy の入出力構成



(a) 入力

- 入力ディレクトリ
複写の入力となる、複写元のディレクトリ。
- 制御文 (SYSIN)
bjexu_pcopy の処理内容を指定する。制御文には次のものがある。
 - COPY 文
 - SELECT 文
 - EXCLUDE 文
 - INDD 文

(b) 出力

- 出力ディレクトリ
複写の出力となる、複写先のディレクトリ。
- メッセージ (SYSPRINT)
bjexu_pcopy が出力するメッセージの出力先。

(c) 終了コード

bjexu_pcopy の終了コードを次の表に示す。

表 3-39 bjexu_pcopy の終了コード

終了コード	意味
0	正常に処理を終了した。*1
4	エラーが発生したが続行できるため、エラーを無視して処理を続行した。詳細については、出力されたメッセージを参照のこと。*2
8	エラーが発生した。エラーの発生したコピーオペレーションの処理を終了し、次のコピーオペレーションの処理を続行する。詳細については、出力されたメッセージを参照のこと。 なお、ファイルの新規作成やファイルの置き換え指定の複写で、文法解析終了後の複写処理中にエラーが発生した場合、出力ファイルの内容は保証されない。
12	次に示すような回復できないエラーが発生したため、処理を打ち切り、ジョブステップを終了する。ファイルの新規作成やファイルの置き換え指定の複写で、文法解析終了後の複写処理中にエラーが発生した場合、出力ファイルの内容は保証されない。 <ul style="list-style-type: none">• 制御文の指定誤り• 制御文や入出力ファイルのオープンエラー、I/O エラー

注※1 入力ディレクトリと出力ディレクトリが同じときは何もしないで処理を終了する。この場合も終了コードは0になる。

注※2 文法的に正しいが、要求された処理を実行できなかった場合、または SELECT 文で指定したファイルがないので処理できなかった場合や SYSPRINT でオープンエラーが発生した場合の終了コードは4になる。

(3) ジョブ定義 XML ファイル

bjexu_pcopy を実行する場合は、次に示す EXEC 要素と DD 要素の指定が必要である。

(a) EXEC 要素

EXEC 要素の PGM に bjexu_pcopy を指定する。

```
<EXEC PGM="bjexu_pcopy" />
```

(b) SYSPRINT の DD 要素

bjexu_pcopy が発行するメッセージの出力先を指定する。なお、DD 連結指定はできない。出力先には、SYSOUT データファイルや恒久ファイルを指定できる。

SYSPRINT が使用できない状態の場合、経過メッセージは出力されない。出力メッセージの詳細については、「10. メッセージと終了コード」を参照のこと。

(c) 入力ディレクトリの DD 要素

複写元となる入力ディレクトリ (TYPE="DIR") を指定する。DD 連結指定はできない。

(d) 出力ディレクトリの DD 要素

複写先となる出力ディレクトリ (TYPE="DIR") を指定する。DD 連結指定はできない。

(e) SYSIN の DD 要素

bjexu_pcopy 制御文を指定する。または、この制御文が記述されたファイルのパスを指定する。DD 連結指定はできない。

(4) bjexu_pcopy 制御文

bjexu_pcopy 制御文と機能を次の表に示す。

表 3-40 bjexu_pcopy 制御文と機能

制御文名	機能
COPY 文	複写元の入力ディレクトリから複写先の出力ディレクトリへファイルを複写する。
SELECT 文	複写するファイルを指定する。
EXCLUDE 文	複写しないファイルを指定する。
INDD 文	入力ディレクトリを指定する。

bjexu_pcopy 制御文は、コピーオペレーションおよびコピーステップという概念を用いる。

コピーオペレーションとは、COPY 文で始まり、後続の COPY 文または指定した制御文の終わりまでの一連の制御文である。

COPY 文で始まるコピーオペレーションは、1 個以上のコピーステップを含んでおり、SELECT 文や EXCLUDE 文の直後の INDD 文がコピーステップの開始と先行するコピーステップの終了を示す。コピーオペレーションおよびコピーステップの概念については、「[図 3-34 コピーオペレーションおよびコピーステップの概念](#)」を参照のこと。

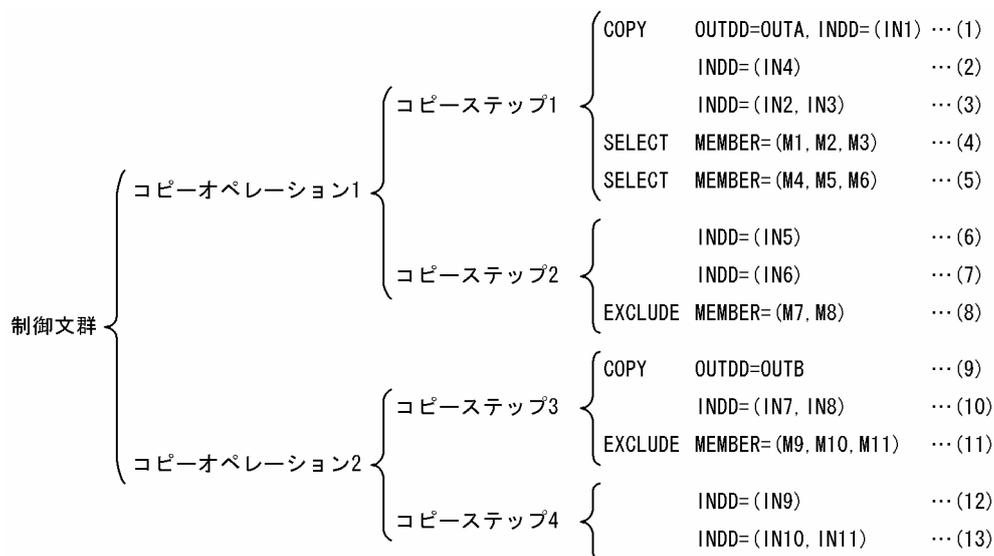
複写処理は、コピーオペレーションの指定順に実行し、複数のコピーステップに分割される場合も、コピーステップ単位に順次実行する。

コピーオペレーションには、1 個の複写先の実出力ディレクトリと、1 個以上の複写元の入力ディレクトリが含まれる。また、コピーオペレーションを幾つかのコピーステップに分割することで、コピーステップ単位に入力ディレクトリを変えることができる。

SELECT 文や EXCLUDE 文は、コピーステップ単位に複写するファイルや複写しないファイルを指定する。SELECT 文や EXCLUDE 文の指定がない場合は、入力ディレクトリにある全ファイルを複写する。なお、出力ディレクトリに複写するファイルと同じ名称のファイルがすでに存在する場合は、ファイルの置き換え指定があるときだけ複写する。置き換え指定は、入力ディレクトリ単位や、SELECT 文のファイル名やファイル総称名でも指定できる。複写元のファイルが、どれかの置き換え指定に該当すれば、出力ディレクトリに同じ名称のファイルがあっても複写される。

各制御文のオペランドは、1 つの制御文中に同じオペランドを複数指定できない。同じオペランドを複数個指定した場合は、最後に指定したオペランドだけが有効になる。

図 3-34 コピーオペレーションおよびコピーステップの概念



(1)~(3)入力ディレクトリ IN1,IN4,IN2,IN3 から、出力ディレクトリ OUTA への複写であることを示す。

(4)~(5)(1)~(3)で指定した入力ディレクトリから、ファイル M1,M2,M3,M4,M5,M6 だけ複写する。

(6)~(7)入力ディレクトリ IN5,IN6 から、出力ディレクトリ OUTA への複写であることを示す。

(8)(6)~(7)で指定した入力ディレクトリから、ファイル M7,M8 を除いて複写する。

(9)出力ディレクトリ OUTB への複写であることを示す。

(10)入力ディレクトリ IN7,IN8 から、(9)で指定した出力ディレクトリへの複写であることを示す。

(11)(10)で指定した入力ディレクトリから、ファイル M9,M10,M11 を除いて複写する。

(12)~(13)入力ディレクトリ IN9,IN10,IN11 から、出力ディレクトリ OUTB への複写であることを示す。後続に SELECT 文と EXCLUDE 文の指定がないため、すべてのファイルを複写する。

注 コピーステップ内の入力ディレクトリは、制御文に指定した順序で処理される。例えばコピーステップ1の場合、最初に IN1 を処理し、続いて IN4,IN2,IN3 の順に処理する。

(a) COPY 文

■ 形式

[記号] Δ ₁ COPY Δ ₁	OUTDD=DD名 [, INDD={({DD名 (DD名,R)}},{DD名 (DD名,R)}]…)] [, LIST=NO]
---	--

■ オペランド

OUTDD=DD 名 ~<環境変数名>((1~31 文字))

出力ディレクトリのパス名を定義した DD 名を指定する。このオペランドは必須である。

INDD={({DD 名 |(DD 名,R)}[, {DD 名 |(DD 名,R)}]…)

- DD 名 ~<環境変数名>((1~31 文字))

入力ディレクトリのパス名を定義した DD 名を指定する。このオペランドの指定は任意であり、後続の INDD 文でも指定できる。

このオペランドを省略する場合、COPY 文の直後に INDD 文の指定が必要である。INDD 文の指定がなければ、エラーメッセージを出力して、このコピーオペレーションを無効とし、次のコピーオペレーションを処理する。

- R

出力ディレクトリ中に、複写するファイルのファイル名または SELECT 文 (MEMBER オペランド指定) で名前を変更して複写する場合は、新しいファイル名と同じ名称のファイルがすでに存在するとき、ファイルの置き換えを指定する。ファイルの置き換えを指定するには、DD 名とキーワード R を括弧で囲むこと。

LIST=NO

処理したファイルのファイル名などの詳細なメッセージを出力する必要があるときに指定する。出力しないメッセージは、KAKD1100-I~KAKD1105-I メッセージである。このオペランドを省略すると、すべてのメッセージを出力する。

■ 注意事項

- COPY 文はコピーオペレーションの開始を示す制御文であるため、コピーオペレーションの先頭にあること。
- INDD オペランドでファイルの置き換えを指定する場合、(DD 名,R) の指定の途中で行を変えないこと。
- INDD オペランドと OUTDD オペランドで同じディレクトリのパス名を指定した場合、SELECT 文や EXCLUDE 文の指定があるとエラーになり、そのコピーオペレーションを終了する。

(b) SELECT 文

■ 形式

[記号] Δ ₁ SELECT Δ ₁	{MEMBER=({ファイル名 (ファイル名,,R) (ファイル名,新ファイル名[,R])} [, {ファイル名 (ファイル名,,R) (ファイル名,新ファイル名[,R])}]…) GROUP=({ファイル総称名 (ファイル総称名,R)} [, {ファイル総称名 (ファイル総称名,R)]…)}
--	---

■ オペランド

MEMBER=({ファイル名|(ファイル名,,R)|(ファイル名,新ファイル名[,R])}[,{ファイル名|(ファイル名,,R)|(ファイル名,新ファイル名[,R])}]…)

- **ファイル名** ~<引用文字列または特殊文字列>
複写するファイル名を指定する。
- **新ファイル名** ~<引用文字列または特殊文字列>
名前を変更して複写するとき、新しいファイル名を指定する。
- **R**
複写先の出力ディレクトリに、複写するファイルのファイル名または名前を変更して複写する場合は、新しいファイル名と同じ名称のファイルがすでに存在するとき、ファイルの置き換えを指定する。

GROUP=({ファイル総称名|(ファイル総称名,R)}[,{ファイル総称名|(ファイル総称名,R)]…)

- **ファイル総称名** ~<引用文字列または特殊文字列>
複写するファイルをファイル群の総称名（ファイル名の頭文字）で指定する。例えば、ABC と指定した場合、ファイル名が ABC で始まるすべてのファイルが対象となる。
- **R**
複写先の出力ディレクトリに、複写するファイルと同じ名称のファイルがすでに存在するとき、ファイルの置き換えを指定する。

■ 注意事項

- 同一コピーステップ内で EXCLUDE 文と同時に指定できない。
- 新ファイル名の指定やファイルの置き換えを指定する場合、(ファイル名,,R)、(ファイル名,新ファイル名,R)、および (ファイル総称名,R) の指定の途中で行を変えないこと。
- 同一コピーステップ内で同一のファイル名（新ファイル名も含む）、およびファイル総称名を指定しないこと。
- 同一コピーステップ内で SELECT 文は複数指定できる。
- 同一コピーステップ内で MEMBER と GROUP を同時に指定できない。
- COPY 文および INDD 文よりあとに記述すること。
- ファイル名およびファイル総称名には、パス名の区切りを示す文字（/：スラント）を指定できない。

- MEMBER オペランド指定とその他の指定では、複写するファイルの選択方法が異なる。
次に SELECT 文の MEMBER オペランド指定以外の場合と SELECT 文の MEMBER オペランド指定の場合の複写するファイルの選択方法を示す。

SELECT 文の MEMBER オペランド指定以外の場合

SELECT 文の GROUP オペランド指定, EXCLUDE 文指定, または SELECT 文および EXCLUDE 文を指定しない全ファイルの複写指定のときは, 入力ディレクトリの指定順に, 入力ディレクトリごとに複写処理する。

<例>入力ディレクトリ IN1→IN2→IN3 の順に, ファイル総称名が A と B のファイルを複写する。ただし, 入力ディレクトリ IN1 配下の対象ファイルは置き換えを行わない。

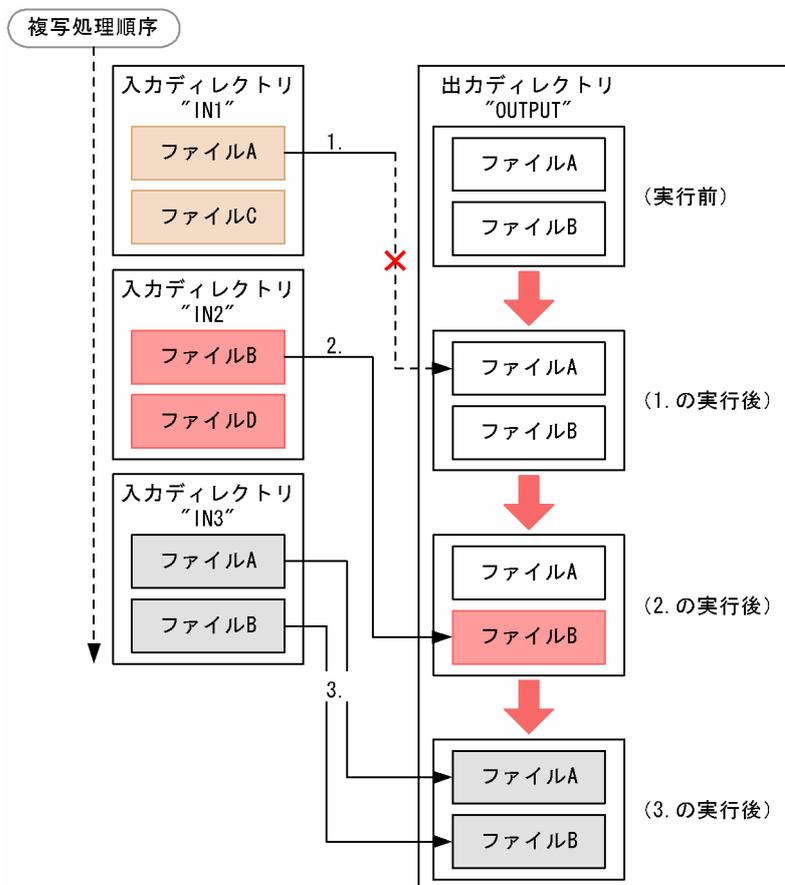
- 制御文

```
COPY  OUTDD=OUTPUT, INDD=(IN1, (IN2, R), (IN3, R))
SELECT GROUP=(A, B)
```

- 処理の流れ

- IN1 を参照する。指定に該当するファイル A は, 出力ディレクトリに同じ名称のファイルが存在し, 置き換え指定がないため複写しない。
- IN2 を参照する。指定に該当するファイル B は, 置き換え指定があるため複写する。
- IN3 を参照する。指定に該当するファイル A と B は, 置き換え指定があるため複写する。

図 3-35 複写処理の流れ (SELECT 文の MEMBER オペランド指定以外の場合)



SELECT 文の MEMBER オペランド指定の場合

SELECT 文の MEMBER オペランド指定は、指定されたファイル名単位に、入力ディレクトリの指定順を検索し、最初に見つかったファイルだけ複写する。

ファイルを置き換える指定がないため、複写しなかった場合も、ファイルは見つかったものとする。

<例>入力ディレクトリ IN1→IN2→IN3 の順に、ファイル名が A と B のファイルを検索し複写する。ただし、ファイル A は置き換えを行わない。

- 制御文

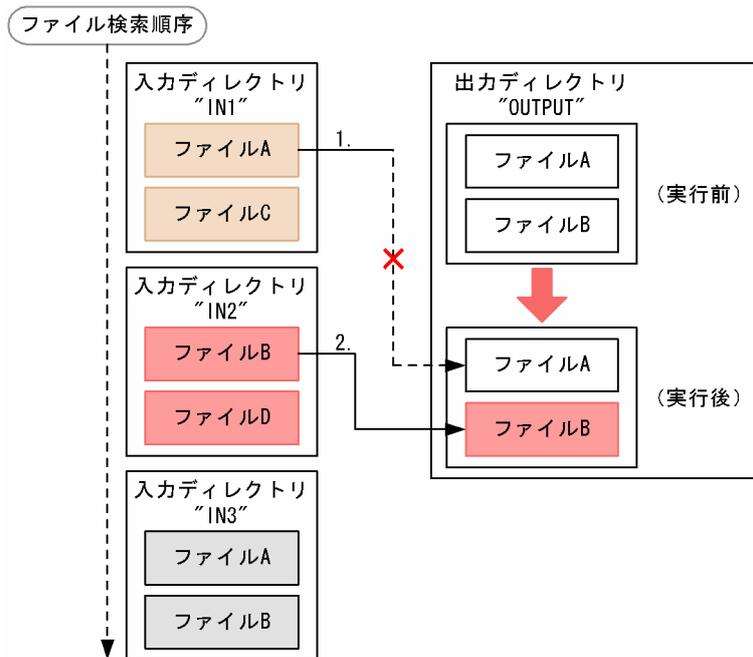
```
COPY OUTDD=OUTPUT, INDD=(IN1, IN2, IN3)
SELECT MEMBER=(A, (B, ., R))
```

- 処理の流れ

1. ファイル A を IN1 から検索する。IN1 でファイル A が見つかったが、出力ディレクトリに同じ名称のファイル A が存在し、置き換え指定がないため複写しない。この時点でファイル A の処理は終了する。

2. ファイル B を IN1 から検索する。IN2 でファイル B が見つかったため、これを複写する。この時点でファイル B の処理は終了する。

図 3-36 複写処理の流れ(SELECT 文の MEMBER オペランド指定の場合)



(c) EXCLUDE 文

■ 形式

[記号] Δ₁ EXCLUDE Δ₁

MEMBER=(ファイル名[, ファイル名]…)

■ オペランド

MEMBER=(ファイル名[,ファイル名]…) ~<引用文字列または特殊文字列>

出力したくないファイル名を指定する。

■ 注意事項

- 同一コピーステップ内で同一のファイル名を指定できない。
- 同一コピーステップ内で SELECT 文と同時に指定できない。
- EXCLUDE 文は複数指定できる。
- 同一コピーステップ内は COPY 文および INDD 文よりあとに記述すること。

(d) INDD 文

■ 形式

【記号】 Δ ₁	INDD=({DD名 (DD名,R)}[, {DD名 (DD名,R)}]…)
---------------------	--

■ オペランド

INDD=({DD名|(DD名,R)}[, {DD名|(DD名,R)}]…)

- DD名 ~<環境変数名>((1~31文字))

入力ディレクトリのパス名が設定された DD 名を指定する。

- R

出力ディレクトリ中に、複写するファイルのファイル名または SELECT 文 (MEMBER オペランド指定) で名前を変更して複写する場合は、新しいファイル名と同じ名称のファイルがすでに存在するとき、ファイルの置き換えを指定する。ファイルの置き換えを指定するには、DD 名と "R" を括弧で囲む。

■ 注意事項

- COPY 文と組み合わせて指定する。
- SELECT 文や EXCLUDE 文の直後に指定した場合は、コピーステップの開始と先行するコピーステップの終了を示す。出力ディレクトリは、先行するコピーステップの COPY 文から引き継がれる。

(5) 使用例

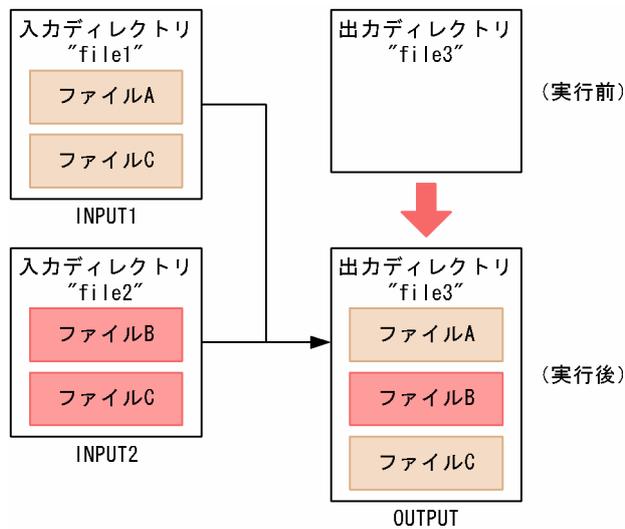
<例 1 > 2つの入力ディレクトリからすべてのファイルを複写する。なお、同じ名称のファイルは置き換えない。

次の手順で複写が実行される。

1. 制御文の COPY 文に指定した INDD オペランドの入力ディレクトリの指定順に、出力ディレクトリ "OUTPUT" へ複写する。

- 最初に入力ディレクトリ"INPUT1"にあるファイル A と C が複写される。
- 入力ディレクトリ"INPUT2"にあるファイル B と C の複写を行うが、ファイル C はすでに同じ名称のファイルが 2. で複写されているため複写しない（置き換え指定がないため）。

図 3-37 入力ディレクトリからすべてのファイルを複写（置き換え指定なし）



```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="FILECOPY" >
    <STEP NAME="STEP0001" >
      <EXEC PGM="bjexu_pcopy" /> ... (1)
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" /> ... (2)
      <DD NAME="INPUT1" TYPE="DIR" DSN="file1" DISP="OLD" /> ... (3)
      <DD NAME="INPUT2" TYPE="DIR" DSN="file2" DISP="OLD" /> ... (4)
      <DD NAME="OUTPUT" TYPE="DIR" DSN="file3" DISP="OLD" /> ... (5)
      <DD NAME="SYSIN" TYPE="DATA" > ... (6)
      <![CDATA[
!
COPY OUTDD=OUTPUT, INDD=(INPUT1, INPUT2) ... (7)
!
      ]]>
    </DD>
  </STEP>
</JOB>
</HitachiBatchJobExec>

```

- (1)bjexu_pcopy を実行する。
- (2)メッセージ出力先に SYSOUT データファイルの A クラスを指定する。
- (3)入力ディレクトリを定義する。ディレクトリは、"file1"を指定する。
- (4)入力ディレクトリを定義する。ディレクトリは、"file2"を指定する。
- (5)出力ディレクトリを定義する。ディレクトリは、"file3"を指定する。
- (6)制御文の入力元を定義する。DD 要素の中に入力データがあることを指定する。

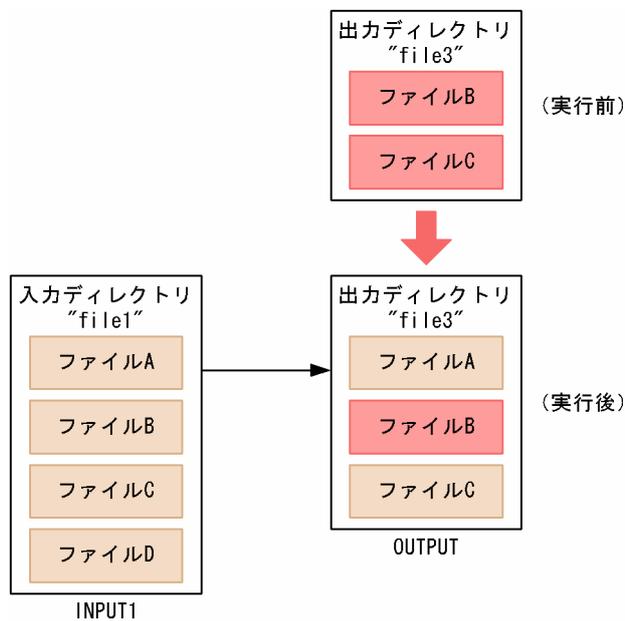
(7)コピーオペレーションの開始を示す。出力ディレクトリとして OUTPUT(ディレクトリ file3)を指定し、入力ディレクトリとして INPUT1 (ディレクトリ file1) と INPUT2 (ディレクトリ file2) を指定する。

<例 2 >入力ディレクトリからファイル名が A と C のファイルを置き換え指定ありで複写する。なお、出力ディレクトリには、ファイル名が B と C のファイルがすでに存在する。

次の手順で複写が実行される。

1. 制御文の COPY 文に指定した入力ディレクトリ"INPUT1"から、出力ディレクトリ"OUTPUT"へ複写する。ただし、複写するファイルは SELECT 文の指定に従う。
2. SELECT 文の指定に従い、"INPUT1"に存在するファイル A と C を複写する。このとき、"OUTPUT"にはすでに同じ名称のファイル C が存在するが、置き換え指定があるため"INPUT1"のファイルで置き換えられる。

図 3-38 入力ディレクトリからすべてのファイルを複写 (置き換え指定あり)



```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="FILECOPY" >
    <STEP NAME="STEP0001" >
      <EXEC PGM="bjexu_pcopy" /> ... (1)
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" /> ... (2)
      <DD NAME="INPUT1" TYPE="DIR" DSN="file1" DISP="OLD" /> ... (3)
      <DD NAME="OUTPUT" TYPE="DIR" DSN="file3" DISP="OLD" /> ... (4)
      <DD NAME="SYSIN" TYPE="DATA" > ... (5)
      <![CDATA[
!
COPY OUTDD=OUTPUT, INDD=(INPUT1) ... (6)
SELECT MEMBER=((A, . R), (C, . R)) ... (7)
!
      ]]>
    </DD>
  </STEP>
</JOB>
</HitachiBatchJobExec>
  
```

(1)bjexu_pcopy を実行する。

(2)メッセージ出力先に SYSOUT データファイルの A クラスを指定する。

(3)入力ディレクトリを定義する。ディレクトリは, "file1"を指定する。

(4)出力ディレクトリを定義する。ディレクトリは, "file3"を指定する。

(5)制御文の入力元を定義する。DD 要素の中に入力データがあることを指定する。

(6)コピーオペレーションの開始を示す。出力ディレクトリとして OUTPUT (ディレクトリ file3) を指定し, 入力ディレクトリとして INPUT1 (ディレクトリ file1) を指定する。

(7)メンバの取り出しを指定する。入力ディレクトリ INPUT1 からメンバ A,C をメンバの置き換え指定で出力ディレクトリに複写する。

また, この例の場合は, 次の制御文でも同じである。

<pre>COPY OUTDD=OUTPUT, INDD=((INPUT1, R))</pre>	... (1)
<pre>SELECT MEMBER=(A, C)</pre>	... (2)

(1)コピーオペレーションの開始を示す。出力ディレクトリとして OUTPUT (ディレクトリ file3) を指定し, 入力ディレクトリとして INPUT1 (ディレクトリ file1) を指定する。また, INPUT1 で指定したディレクトリに存在するファイルは, メンバの置き換え指定を行う。

(2)メンバの取り出しを指定する。入力ディレクトリ INPUT1 からメンバ A,C を取り出し, 出力ディレクトリに複写する。

3.18.7 bjexu_scopy (ファイルのレコード単位の複写)

(1) 機能

ファイルの複写, ファイルの内容をレコードの単位で操作しての複写, および入力ファイルを分割して複数のファイルに複写できる。ファイルを複写するとき, 既にあるファイルを出力ファイルに指定し DD 要素で DISP="MOD"を指定することで, 追加書きができる。

(a) ファイルの複写

- 単純複写

入出力ファイルのファイル情報 (ファイル形式, レコード長) が同一のとき, 入力ファイルの内容をそのまま出力ファイルに複写する。

入力ファイルのファイル情報の指定がないときは, 入力ファイルとして指定されたファイルの実際の形式に関係なく, 入力ファイルの内容をそのまま出力ファイルに複写する。

- 形式変換

入力ファイルのファイル情報と異なるファイル情報を出力ファイルに指定した場合、ファイル形式を変換して複写する。

入力ファイルや出力ファイルには、PREST ファイルも指定できる。

ファイルを複写するとき指定できる入出力ファイル形式の組み合わせを次の表に示す。

表 3-41 入出力ファイル形式の組み合わせ

入力ファイル形式	出力ファイル形式		
	F	V	A
F	◎	◎	×
V	△	◎	×
A	×	×	○

(凡例)

◎：指定できる組み合わせ。

○：指定できる組み合わせ。ただし、単純複写だけができ、形式変換はできない。

×：指定できない組み合わせ。

△：指定できない組み合わせ。ただし、RECORD 文でレコードの編集を指定することで、指定できる組み合わせとなるが、入力レコードを超えないフィールド定義が必要である。詳細については、「(4) [bjexu_scOPY 制御文](#)」を参照のこと。

F：固定長順編成ファイル（COBOL で作成したファイルでファイルサイズがレコード長の整数倍であるバイナリファイル）。

V：可変長順編成ファイル（COBOL で作成したファイル）。

A：テキストファイル（COBOL またはテキストエディタで作成したファイル）。改行コードでレコードが区切られる（UNIX と Windows では改行コードが異なる）。

• 入力ファイルの連結

複数のファイルを DD 要素で連結して入力する。連結するファイルの属性は、同一でなければならない。ただし、コロン（:）は、連結されたデータセット名が環境変数に設定されるとき区切りとして使用しているため、ファイル名に使用できない。

入力ファイルのファイル情報の指定がないときは、可変長順編成ファイルを連結することはできない。可変長順編成ファイルを連結するには、入力ファイルのファイル形式を指定し、GENERATE 文を指定する必要がある。

PREST ファイルは、連結できない。

• 入力レコードの連結

固定長順編成ファイルの複数のレコードを連結して、連結した複数のレコードを論理的な 1 入力レコードとして処理する。

• 出力ファイルへの追加書き

環境変数 BJEX_SCOPY_DISP_MOD の定義によって、MOD として処理するか、または OLD として処理するかを選択できる。環境変数で MOD を選択したとき、DD 要素で DISP="MOD" を指定すると、該当するファイルに対して追加書きができる。追加書きが有効なファイル種別などの詳細は、「3.4.2(1)(e) MOD パラメータ」を参照。

環境変数 BJEX_SCOPY_DISP_MOD の設定値ごとの内容を次の表に示す。

表 3-42 環境変数 BJEX_SCOPY_DISP_MOD の設定と内容

環境変数 BJEX_SCOPY_DISP_MOD の設定	内容
MOD	MOD として処理する
OLD	OLD として処理する
MOD と OLD 以外、または設定されていない	MOD として処理する

(b) ディレクトリにファイル作成

制御文で順編成の入力をグループに分割すると、各グループを 1 個のファイルにして、ディレクトリにファイルが作成される。

また、出力のディレクトリが既存のものであれば、これら各グループは追加のファイルとして、ディレクトリに新たに作成される。ただし、出力ディレクトリに同一ファイル名がある場合には、そのファイルは作成されない。

レコードのフィールドと、そのフィールド内の文字の並びを指定することでグループが分割され、指定した文字の並びをレコード内に見つけると、そこまでのレコードをグループとして 1 個のファイルが作成される。この順編成の入力を分割してできた 1 つのグループをレコードグループと呼ぶ。

(c) レコードの編集

入力ファイルの各レコードの指定したフィールドを編集して、出力ファイルのレコードを作成する。指定した文字の並びをレコード内に見つけると、そこまでのレコードをレコードグループとして、1 個のファイルが作成される。レコードグループの指定は、ディレクトリにファイルを作成する場合と同じである。

レコード編集できるのは、固定長順編成ファイルと可変長順編成ファイルである。テキストファイルはレコード編集できない。

レコードの編集には、次の方法がある。

図 3-39 レコードの編集方法



(1)入力レコードの指定位置から指定した長さのフィールド（内の情報）を、出力レコードの指定位置へ移す。

(2)入力レコードの指定位置から指定した長さのフィールド（内の情報）を変換して、出力レコードの指定位置へ移す。変換には次の2とおりがあある。

- PZ 変換

入力フィールドのデータを内部 10 進形式と見なし、外部 10 進形式に変換する。このとき、入力フィールドの長さを L とすると、出力フィールド長は、出力長の指定がなければ $2L-1$ となる。なお、入力データの妥当性はチェックしない。

- ZP 変換

入力フィールドのデータを外部 10 進形式と見なし、内部 10 進形式に変換する。このとき、入力フィールドの長さを L とすると、出力フィールド長は、出力長の指定がなければ $(L+2) / 2$ を超えない最大の整数となる。なお、入力データの妥当性はチェックしない。

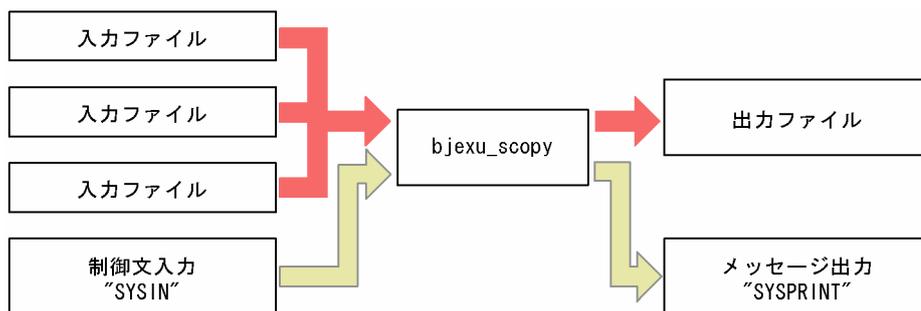
(3)出力レコードの指定位置から指定の文字列を埋め込む。

(4)未編集フィールドへ埋め込み文字をセットする。未編集フィールド（出力レコード上で編集が指定されなかった部分）には、空白または制御文での指定文字を埋め込む。

(2) 入出力構成

bjexu_scopy の入出力構成を次の図に示す。

図 3-40 bjexu_scopy の入出力構成



(a) 入力

- 入力ファイル

複写の入力となる、複写元のファイル。

PREST ファイルやダミーファイルも指定できる。

- 制御文 (SYSIN)

bjexu_scopy の処理内容を指定する。制御文には次のものがある。

- GENERATE 文
- RECORD 文
- MEMBER 文
- INPFILE 文

- OUTFILE 文

(b) 出力

- 出力ファイル
複写の出力となる、複写先のファイル。
PREST ファイルやダミーファイルも指定できる。
- メッセージ (SYSPRINT)
bjexu_scopy が出力するメッセージの出力先。

(c) 終了コード

bjexu_scopy の終了コードを次の表に示す。

表 3-43 bjexu_scopy の終了コード

終了コード	意味
0	正常に処理を終了した。
4	エラーが発生したが無視して続行した。詳細については、出力されたメッセージを参照のこと。*
12	次に示すような回復できないエラーが発生したため、処理を打ち切り、ジョブステップを終了する。文法解析終了後の複写処理中にエラーが発生した場合、出力ファイルの内容は保証されない。 <ul style="list-style-type: none"> • 実行できない制御文を指定している。 • 制御文や入出力ファイルのオープンエラー、I/O エラー

注※ 文法的に正しいが、要求された処理を実行できなかった場合、または SYSPRINT でオープンエラーが発生した場合の終了コードは 4 になる。

(3) ジョブ定義 XML ファイル

bjexu_scopy を実行する場合は、次に示す EXEC 要素と DD 要素の指定が必要である。

(a) EXEC 要素

EXEC 要素の PGM に bjexu_scopy を指定する。

```
<EXEC PGM="bjexu_scopy" />
```

(b) SYSPRINT の DD 要素

bjexu_scopy が発行するメッセージの出力先を指定する。なお、DD 連結指定はできない。出力先には、SYSOUT データファイルや恒久ファイルを指定できる。

SYSPRINT が使用できない状態の場合、経過メッセージは出力されない。出力メッセージの詳細については、「10. メッセージと終了コード」を参照のこと。

(c) SYSUT1 の DD 要素

複写元となる入力ファイルを指定する。

複数のファイルを連結して指定できる。このとき、各ファイルの属性は一致していなければならない。なお、ファイルの属性が一致しているかのチェックは行わない。固定長順編成ファイルの場合、指定されたレコード長で処理し、最終レコードの長さが不一致のときは、終了コードが 12 で終了する。

PREST ファイルは、連結できない。

レコード数が 0 件のファイルを指定してもエラーにはならないで、終了コード 0 で終了する。

(d) SYSUT2 の DD 要素

複写先となる出力ファイルを指定する。また、複数のファイルに分割して出力するときは、複写先となる出力ディレクトリを指定し、ファイル名を MEMBER 文で指定する。ディレクトリを指定し、MEMBER 文で指定しないとエラーになる。なお、DD 連結指定はできない。

(e) SYSIN の DD 要素

bjexu_scopy 制御文を指定する。または、この制御文が記述されたファイルのパスを指定する。DD 連結指定はできない。

(4) bjexu_scopy 制御文

bjexu_scopy 制御文と機能を次の表に示す。

表 3-44 bjexu_scopy 制御文と機能

制御文名	機能
INPFILE 文	入力ファイルのファイル形式を指定する。
OUTFILE 文	出力ファイルのファイル形式を指定する。
GENERATE 文	ファイルの複写、編集の開始を指定する。
MEMBER 文	ファイルを指定する。
RECORD 文	グループに分割するための識別文字とレコード編集の情報を指定する。

bjexu_scopy 制御文に指定できる組み合わせと処理内容の対応を次の表に示す。

表 3-45 bjexu_scOPY 制御文に指定できる組み合わせと処理内容

制御文					入力ファイル形式			処理内容	
INP FILE	OU TFIL E	GENERATE		ME MBE R	REC OR D	F	V		A
		LEN GTH あり	LEN GTH なし						
×	×	×	×	×	×	指定できる	指定できる	指定できる	単純複写
×	×	×	○	×	×	指定できる	指定できる	指定できる	単純複写
○	○*1	×	×	×	×	指定できる	指定できる	指定できる	形式変換/単純複写*2
○	○*3	○*4	—	×	×	指定できる	指定できない	指定できない	形式変換
○	○*1	—	○	×	×	指定できる	指定できる	指定できる	形式変換/単純複写*2
○	○*3	○*4	—	○	○*5	指定できる	指定できない	指定できない	ディレクトリにファイル作成
○	○*1	—	○	○	○*5	指定できる	指定できる	指定できない	ディレクトリにファイル作成
○	○*3	○*4	—	×	○	指定できる	指定できない	指定できない	レコード編集
○	○*1	—	○	×	○	指定できる	指定できる	指定できない	レコード編集

(凡例)

- ：指定あり。
- ×：指定なし。
- ：同時指定できない。
- F：固定長順編成ファイル。
- V：可変長順編成ファイル。
- A：テキストファイル。

注※1

省略できる。INPFILE 文の指定値を仮定する。

注※2

次のどちらかに該当するとき、単純複写になる。ただし、入力ファイルが可変長順編成ファイルで DD 連結されている場合を除く。

- INPFILE 文と OUTFILE 文の指定が同じ。
- OUTFILE 文でレコード長の指定を省略、または OUTFILE 文の指定を省略。

注※3

省略できる。GENERATE 文の LENGTH オペランドの指定値を仮定する。

注※4

INPFILE が固定長順編成ファイルのとき、GENERATE 文の LENGTH オペランドを指定できる。可変長順編成ファイルやテキストファイルのときは指定しないこと。

注※5

省略できる。

(a) INPFILE 文

■ 形式

[記号] Δ ₁ INPFILE Δ ₁	FILEORG=ファイル形式[, RECLEN=レコード長][, RECCOUNT][, CHKEMPTY]
--	--

■ オペランド

FILEORG=ファイル形式

bjexu_scopy 制御文の入力ファイルのファイル形式を指定する。

- ファイル形式～((F | V | A))

F：固定長順編成ファイル（COBOL で作成したファイル、ファイルサイズがレコード長の整数倍であるバイナリファイル）であることを示す。

V：可変長順編成ファイル（COBOL で作成したファイル）であることを示す。

A：テキストファイル（COBOL またはテキストエディタで作成したファイル）であることを示す。テキストファイルの各行（テキストレコード）は、改行文字（X'0A'）で区切られている。

RECLEN=レコード長

bjexu_scopy 制御文の入力ファイルのレコード長を指定する。

- レコード長 ～<符号なし整数>((1～65535))

レコード長を指定する。FILEORG=F を指定した場合は、必ず指定する必要がある。その他の形式のときは指定しないこと。指定した場合はエラーとなる。

RECCOUNT

単純複写時、ファイルのレコードサイズに基づいたレコード数を KAKD2138-I メッセージに表示することを指定する。省略時は、レコード長が 65535 バイトのレコードを出力したときのレコード数を表示する。

固定長順編成ファイルの場合、FILEORG=F と RECLEN オペランドとともに指定する必要がある。

可変長順編成ファイルを単純複写する場合、レコード単位に入出力を行うので、RECCOUNT オペランド指定がない単純複写と比較して性能が低下する。

入力ファイルがテキストファイルのときは、指定できない。指定すると、エラーとなる。

CHKEMPTY

入力が可変長順編成ファイルのとき、レコード数が 0 件のファイルであればエラーにすることを指定する。省略すると、入力ファイルのレコード数が 0 件であってもエラーにしないで、レコード数が 0 件の出力ファイルを作成する。

入力ファイルが可変長順編成ファイル以外の場合は、指定できない。指定すると、エラーとなる。単純複写のときは、指定しても無効で、エラーにならない。

■ 注意事項

- 制御文の先頭に指定すること。
- ほかの制御文が1つも指定されていないとき、および LENGTH オペランド指定なしの GENERATE 文だけを指定したときは省略できる。その他の場合は省略できない。
- すべての制御文が省略されたとき、および LENGTH オペランド指定なしの GENERATE 文だけを指定したときは、入力ファイルの実際の形式に関係なく、入力ファイルをそのまま出力ファイルに複写する。可変長順編成ファイルを DD 連結するときは、INPFIL 文で FILEORG=V を指定する必要がある。

(b) OUTFILE 文

■ 形式

[記号] Δ ₁ OUTFILE Δ ₁	FILEORG=ファイル形式[, RECLEN=レコード長][, MINLEN=最小レコード長]
--	--

■ オペランド

FILEORG=ファイル形式

bjexu_scopy 制御文の出力ファイルのファイル形式を指定する。

- ファイル形式～((F | V | A))

F：固定長順編成ファイル（COBOL で作成したファイル、ファイルサイズがレコード長の整数倍であるバイナリファイル）であることを示す。

V：可変長順編成ファイル（COBOL で作成したファイル）であることを示す。

A：テキストファイル（COBOL またはテキストエディタで作成したファイル）であることを示す。テキストファイルの各行（テキストレコード）は、改行文字（X'0A'）で区切られている。

RECLEN=レコード長

bjexu_scopy 制御文の出力ファイルのレコード長を指定する。

- レコード長 ～<符号なし整数>((1~65535))

レコード長を指定する。可変長順編成ファイルの場合は最大レコード長を指定する。テキストファイルの場合は指定しないこと。指定した場合はエラーとなる。

出力が固定長順編成ファイルの場合

- RECLEN オペランドの指定値が出力ファイルのレコード長になる。

単純複写を除く複写や編集時に、RECLEN オペランドを省略した場合、次のように仮定する。

- 入力が可変長順編成ファイルのときは、FIELD 編集の結果のレコードの長さがレコード長になる。

- 入力が固定長順編成ファイルのときは、環境変数 BJEX_SCOPY_FIXRECLEN の定義によって出力レコード長の求め方を選択できる。指定値ごとの出力レコード長を決める優先順位を次の表に示す。

表 3-46 出力レコード長の決定方法

環境変数 BJEX_SCOPY_FIXRECLEN の設定	出力レコード長の優先順位		
	FIELD	GENERATE	INPFILE
FIELD	1	2	3
NOFIELD	—	1	2
FIELD と NOFIELD 以外、または設定されていない	1	2	3

(凡例)

FIELD：RECORD 文の FIELD オペランドで編集した結果の長さ。

GENERATE：GENERATE 文の LENGTH オペランドの指定値。

INPFILE：INPFILE 文の RECLEN オペランドの指定値。

1～3：出力レコード長を決定する優先順位（優先順位は 1 が最も高い）。

—：対象外。

- RECORD 文の FIELD オペランドで編集した結果の長さが出力レコード長を超えるとエラーになる。
- RECORD 文や RECORD 文の FIELD オペランドの指定がない場合、入力レコード長が出力レコード長と一致しないときは、次に示す処理となる。
「入力レコード長>出力レコード長」の場合、後ろを切り捨てる。
「入力レコード長<出力レコード長」の場合、残りは RECORD 文の FILL オペランドで指定した文字を埋め込む。FILL オペランドの指定を省略した場合は空白 (X'20') を埋め込む。

出力が可変長順編成ファイルの場合

- 単純複写を除く複写や編集処理で RECLEN オペランド省略時は、最大レコード長に 65535 を仮定する。
- 出力ファイルの最大レコード長を超える編集はエラーになる。次に示す処理となる。
「入力または編集結果のレコード長>出力ファイルの最大レコード長」の場合はエラーとなる。
「入力または編集結果のレコード長≤出力ファイルの最大レコード長」の場合、入力または編集結果のレコード長の長さで出力する。

MINLEN=最小レコード長

出力ファイルが可変長順編成ファイルのとき、最小レコード長を指定する。このオペランドを指定すると、最小レコード長より短いレコードを出力しようとしたときにエラーになるので、必要な情報が含まれていることをチェックできる。

固定長順編成ファイルやテキストファイルの場合は指定しないこと。指定した場合はエラーとなる。

• 最小レコード長 ～<符号なし整数>((1～65535))

単純複写を除く複写や編集処理で MINLEN オペランド省略時は、最小レコード長に 1 を仮定する。出力ファイルの最小レコード長に満たない編集はエラーとなる。

■ 注意事項

- INPFILE 文の次に指定すること。
- RECLEN および MINLEN オペランドを指定するときは、「最大レコード長 \geq 最小レコード長」になるように指定する。指定に誤りがあると、エラーとなる。

(c) GENERATE 文

■ 形式

[記号] Δ_1 GENERATE Δ_1	[LENGTH=入力論理レコード長]
-------------------------------------	--------------------

■ オペランド

LENGTH=入力論理レコード長

- 入力論理レコード長 \sim <符号なし整数>((1~65535))

固定長順編成ファイルのレコードを連結、または切り捨てるときに指定する。

入力論理レコード長を指定する。このオペランドで、入力ファイルのレコード長をオペランドで指定した入力論理レコード長に変更し、複数の入力レコードを連結して処理できる。

入力論理レコードは、複写や編集のための新たな入力レコードになり、その編集結果が出力レコードになる。

このオペランドを指定した場合、次の手順でレコード連結される。

1. 指定した入力論理レコード長に達するまで、複数の入力レコードを連結する。
2. 入力論理レコード長が入力レコード長の倍数でない場合、連結後の端数部分を切り捨てる。
3. 連結後の入力レコードの長さが入力論理レコード長に達しない場合、出力しない。
4. 上記の手順で連結したレコードを1つの入力論理レコードとして処理する。したがって、レコードを編集する場合、入力位置は連結後の入力論理レコードに対するものを指定する。

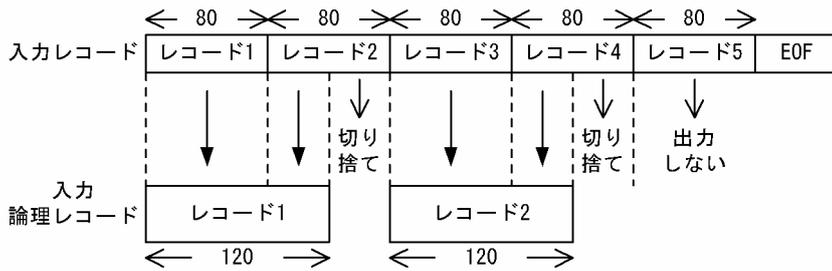
このオペランドは、入力ファイル形式が固定長順編成ファイルのときだけ指定できる。省略した場合はレコードを連結しない。

このオペランドを指定して出力ファイルのレコード長を省略した場合、出力ファイルのレコード長にこのオペランドで指定した値が仮定される。なお、可変長順編成ファイルおよびテキストファイルにこのオペランドを指定した場合はエラーとなる。

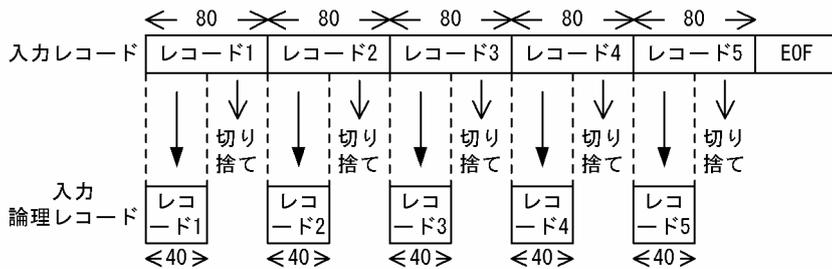
このオペランドを使用した例を次に示す。

図 3-41 LENGTH オペランドを使用した例

●LENGTH=120を指定したとき



●LENGTH=40を指定したとき



■ 注意事項

- INPFILE 文と OUTFILE 文の次に記述すること。
- GENERATE 文は 1 行だけ指定できる。1 つのステップ内に 2 行以上記述するとエラーになる。

(d) MEMBER 文

■ 形式

[記号] Δ ₁ MEMBER Δ ₁	NAME=(ファイル名)
---	--------------

■ オペランド

NAME=(ファイル名) ~<引用文字列または特殊文字列>

作成するファイル名を指定する。SYSUT2 の指定がディレクトリの場合に指定できる。

この制御文に続き、RECORD 文でレコードグループを指定する。1 つの MEMBER 文に対して複数個の RECORD 文を指定できる。RECORD 文がなければ、入力ファイルの終わりまでが 1 つのファイルとして、このオペランドで指定したファイル名が与えられる。

ファイル名には、パス名の区切りを示す文字 (/ : スラント) を指定できない。

■ 注意事項

- 出力がディレクトリでない場合、オープンエラーになる。
- 複数の MEMBER 文を指定するときは、RECORD 文の IDENT オペランドでレコードグループを指定する必要がある。MEMBER 文を 1 つだけ指定したとき、および複数の MEMBER 文を指定したときの最後の RECORD 文の IDENT オペランドは省略できる。IDENT オペランドを省略すると、ファイルの終わりまでが同一のレコードグループとして扱われる。

- MEMBER 文を 1 つだけ指定するときは、RECORD 文を省略できる。また、複数の MEMBER 文を指定するとき、最後の MEMBER 文に対する RECORD 文は省略できる。RECORD 文を省略すると、ファイルの終わりまでが同一のレコードグループとして扱われる。
- 先頭の MEMBER 文の前に RECORD 文を指定した場合、エラーになる。
- MEMBER 文で指定したファイルがすでに存在するときは、エラーになる。

(e) RECORD 文

■ 形式

[記号] Δ ₁ RECORD Δ ₁	[IDENT=(文字列長, 最終レコード情報, 入力位置)] [, FIELD=([フィールド長], [入力位置 フィールド情報], [PZ ZP] [, 出力位置], [出力長])]… [, FILL=埋め込み文字 X'埋め込み文字']
---	--

■ オペランド

IDENT=(文字列長, 最終レコード情報, 入力位置)

順編成の入力をレコードグループに分割するとき、レコードグループの最終レコードを識別する情報を指定する。

- **文字列長** ~<符号なし整数>((1~8))
最終レコード情報の文字列長を示す。
- **最終レコード情報** ~<引用文字列>((1~8 バイト))
最終レコードを識別する文字列を指定する。文字列長サブオペランド値を指定した場合、このオペランドに指定した文字列（先頭と末尾の引用符は除く）の長さは、文字列長サブオペランド値と一致しなければならない。
- **入力位置** ~<符号なし整数>((1~65535))
レコードの先頭からの位置を指定する。GENERATE 文で LENGTH オペランドを指定しているときは、入力論理レコードの先頭からの位置を指定する。
レコードの指定された入力位置から指定された長さが、最終レコードの情報で示された文字列と一致すれば、それが最終レコードとなる。
複数の RECORD 文を連続して指定できる。複数の RECORD 文を指定するときは、IDENT オペランドでレコードグループを指定する必要がある。最後の RECORD 文だけは、IDENT オペランドを省略できる。

IDENT オペランドを省略すると、ファイルの終わりまでが同一のグループとして扱われる。

FIELD=([フィールド長], [入力位置 | フィールド情報], [PZ | ZP], [出力位置], [出力長])

レコードの編集方法を指定する。1 つの RECORD 文に複数指定できる。

出力ファイルが可変長順編成ファイルのとき、編集してできたレコードの長さがレコード長になる。

入力レコードまたは入力論理レコードの内容を編集しないで出力レコードの先頭に複写するときは、FIELD オペランドを省略できる。それ以外の場合は、FIELD オペランドで編集方法を指定する必要がある。

FIELD オペランドを省略したとき、環境変数 BJEX_SCOPY_NOFIELD_TRUNC の定義によって複写する入力レコードを 80 バイトに切り詰めるかどうかを選択できる。指定値ごとの動作を次の表に示す。

表 3-47 環境変数 BJEX_SCOPY_NOFIELD_TRUNC の設定ごとの制限

環境変数 BJEX_SCOPY_NOFIELD_TRUNC の設定	出力レコード長
YES	80 バイトに切り詰める
NO	80 バイトに切り詰めない
YES と NO 以外、または設定されていない	80 バイトに切り詰めない

- フィールド長** ～<符号なし整数>《80》((1～65535))
 入力フィールドの長さを示す。複写の場合は、出力フィールドの長さに一致する。変換の場合には、出力フィールドの長さが変化する。
 出力位置からレコード長の最後までまでの長さが指定できる最大値になる。
- 入力位置** ～<符号なし整数>《1》((1～65535))
 入力フィールドの位置を入力レコードの先頭からのバイト数で示す（レコードの先頭を 1 とする）。GENERATE 文で LENGTH オペランドを指定しているときは、入力論理レコードの先頭からの位置を指定する。
- フィールド情報** ～<引用文字列>((1～40 バイト))
 文字列を埋め込む場合、埋め込む文字列をアポストロフィで囲んで指定する。最大で 40 バイト指定できる。

PZ | ZP

PZ を指定した場合、入力フィールドのデータを内部 10 進形式とし、外部 10 進形式に変換する。

ZP を指定した場合、外部 10 進形式から内部 10 進形式に変換する。

PZ も ZP も指定しなかった場合、入力フィールドのデータは文字列となり、変換しないで転送する。変換での妥当性はチェックしない。あらかじめ正しい形式のデータを設定しておく必要がある。

変換後の内部 10 進形式の符号（最終バイト下位 4 ビット）は、正の場合 (1100)₂、負の場合は (1101)₂ とする。変換元内部 10 進数の最終バイト下位 4 ビットが (1101)₂ の場合は負とし、(1101)₂ 以外であれば正となる。

外部 10 進形式の符号位置は、最終バイト上位 4 ビットとする。変換後は正の場合は 3、負の場合は 7 とする。変換前の外部 10 進数の最終バイト上位 4 ビットが 7 の場合は負とし、7 以外であれば正となる。外部 10 進形式のゾーン部（各バイトの上位 4 ビット）は 3 とする。

外部 10 進数は次に示す形式とする。

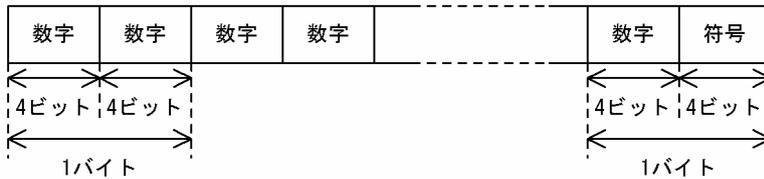


ゾーン部には 3 の値を，数字部には 0~9 の値を指定する。
また，符号部は次に示す値のどちらかを指定する。

3：正

7：負

内部 10 進数は次に示す形式とする。



数字部には 0~9 の値を指定する。

また，符号部には次に示す値のどれかを指定する。

(1100)₂：正

(1101)₂：負

(1111)₂：絶対値（変換した結果がこの値になることはない）

出力位置 ～<符号なし整数>《1》((1~65535))

編集後のデータの出力フィールドの開始位置を示す（レコードの先頭を 1 とする）。

FIELD オペランドの指定順に処理する。直前の結果と重なったときは，上書きされる。

出力長 ～<符号なし整数>《80》((1~65535))

入力フィールドのデータ変換（PZ，ZP 変換）を指定した場合，出力フィールドの長さを指定する。

この指定を省略すると，次の値が仮定される。なお，「↓（数値）↓」は小数点切り捨てを表す。

PZ 変換の時：入力フィールド長×2-1

ZP 変換の時：↓（入力フィールド長+2）÷2↓

出力長が上記の値より短い場合，変換後のフィールドの右端から指定した長さのフィールドが出力される。

出力長が上記の値より長い場合，PZ 変換のときは (30)₁₆ が，ZP 変換のときは (00)₁₆ が左端から埋め込まれる。

データ変換しない場合，出力長を指定しても無視される。

PZ 変換と ZP 変換時の入力フィールドと出力フィールドの関連を次に示す。なお，フィールド内の数値は 16 進数で記述している。

・ PZ変換のフィールド状態

入力フィールド

12	34	5C
----	----	----

出力長指定なし

出力長=3

出力長=7

出力フィールド

31	32	33	34	35
----	----	----	----	----

33	34	35
----	----	----

30	30	31	32	33	34	35
----	----	----	----	----	----	----

・ ZP変換のフィールド状態

入力フィールド

31	32	33	34
----	----	----	----

出力長指定なし

出力長=2

出力長=5

出力フィールド

01	23	4C
----	----	----

23	4C
----	----

00	00	01	23	4C
----	----	----	----	----

FILL=埋め込み文字 | X'埋め込み文字'

- 埋め込み文字 ~<引用文字列> 《空白》 ((1 文字))
- X'埋め込み文字' ~< 16 進数字> 《20》 ((2 文字))

未編集フィールドに埋め込む文字を指定する。X を指定した場合には 16 進数字で指定する。

■ 注意事項

- IDENT オペランドは、RECORD 文に複数指定されたとき、最後の指定を有効にする。
- FILL オペランドは、RECORD 文に複数指定されたとき、最後の指定を有効にする。
- 入力位置とフィールド長は、入力レコードまたは入力論理レコード外を指定すると、エラーになる。
- フィールド長、出力位置、および出力長は、編集の結果が出力レコード長を超えない値を指定すること。出力ファイルが固定長順編成ファイルの場合、OUTFILE 文の RECLLEN オペランドで出力レコード長を指定したとき、FIELD オペランドの指定がレコード長を超える指定はエラーになる。RECLLEN オペランドの指定がないときは、環境変数 BJEX_SCOPY_FIXRECLLEN の定義によって出力レコード長の求め方を選択できる。詳細については、「3.18.7(4)(b) OUTFILE 文」を参照のこと。出力ファイルが可変長順編成ファイルのとき、出力ファイルの最大レコード長を超える編集はエラーになる。
- 入力レコードに有効なデータがない場合、空（ゼロバイト）のファイルができることがある。入力ファイルや編集結果のレコードがゼロバイトの場合などが該当する。

(5) 使用例

<例 1> 入力の固定長順編成ファイルを出力ファイルに単純複写する。

図 3-42 入力の固定長順編成ファイルを出力ファイルに単純複写

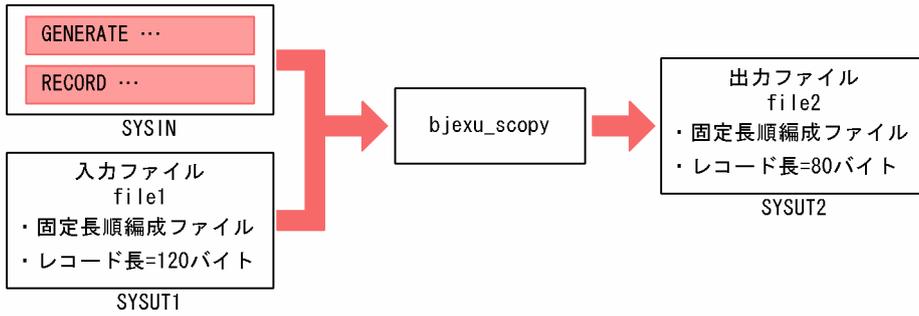


```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB_NAME="SAMPLE1" >
    <STEP_NAME="STEP0001" >
      <EXEC_PGM="bjexu_scopcy" /> ... (1)
      <DD_NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" /> ... (2)
      <DD_NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="OLD" /> ... (3)
      <DD_NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" /> ... (4)
      <DD_NAME="SYSIN" TYPE="DATA" > ... (5)
        <![CDATA[
!
  INPFILE FILEORG=F, RECLEN=80 ... (6)
!
        ]]>
      </DD>
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

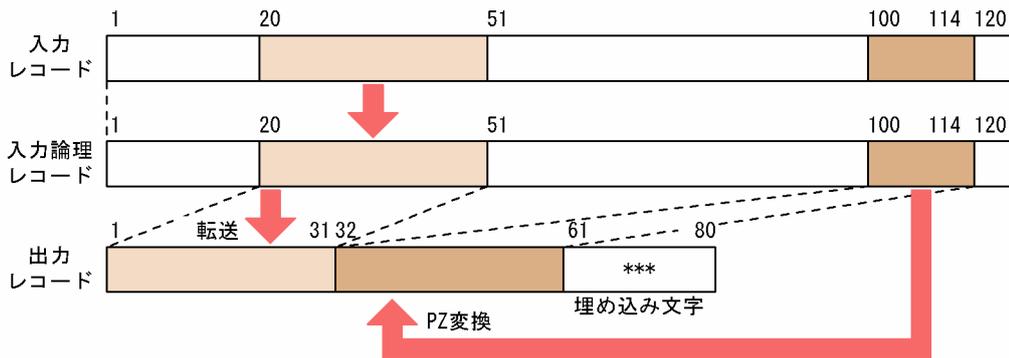
- (1)bjexu_scopcy を実行する。
- (2)メッセージ出力先に SYSOUT データファイルの A クラスを指定する。
- (3)入力ファイルを定義する。ファイルは, "file1"を指定する。
- (4)出力ファイルを定義する。ファイルは, "file2"を指定する。
- (5)制御文の入力元を定義する。DD 要素の中に入力データがあることを指定する。
- (6)入力ファイルの形式を指定する。入力ファイルは, 固定長順編成ファイルでレコード長が 80 バイトであることを指定する。OUTFILE 文の指定を省略した場合, INPFILE 文で指定した形式を仮定する。

<例 2> 入力の固定長順編成ファイルをレコード編集して出力ファイルに複写する。

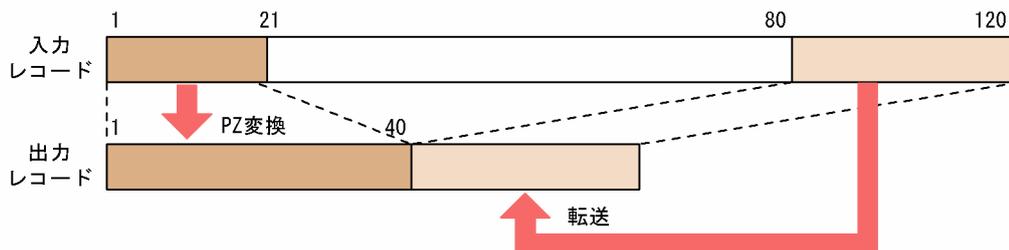
図 3-43 入力の固定長順編成ファイルをレコード編集して出力ファイルに複写



・最初のレコードグループの編集 (GROUP1)



・第2レコードグループの編集 (GROUP2)



```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="SAMPLE2" >
    <STEP NAME="STEP0001" >
      <EXEC PGM="bjexu_scopy" /> ... (1)
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" /> ... (2)
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="OLD" /> ... (3)
      <DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" /> ... (4)
      <DD NAME="SYSIN" TYPE="DATA" > ... (5)
        <![CDATA[
!
      INPFILE FILEORG=F, RECLEN=120 ... (6)
      OUTFILE FILEORG=F, RECLEN=80 ... (7)
      GENERATE LENGTH=120 ... (8)
GROUP1 RECORD IDENT=(8, 'FIRSTEND', 1), ... (9)
      FIELD=(31, 20, . 1),
      FIELD=(15, 100, PZ, 32),
      FILL='*'
GROUP2 RECORD FIELD=(20, 1, PZ, 1), ... (10)
      FIELD=(41, 80, . 40)
!
      ]]>
        </DD>
      </STEP>
    </JOB>
  </HitachiBatchJobExec>

```

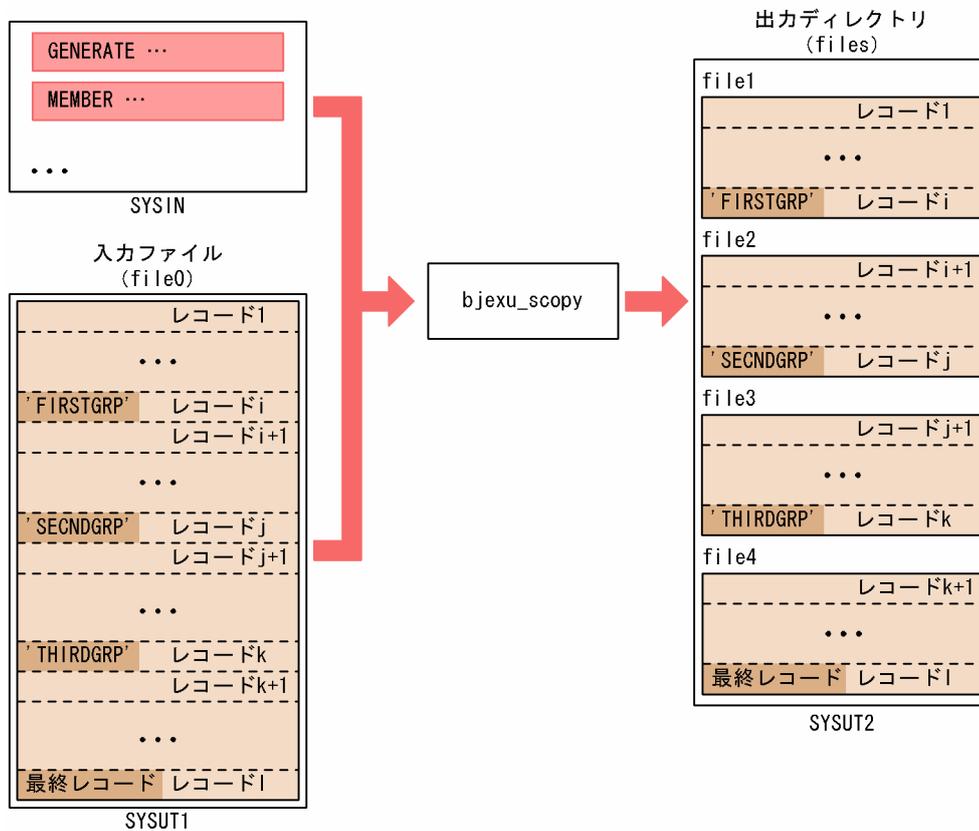
- (1)bjexu_scopy を実行する。
- (2)メッセージ出力先に SYSOUT データファイルの A クラスを指定する。
- (3)入力ファイルを定義する。ファイルは, "file1"を指定する。
- (4)出力ファイルを定義する。ファイルは, "file2"を指定する。
- (5)制御文の入力元を定義する。DD 要素の中に入力データがあることを指定する。
- (6)入力ファイルの形式を指定する。入力ファイルのレコード長は, 120 バイトであることを指定する。
- (7)出力ファイルの形式を指定する。出力ファイルのレコード長を 80 バイトにすることを指定する。
- (8)ファイルの複写と編集の開始を指定する。入力レコードを連結しないで入力論理レコードにする。
- (9)最初のレコードグループの編集内容を指定する。
 - (a)IDENT オペランドで, 最初のレコードグループに分割する最後のレコードの識別情報を指定する。レコードの 1~8 バイトの内容が'FIRSTEND'のレコードを示す。
 - (b)入力レコードの 20 バイト目からの 31 バイトを出力レコードの 1 バイト目以降へ変換しないで転送する。
 - (c)入力レコードの 100 バイト目からの内部 10 進形式データ 15 バイトを外部 10 進形式データに変換して出力レコードの 32 バイト目以降へ転送する。
 - (d)FILL オペランドで, 未編集領域に埋め込み文字として"*"を埋め込むことを指定する。
- (10)第 2 レコードグループ (残りのレコード) の編集内容を指定する。
 - (a)入力レコードの 1 バイト目からの内部 10 進形式データ 20 バイトを外部 10 進形式データに変換して出力レコードの 1 バイト目以降へ転送する。

(b)入力レコードの 80 バイト目からの 41 バイトを出力レコードの 40 バイト目以降へ転送する。

(c)IDENT オペランドを指定しないで，入力ファイルの最後まで処理することを指定する。

<例 3> 入力の固定長順編成ファイルをレコード編集してレコードグループ単位に出力ファイルを作成する。

図 3-44 入力の固定長順編成ファイルをレコード編集してレコードグループ単位に出力ファイルを作成



```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="SAMPLE3" >
    <STEP NAME="STEP0001" >
      <EXEC PGM="bjexu_scopy" /> ... (1)
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" /> ... (2)
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file0" DISP="OLD" /> ... (3)
      <DD NAME="SYSUT2" TYPE="DIR" DSN="files" DISP="OLD" /> ... (4)
      <DD NAME="SYSIN" TYPE="DATA" > ... (5)
      <![CDATA[
!
      INPFILE FILEORG=F, RECLEN=80 ... (6)
      OUTFILE FILEORG=F, RECLEN=80 ... (7)
      GENERATE LENGTH=80 ... (8)
      MEMBER NAME=(file1) ... (9)
GROUP1 RECORD IDENT=(8,'FIRSTGRP',1) ... (10)
      MEMBER NAME=(file2) ... (11)
GROUP2 RECORD IDENT=(8,'SECNDGRP',1) ... (12)
      MEMBER NAME=(file3) ... (13)
GROUP3 RECORD IDENT=(8,'THIRDGRP',1) ... (14)
      MEMBER NAME=(file4) ... (15)
!
      ]]>
      </DD>
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

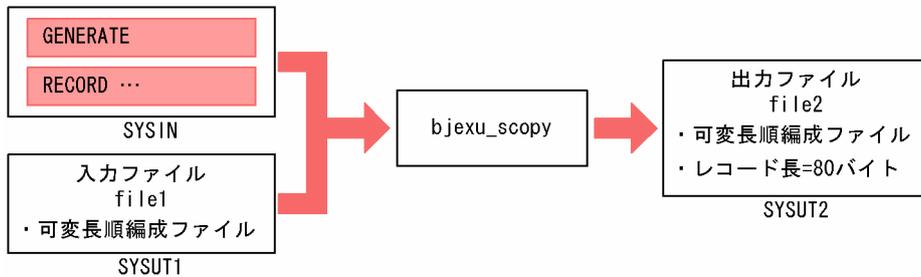
- (1)bjexu_scopy を実行する。
- (2)メッセージ出力先に SYSOUT データファイルの A クラスを指定する。
- (3)入力ファイルを定義する。ファイルは, "file0"を指定する。
- (4)出力ディレクトリを定義する。ディレクトリは, "files"を指定する。
- (5)制御文の入力元を定義する。DD 要素の中に入力データがあることを指定する。
- (6)入力ファイルの形式を指定する。入力ファイルのレコード長は, 80 バイトであることを指定する。
- (7)出力ファイルの形式を指定する。出力ファイルのレコード長は, 80 バイトであることを指定する。
- (8)ファイルの複写と編集の開始を指定する。入力論理レコード長は, 80 バイトであることを指定する。
- (9)最初のレコードグループに付けるファイル名 file1 を指定する。
- (10)最初のレコードグループの最後のレコードを指定する。レコードの 1 バイト目から 8 バイトの内容が'FIRSTGRP'のレコードを示す。
- (11)第 2 レコードグループに付けるファイル名 file2 を指定する。
- (12)第 2 レコードグループの最後のレコードを指定する。レコードの 1 バイト目から 8 バイトの内容が'SECNDGRP'のレコードを示す。
- (13)第 3 レコードグループに付けるファイル名 file3 を指定する。

(14)第3レコードグループの最後のレコードを指定する。レコードの1バイト目から8バイトの内容が'THIRDGRP'のレコードを示す。

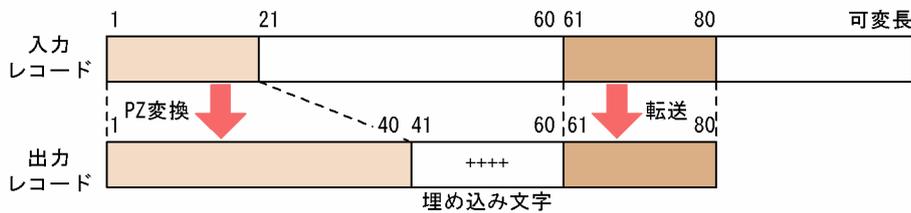
(15)第4レコードグループに付けるファイル名 file4 を指定する。RECORD 文を省略し、入力ファイルの最後までを第4レコードグループとして出力することを指定する。

<例4> 入力の可変長順編成ファイルをレコード編集して出力ファイルに複製する。

図 3-45 入力の可変長順編成ファイルをレコード編集して出力ファイルに複製



・レコードの編集



```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="SAMPLE2" >
    <STEP NAME="STEP0001" >
      <EXEC PGM="bjexu_scopy" /> ... (1)
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" /> ... (2)
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="OLD" /> ... (3)
      <DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" /> ... (4)
      <DD NAME="SYSIN" TYPE="DATA" > ... (5)
      <![CDATA[
!
      INPFILE FILEORG=V ... (6)
      OUTFILE FILEORG=V, RECLEN=256 ... (7)
      GENERATE ... (8)
      RECORD FIELD=(20, 1, PZ, 1, 40), ... (9)
              FIELD=(20, 61, . 61),
              FILL='+'
!
      ]]>
    </DD>
  </STEP>
</JOB>
</HitachiBatchJobExec>

```

(1)bjexu_scopy を実行する。

(2)メッセージ出力先に SYSOUT データファイルの A クラスを指定する。

(3)入力ファイルを定義する。ファイルは, "file1"を指定する。

(4)出力ファイルを定義する。ファイルは、"file2"を指定する。

(5)制御文の入力元を定義する。DD 要素の中に入力データがあることを指定する。

(6)入力ファイルの形式を指定する。入力ファイルは、可変長順編成ファイルであることを指定する。

(7)出力ファイルの形式を指定する。出力ファイルは、可変長順編成ファイルで、最大レコード長を 256 バイトにすることを指定する。

(8)ファイルの複写と編集の開始を指定する。

(9)レコードグループの編集内容を指定する。

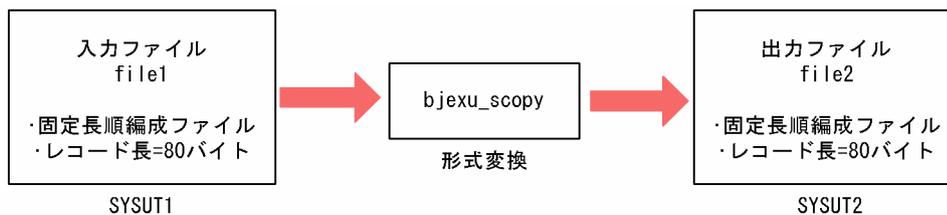
(a)入力レコードの 1 バイト目からの内部 10 進形式データ 20 バイトを外部 10 進形式データに変換し、出力レコードの 1 バイト目以降へ転送する。

(b)入力レコードの 61 バイト目からの 20 バイトを出力レコードの 61 バイト目以降へ変換しないで転送する。

(c)FILL オペランドで、未編集領域に埋め込み文字として"+"を埋め込むことを指定する。

<例 5 > 入力の固定長順編成ファイルを可変長順編成ファイルに形式変換して複写する。

図 3-46 入力の固定長順編成ファイルを可変長順編成ファイルに形式変換して複写



```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="SAMPLE1" >
    <STEP NAME="STEP0001" >
      <EXEC PGM="bjexu_scopy" /> ... (1)
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" /> ... (2)
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="OLD" /> ... (3)
      <DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" /> ... (4)
      <DD NAME="SYSIN" TYPE="DATA" > ... (5)
      <![CDATA[
!
      INPFILE FILEORG=F, RECLEN=80 ... (6)
      OUTFILE FILEORG=V, RECLEN=80 ... (7)
      GENERATE ... (8)
!
      ]]>
    </DD>
  </STEP>
</JOB>
</HitachiBatchJobExec>
```

(1)bjexu_scopy を実行する。

(2)メッセージ出力先に SYSOUT データファイルの A クラスを指定する。

(3)入力ファイルを定義する。ファイルは、"file1"を指定する。

(4)出力ファイルを定義する。ファイルは、"file2"を指定する。

(5)制御文の入力元を定義する。DD 要素の中に入力データがあることを指定する。

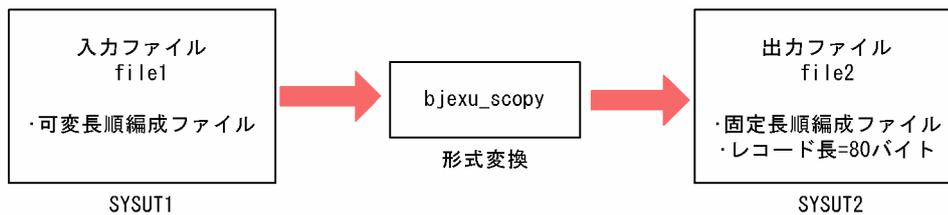
(6)入力ファイルの形式を指定する。入力ファイルは、固定長順編成ファイルでレコード長が 80 バイトであることを指定する。

(7)出力ファイルの形式を指定する。出力ファイルは、可変長順編成ファイルで、最大レコード長を 80 バイトにすることを指定する。

(8)ファイルの複写と編集の開始を指定する。

<例 6 >入力の可変長順編成ファイルを固定長順編成ファイルに形式変換して複写する。

図 3-47 入力の可変長順編成ファイルを固定長順編成ファイルに形式変換して複写



```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="SAMPLE1" >
    <STEP NAME="STEP0001" >
      <EXEC PGM="bjexu_scopcy" /> ... (1)
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" /> ... (2)
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="OLD" /> ... (3)
      <DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" /> ... (4)
      <DD NAME="SYSIN" TYPE="DATA" > ... (5)
      <![CDATA[
!
        INPFILE FILEORG=V ... (6)
        OUTFILE FILEORG=F, RECLEN=80 ... (7)
        GENERATE ... (8)
        RECORD FIELD=(80,1,,1) ... (9)
!
      ]]>
    </DD>
  </STEP>
</JOB>
</HitachiBatchJobExec>
```

(1)bjexu_scopcy を実行する。

(2)メッセージ出力先に SYSOUT データファイルの A クラスを指定する。

(3)入力ファイルを定義する。ファイルは、"file1"を指定する。

(4)出力ファイルを定義する。ファイルは、"file2"を指定する。

(5)制御文の入力元を定義する。DD 要素の中に入力データがあることを指定する。

(6)入力ファイルの形式を指定する。入力ファイルは、可変長順編成ファイルであることを指定する。

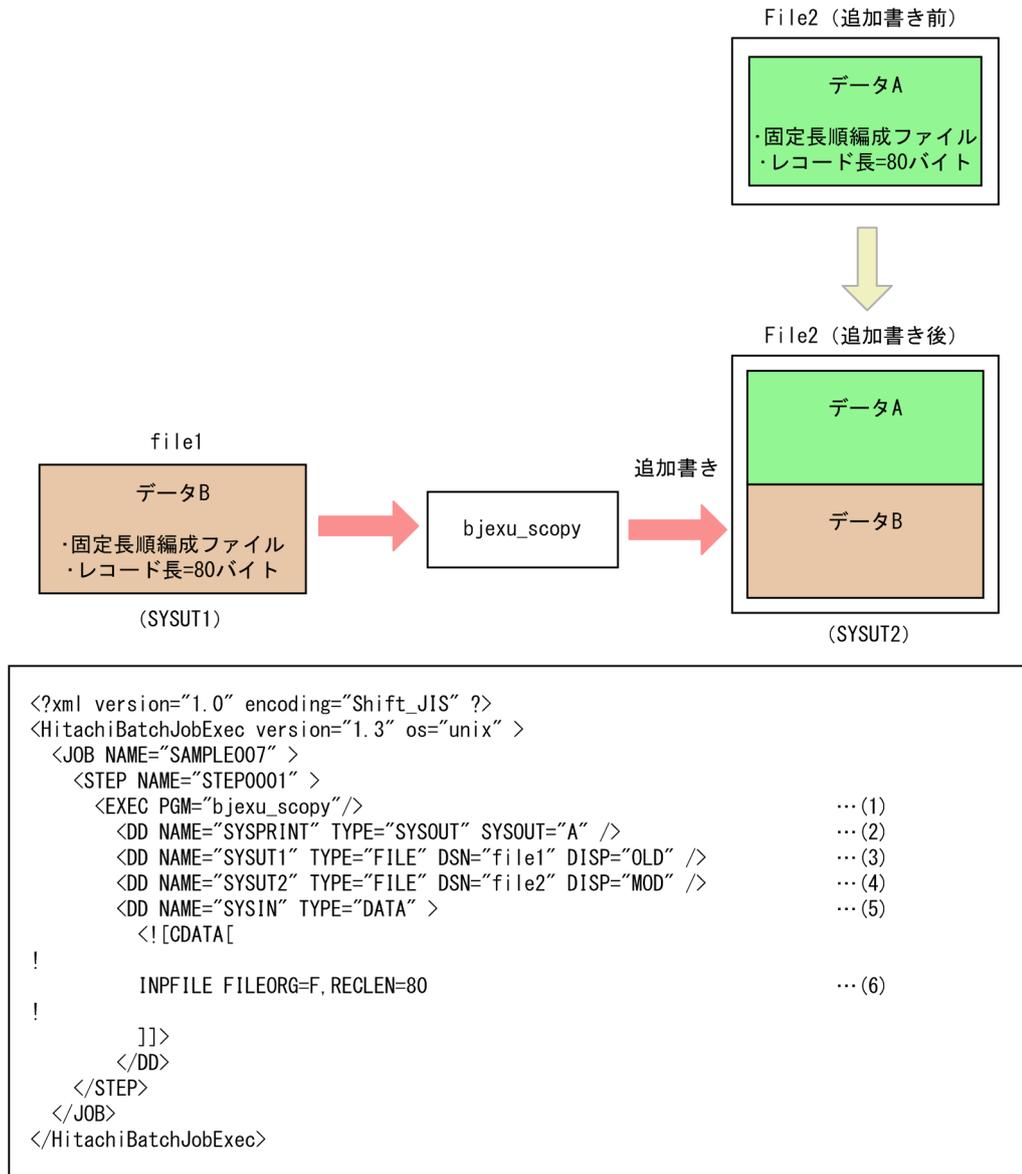
(7)出力ファイルの形式を指定する。出力ファイルは、固定長順編成ファイルで、レコード長を 80 バイトにすることを指定する。

(8)ファイルの複写と編集の開始を指定する。

(9)入力レコードの 1 バイト目からの 80 バイトを出力レコードの 1 バイト目以降へ変換しないで転送する。

<例 7 > 入力の固定長順編成ファイルを出力ファイルに追加書きで単純複写する。

図 3-48 入力の固定長順編成ファイルを出力ファイルに追加書きで単純複写



(1)bjexu_scopy を実行する。

(2)メッセージ出力先に SYSOUT データファイルの A クラスを指定する。

(3)入力ファイルを定義する。ファイルは、"file1"を指定する。

(4)出力ファイルを定義する。ファイルは、"file2"を指定する。追加書きするために DISP="MOD"を指定する。

(5)制御文の入力元を定義する。DD 要素の中に入力データがあることを指定する。

(6)入力ファイルの形式を指定する。入力ファイルは、固定長順編成ファイルでレコード長が 80 バイトであることを指定する。OUTFILE 文の指定を省略した場合、INPFIL 文で指定した形式を仮定する。

3.18.8 bjexu_ismut および bjexu_ismut64 (索引順編成ファイルの作成, 削除, 索引順編成ファイルと順編成ファイルの複写, 変換および併合)

(1) 機能

索引順編成ファイルの作成, 削除, 索引順編成ファイルおよび順編成ファイルの複写, 変換および併合ができる。

(a) ファイルの作成 (DEFINE 文)

索引順編成ファイルを作成する。

- 可変長索引順編成ファイルを作成する場合

レコード長が 100 バイト, 最小レコード長が 10 バイト, キー長が 8 バイトで, キーはレコードの先頭にあり, キーの属性が文字型であるファイルを作成する。

```
DEFINE ISAM, DD=索引順編成ファイル用dd要素名,  
  ISRECL=(100, 10),  
  ISRECFM=V,  
  ISKEY=(8, 0, C)
```

- 固定長索引順編成ファイルを作成する場合

レコード長が 100 バイト, キー長が 4 バイトで, キーはレコードの先頭にあり, キーの属性が 4 バイト整数型であるファイルを作成する。

```
DEFINE ISAM, DD=索引順編成ファイル用dd要素名,  
  ISRECL=(100),  
  ISRECFM=F,  
  ISKEY=(4, 0, L)
```

(b) 索引順編成ファイルおよび順編成ファイルの複写, 変換および併合 (REPRO 文)

ファイルの複写とは, 同じ編成のファイルにデータを入れることである。ファイルの変換とは, 編成の異なるファイルにデータを入れることである。また, ファイルの併合とは, 2つのファイルを1つのファイルにすることである。REPRO 文の機能の共通の内容について説明する。個別の内容については, (c)~(e)の説明を参照のこと。

入出力ファイルの組み合わせと操作内容を次の表に示す。

表 3-48 入出力ファイルの組み合わせと操作内容

入力		出力			
ファイル編成	レコード種別	索引順編成ファイル		順編成ファイル	
		固定長レコード	可変長レコード	固定長レコード	可変長レコード
索引順編成ファイル	固定長レコード	◎	×	○	×
	可変長レコード	×	◎	×	○
順編成ファイル	固定長レコード	○	×	◎	×
	可変長レコード	×	○	×	◎

(凡例)

- ◎：複写と併合ができる。
- ：変換と併合ができる。
- ×：指定できない。

• 注意事項

(1)出力ファイルが索引順編成ファイルの場合は、出力ファイルにデータがないときに複写または変換となる。すでにデータがあるときは併合となる。

(2)出力ファイルが順編成ファイルの場合はデータの有無に関係なく、DD 要素で DISP="OLD", "NEW"または"RNW"を指定したときに複写または変換となる。このとき、データがあっても削除される。順編成ファイルへ併合するときは、DD 要素で DISP="MOD"を指定する必要がある。

(3)レコード長は変換しない。固定長で入力ファイルと出力ファイルのレコード長が異なる場合は解析エラーとなる。

(4)出力ファイルが索引順編成ファイルの場合、入力ファイルの各レコードには出力ファイルで設定されているキーがすべて存在することが必要である。そのため、レコード長が出力ファイルの最小レコード長以上である必要がある。

(5)索引順編成ファイルを索引順編成ファイルに複写した場合、データファイルのレコード格納順は、入力ファイルで指定したキーの昇順になる。副キーを指定する場合はキー番号で指定する。

(6)ファイルの複写、変換および併合の範囲指定では、入力ファイルの読み込みで指定したキーの属性に従った値を指定できる。

(7)キー属性は、文字型 (CHARTYPE)、2 バイト整数型 (INTTYPE) または 4 バイト整数型 (LONGTYPE) のどれかを指定できる。文字型のキーは、コード体系では比較されないで、バイナリで比較される。

(8)索引順編成ファイルに併合する場合、出力ファイルに入力レコードと同じ主キー値を持つレコードが存在したとき、既存のレコードを置き換えるかどうかは制御文の指定に従う。置き換えの指定がない場合、重複を許可しないキーで重複があったときはエラーとなり、レコードは出力されない。

(9)併合の場合、追加するレコードは出力ファイルの物理的な最終レコードのあとに入力順で追加される。索引順編成ファイルへの併合で、出力ファイルがデータファイルの削除レコードを再利用する指定の場合、併合後の物理的なレコードの順番は不定となる。順番は、ISAMの仕様に従う。

(10)コマンド実行時はISAMの排他モードでファイルを占有する。ISAMユーティリティなどでほかの端末から同じファイルに同時にアクセスできない。ファイルを占有できなかった場合は処理を中止する。

(11)出力ファイルが索引順編成ファイルの場合、主キーがないファイルを指定できない。また、主キーの重複を許可しているファイルを指定できない。

(c) ファイルの複写

索引順編成ファイル中のデータを、同じ編成の索引順編成ファイル用に複写する。または、順編成ファイル中のデータを、同じ編成の順編成ファイル用に複写する。複写範囲も指定できる。

索引順編成ファイルの複写では、複写先ファイルのレコード格納順は複写元ファイルで指定しているキーの順序になる。

- 索引順編成ファイルまたは順編成ファイル全体を複写する場合

```
REPRO INDD=複写元ファイル用dd要素名, ※1
      OUTDD=複写先ファイル用dd要素名 ※1
```

- 索引順編成ファイルまたは順編成ファイルを部分的に複写する場合

```
REPRO INDD=複写元ファイル用dd要素名, ※1
      OUTDD=複写先ファイル用dd要素名, ※1
      SKIP=複写処理開始前に読み飛ばすレコード数, ※2
      COUNT=複写処理するレコード数 ※2
```

- 索引順編成ファイルを部分的に複写する場合

```
REPRO INDD=複写元ファイル用dd要素名, ※1
      OUTDD=複写先ファイル用dd要素名, ※1
      FROMKEY=複写処理を開始するキーの値, ※2
      TOKEY=複写処理を終了するキーの値 ※2
```

注※1 複写元ファイルと複写先ファイルはファイル編成が同じものを指定する。

注※2 複写範囲を指定する。複写範囲を指定するオペランドには、次の表に示すものがある。

表 3-49 複写範囲を指定するオペランド

入力ファイル	開始位置の指定	終了位置の指定
索引順編成ファイル	FROMKEY	TOKEY
	SKIP	COUNT

入力ファイル	開始位置の指定	終了位置の指定
順編成ファイル	SKIP	COUNT

注

- 索引順編成ファイルの場合、複写先のファイルをあらかじめ作成しておく必要がある。
- レコード単位で複写しているため、複写範囲を指定しない複写であれば ISAM が提供するユーティリティコマンドを使用した方が処理時間を短くできる。

(d) ファイルの変換

複写元ファイルのファイル編成を変えて、複写先ファイルに複写する。変換範囲も指定できる。

- 索引順編成ファイル中のデータを、順編成ファイルに変換する。
- 順編成ファイル中のデータを、索引順編成ファイルに変換する。変換後の変換先ファイルのレコード格納順は変換元ファイルと同じ順序になる。
- 変換元ファイル全体を変換する場合

```
REPRO INDD=変換元ファイル用dd要素名,
      OUTDD=変換先ファイル用dd要素名
```

- 変換元ファイルを部分的に変換する場合

```
REPRO INDD=変換元ファイル用dd要素名,
      OUTDD=変換先ファイル用dd要素名,
      SKIP=変換処理開始前に読み飛ばすレコード数, ※
      COUNT=変換処理するレコード数 ※
```

- 索引順編成ファイルを部分変換する場合

```
REPRO INDD=変換元ファイル用dd要素名,
      OUTDD=変換先ファイル用dd要素名,
      FROMKEY=変換処理を開始するキーの値, ※
      TOKEY=変換処理を終了するキーの値 ※
```

注※

変換範囲を指定する。変換範囲を指定するオペランドには、次のものがある。

表 3-50 変換範囲を指定するオペランド

変換元ファイル	開始位置の指定	終了位置の指定
索引順編成ファイル	FROMKEY	TOKEY
	SKIP	COUNT
順編成ファイル	SKIP	COUNT

注

- 索引順編成ファイルの場合、変換先のファイルをあらかじめ作成しておく必要がある。

- 順編成ファイルを索引順編成ファイルに変換する場合、変換元ファイルのレコード順で出力される。索引順編成ファイルに変換する場合、変換先ファイルに指定できる最大レコード長は 65,503 バイトである。
- 索引順編成ファイルを順編成ファイルに変換する場合、変換先ファイルに指定できる最大レコード長は 65,535 バイトである。
- レコード単位で変換しているため、変換範囲を指定しない変換であれば ISAM が提供するユティリティコマンドを使用した方が処理時間を短くできる。

(e) ファイルの併合

ファイルの併合とは、併合元ファイルと併合先ファイル（併合先ファイルにもレコードが入っている）を合わせて 1 つのファイルにすることである。併合元ファイルには、索引順編成ファイルおよび順編成ファイルのどちらかを指定できる。

併合元ファイルの併合範囲を指定できる。また、索引順編成ファイルへの併合では、入力レコードと出力先のレコードのキー状態によって併合方法を指定できる。併合方法を指定するオペラントと機能を次の表に示す。

表 3-51 併合方法を指定するオペラントと機能

オペラント	機能
REPLACE	同一キーを持ったレコードが入力された場合、レコードを置き換える。
NOREPLACE	同一キーを持ったレコードが入力された場合、処理を中止する。
IGNORE	同一キーを持ったレコードが入力された場合、キーが重複するレコードは出力しないで以降の処理を続行する。

• 索引順編成ファイルへの併合

索引順編成ファイルへの併合では、併合元ファイルのレコードのキーと併合先ファイルのレコードのキーを比較する。キーが一致して REPLACE オペラントの指定がある場合は、併合先ファイルのレコードを置き換える。キーが不一致の場合は、併合先ファイルにレコードを挿入または追加する。

```
REPRO INDD=併合元ファイル用dd要素名,
      OUTDD=併合先ファイル用dd要素名,
      REPLACE
```

• 順編成ファイルへの併合

順編成ファイルへの併合では、併合先ファイルの最終レコードに続けて併合元ファイルのレコードを追加する。

```
REPRO INDD=併合元ファイル用dd要素名,
      OUTDD=併合先ファイル用dd要素名
```

• 併合範囲を指定した併合

ファイルの併合では、併合元ファイルの範囲を指定できる。

REPRO INDD=併合元ファイル用dd要素名,
 OUTDD=併合先ファイル用dd要素名,
 SKIP=併合処理開始前に読み飛ばすレコード数,※
 COUNT=併合処理するレコード数※

注※

併合元ファイルの併合範囲を指定する。併合範囲を指定するオペランドには、次のものがある。

表 3-52 併合範囲を指定するオペランド

併合元ファイル	開始位置の指定	終了位置の指定
索引順編成ファイル	FROMKEY	TOKEY
	SKIP	COUNT
順編成ファイル	SKIP	COUNT

注

- 索引順編成ファイルの場合、併合先のファイルをあらかじめ作成しておく必要がある。
- 順編成ファイルに併合する場合、併合先ファイルを定義する DD 要素で DISP="MOD"を指定する。
- 順編成ファイルを索引順編成ファイルに併合する場合、併合元ファイルのレコード順で出力される。
- 順編成ファイルに併合する場合、併合先ファイルに指定できる最大レコード長は 65,535 バイトである。
- 索引順編成ファイルに併合する場合、併合先ファイルに指定できる最大レコード長は 65,503 バイトである。
- 併合する場合、併合処理中に障害が発生したときに備えて、あらかじめバックアップを取得することを推奨する。併合処理中に障害が発生すると、ファイル操作ユーティリティは終了コード 8 で終了するため、バックアップから回復できる。

(f) ファイルの複写と変換の処理時間短縮

入力ファイルをそのまま複写や変換する指定のとき、ISAM のユーティリティコマンドを実行して複写や変換の処理時間を短縮できる。

bjexu_ismut および bjexu_ismut64 は、ISAM ファイル中の一部のキー範囲のレコードを複写、変換、併合や順編成ファイルをすでにある ISAM ファイルに併合できる。このため、1 レコードずつキーをチェックして、条件に合致したレコードを処理する。対して、ISAM ユティリティコマンドの iscpy および iscpy64 や isconv および isconv64 は、複数レコードをまとめてそのまま複写や変換するため、bjexu_ismut および bjexu_ismut64 より複写や変換の処理時間が短い。

入力ファイルをそのまま複写や変換する場合の処理時間を短縮したいとき、環境変数 BJEX_ISMUT_HISPEED=YES を指定する。この指定があると、bjexu_ismut および bjexu_ismut64 は、入力ファイルをそのまま複写や変換する指定のとき、ISAM のユーティリティコマンドを実行する。次の表に入出力ファイルの組み合わせと ISAM ユティリティコマンドの使用可否を示す。

表 3-53 入出力ファイルの組み合わせと ISAM ユティリティコマンドの使用可否

入力		出力			
ファイル編成	レコード種別	索引順編成ファイル		順編成ファイル	
		固定長レコード	可変長レコード	固定長レコード	可変長レコード
索引順編成ファイル	固定長レコード	◎	—	○	×
	可変長レコード	—	◎	—	○
順編成ファイル	固定長レコード	○	—	×	—
	可変長レコード	—	○	—	×

(凡例)

- ◎：入力ファイルをそのまま複製するとき ISAM ユティリティコマンド (iscpy および iscpy64) を使用できる。
- ：入力ファイルをそのまま変換するとき ISAM ユティリティコマンド (isconv および isconv64) を使用できる。
- ×：ISAM ユティリティコマンドを使用できない。
- ：指定できない組み合わせ

• 共通事項

ファイルの変換や複製のとき、ISAM ユティリティコマンドのエラーメッセージは、標準エラー出力に出力される。

• 複製機能

索引順編成ファイルを複製する場合、次のすべての条件を満たすとき、ISAM ユティリティコマンド (iscpy および iscpy64) を使用する。

- INDD, OUTDD および COPYDD オペランド以外のオペランドの指定がない。
- 出力先の索引順編成ファイル中のレコード数が 0 件。

• 変換機能

順編成ファイルを索引順編成ファイルに変換 (索引順編成ファイル変換機能)、または索引順編成ファイルのデータファイルを順編成ファイルに変換 (順編成ファイル変換機能) するとき、ISAM ユティリティコマンド (isconv および isconv64) を使用する。

順編成ファイルを索引順編成ファイルに変換する場合、キーファイルをソートするために、ISAM コマンドでワークファイルを使用する。そのため、ワークファイルを作成するディレクトリを DD 要素で指定できる。DD 名が"TEMPDIR"である DD 要素でディレクトリを指定する。省略時は、/tmp または /usr/tmp を使用する。

次のすべての条件を満たすとき、ISAM ユティリティコマンド (isconv および isconv64) を使用する。

- INDD, OUTDD および COPYDD オペランド以外のオペランドの指定がない。
- 入力ファイルが連結されていない。

- 出力先の索引順編成ファイル中のレコード数が0件である。または、出力先が順編成ファイルのとき、DD 要素で DISP="MOD"以外を指定しているか、DISP="MOD"指定のときは、ファイルサイズが0バイトである。
- 環境変数
環境変数 BJEX_ISMUT_HISPEED の設定値ごとの内容を次の表に示す。

表 3-54 環境変数 BJEX_ISMUT_HISPEED の設定ごとの内容

環境変数 BJEX_ISMUT_HISPEED の設定	内容
YES	ISAM のユティリティコマンドを実行し、ファイルの複写や変換の処理時間を短縮する。
NO	ISAM のユティリティコマンドを実行しない。 レコード単位に複写、変換および併合する。
YES と NO 以外、または設定されていない	ISAM のユティリティコマンドを実行しない。 レコード単位に複写、変換および併合する。

(g) ファイルの削除 (DELETE 文)

索引順編成ファイルを削除する。

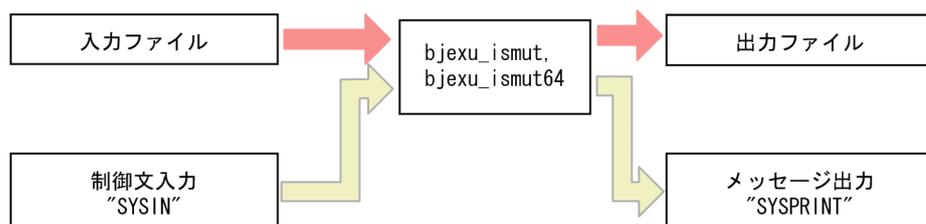
- 可変長索引順編成ファイルを削除する場合
索引順編成ファイルの dd 要素名を指定する。

```
DELETE ISAM, DD=索引順編成ファイル用dd要素名
```

(2) 入出力構成

bjexu_ismut および bjexu_ismut64 の入出力構成を次の図に示す。

図 3-49 bjexu_ismut および bjexu_ismut64 の入出力構成



(a) 索引順編成ファイルを作成する場合

1. 入力

- 入力ファイル
ファイルの作成のため、入力ファイルはない。
- 制御文 (SYSIN)

bjexu_ismut および bjexu_ismut64 の処理内容を次の制御文で指定する。制御文は、(b)の制御文も含めて指定した順に実行される。

- ・ DEFINE 文

2. 出力

- 出力ファイル
新しく作成するファイルが出力ファイルとなる。
- メッセージ (SYSPRINT)
bjexu_ismut および bjexu_ismut64 が出力するメッセージの出力先となる。

(b) 索引順編成ファイル, 順編成ファイルを複写, 変換および併合する場合

1. 入力

- 入力ファイル
複写, 変換または併合の入力となるファイルとなる。
固定長順編成ファイルとして, ダミーファイルも指定できる。
順編成ファイル (固定長順編成ファイルと可変長順編成ファイル) のときは, PREST ファイルを指定できる。
順編成ファイルの場合は, 連結して複数のファイルを指定できる。
PREST ファイルは, 連結できない。
- 制御文 (SYSIN)
bjexu_ismut および bjexu_ismut64 の処理内容を次の制御文で指定する。制御文は, (a)の制御文も含めて指定した順に実行される。
 - ・ INPFILE 文
 - ・ OUTFILE 文
 - ・ REPRO 文

2. 出力

- 出力ファイル
複写, 変換または併合先の出力ファイルとなる。出力先ファイルは, 指定によっては同じ形式で2つのファイルに出力できる。
ダミーファイルも指定できる。
順編成ファイル (固定長順編成ファイルと可変長順編成ファイル) の場合は, PREST ファイルを指定できる。
- メッセージ (SYSPRINT)
bjexu_ismut および bjexu_ismut64 が出力するメッセージの出力先となる。

(c) 索引順編成ファイルを削除する場合

1. 入力

- 入力ファイル
削除するファイル。
- 制御文 (SYSIN)
bjexu_ismut および bjexu_ismut64 の処理内容を次の制御文で指定する。制御文は、(b)の制御文も含めて指定した順に実行される。
 - DELETE 文

2. 出力

- 出力ファイル
ファイルの削除のため、出力ファイルはない。
- メッセージ (SYSPRINT)
bjexu_ismut および bjexu_ismut64 が出力するメッセージの出力先となる。

(d) 終了コード

bjexu_ismut および bjexu_ismut64 の終了コードを次の表に示す。

表 3-55 bjexu_ismut および bjexu_ismut64 の終了コード

終了コード	意味
0	正常に処理を終了した。
4	エラーが発生したが無視して続行した。詳細については、出力されたメッセージを参照のこと。*
8	コマンド処理中にエラーが発生した。エラーの発生したコマンドの処理を終了し、ジョブステップを終了する。出力処理中にエラーが発生しており、出力ファイルの内容は保証されない。必要であればあらかじめ出力ファイルのバックアップを取っておく必要がある。詳細については、出力されたメッセージを参照のこと。
12	次のような回復できないエラーが発生したため、コマンド処理を実行しないでジョブステップを終了する。 <ul style="list-style-type: none"> • 制御文の指定誤り • 制御文 (SYSIN) や入出力ファイルのオープンエラー

注※ 文法的に正しいが、要求された処理を実行できなかった場合、終了コードは 4 になる。例えば、DELETE 文で指定したファイルが存在しないためにファイルの削除ができなかった場合がある。

(3) ジョブ定義 XML ファイル

bjexu_ismut および bjexu_ismut64 を実行する場合は、次に示す EXEC 要素と DD 要素の指定が必要である。

(a) EXEC 要素

- AIX で ISAM ライブラリの 32 ビット版を使用する場合
EXEC 要素の PGM に bjexu_ismut を指定する。

```
<EXEC PGM="bjexu_ismut" />
```

- AIX で ISAM ライブラリの 64 ビット版を使用する場合

EXEC 要素の PGM に bjexu_ismut64 を指定する。

```
<EXEC PGM="bjexu_ismut64" />
```

- Linux の場合

Linux では、ISAM ライブラリは 64 ビット版だけを使用する。EXEC 要素では、PGM に bjexu_ismut を指定する。

```
<EXEC PGM="bjexu_ismut" />
```

bjexu_ismut64 は bjexu_ismut へのシンボリックリンクであり、EXEC 要素の PGM に bjexu_ismut64 を指定しても bjexu_ismut が実行される。

(b) SYSPRINT の DD 要素

bjexu_ismut および bjexu_ismut64 が発行するメッセージの出力先を指定する。DD 連結指定はできない。出力先には、SYSOUT データファイルや恒久ファイルを指定できる。

SYSPRINT が使用できない状態の場合、どこにも経過メッセージは出力されない。メッセージの詳細については、「10. メッセージと終了コード」を参照のこと。

(c) 入力ファイルの DD 要素

複写、変換および併合の元となる入力ファイルを指定する。

索引順編成ファイルの場合、ファイル拡張子を除いたファイル名を指定する。また、ジョブ定義 XML ファイルに、TYPE="FILE",DISP="OLD"または DISP="SHR"を指定する。また、第 2, 第 3 オペランドで DELETE を指定してはならない。指定したファイル名と一致するファイルがある場合は、ジョブステップ終了時にそのファイルは削除される。指定したファイル名に一致するファイルがない場合は、ジョブステップ終了時に警告メッセージが出力されるが、終了コードには反映されない。

DISP="SHR"を指定すると、ファイルを占有しない指定でオープンする (ISAM の open 関数で ISMANULOCK を指定)。このため、DISP="SHR"を指定した複数のジョブで同じファイルを同時に入力できる。

順編成ファイルの場合、複数のファイルを連結して指定できる。このとき、各ファイルの属性は一致させる必要がある。

PREST ファイルは、連結できない。

ファイルの属性が一致しているかどうかはチェックされない。固定長順編成ファイルの場合に、ファイルサイズがレコード長の整数倍でないときはエラーになる。このため、入力ファイルを連結している場合、各ファイルのファイルサイズを確認すること。指定できる DISP オペランドを次の表に示す。

表 3-56 入力ファイルの編成と DISP オペランドの関係

入力ファイル編成	DISP オペランド				
	OLD	NEW	RNW	MOD	SHR
索引順編成ファイル	○	—	—	—	○*
順編成ファイル	○	—	—	—	○

(凡例)

- ：指定できる。
- ：指定できない。ただし、指定してもエラーにはならない。
指定に従ってファイルが生成される。

注※

DD 要素で DISP="SHR"を指定すると、ISAM ファイルを占有しない指定でオープンする。
ただし、RENAME を指定したときは、ファイルを占有する指定でオープンする。

(d) 出力ファイルの DD 要素

索引順編成ファイルを作成する場合、新しく作成するファイルの出力先となる出力ファイルを指定する。
DD 連結指定はできない。

索引順編成ファイルおよび順編成ファイルを操作する場合、複写、変換または併合の出力先となる出力ファイルを指定する。DD 連結指定はできない。

順編成ファイルに出力する場合は、出力ファイルの DD 要素にディレクトリを指定できる。ディレクトリを指定する場合、入力ファイルと同じ名称のファイルに出力する。すでに入力ファイルと同じ名称のファイルがある場合は、エラーとなる。

索引順編成ファイルに出力する場合、ファイル拡張子を除いたファイル名を指定し、ジョブ定義 XML ファイルの DD 要素では、TYPE="FILE",DISP="OLD"を指定する。DISP="NEW", "RNW"または"MOD"を指定した場合、バッチジョブ実行システムは指定されたパス名に従ってファイルを作成する。しかし、bjexu_ismut および bjexu_ismut64 からは索引順編成ファイルとして使用できない。また、第 2, 第 3 オペランドで DELETE を指定してはならない。指定したファイル名と一致するファイルがある場合は、ジョブステップ終了時にそのファイルは削除される。指定したファイル名に一致するファイルがない場合は、ジョブステップ終了時に警告メッセージが出力されるが、終了コードには反映されない。

バッチジョブ実行システムでは、ジョブ定義 XML ファイルに指定することで、索引順編成ファイルを作成できない。複写または変換をする場合、REPRO 文を実行する前に、ISAM ユティリティや DEFINE 文で索引順編成ファイルを作成しておく必要がある。

順編成ファイルに併合する場合は、ジョブ定義 XML ファイルの DD 要素で DISP="MOD"を指定する。それ以外を指定した場合、複写または変換になり、出力ファイルにあったレコードは、削除される。

COPYDD オペランドを指定することで、複写または変換を行う出力先として 2 つの異なる出力先を登録できる。この場合、COPYDD オペランドで指定するファイルは、OUTDD オペランドで指定したファイルと同じ形式である必要がある。異なる形式の場合は処理されないでエラーとなる。併合の場合に COPYDD オペランドを指定したときは、処理しないでエラーとなる。

入力ファイルと出力ファイル（OUTDD オペランドおよび COPYDD オペランドの指定）には、すべて異なるファイルを指定すること。同じファイルを指定した場合は解析エラーとなる。

出力ファイル編成と DD 要素の関係を次の表に示す。

表 3-57 出力ファイル編成と DD 要素の関係

出力ファイル 編成	DSN オペランド		DISP オペランド				
	ファイル	ディレクトリ	OLD	NEW	RNW	MOD	SHR
索引順編成 ファイル	○	×	○	—*	—*	—*	—
順編成ファ イル	○	○	○	○	○	○	—

(凡例)

- ：指定できる。
- ×：指定できない。指定するとエラーとなる。
- ：指定できない。ただし、指定してもエラーにはならない。

注※ 指定するとファイルが割り当てられるが、ISAM ファイルとしては使用できない。

(e) SYSIN の DD 要素

bjexu_ismut または bjexu_ismut64 制御文を指定する。または、この制御文が記述されたファイルのパスを指定する。DD 連結指定はできない。

(f) 索引順編成ファイルの指定

索引順編成ファイルを使用するバッチジョブ間の排他制御は、バッチジョブ実行システムの排他制御機能を使用する。そのため、バッチジョブで使用する索引順編成ファイルは、ジョブ定義 XML ファイルの DD 要素の DSN オペランドでファイル拡張子を除いたファイル名を指定し、DISP="OLD"を指定する必要がある。指定したファイル名と同じ名称の索引順編成以外のファイルがある場合、そのファイルも排他制御の対象になる。バッチジョブ実行システムは、索引順編成ファイルを作成または再作成しないため、DISP="NEW", DISP="RNW"および DISP="MOD"オペランドを指定しないこと。また、第 2, 第 3 オペランドに DELETE を指定しないこと。指定したファイル名と一致するファイルがある場合は、ジョブステップ終了時にそのファイルは削除される。指定したファイル名に一致するファイルがない場合は、ジョブステップ終了時に警告メッセージが出力される。ただし、終了コードには影響しない。

bjexu_ismut または bjexu_ismut64 プログラムでは、同時に ISAM の排他制御も使用する。この排他制御によって、索引順編成ファイルを使用する端末からの操作、およびバッチジョブ実行システムの環境以外からのジョブ実行によるファイルの同時アクセスを防止する。

(4) bjexu_ismut および bjexu_ismut64 制御文

bjexu_ismut および bjexu_ismut64 制御文と機能を次の表に示す。

表 3-58 bjexu_ismut および bjexu_ismut64 制御文と機能

制御文名	機能
DEFINE 文	作成する索引順編成ファイルの形式を指定する。
INPFILE 文	入力ファイルのファイル形式を指定する。
OUTFILE 文	出力ファイルのファイル形式を指定する。
REPRO 文	ファイルの複写、変換または併合の開始を指定する。
DELETE 文	索引順編成ファイルの削除を指定する。
SET 文	終了コードを変更する。

DEFINE 文、DELETE 文および SET 文は、単独または複数で指定できる。

INPFILE 文、OUTFILE 文、REPRO 文は、この順で連続して指定する。INPFILE 文と OUTFILE 文は、省略できない。INPFILE 文、OUTFILE 文、REPRO 文を 1 組と考えて、複数組の指定ができる。

DEFINE 文、DELETE 文および SET 文と INPFILE 文、OUTFILE 文、REPRO 文の組を組み合わせて指定することもできる。

INPFILE 文と OUTFILE 文で指定できるオペランドの組み合わせと処理内容の対応を次の表に示す。

表 3-59 INPFILE 文と OUTFILE 文で指定できるオペランドの組み合わせと処理内容

項番	INPFILE 文の指定	OUTFILE 文の指定	処理内容 ※
1	FILEORG=I	FILEORG=I	索引順編成ファイルの複写または併合
2	FILEORG=I	FILEORG=F	変換または併合
3	FILEORG=X	FILEORG=X	索引順編成ファイルの複写または併合
4	FILEORG=X	FILEORG=V	変換または併合
5	FILEORG=F	FILEORG=I	変換または併合
6	FILEORG=F	FILEORG=F	複写または併合
7	FILEORG=V	FILEORG=X	変換または併合
8	FILEORG=V	FILEORG=V	複写または併合

(凡例)

FILEORG オペランドの指定値の意味を次に示す。

- I：固定長索引順編成ファイル
- X：可変長索引順編成ファイル
- F：固定長順編成ファイル
- V：可変長順編成ファイル

注※

- 入力ファイルのレコードは、出力ファイルに同じ形式で出力される。レコード長の変更はできない。
- 出力ファイルが固定長順編成ファイルの場合、OUTFILE 文の RECLEN オペランドに入力ファイルと同じレコード長を指定する、または OUTFILE 文の RECLEN オペランドを省略する必要がある。
- 出力ファイルが可変長順編成ファイルの場合、OUTFILE 文の RECLEN オペランドに入力レコード長より大きい値を指定する、または OUTFILE 文の RECLEN オペランドを省略する必要がある。
- 出力ファイルが索引順編成ファイルの場合、出力ファイルにデータがないときは複写または変換で、出力ファイルにデータがすでにあるときは併合になる。
- 出力ファイルが順編成ファイルの場合は、データの有無に関係なく、DD 要素で DISP="OLD", "NEW"または"RNW"を指定すると複写または変換となる（データがあっても削除される）。DD 要素で DISP="MOD"を指定し、該当ファイルが存在すると併合となり、最終レコードに続いて出力される。

(a) DEFINE 文

■ 形式

[記号] Δ ₁ DEFINE Δ ₁	ISAM , DD=DD名 , ISKEY=(キー長, キー位置[, {C I L}]) , ISRECFM={F V} , ISRECL=(レコード長[, 最小レコード長])
---	--

■ オペランド

ISAM

索引順編成ファイルを作成することを指定する。主キーファイルだけを作成する。副キーファイルは、作成できない。ISKEY オペランド、ISRECFM オペランドおよび ISRECL オペランドで定義する情報以外は、次に示す内容で定義される。

表 3-60 ISAM ファイルの定義情報と内容

ISAM ファイルの定義情報	内容
障害発生時のデータファイル内容保証	保証されない
主キーのキー順序	昇順となる

ISAM ファイルの定義情報	内容
主キーの重複キー許可	許可しない
主キーの重複キー順序保証	保証する
主キーの圧縮レベル	圧縮しない
主キーのスペース文字許可	許可しない

DD=DD 名

- DD 名 ~<環境変数名>((1~31 文字))
作成するファイルのファイルパス名を定義した DD 名を指定する。

ISKEY=(キー長,キー位置[, {C | I | L}])

索引順編成ファイルのキー長, 相対キー位置およびキー属性を指定する。

なお, キー位置とキー長の合計は, ISRECL 属性で指定するレコード長 (固定長順編成ファイルの場合) または最小レコード長 (可変長順編成ファイルの場合) 以下にする必要がある。

- キー長 ~<整数>((1~255))
キー属性に応じて, 次に示すように指定する。
C (文字型) の場合: 1~255 の範囲
I (2 バイト整数型) の場合: 2
L (4 バイト整数型) の場合: 4
- キー位置 ~<整数>((0~65502))
ISRECFM 属性で指定するレコード種別に応じて, 次に示すように指定する。
ISRECFM=F の場合: 0~ (ISRECL 属性で指定するレコード長-1) の範囲
ISRECFM=V の場合: 0~ (ISRECL 属性で指定する最小レコード長-1) の範囲
- {C | I | L}
キー属性を次に示すどれかのパラメータで指定する。
C: 文字型
I: 2 バイト整数型
L: 4 バイト整数型

ISRECFM={F | V}

索引順編成ファイルを作成する場合のレコード種別を指定する。

- F
レコード種別が固定長であることを示す。
- V
レコード種別が可変長であることを示す。

ISRECL=(レコード長[,最小レコード長])

索引順編成ファイルを作成する場合に, 索引順編成ファイルのレコード長を指定する。

ISRECFM 属性に V を指定した場合は、最小レコード長も指定する必要がある。

ISRECFM 属性に F を指定した場合は、最小レコード長を指定しないこと。指定しても無効となる。

- **レコード長** ～<整数>((1~65503))

索引順編成ファイルのレコード長を指定する。

- **最小レコード長** ～<整数>((1~65503))

索引順編成ファイルの最小レコード長を指定する。なお、最小レコード長 ≤ レコード長にする必要がある。

■ 注意事項

- 作成される索引順編成ファイルは、主キーの重複を許可しない。
- 同じ名称の索引順編成ファイルがすでにある場合は、エラーになる。

(b) INPFILE 文

■ 形式

[記号] Δ ₁ INPFILE Δ ₁	FILEORG=ファイル形式[, RECLEN=レコード長][, CHKEMPTY]
--	--

■ オペランド

FILEORG=ファイル形式

REPRO 文で使用する入力ファイルのファイル形式を指定する。

- **ファイル形式** ～((I | X | F | V))

I: 固定長索引順編成ファイルであることを示す。

X: 可変長索引順編成ファイルであることを示す。

F: 固定長順編成ファイル (COBOL またはアプリケーションなどで作成したファイル, ファイルサイズがレコード長の整数倍であるバイナリファイル) であることを示す。

V: 可変長順編成ファイル (COBOL で作成したファイル) であることを示す。

RECLEN=レコード長

入力ファイルのファイル形式に固定長順編成ファイルを指定した場合, レコード長を指定する。

- **レコード長** ～<符号なし整数>((1~65535))

レコード長を指定する。固定長順編成ファイルを指定した場合は, 必ず指定する。その他の形式のときは指定しないこと。指定した場合はエラーとなる。

ファイルサイズがレコード長の整数倍でないときはエラーとなる。

CHKEMPTY

入力が可変長順編成ファイルのとき, レコード数が 0 件のファイルであればエラーにすることを指定する。省略すると, 入力ファイルのレコード数が 0 件であってもエラーにしないで, レコード数が 0 件の出力ファイルを作成する。

入力ファイルが可変長順編成ファイル以外のときは, 指定できない。指定すると, エラーとなる。

■ 注意事項

- INPFILE 文の指定内容とジョブ定義 XML ファイルの DD 要素で指定したファイルの形式が異なる場合、エラーとなる。

(c) OUTFILE 文

■ 形式

[記号] Δ ₁ OUTFILE Δ ₁	FILEORG=ファイル形式[, RECLEN=レコード長][, MINLEN=最小レコード長]
--	--

■ オペランド

FILEORG=ファイル形式

REPRO 文で使用する出力ファイルのファイル形式を指定する。

- ファイル形式～((I | X | F | V))

I：固定長索引順編成ファイルであることを示す。

X：可変長索引順編成ファイルであることを示す。

F：固定長順編成ファイル（COBOL またはアプリケーションなどで作成したファイル、ファイルサイズがレコード長の整数倍であるバイナリファイル）であることを示す。

V：可変長順編成ファイル（COBOL で作成したファイル）であることを示す。

RECLEN=レコード長

出力ファイルに固定長順編成ファイルまたは可変長順編成ファイルを指定した場合、レコード長を指定する。

- レコード長 ～<符号なし整数>((1～65535))

レコード長を指定する。固定長順編成ファイルの場合はレコード長を、可変長順編成ファイルの場合は、最大レコード長を指定する。索引順編成ファイルの場合は指定しないこと。指定した場合はエラーとなる。

固定長順編成ファイルの場合

次に示すレコード長の出力ファイルとなる。

- RECLEN オペランドの指定値が出力ファイルのレコード長になる。
- RECLEN オペランドを省略すると、INPFILE 文の RECLEN オペランドの指定値を仮定する。

可変長順編成ファイルの場合

次に示す入力ファイルのファイル形式となる。

- 入力が可変長順編成ファイルのとき

可変長順編成ファイルから新規に作成した可変長順編成ファイルへ複写するときは、ヘッダ情報の内容を複写し、RECLEN オペランドを省略すると最大レコード長の内容を引き継ぐ。RECLEN オペランドの指定があれば、指定された値を最大レコード長に設定する。既存ファイルの場合は、最大レコード長を変更しない。

- 入力が可変長索引順編成ファイルのとき

可変長索引順編成ファイルから新規に作成した可変長順編成ファイルへ変換するときは、RECLLEN オペランドを省略すると、最大レコード長として 65535 を仮定する。RECLLEN オペランドの指定があれば、指定された値を最大レコード長に設定する。既存ファイルの場合は、最大レコード長を変更しない。

MINLEN=最小レコード長

出力ファイルに可変長順編成ファイルを指定したとき、最小レコード長を指定する。このオペランドを指定すると、最小レコード長より短いレコードを出力しようとしたときにエラーになるので、キーデータなど必須の情報が含まれていることをチェックできる。

- **最小レコード長** ~<符号なし整数>((1~65535))

最小レコード長を指定する。固定長順編成ファイルや索引順編成ファイルの場合は指定しないこと。指定した場合はエラーとなる。

- 入力が可変長順編成ファイルのとき

可変長順編成ファイルから新規に作成した可変長順編成ファイルへ複写するときは、ヘッダ情報の内容を複写し、MINLEN オペランドを省略すると最小レコード長の内容を引き継ぐ。MINLEN の指定があれば、指定された値を最小レコード長に設定する。既存ファイルの場合は、最小レコード長を変更しない。

- 入力が可変長索引順編成ファイルのとき

可変長索引順編成ファイルから新規に作成した可変長順編成ファイルへ変換するときは、MINLEN オペランドを省略すると、最小レコード長に 1 を設定する。MINLEN オペランドの指定があれば、指定された値を最小レコード長に設定する。既存ファイルの場合は、最小レコード長を変更しない。

- 出力ファイルの最小レコード長に満たないレコードを入力した場合はエラーとなる。

■ 注意事項

- OUTFILE 文の指定内容とジョブ定義 XML ファイルの DD 要素で指定したファイルの形式が異なる場合、エラーとなる。
- 索引順編成ファイルの場合、主キーを持たないファイルは指定できない。指定するとエラーとなる。
- 主キーの重複を許可しているファイルは指定できない。指定するとエラーとなる。

- **固定長索引順編成ファイルまたは固定長順編成ファイルの場合**

固定長索引順編成ファイルまたは固定長順編成ファイルに複写する場合、入力ファイルと出力ファイルは、同じレコード長でなければならない。レコード長が異なるとエラーとなる。

- **可変長索引順編成ファイルまたは可変長順編成ファイルの場合**

可変長索引順編成ファイルまたは可変長順編成ファイルに複写する場合、入力ファイルの各レコードのレコード長に従って処理する。

入力ファイルが可変長索引順編成ファイルや可変長順編成ファイルのとき、入力ファイルと出力ファイルの最小レコード長と最大レコード長は、異なってもよい。

入力ファイルのレコード長が、出力先ファイルの最小レコード長と最大レコード長の範囲に含まれていない場合は、入力元のレコード長を不正と見なし、エラーとする。

「入力ファイルのレコード長>出力ファイルの最大レコード長」の場合、エラーとなる。

「入力ファイルのレコード長<出力ファイルの最小レコード長」の場合、エラーとなる。

「最小レコード長≤入力≤出力ファイルの最大レコード長」の場合、入力したレコードの長さで出力する。

- RECLLEN および MINLEN オペランドを指定するときは、「レコード長≥最小レコード長」になるように指定する。指定に誤りがあると、エラーとなる。

(d) REPRO 文

■ 形式

[記号] Δ ₁ REPRO Δ ₁	INDD=DD名 , OUTDD=DD名 [, COPYDD=DD名] [, KEY=キー番号] [, REPLACE NOREPLACE IGNORE] [, FROMKEY=キー SKIP=レコード数] [, TOKEY=キー COUNT=レコード数]
--	--

■ オペランド

INDD=DD 名

- DD 名 ~<環境変数名>((1~31 文字))
入力ファイルのファイルパス名を定義した DD 名を指定する。

OUTDD=DD 名

- DD 名 ~<環境変数名>((1~31 文字))
出力ファイルのファイルパス名を定義した DD 名を指定する。入力ファイルと同じファイルを指定できない。指定するとエラーとなる。
出力ファイルが順編成ファイルの場合、出力先のディレクトリパス名を定義した DD 名を指定できる。この場合、入力ファイルと同名のファイルを作成する。入力ファイルがあるディレクトリは指定できない。指定するとエラーとなる。

COPYDD=DD 名

- DD 名 ~<環境変数名>((1~31 文字))
同時に 2 つのファイルに出力する場合、出力ファイルのファイルパス名を定義した DD 名を指定する。複写または変換する場合に指定できる。併合の場合に指定するとエラーとなる。また、DISP="MOD"を指定した場合は、出力ファイルの指定に関係なく、エラーとなる。
出力ファイルが順編成ファイルの場合、出力先のディレクトリパス名を定義した DD 名を指定できる。入力ファイルと同じ名称のファイルがあるディレクトリは指定できない。指定するとエラーとなる。
OUTDD オペランドとともに指定する。INDD オペランドおよび OUTDD オペランドと異なる出力先を指定すること。同じ出力先を指定するとエラーとなる。

KEY=キー番号

- キー番号 ～<整数><<1>>((1～99))

入力ファイルが索引順編成ファイルの場合、読み込みに使用するキー番号を指定する。キー番号は、ISAM のユティリティコマンドである isinfo コマンドで表示されるキーファイル情報に対応する番号で、先頭を 1 とした通し番号である。省略した場合は 1 が仮定され、主キーが選択される。

重複を許可するキーを指定した場合、レコードの読み込み順序は、ISAM のときと同じである。詳細については、マニュアル「索引順編成ファイル管理 ISAM」を参照のこと。

キー番号は、ファイル拡張子".Knn"の番号 nn とは対応しないことがあるので、注意が必要である。入力ファイルが順編成ファイルのときに指定すると、エラーとなる。

REPLACE | NOREPLACE | IGNORE

索引順編成ファイルへ併合する場合に併合先の主キー定義で比較して、入力レコードと同じキー値の出力レコードが存在したときの扱いを指定する。

重複を許可しない副キーに対して重複したキーのレコードを追加しようとした場合、IGNORE 以外を指定したときは、エラーとなり処理を中止する。

- REPLACE

索引順編成ファイルに併合する場合に、入力レコードの主キーと出力レコードの主キーが一致するときは、出力レコードを置き換える。一致しないときは、出力ファイルへレコードを挿入または追加する。

- NOREPLACE

索引順編成ファイルに併合する場合に、入力レコードのキーと出力レコードのキーが一致するときは、併合処理を中止する。

- IGNORE

索引順編成ファイルに併合する場合に、入力レコードのキーと出力レコードのキーが一致するときは、キーが重複するレコードを出力しないで以降の処理を続行する。

主キーが重複する場合または重複を許可しない副キーが重複した場合、キーが重複するレコードを出力しないで処理を続行する。

FROMKEY=キー | SKIP=レコード数

入力ファイル中の複写、変換または併合処理の開始位置を指定する。省略すると先頭から処理する。

- FROMKEY=キー

入力が索引順編成ファイルの場合に指定できる。複写、変換または併合処理を開始するレコードのキーを指定する。指定したキーと一致するレコードがない場合、それより大きい最小キーを持つレコードから処理する。

キー

文字型の場合 ～<引用文字列>((1～255 バイト))

2 バイト整数型の場合 ～<整数>((-32768～32767))

4 バイト整数型の場合 ～<整数>((-2147483647～2147483647))

KEY オペランドで指定したキー番号に対応したレコードのキーを指定する。

キー属性が文字型と整数型の場合に指定できる。ファイルに定義されているキー属性に合わせてキーの値を指定する。指定したキーとキー属性が異なると、エラーとなる。また、読み込みで指定したキー番号の定義が合成キーの場合は、指定できない。指定すると、エラーとなる。

キー属性が文字型の場合、引用文字列で指定する。整数を指定するとエラーとなる。指定した文字列とレコードのキーは上位バイトから比較する。指定した文字列がファイルに定義されているキーの長さより短い場合、指定した文字列を含む最小キーを持つレコードから処理する。指定した文字列がファイルに定義されているキーの長さよりも長いと、エラーとなる。

例 文字型のキーを指定したときの処理対象

オペランドで指定した文字列とファイルのキーは上位バイトから比較する。

・ FROMKEY='ABC' を指定
・ ファイルのキー (8バイト)
上位→下位
'ABAAAAAA' ← 処理されない
'ABCAAAAA' ← 処理対象
'ABCBBBBB' ← 処理対象
'ABDAAAAA' ← 処理対象

キー属性が2バイトの整数型の場合は、-32768~32767の範囲で指定する。範囲外の値を指定すると、エラーとなる。また、引用文字列を指定するとエラーとなる。

キー属性が4バイトの整数型の場合は、-2147483647~2147483647の範囲で指定する。-2147483647より小さい値を指定すると、-2147483647を仮定する。また、2147483647より大きい値を指定すると、2147483647を仮定する。引用文字列を指定するとエラーとなる。

• SKIP=レコード数

REPRO文で指定できる、すべての形式の入力ファイルに指定できる。複写、変換または併合処理を開始する前に、入力したレコードの先頭からスキップするレコード数を指定する。なお、複数の入力ファイルを指定した場合は、最初に指定したファイルからスキップする。最初のファイルのレコードがすべてスキップの対象となった場合、二番目に指定したファイルが引き続きスキップの対象となる。

レコード数 ~<符号なし整数>((0~2147483647))

スキップするレコード数を示す。省略した場合は、すべてのレコードが入力の対象になる。

入力レコードの件数より大きい数を指定した場合は、出力レコードの件数は0件となる。

2147483647より大きい値を指定すると、2147483647を仮定する。

TOKEY=キー | COUNT=レコード数

入力ファイル中の複写、変換または併合処理の終了位置を指定する。省略すると最後まで処理する。開始位置≤終了位置の関係が成り立つように指定する。

• TOKEY=キー

入力が索引順編成ファイルの場合に指定できる。複写、変換または併合処理を終了するレコードのキーを指定する。指定したキーと一致するレコードがない場合、それより小さい最大キーを持つレコードまで処理をする。

合成キーは、指定できない。

キー

文字型の場合 ～<引用文字列>((1～255バイト))

2バイト整数型の場合 ～<整数>((-32768～32767))

4バイト整数型の場合 ～<整数>((-2147483647～2147483647))

読み込みで指定したキー番号の定義に従ったキーの値を示す。

キー属性が文字型と整数型の場合に指定できる。ファイルに定義されているキー属性に合わせてキーの値を指定する。指定したキーとキー属性が異なると、エラーとなる。また、読み込みで指定したキー番号の定義が合成キーのときは、指定できない。指定すると、エラーとなる。

キー属性が文字型の場合、引用文字列で指定する。整数を指定するとエラーとなる。指定した文字列とレコードのキーは上位バイトから比較する。指定した文字列が、定義されているキーの長さより短いときは、指定した文字列を含む最大キーを持つレコードまで処理する。指定した文字列がファイルに定義されているキーの長さよりも長いと、エラーとなる。

例 文字型のキーを指定したときの処理対象

オペランドで指定した文字列とファイルのキーは上位バイトから比較する。

・ TOKEY='ABC' を指定	
・ ファイルのキー (8バイト)	
上位→下位	
'ABAAAAAA'	← 処理対象
'ABCAAAAA'	← 処理対象
'ABCBBBBB'	← 処理対象
'ABDAAAAA'	← 処理されない

キー属性が2バイトの整数型のときは、-32768～32767の範囲で指定する。範囲外の値を指定すると、エラーとなる。また、引用文字列を指定するとエラーとなる。

キー属性が4バイトの整数型の場合は、-2147483647～2147483647の範囲で指定する。-2147483647より小さい値を指定すると、-2147483647を仮定する。また、2147483647より大きい値を指定すると、2147483647を仮定する。引用文字列を指定するとエラーとなる。

• COUNT=レコード数

REPRO文で指定できるすべての形式の入力ファイルに指定できる。複写、変換または併合処理を終了するまでのレコード数を指定する。

レコード数 ～<符号なし整数>((1～2147483647))

複写、変換または併合処理を終了するまでのレコード数を示す。省略した場合は、すべてのレコードが入力の対象になる。入力レコードの件数より大きい数を指定した場合は、すべてのレコードが入力の対象になる。

2147483647より大きい値を指定すると、2147483647を仮定する。

■ 注意事項

- 主キーがない索引順編成ファイルに FROMKEY オペランドまたは TOKEY オペランドを指定すると、エラーになる。

- REPRO 文の前には、必ず INPFILE 文と OUTFILE 文を指定する。INPFILE 文、OUTFILE 文、REPRO 文の順で指定する。この組み合わせの REPRO 文を複数指定できる。制御文は、指定した順に実行される。
- FROMKEY オペランドと TOKEY オペランドは、昇順に値を指定する必要がある。入力ファイルのキー要素が昇順型のときに有効で、入力ファイルのキー要素が降順型のときはエラーとなる。
- REPLACE オペランド、NOREPLACE オペランドおよび IGNORE オペランドは、出力ファイルが索引順編成ファイルの場合に有効である。順編成ファイルの場合は、指定してもエラーにならないが無効である。

(e) DELETE 文

■ 形式

[記号] Δ ₁ DELETE Δ ₁	ISAM, DD=DD名
---	--------------

■ オペランド

ISAM

索引順編成ファイルを削除することを指定する。

削除対象のファイルが存在しないときはエラーになるが、処理を続行する。

DD=DD 名

- DD 名 ~<環境変数名>((1~31 文字))

削除するファイルのファイルパス名を定義した DD 名を指定する。

■ 注意事項

- 削除対象ファイルがほかのプログラムやプロセスでオープンされている場合は、エラーとなる。

(f) SET 文

■ 形式

[記号] Δ ₁ SET Δ ₁	[MAXCC LASTCC]=終了コード
--	------------------------

制御文で指定した機能の実行時の終了コードを変更するときに指定する。ただし、制御文で文法解析エラー発生時には、終了コードを変更することはできない。

直前の制御文の実行時にエラーがあっても、後続の制御文を実行したいときに指定する。例えば、直前の制御文で終了コード 12 になっても、終了コードを 0 に変更すると後続の制御文を実行できる。

■ オペランド

[MAXCC | LASTCC]=終了コード

- MAXCC

現在まで実行した制御文の終了コードのうち最大の値を変更するときに指定する。

- LASTCC

直前に実行した制御文の終了コードの値を変更するときに指定する。

- 終了コード

終了コード 0, 4, 8 または 12 を指定する。

(5) 使用例

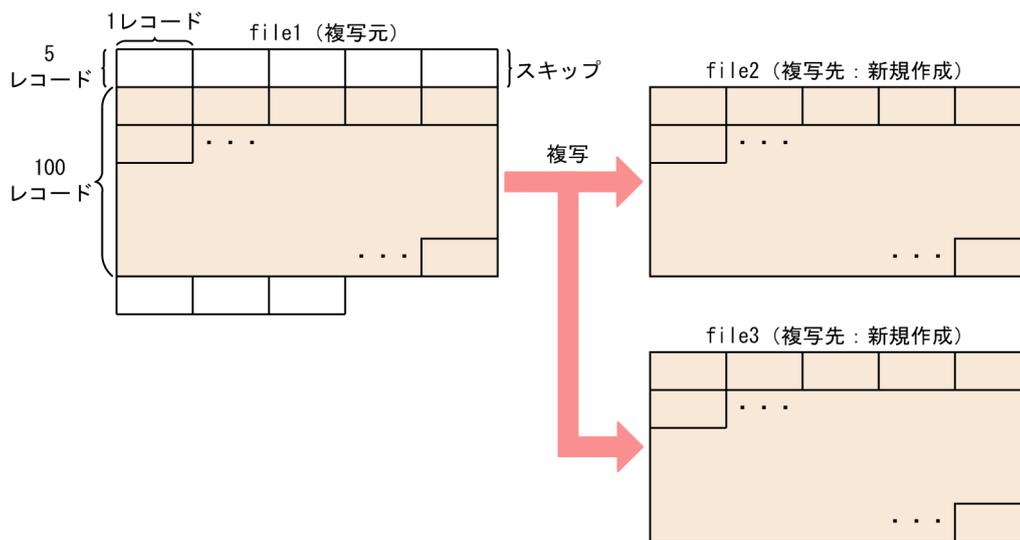
(a) ファイルの複写

索引順編成ファイルを部分的に複写する例を次に示す。

<例 1> 複写元の索引順編成ファイルから先頭の 5 レコードをスキップし、そこから 100 レコード分を新規作成ファイルに複写する。また、複写先を 2 か所にする。

なお、索引順編成ファイルは使用するキーによって入力するレコードの順序が異なる。次の図に、使用するキーの順に入力するレコードの配置のイメージを示す。物理的なレコードの位置は示していないことに注意すること。

図 3-50 索引順編成ファイルの一部を複写



```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="SAMPLE1" >
    <STEP NAME="STEP001" >
      <EXEC PGM="bjexu_ismut" /> : (1)
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="OLD" /> : (2)
      <DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" /> : (3)
      <DD NAME="SYSUT3" TYPE="FILE" DSN="file3" DISP="OLD" /> : (4)
      <DD NAME="SYSIN" TYPE="DATA" >
        <![CDATA[
```

!

```

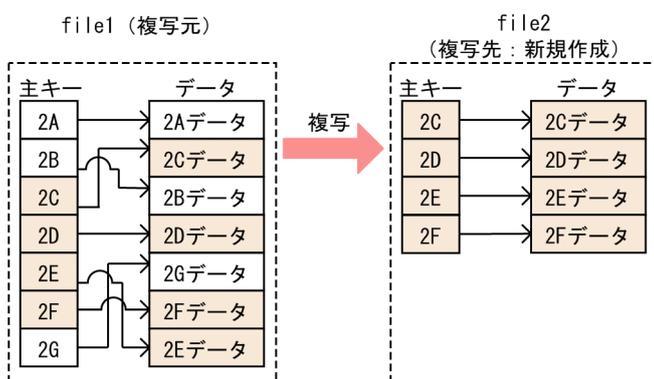
DEFINE ISAM,                                     :(5)
    DD=SYSUT2,                                   :(6)
    ISRECL=(100),                               :(7)
    ISRECFM=F,                                  :(8)
    ISKEY=(5,0,C)                               :(9)
DEFINE ISAM,                                     :(10)
    DD=SYSUT3,                                   :(11)
    ISRECL=(100),                               :(12)
    ISRECFM=F,                                  :(13)
    ISKEY=(5,0,C)                               :(14)
INPFIL FILEORG=I                               :(15)
OUTFIL FILEORG=I                               :(16)
REPRO KEY=1,                                    :(17)
    INDD=SYSUT1,                                :(18)
    OUTDD=SYSUT2,                              :(19)
    COPYDD=SYSUT3,                             :(20)
    SKIP=5,                                     :(21)
    COUNT=100                                  :(22)
!
]]>
</DD>
</STEP>
</JOB>
</HitachiBatchJobExec>

```

- (1)bjexu_ismut プログラムを指定する。
- (2)入力ファイルを定義する。
- (3)新たに作成する索引順編成ファイルを定義する。このファイルを出力ファイルとして使用する。
- (4)新たに作成する索引順編成ファイルを定義する。もう 1 つの出力ファイルとして使用する。
- (5)DEFINE 文で索引順編成ファイルを作成する。
- (6)DD オペランドに、新たに作成する索引順編成ファイルのファイルパス名を定義した DD 名を指定する。
- (7)ISRECL オペランドにレコード長を指定する。
- (8)ISRECFM オペランドに作成する索引順編成ファイルのレコード種別を指定する。ISRECFM=F は、固定長であることを示す。
- (9)ISKEY オペランドに主キーのキー長、相対キー位置およびキー属性を指定する。ISKEY=(5,0,C)は、キー長が 5 バイト、相対キー位置が 0、キー属性が文字型であることを示す。
- (10)DEFINE 文で索引順編成ファイルを作成する。
- (11)DD オペランドに、作成する索引順編成ファイルのファイルパス名を定義した DD 名を指定する。
- (12)ISRECL オペランドにレコード長を指定する。
- (13)ISRECFM オペランドに作成する索引順編成ファイルのレコード種別を指定する。ISRECFM=F は、固定長であることを示す。

- (14) ISKEY オペランドにキー長, 相対キー位置およびキー属性を指定する。ISKEY=(5,0,C)は, キー長が 5 バイト, 相対キー位置が 0, キー属性が文字型であることを示す。
- (15) 入力ファイル形式を指定する。FILEORG=I は固定長索引順編成ファイルであることを示す。
- (16) 出力ファイル形式を指定する。FILEORG=I は固定長索引順編成ファイルであることを示す。
- (17) REPRO 文でファイルを複写する。入力ファイルが索引順編成ファイルの場合, KEY オペランドに入力ファイルで使用するキーを指定する。KEY=1 は主キーファイル (ファイル拡張子 ".K01") を使用することを示す。このファイルの主キーは, キー長が 2 バイト, 相対キー位置が 0 で, キー属性は文字型と定義してある。
- (18) INDD オペランドに入力ファイルを定義した DD 名を指定する。
- (19) OUTDD オペランドに出力ファイルを定義した DD 名を指定する。
- (20) COPYDD オペランドは, もう 1 つの出力ファイルを定義した DD 名を指定する。
- (21) SKIP オペランドに複写元のレコードの先頭からスキップするレコード数を指定する。
- (22) COUNT オペランドに複写するレコード数を指定する。
- <例 2> 複写元ファイルで主キーが"2C"から"2F"までのレコードを新規作成ファイルに複写する。

図 3-51 索引順編成ファイルで特定の主キーであるレコードを複写



```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="SAMPLE2" >
    <STEP NAME="STEP001" >
      <EXEC PGM="bjexu_ismut" />
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="OLD" />           :(1)
      <DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" />         :(2)
      <DD NAME="SYSIN" TYPE="DATA" >
        <![CDATA[
!
      DEFINE ISAM,                                                     :(3)
              DD=SYSUT2,                                             :(4)
              ISRECL=(100),                                           :(5)
        ]>
      ]>
    ]>
  ]>
</HitachiBatchJobExec >

```

```

        ISRECFM=F,                :(6)
        ISKEY=(2,0,C)            :(7)
INPFIL FILEORG=I                :(8)
OUTFIL FILEORG=I                :(9)
REPRO KEY=1,                    :(10)
        INDD=SYSUT1,            :(11)
        OUTDD=SYSUT2,          :(12)
        FROMKEY='2C',          :(13)
        TOKEY='2F'             :(14)
!
]]>
</DD>
</STEP>
</JOB>
</HitachiBatchJobExec>

```

- (1)入力ファイルを定義する。
- (2)新たに作成する索引順編成ファイルを定義する。このファイルを出力ファイルとして使用する。
- (3)DEFINE 文で索引順編成ファイルを作成する。
- (4)DD オペランドに、新たに作成する索引順編成ファイルのファイルパス名を定義した DD 名を指定する。
- (5)ISRECL オペランドにレコード長を指定する。
- (6)ISRECFM オペランドに作成する索引順編成ファイルのレコード種別を指定する。ISRECFM=F は、固定長であることを示す。
- (7)ISKEY オペランドに主キーのキー長、相対キー位置およびキー属性を指定する。ISKEY=(2,0,C)は、キー長が 2 バイト、相対キー位置が 0、キー属性が文字型であることを示す。
- (8)入力ファイル形式を指定する。FILEORG=I は固定長索引順編成ファイルであることを示す。
- (9)出力ファイル形式を指定する。FILEORG=I は固定長索引順編成ファイルであることを示す。
- (10)REPRO 文でファイルを複写する。入力ファイルが索引順編成ファイルの場合、KEY オペランドに入力ファイルで使用するキーを指定する。KEY=1 は主キーファイル（ファイル拡張子".K01"）を使用することを示す。このファイルの主キーは、キー長が 2 バイト、相対キー位置が 0 で、キー属性は文字型と定義してある。
- (11)INDD オペランドに入力ファイルを定義した DD 名を指定する。
- (12)OUTDD オペランドに出力ファイルを定義した DD 名を指定する。
- (13)FROMKEY オペランドに複写を開始するレコードのキーの値を指定する。
- (14)TOKEY オペランドに複写を終了するレコードのキーの値を指定する。

(b) ファイルの併合

索引順編成ファイルの併合の例を次の図に示す。

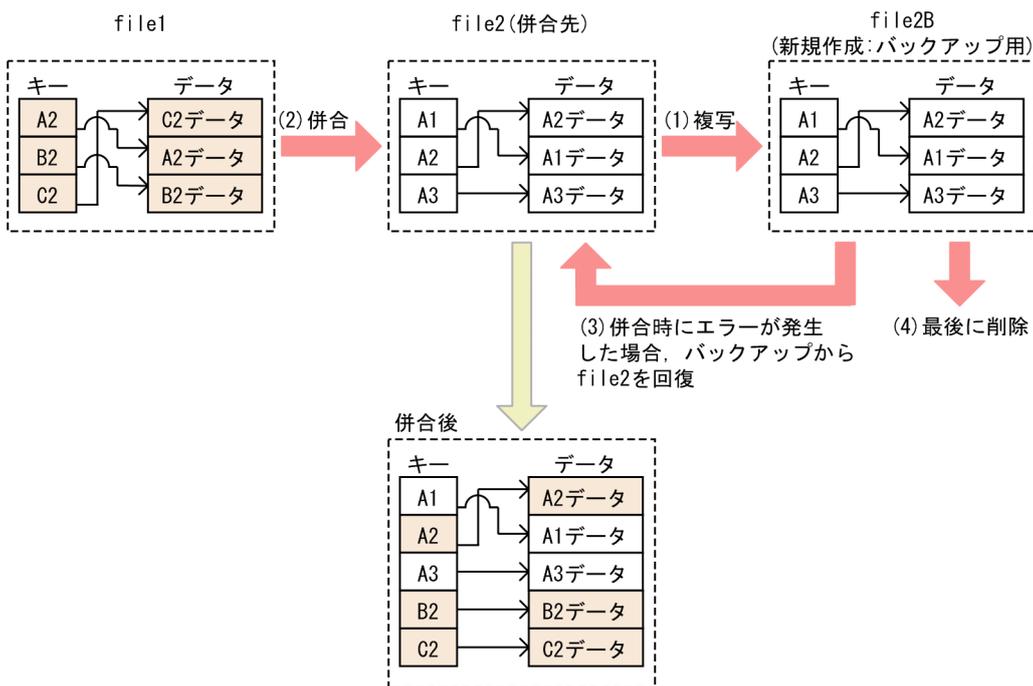
ファイルの併合は、レコードを複写する点ではファイルの複写と同じだが、併合先の出力ファイルにすでにレコードがある点が異なる。ファイルの併合の例では、次の処理を実施している。

1. 併合途中のエラー発生によるファイル破壊に備えて、バックアップを取得する。
2. ファイルの併合を実行する。
3. エラー（リターンコード 8）が発生した場合は、バックアップからファイルを回復する。
4. バックアップ用のファイルを削除する。

複写元と併合先のファイルで主キーが重複するレコードがある場合、複写元のレコードに置き換えるには、REPRO 文で REPLACE オペランドを指定する。

なお、索引順編成ファイルでは、キーの順に処理され、新たにレコードを追加する場合には、レコード削除後に残る無効レコード（空き領域）を使用することがある。このため、レコードの物理的な位置は不定となる。

図 3-52 索引順編成ファイルで特定の主キーのレコードを複写



```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="SAMPLE3" >
    <STEP NAME="STEP0000" >
      <EXEC PGM="*" >
        <![CDATA[! iscpy ${DDN_SYSUT2} ${DDN_SYSUT2B} !]]>           :(1)
      </EXEC>
```

```

<DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" />
<DD NAME="SYSUT2B" TYPE="FILE" DSN="file2B" DISP="OLD" />
</STEP>
<STEP NAME="STEP0001" >
  <EXEC PGM="bjexu_ismut" />
  <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
  <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="OLD" />
  <DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" />
  <DD NAME="SYSIN" TYPE="DATA" >
    <![CDATA[
!
      INPFILE FILEORG=I
      OUTFILE FILEORG=I
      REPRO INDD=SYSUT1,OUTDD=SYSUT2,                : (2)
        REPLACE                                     : (3)
!
    ]]>
  </DD>
</STEP>
<STEP NAME="STEP0002" >
  <EXEC COND="8,NE,STEP0001" PGM="*" >                : (4)
    <![CDATA[! iscpy ${DDN_SYSUT2B} ${DDN_SYSUT2} !]]> : (5)
  </EXEC>
  <DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" />
  <DD NAME="SYSUT2B" TYPE="FILE" DSN="file2B" DISP="OLD" />
</STEP>
<STEP NAME="STEP0003" >
  <EXEC PGM="bjexu_ismut" />
  <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
  <DD NAME="SYSUT2B" TYPE="FILE" DSN="file2B" DISP="OLD" />
  <DD NAME="SYSIN" TYPE="DATA" >
    <![CDATA[
!
      DELETE ISAM,DD=SYSUT2B                          : (6)
!
    ]]>
  </DD>
</STEP>
</JOB>
</HitachiBatchJobExec>

```

(1)ISAM のユーティリティコマンドで、バックアップ用に索引順編成ファイルを複写する。ISAM についての詳細は、マニュアル「索引順編成ファイル管理 ISAM」を参照のこと。

(2)REPRO 文を実行する。出力ファイルには既存のレコードがあるため、ファイルの併合を実施する。

(3)REPLACE 指定がある場合、同じ主キーのレコードが出力先にあると、そのレコードを置き換える。

(4)先行の bjexu_ismut のジョブステップでリターンコード 8 のエラーが発生した場合は、ファイル回復のジョブステップを実行する。

(5)ISAM のユーティリティコマンドで、バックアップ用の索引順編成ファイルから複写して回復する。

(6)バックアップ用の索引順編成ファイルを削除する。

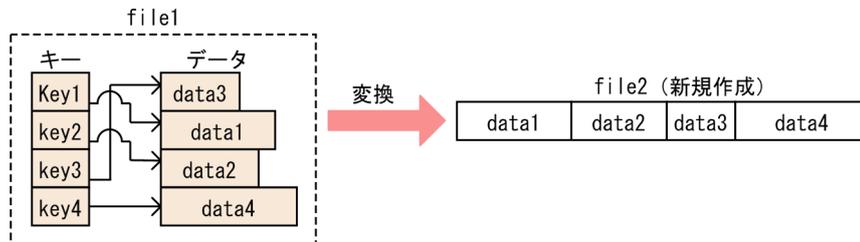
注 この例では、複写するレコードを指定するオペランドが省略されているため、主キーの順にすべてのレコードが複写される。

(c) ファイルの変換

可変長索引順編成ファイルを可変長順編成ファイルに変換する例を次に示す。

<例 1 > 入力元の可変長索引順編成ファイルからキー順にデータを読み込み、出力先へ書き込んでいる。

図 3-53 索引順編成ファイルを順編成ファイルに変換



```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="SAMPLE4" >
    <STEP NAME="STEP0001" >
      <EXEC PGM="bjexu_ismut" />
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="OLD" />
      <DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="NEW" />
      <DD NAME="SYSIN" TYPE="DATA" >
        <![CDATA[
!
      INPFILE FILEORG=X                               :(1)
      OUTFILE FILEORG=V, RECLEN=300                   :(2)
      REPRO INDD=SYSUT1,                               :(3)
              OUTDD=SYSUT2                             :(4)
!
        ]]>
      </DD>
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

(1) 入力ファイル形式を指定する。FILEORG=X は可変長索引順編成ファイルであることを示す。

(2) 出力ファイル形式を指定する。FILEORG=V は可変長順編成ファイルであることを示す。RECLEN オペランドには最大レコード長を指定する。

(3) REPRO 文でファイルの変換および複写を実行する。INDD オペランドに入力ファイルを定義した DD 名を指定する。

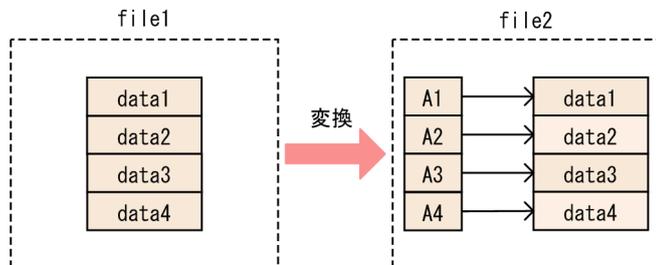
(4) OUTDD オペランドに出力ファイルを定義した DD 名を指定する。出力するレコードは、入力ファイルのレコード長で出力される。

注 この例では、複写するレコードを指定するオペランドが省略されているため、主キーの順にすべてのレコードを変換し、複写する。

順編成ファイルを索引順編成ファイルに変換する例を次に示す。

<例 2> bjexu_ismut64 プログラムを指定して、入力元の順編成ファイルからデータを読み込み、出力先へ書き込んでいる。

図 3-54 順編成ファイルを索引順編成ファイルに変換



```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="SAMPLE5" >
    <STEP NAME="STEP0001" >
      <EXEC PGM="bjexu_ismut64" />                                : (1)
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="OLD" />
      <DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" />
      <DD NAME="SYSIN" TYPE="DATA" >
        <![CDATA[
!
      INPFILE FILEORG=F, RECLLEN=80                                : (2)
      OUTFILE FILEORG=I                                          : (3)
      REPRO INDD=SYSUT1,                                         : (4)
              OUTDD=SYSUT2                                       : (5)
!
      ]]>
    </DD>
  </STEP>
</JOB>
</HitachiBatchJobExec>
```

(1)bjexu_ismut64 プログラムを指定する。

(2)入力ファイル形式を指定する。FILEORG=F は順編成ファイルであることを示す。RECLLEN=80 は、レコード長が 80 バイトであることを示す。

(3)出力ファイル形式を指定する。FILEORG=I は索引順編成ファイルであることを示す。

(4)REPRO 文でファイルの変換を実行する。INDD オペランドに入力ファイルを定義した DD 名を指定する。

(5)OUTDD オペランドに出力ファイルを定義した DD 名を指定する。出力するレコードは、入力ファイルのレコード長で出力される。

(d) ファイルの削除

索引順編成ファイルを作り直す例を次に示す。

<例 1 >

既存の索引順編成ファイルを削除した後、同じ索引順編成ファイルを再作成する。

なお、削除対象のファイルがなかった場合、それまでのエラーを無視する。このため、それまでに実行した制御文の終了コードのうち最大の値は0にしておく。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="SAMPLE6" >
    <STEP NAME="STEP0001" >
      <EXEC PGM="bjexu_ismut" />
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="OLD" />           :(1)
      <DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" />           :(1)
      <DD NAME="SYSIN" TYPE="DATA" >
        <![CDATA[
!
      DELETE ISAM, DD=SYSUT1                                           :(2)
      DELETE ISAM, DD=SYSUT2                                           :(2)
      SET MAXCC=0                                                       :(3)
      DEFINE ISAM,                                                       :(4)
          DD=SYSUT1,
          ISRECL=(100),
          ISRECFM=F,
          ISKEY=(5, 0, C)
      DEFINE ISAM,                                                       :(4)
          DD=SYSUT2,
          ISRECL=(100),
          ISRECFM=F,
          ISKEY=(5, 0, C)
!
        ]]>
      </DD>
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

(1)削除する索引順編成ファイルを定義する。

(2)索引順編成ファイルの削除を行う。

(3)現在までに実行した終了コードのうち最大の値を0とし、それまでのエラーを無視する。

(4)索引順編成ファイルを作成する。

<例 2 >

既存の索引順編成ファイルを削除した後、同じ索引順編成ファイルを再作成する。

なお、再作成するファイルがすでになかった場合、DELETE 文のエラーを無視する。このため、直前に実行した制御文の終了コードは0にしておく。

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="SAMPLE7" >
    <STEP NAME="STEP0001" >
      <EXEC PGM="bjexu_ismut" />
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="OLD" />           : (1)
      <DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" />           : (1)
      <DD NAME="SYSIN" TYPE="DATA" >
        <![CDATA[
!
      DELETE ISAM, DD=SYSUT1           : (2)
      DELETE ISAM, DD=SYSUT2           : (3)
      SET LASTCC=0                     : (4)
      DEFINE ISAM,                     : (5)
          DD=SYSUT1,
          ISRECL=(100),
          ISRECFM=F,
          ISKEY=(5, 0, C)
!
        ]]>
      </DD>
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

(1)削除する索引順編成ファイルを定義する。

(2), (3)索引順編成ファイルの削除を行う。

(4)直前に実行した(3)の終了コードの値を0にする。

それ以前に実行された制御文の終了コードは変更しない。

例2では、(2)の終了コードが4のとき、最大の終了コードは4になる。

(5)索引順編成ファイルを作成する。

(e) 入力の索引順編成ファイルを複数ジョブで共用

JOB0001 と JOB0002 で、入力の索引順編成ファイルを共用することで、排他/共用ロック確保待ちになることなく同時に実行できるようにする。このため、入力ファイルを指定する DD 要素で DISP="SHR" を指定する。

<JOB : JOB0001>

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="JOB0001" >
    <STEP NAME="S1" >
      <EXEC PGM="bjexu_ismut" />
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="SHR" />           : (1)
      <DD NAME="SYSUT2" TYPE="FILE" DSN="file2" DISP="OLD" />           : (2)
      <DD NAME="SYSIN" TYPE="DATA" >

```

```

<![CDATA[
!
    INPFILE FILEORG=I           : (3)
    OUTFILE FILEORG=I          : (4)
    REPRO KEY=1,
        INDD=SYSUT1,          : (5)
        OUTDD=SYSUT2         : (6)
!
]]>
</DD>
</STEP>
</JOB>
</HitachiBatchJobExec>

```

- (1)DD 要素 SYSUT1 でファイル file1 を共用使用するために DISP="SHR"を指定
- (2)DD 要素 SYSUT2 でファイル file2 を排他使用するために DISP="OLD"を指定
- (3)INPFILE 文で、入力が索引順編成ファイルであることを指定
- (4)OUTFILE 文で、出力が索引順編成ファイルであることを指定
- (5)INDD 文で入力ファイルとして SYSUT1 を指定
- (6)OUTDD 文で出力ファイルとして SYSUT2 を指定

<JOB : JOB0002>

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix" >
  <JOB NAME="JOB0002" >
    <STEP NAME="S1" >
      <EXEC PGM="bjexu_ismut" />
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" DISP="SHR" />      : (1)
      <DD NAME="SYSUT2" TYPE="FILE" DSN="file3" DISP="OLD" />    : (2)
      <DD NAME="SYSIN" TYPE="DATA" >
        <![CDATA[
!
            INPFILE FILEORG=I           : (3)
            OUTFILE FILEORG=I          : (4)
            REPRO KEY=1,
                INDD=SYSUT1,          : (5)
                OUTDD=SYSUT2         : (6)
!
        ]]>
        </DD>
      </STEP>
    </JOB>
  </HitachiBatchJobExec>

```

- (1)DD 要素 SYSUT1 でファイル file1 を共用使用するために DISP="SHR"を指定
- (2)DD 要素 SYSUT2 でファイル file3 を排他使用するために DISP="OLD"を指定

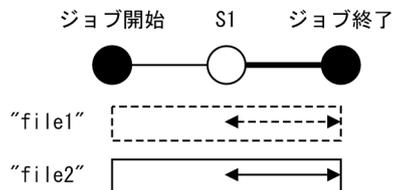
(3)INPFILE 文で、入力が索引順編成ファイルであることを指定

(4)OUTFILE 文で、出力が索引順編成ファイルであることを指定

(5)INDD 文で入力ファイルとして SYSUT1 を指定

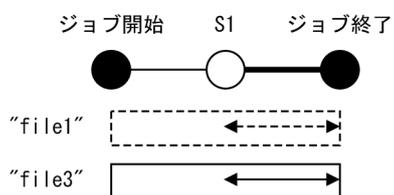
(6)OUTDD 文で出力ファイルとして SYSUT2 を指定

<JOB : JOB0001>



JOB0001とJOB0002でfile1を入力ファイルとして指定していて、DISP=SHR指定のため、排他/共用ロック確保待ちにならない。
JOB0001とJOB0002の出力ファイルは、それぞれfile2とfile3と異なるので、排他/共用ロック確保待ちにならない。
これによって、JOB0001とJOB0002が同時に実行できる。

<JOB : JOB0002>



(凡例)

- : ジョブの実行区間を示す。
- : ジョブステップの実行区間を示す。
- ←- - -> : 共用要求のDD要素のスコープを示す。
- ←——> : 排他要求のDD要素のスコープを示す。
- [- - - -] : 共用ロック確保区間を示す。
- [———] : 排他ロック確保区間を示す。

3.18.9 メッセージの出力先

ファイル操作ユーティリティが出力するメッセージは、次の出力先に出力する。

(1) ジョブ定義 XML ファイルの DD 要素で NAME="SYSPRINT"を指定している場合

ファイル操作ユーティリティが出力するすべてのメッセージを DD 要素で指定した出力先に出力する。

(2) ジョブ定義 XML ファイルの DD 要素で NAME="SYSPRINT"を指定していない場合

タイプコードが I (情報) 以外のメッセージは stderr に出力する。ジョブ終了時、ジョブの終了コードで実行結果を確認し、エラーが発生している場合、経過メッセージが出力されない。そのため、出力先のディレクトリやファイルの内容を調査し、実行結果を確認する必要がある。

3.19 Cosminexus との連携によるバッチアプリケーション実行

バッチジョブ実行システムでは、Cosminexus と連携して Java バッチアプリケーションが実行できる。また、バッチジョブ実行システムに定義したファイルの情報を Java バッチアプリケーションからも利用できる。

バッチジョブ実行システムから Java バッチアプリケーションを実行する方法は、EXEC 要素に LANG="Java" を指定する方法と EXEC 要素に LANG="Java" を指定しない方法の 2 とおりある。

EXEC 要素に LANG="Java" を指定した場合は、Cosminexus 08-00 以降が提供する CTM (Component Transaction Monitor) のジョブスケジューリング機能と連携して、Java バッチアプリケーションを実行できる。この場合、バッチジョブ実行システムは内部的に cjexecjob コマンドを呼び出すため、EXEC 要素の PGM 属性に cjexecjob コマンドを記述する必要はない。

EXEC 要素に LANG="Java" を指定しないで Java バッチアプリケーションを実行する方法については「3.19.2 LANG="Java" を指定しないで Java バッチアプリケーションを実行する方法」を参照のこと。

EXEC 要素に LANG="Java" を指定した場合、バッチジョブ実行システムでは、次の機能を利用できる。

- DD 要素に指定したファイルの属性を、Java バッチアプリケーション内でシステムプロパティ値として取得する。
- SYSOUT の機能を利用して、PDE との連携による帳票を出力する。
- JP1/AJS からの強制停止要求時に、Java バッチアプリケーションに対して内部的に ckilljob コマンドを発行して、Java バッチアプリケーションを停止する。

Cosminexus の Java バッチアプリケーション実行基盤との連携可否を次の表に示す。

表 3-61 Cosminexus の Java バッチアプリケーション実行基盤との連携可否

連携できる機能	LANG="Java"を指定するとき	LANG="Java"を指定しないとき
ジョブステップ実行	○	○
ジョブステップの条件付実行	○	○
一時ファイル	○	×
ダミーファイル	○	×
世代ファイル	○※1	×
バッチアプリケーションの環境変数への受け渡し	×	×
SYSIN	○	×
SYSOUT (PDE 連携)	○	×
スプール制御/ジョブログ出力機能	○	○※2
JP1/AJS からの強制終了時の連携	○	×

連携できる機能	LANG="Java"を指定するとき	LANG="Java"を指定しないとき
JOBLIB/STEPLIB	×	×
DD 連結	×	×

(凡例)

- ：連携できる。
- ×

注※1 cjexecjob コマンドの引数には文字数制限があるため、世代データ群に属する世代ファイルを作成するときは、作成数に注意すること。詳細については「3.19.1(5) 注意事項」を参照のこと。

注※2 ジョブステップの CPU 使用時間は出力できない。cjexecjob コマンドの CPU 使用時間が出力される。

CTM, ジョブスケジューリング機能の詳細, および cjexecjob コマンドと ckilljob コマンドについては, Cosminexus 08-00 以降のマニュアルを参照のこと。

3.19.1 LANG="Java"を指定した場合の Java バッチアプリケーションを実行する方法

EXEC 要素に LANG="Java"を指定した場合の Java バッチアプリケーションの実行に必要な情報の定義方法について示す。

(1) EXEC 要素

バッチジョブ実行システムから Java バッチアプリケーションを実行する場合には、次に示す情報をジョブ定義 XML ファイルに指定する。バッチジョブ実行システムは、これらの情報を指定して cjexecjob コマンドを呼び出し、Java バッチアプリケーションを実行する。

表 3-62 ジョブ定義 XML ファイルに指定する情報 (Java バッチアプリケーションを実行する場合)

要素	属性	指定する値	最大長	説明
EXEC	PGM	Java アプリケーションクラス名(必須)	1,023 文字	パッケージ名を含む, Java アプリケーションのクラス名を指定する。 例: PGM="jp.co.Hitachi.MyBank.MyBatchJob"
	LANG	"Java"(必須)	4 文字	ジョブステップで実行するプログラムが Java アプリケーションであることを示す。常に"Java"を指定する。 例: LANG="Java"
	Group	スケジュールグループ名	63 文字	ジョブスケジューリング機能で, CTM がバッチアプリケーションの実行をスケジューリングするときに割り当てるバッチサーバのグループを指定する。

要素	属性	指定する値	最大長	説明
EXEC	Group	スケジュールグループ名	63 文字	例: Group="MyScheduleGroup"
	Option	Java オプション	1,022 文字	Java VM に渡すオプションを指定する。次のオプションが指定できる。 -cp, -classpath : Java アプリケーション実行時のクラスパスを指定する。-D<name>=<value> : システムプロパティを指定する。 例: Option="-cp /mydir/hello.jar -DHELLO=WORLD"
	PARM	main メソッドに渡す引数	958 文字	Java アプリケーションの main メソッドに渡す引数を指定する。

指定例を次に示す。

```
<EXEC PGM="jp.co.Hitachi.MyBank.MyJob" LANG="Java" Group="MyGroup" Option="-cp /mydir/hello.jar -DHello=World" PARM="myjob check_transaction" />
```

(2) DD 要素

DD 要素で指定した内容は、バッチジョブ実行システムによって-D オプションとして Java オプションに指定される。指定例を次に示す。

```
<DD NAME="WRITEDATA" TYPE="FILE" DSN="/myinfo/mydata.txt" DISP="OLD" />
```

ジョブ定義 XML ファイル中の LANG="Java" のジョブステップで上記 DD 要素が定義された場合、バッチジョブ実行システムは NAME 属性の指定値に "DDN_" を付与した文字列と DSN 属性の指定値を、それぞれ次に示すように-D オプションの <name> および <value> として Java オプションに設定する。指定例を次に示す。

```
-DDN_WRITEDATA=/myinfo/mydata.txt
```

これによって、Java アプリケーション内で <name> の値をキーとして、DD 要素に指定されたファイルのパス名を取得できる。指定例を次に示す。

```

:
java.lang.String mystr;
mystr = System.getProperty("DDN_WRITEDATA");

// これによってmystrに/myinfo/mydata.txtが格納される。
:

```

(3) バッチジョブ実行システムによる cjexecjob コマンドラインの構築内容

バッチジョブ実行システムは、ジョブ定義 XML ファイルに指定された Java バッチアプリケーション用の属性値に基づいて cjexecjob コマンド用の引数を作成し、cjexecjob コマンドを呼び出す。

ジョブ定義 XML ファイルでの Java バッチアプリケーション用の XML 属性の指定方法を次に示す。

```
<EXEC PGM="Javaアプリケーションクラス名" LANG="Java"  
Group="スケジュールグループ名" Option="Javaオプション"  
PARM="mainメソッドに渡す引数"/>  
<DD NAME="NAME属性値" DSN="DSN属性値"/>
```

上記の XML 定義ファイルを指定した場合、バッチジョブ実行システムは次のように cjexecjob コマンドを呼び出す。

```
/opt/Cosminexus/CC/batch/bin/cjexecjob <スケジュールグループ名>  
-jobID <ジョブID> <Javaオプション> -DDD_<NAME属性値>=<ファイルパス>  
"<Javaアプリケーションクラス名> <mainメソッドに渡す引数>"
```

DD 要素に RENAME 属性を指定した場合は、-D オプションは次の形式になる。

```
-D<RENAME属性値>=<ファイルパス>
```

構築例を次に示す。

```
<EXEC PGM="MyJob" LANG="Java" Group="MyGroup"  
Option="-cp /home/user/Java -DHello=World" PARM="arg1 arg2"/>  
<DD NAME="WRITEDATA" TYPE="FILE" DSN="/tmp/mydata.txt" DISP="OLD"/>
```

上記のジョブ定義 XML を指定した場合、次のような cjexecjob コマンドラインが構築される。

```
/opt/Cosminexus/CC/batch/bin/cjexecjob MyGroup -jobID  
<BJEXが自動割り当てした29けたの英数字> -cp /home/user/Java -DHello=World  
-DDD_WRITEDATA=/tmp/mydata.txt MyJob arg1 arg2
```

(4) Java バッチアプリケーションに対するジョブ ID の自動割り当て

ジョブスケジューリング機能を利用した Java アプリケーションの実行には、ジョブ ID の情報が必要となる。LANG="Java"指定の Java バッチアプリケーションの実行ではバッチジョブ実行システムが、実行した Java バッチアプリケーションを識別するために独自にジョブ ID を割り当てる。その機能によって、ジョブステップごとにユニークなジョブ ID を保証するため、Option 属性に cjexecjob コマンドの"-jobID"オプションを指定しないこと。指定した場合、バッチジョブ実行システムから Java バッチアプリケーションの強制停止はできない。バッチジョブ実行システムは、次の形式でジョブ ID を作成して Java アプリケーションに割り当てるため、運用でほかの Java バッチアプリケーションと重複しないようにすること。Java アプリケーションに対するジョブ ID の割り当て形式を次の図に示す。

図 3-55 Java アプリケーションに対するジョブ ID の割り当て形式

B	J	E	X	ジョブ識別子	ステップ番号	時刻情報	ID
---	---	---	---	--------	--------	------	----

Java アプリケーションに対するジョブ ID の割り当て形式の項目と内容を次の表に示す。

表 3-63 Java アプリケーションに対するジョブ ID の割り当て形式の項目と内容

項目	長さ	内容
BJEX	4 けた	バッチジョブ実行システムが割り当てるジョブ ID に自動的に付与される。"BJEX"固定である。
ジョブ識別子	6 けた	バッチジョブ実行システムがジョブごとに割り当てるジョブ識別子である。
ステップ番号	3 けた	バッチジョブ実行システムが割り当てるジョブステップの通し番号である。
時刻情報	8 けた	内部的な時刻識別用情報である。
ID	8 けた	内部的な識別用 ID である。

(5) 注意事項

- cjexecjob, ckilljob の権限は 550 であるため、root か同じグループのユーザアカウントで実行すること。
- Java オプションとして、-cp, -classpath, -D 以外はサポートしないため指定しないこと。詳細については、マニュアル「Cosminexus アプリケーションサーバ V8 リファレンス コマンド編」の cjexecjob コマンドを参照のこと。
- LANG="Java"を指定しないで Group や Option を指定した場合は、エラーとなる。
- DD 要素に指定できる属性に制限はないが、ISAM ファイルなどは Java から利用することは困難なため、指定するファイルについて考慮すること。
- LANG="Java"を指定する場合、従来のバッチサーバ名称指定による Java バッチアプリケーションの実行はできない。スケジュールグループを指定すること。
- cjexecjob コマンドの引数は「Java オプション」, 「Java アプリケーションクラス名」, および「main メソッドに渡す引数」の文字数の合計が 1,024 バイト以下であること。そのため、ジョブ定義 XML ファイルに指定する値の文字数には注意すること（特に世代ファイルの作成数には注意すること）。cjexecjob コマンドの引数の詳細については、マニュアル「Cosminexus アプリケーションサーバ V8 リファレンス コマンド編」を参照のこと。
- cjexecjob コマンドが異常終了した場合、cjexecjob が出力するエラーメッセージが、ジョブステップの標準エラー出力としてジョブログに出力される。
- 使用する OS ごとに、コマンドラインに入力できる文字数に上限がある。そのため、各属性に指定した値の長さが上限値以内であっても、各属性の長さの合計が OS ごとの上限を超えないように注意すること。

3.19.2 LANG="Java"を指定しないで Java バッチアプリケーションを実行する方法

EXEC 要素に LANG="Java"を指定しないで、直接 cjexecjob コマンドを呼び出して Java バッチアプリケーションを実行できる。この場合は、通常のコマンドやプログラムと同様に、cjexecjob コマンドを EXEC 要素の PGM 属性に指定し、cjexecjob コマンドへの引数を PARM 属性に指定する。なお、EXEC 要素に LANG="Java"を指定しない場合は、バッチジョブ実行システムが提供する機能との連携が制限されることに注意すること。

3.19.3 Java バッチアプリケーションの強制終了

LANG="Java"を指定して Java バッチアプリケーションを実行した場合、バッチジョブ実行システムが JP1/AJS からの強制終了要求を受け、自動的に CTM と連携して Java アプリケーションを停止するため、手動で ckilljob コマンドを入力する必要はない。

LANG="Java"を指定しないで Java バッチアプリケーションを実行した場合、バッチジョブの強制終了時もバッチジョブ実行システムは Java バッチアプリケーションを停止しない。LANG="Java"を指定しないでバッチジョブ実行システムから起動した Java バッチアプリケーションを強制停止する場合は、手動で ckilljob コマンドを実行すること。コマンドは次の形式になる。

```
/opt/Cosminexus/CC/batch/bin/cckilljob [<スケジュールグループ名>] -jobID <ジョブID>
```

<スケジュールグループ名>は EXEC 要素の Group 属性が指定されていれば、Group 属性値となる。EXEC 要素の Group 属性が指定されていなければ、省略される。<ジョブ ID>は cjexecjob 実行時にバッチジョブ実行システムが作成したジョブ ID である。

3.20 JP1/Script の実行

バッチジョブ実行システムでは、JP1/Script を実行できる。通常、JP1/Script は、スクリプト実行プログラム (sptxe) を指定して引数にスクリプトファイルを指定して実行する。バッチジョブ実行システムでは、ジョブ定義 XML ファイルの EXEC 要素にスクリプトファイル名と LANG="JP1Script" を指定することで、スクリプト実行プログラムを起動して実行する。

3.20.1 JP1/Script を実行する方法

バッチジョブ実行システムで JP1/Script を実行するためには、EXEC 要素に必要な情報を定義する。

(1) EXEC 要素

バッチジョブ実行システムで JP1/Script を実行する場合には、次に示す情報をジョブ定義 XML ファイルに指定する。バッチジョブ実行システムは、これらの情報を指定して sptxe プログラムを呼び出し、JP1/Script を実行する。

表 3-64 ジョブ定義 XML ファイルに指定する情報 (JP1/Script を実行する場合)

要素	属性	指定する値	最大長	説明
EXEC	PGM	スクリプトファイル名(必須)	1,023 文字	JP1/Script のスクリプトファイル名を指定する。* 例: PGM="/home/user/script/echo"
	LANG	"JP1Script"(必須)	9 文字	実行するプログラムが JP1/Script であることを示す。常に"JP1Script"を指定する。 例: LANG="JP1Script"
	PARM	実行するプログラムに渡す引数	958 文字	JP1/Script に渡す引数として sptxe の位置変数とオプションを指定する。位置変数とオプションは、空白で区切って指定する。 例: PARM="Hello -SPT:SPXLV=3"

注※ 環境変数 PATH にスクリプトファイルが格納されているディレクトリを設定した場合、スクリプトファイル名だけの指定ができる。また、STEPLIB 要素でスクリプトファイルが格納されているディレクトリを指定した場合、スクリプトファイル名だけの指定ができる。

指定方法を次に示す。

```
<EXEC PGM="<スクリプトファイル名>" LANG="JP1Script"  
PARM="<位置変数> <sptxeのオプション>" />
```

上記の XML 定義ファイルを指定した場合、バッチジョブ実行システムは次のように sptxe コマンドを実行する。sptxe のパスは次のパスで固定である。スクリプトファイル名を相対パスで記述した場合、カレントディレクトリからの相対パスとなる。

```
/opt/jp1script/bin/sptxe <スクリプトファイル名> <位置変数> <sptxeのオプション>
```

JP1/Script の実行例を次に示す。

```
<EXEC PGM="/home/user/script/echo" LANG="JP1Script" PARM="Hello -SPT:SPXLV=3" />
```

上記の EXEC 要素を指定した場合、次のコマンドライン文字列が生成される。

```
/opt/jp1script/bin/sptxe /home/user/script/echo Hello -SPT:SPXLV=3
```

スクリプトファイル名に拡張子を省略した場合、拡張子として".SPT"が仮定される。

(2) 注意事項

(a) JP1/Script から COBOL プログラムを実行する場合

バッチジョブ実行システムから COBOL プログラムを実行する場合と比べて、バッチジョブ実行システムから JP1/Script を実行し、JP1/Script から COBOL プログラムを実行する場合、次の点が異なる。

- DD 要素で割り当てたファイルのパスが環境変数 CBL_<DD 名>に格納されない。DDN_<DD 名>に格納される。COBOL プログラムが CBL_<DD 名>にファイルのパスが格納されていることを期待している場合、DD 要素の RENAME 属性でファイルのパスが格納される環境変数名を CBL_<DD>名に変更するか、JP1/Script 内で環境変数 CBL_<DD 名>に環境変数 DDN_<DD 名>の値をコピーすること。
- PDE ファイル形式 2 の帳票を作成できない。PDE ファイル形式 2 の帳票を作成したい場合、バッチジョブ実行システムから COBOL プログラムを実行すること。
- SYSOUT データファイルに追加して書き込みができない。SYSOUT データファイルに追加して書き込みたい場合、バッチジョブ実行システムから COBOL プログラムを実行すること。

3.21 環境変数

3.21.1 ジョブコントローラが使用する環境変数

ジョブコントローラが使用する環境変数を次の表に示す。

表 3-65 ジョブコントローラが使用する環境変数

環境変数名	内容
AJS_BJEX_STOP	AJS_BJEX_STOP=TERM を指定すると、JP1/AJS の強制終了機能使用時プロセスグループに対して、SIGKILL ではなく SIGTERM が送付される。 この環境変数は必ず設定しなければならない。
BJEX_DISPLAY_REPLY	応答要求メッセージへの応答内容を JP1/IM-View の画面に出力するために設定する。YES を設定した場合に有効となる。
BJEX_SUPPRESS_CBLDISPLAY	COBOL2002 で作成したアプリケーションプログラムの DISPLAY 文による JP1 イベント発行を抑止するために設定する。YES を設定した場合に有効となる。
shell または SHELL	ジョブステップを実行時、EXEC 要素内の PGM 属性値に "*" の指定と EXEC 要素の内容にシェルの指定をした場合、シェルスクリプトとして実行するために shell または SHELL 環境変数を使用する。 ジョブ実行前に OS 標準のシェル（AIX の場合 ksh, Linux の場合 bash）を設定しておく必要がある。
XML ファイルへの環境変数指定機能で使用する環境変数	ジョブ定義 XML ファイル内に指定した "%{" と "}" で囲んだ環境変数を設定されている値に置換する。

3.21.2 ジョブコントローラから実行されるプログラムに設定される環境変数

ジョブコントローラから実行されるプログラムに設定される環境変数を次の表に示す。

表 3-66 ジョブコントローラから実行されるプログラムに設定される環境変数

環境変数名	内容
JP1 が設定する環境変数	JP1 のマニュアルを参照のこと。
PATH	DD 要素で TYPE="LIB" を指定した場合、DSN 属性に指定したディレクトリのパス名が追加される。
BJEX_DIR_で始まる名称	設定ファイルに記述されたパス名が設定される。
BJEX_JOB_NAME	JOB 要素の NAME 属性に指定したジョブ名が設定される。
BJEX_JOBID	ジョブ識別子（前 0 を付加した 6 けた固定の 10 進数）が設定される。
BJEX_STEP_NAME	STEP 要素の NAME 属性に指定したジョブステップ名が設定される。

環境変数名	内容
BJEX_STEP_NAME	(ジョブ定義 XML ファイルでステップ名を省略している場合は、環境変数自体を設定しない)。
DDN_で始まる名称 ^{※1}	LANG 属性の指定がない DD 要素のパス名が設定される。 LANG 属性の指定がない DD 要素の NAME 属性名称の環境変数があり、DSN 属性が環境変数の値となる。
CBL_で始まる名称 ^{※1}	LANG="COBOL"の指定がある DD 要素のパス名が設定される。 LANG="COBOL"の指定がある DD 要素の NAME 属性名称の環境変数があり、DSN 属性が環境変数の値となる。
RENAME 属性で指定した名称 ^{※1}	RENAME 属性を指定した DD 要素のパス名が設定される。
任意	SETENV 要素および SETENVFILE 要素で定義した環境変数が設定される。
CBLPDEFTYPE_DD 名 ^{※1※2}	PDE ファイル形式 1 を使用するかファイル形式 2 を使用するかが設定される。
CBLPDERECFM_DD 名 ^{※1※2}	PDE ファイル形式 2 の場合のプリンタ制御文字が設定される。
CBLPDETRUNC_DD 名 ^{※1※2}	PDE ファイル形式 2 の場合に各レコードの後部にある半角空白をファイル出力するかどうかを設定される。
CBLDISP_DD 名 ^{※1※3}	COBOL2002 で作成したアプリケーションプログラムの OUTPUT 指定の OPEN 文で順ファイルを開くとき、ファイルを上書きで開くか、追加書きで開くかが設定される。

注※1 ジョブステップ開始時に設定され、ジョブステップ終了時に削除される。

注※2 詳細は、マニュアル「PRINT DATA EXCHANGE for Open」を参照のこと。

注※3 詳細は、「COBOL2002 マイグレーションライブラリ」の取扱説明書「マイグレーション支援機能」を参照のこと。

3.21.3 ファイル操作ユーティリティの機能を定義する環境変数

ファイル操作ユーティリティが使用する環境変数を次の表に示す。

表 3-67 ファイル操作ユーティリティが使用する環境変数

環境変数名	内容
BJEX_ISMUT_HISPEED	bjexu_ismut/bjexu_ismut64 でファイルの複写や変換をするとき、処理時間を短縮することを指定する。この指定をするとき、複写や変換時、ISAM ユティリティコマンドを実行して処理時間を短縮できる。
BJEX_SCOPY_FIXRECLN	bjexu_scopy で OUTFILE 文に RECLN オペランドの指定がないときの出力レコード長の求め方を指定する。
BJEX_SCOPY_NOFIELD_TRUNC	bjexu_scopy で RECORD 文の FIELD オペランドを省略すると、入力レコード長または入力論理レコード長をフィールド長とし、出力レコードの先頭に複写する。入力レコードの長さが 80 バイトを超えるとき、80 バイトに切り詰めるために設定する。YES を設定した場合に有効になる。

環境変数名	内容
BJEX_SCOPY_DISP_MOD	bjexu_scopy で DD 要素で DISP="MOD"を指定したとき、MOD として処理するか、または OLD として処理するかを指定する。

3.21.4 XML ファイルへの環境変数指定機能

ジョブ定義 XML ファイルおよびカタログドプロシジャ内の任意の位置に指定した環境変数をその環境変数の値に置換する。環境変数をジョブ定義 XML ファイルおよびカタログドプロシジャ内に指定しておくことで、構文の解析前に文字列に置換することができる。

この機能は複数のノードに配布された、ジョブ定義 XML ファイル内の環境変数を一括して変更したい場合に利用できる。

なお、XML ファイルの環境変数による置換は、SETSYMBOL による置換よりも先に置換する。実行結果およびスプールに出力される XML には、環境変数を置換したあとのジョブ定義 XML ファイルが出力される。

(1) 指定方法

XML ファイル中に"%{"と"}"で囲んだ環境変数名を指定した場合、"%{"から"}"までの文字列を環境変数の値に置換する。環境変数は、XML ファイルの任意の位置に指定できる。

```
%{環境変数名}
```

(a) 注意事項

- 指定した環境変数が設定されていない場合は""（空文字）に置換する。
- XML ファイル環境変数による一括置換は、XML 解析処理の前に実施するため、バッチジョブ実行システムが設定する環境変数には使用できないものがある。使用できない環境変数を次に示す。
 - ファイル割り当て機能で設定した環境変数
 - SETENV 要素および SETENVFILE 要素で設定した環境変数
 - BJEX_JOB_NAME
 - BJEX_STEP_NAME
 - BJEX_JOBID

使用できる環境変数を次に示す。

- 設定ファイル BJEX_DIR_xxx パラメータで設定した環境変数
設定ファイル BJEX_DIR_xxx と同じ名前の環境変数が設定されている場合、設定ファイル BJEX_DIR_xxx が優先される。

(2) 使用例

(a) ジョブ定義 XML ファイル

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version="1.4" os="unix" >
  <JOB NAME="ENV_REPLACE">
    <STEP NAME="STEP01" >
      <EXEC PGM="%{BJEX_DIR_SHELLDIR}%{EXEC_PGM}" />
      <DD TYPE="%{TYPEVAL}" NAME="DDNAME" %{DSNATTR} />
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

(b) 環境変数の設定

```
$ env BJEX_DIR_SHELLDIR
abc123
$ grep BJEX_DIR_SHELLDIR /opt/hitachi/bjex/conf/bjex.conf
BJEX_DIR_SHELLDIR /home/user/shell/
$ env EXEC_PGM
mkDD.sh
$ env TYPEVAL
TEMP
$ env DSNATTR
DSNATTR: このパス名のファイルまたはディレクトリは存在しません。
```

(c) 実行結果 (抜粋)

環境変数を値に置換するとき、KAKC1175-I のメッセージを出力する。

```
***** XML IMAGE *****
000001 <?xml version="1.0" encoding="Shift-JIS" ?>
000002 <HitachiBatchJobExec version="1.4" os="unix" >
000003 <JOB NAME="ENV_REPLACE">
000004 <STEP NAME="STEP01" >
000005 <EXEC PGM="/home/user/shell/mkDD.sh" />
000006 <DD TYPE="TEMP" NAME="DDNAME" />
000007 </STEP>
000008 </JOB>
000009 </HitachiBatchJobExec>

***** SYSTEM MESSAGE *****
16:29:17 002404 KAKC0900-I Parsing of xml file "test.xml" started.
16:29:17 002404 KAKC1175-I Environment variable %{BJEX_DIR_SHELLDIR} was replaced to "/home/
user/shell/".
16:29:17 002404 KAKC1175-I Environment variable %{EXEC_PGM} was replaced to "mkDD.sh".
16:29:17 002404 KAKC1175-I Environment variable %{TYPEVAL} was replaced to "TEMP".
16:29:17 002404 KAKC1175-I Environment variable %{DSNATTR} was replaced to "".
16:29:17 002404 KAKC0909-I Parsing of xml file "test.xml" ended. code=0
```

3.21.5 環境変数一括設定機能

ファイルに格納された複数の環境変数を、ジョブステップ実行前に一括して設定することができる。

設定した環境変数は、それ以降に実行される全ジョブステップで有効となる。

(1) 指定方法

(a) 環境変数の指定方法

ジョブステップ実行前に設定したい環境変数を次の形式で任意のファイルに出力する。

環境変数の形式

```
環境変数=環境変数値
```

上記の環境変数が格納されたファイルを環境変数格納ファイルと呼ぶ。

(b) 環境変数格納ファイルの指定方法

環境変数格納ファイルは、環境変数を設定したいジョブステップの DD 要素で定義する。併せて、ジョブ定義 XML ファイルの SETENVFILE 要素で、環境変数格納ファイルを定義した DD 要素の DD 名を指定する。

(i)SETENVFILE 要素

```
<SETENVFILE DDNAME="環境変数格納ファイルを定義しているDD要素のDD名" />
```

- DDNAME="環境変数格納ファイルを定義している DD 要素の DD 名"
DDNAME 属性で指定された DD 名で定義されているファイルを環境変数格納ファイルとする。
環境変数格納ファイルに格納されている環境変数がジョブステップ実行前に設定される。

(2) 環境変数格納ファイルの詳細

(a) 設定する環境変数の書き方

設定する環境変数は、1 行につき 1 つの環境変数を次の形式で任意のファイルに出力する。

```
△0環境変数名=環境変数値
```

(凡例)

△0：0 バイト以上の空白文字を指定する。空白文字の省略は可能。

複数の環境変数を設定する場合には、環境変数分の行を設定する。行数の制限はなく、ファイルの終わりまでを有効とする。

複数の環境変数を設定する例を次に示す。

```

NAME1=VALUE_A   ... 1つの環境変数が1行
NAME2=VALUE_B
NAME3=VALUE_C
:
NAME10=VALUE_J

```

} 必要な行数分格納する。

(b) 環境変数格納ファイルの長さ

環境変数格納ファイルの1行の長さは改行コードを含めて2,047バイト以内である。長さを超える行があった場合にはエラーとし、その時点でジョブの実行は打ち切られる。

(c) 環境変数格納ファイル中の環境変数の扱い

環境変数格納ファイル中の環境変数名および環境変数値の形式チェックは行わない。行内の先頭の空白を削除した後、環境変数名の先頭から改行コードの前までをOSの環境変数設定関数に渡す。また、行の後部の空白は削除されない。なお、空白行だけの行は何もしないで、次の行を処理する。

バッチジョブ実行システムが使用/設定する環境変数は、環境変数格納ファイルでの設定を行ってはいらない。

(d) 環境変数格納ファイルのDD要素での指定方法

環境変数格納ファイルは通常のファイル形式であり、ジョブ定義XMLのDD要素のTYPE属性では次の属性が指定可能である。

表 3-68 指定可能なDD要素のTYPE属性値

ファイル種別	DATA	DUMMY	FILE	TEMP	TEMPISAM	PREST	TEMPPREST	GDG	LIB	SYSOUT	DIR
指定可否	○*	×	○*	○*	×	×	×	○	×	×	×

注※ 逆方向参照も可能。

なお、上記以外の属性の場合、XML解析時にエラーとなる。

(e) 環境変数格納ファイルの連結

複数の環境変数格納ファイルを連結した場合、ファイルの連結順に環境変数を設定する。

このため、複数のファイルで同一の環境変数を設定している場合、最後に設定したものが有効となる。また、世代データ群名を指定した場合も同様に、連結された世代ファイル順に環境変数を設定する。

(3) 環境変数設定エラー時のジョブの扱い

ジョブステップ実行前の環境変数設定処理でエラーが発生した場合、その時点でジョブの実行を中止する。

(4) 使用例

(a) あらかじめ環境変数が格納されているファイルを使用

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.4" os="unix" >
  <JOB NAME="STEP_ENV">
    <STEP NAME="STEP01" >
      <EXEC PGM="program1" />
      <DD NAME="ENVFILE" TYPE="FILE" DSN="Envvarfile" DISP="SHR,KEEP" />.....※1
      <SETENVFILE DDNAME="ENVFILE" />.....※2
    </STEP>
    <STEP NAME="STEP02" >
      <EXEC PGM="program2" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

注※1 DD 要素で、一括設定したい環境変数が格納されている環境変数格納ファイルを定義する。

注※2 環境変数格納ファイルを定義した DD 要素の DD 名を指定する。

(b) 先行ジョブステップで後続ジョブステップで設定する環境変数をファイルに出力

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.4" os="unix" >
  <JOB NAME="STEP_ENV">
    <STEP NAME="STEP01" >
      <EXEC PGM="program1" />
      <DD NAME="ENVOUT" TYPE="TEMP" DSN="Envvarfile" DISP="NEW,PASS" /> .....※1
    </STEP>
    <STEP NAME="STEP02" >
      <EXEC PGM="program2" />
      <DD NAME="ENVFILE" TYPE="TEMP" DSN="Envvarfile" DISP="OLD,DELETE" />.....※2
      <SETENVFILE DDNAME="ENVFILE" /> .....※3
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

注※1 PGM="program1"が、一括設定したい環境変数をファイルに出力。

注※2 DD 要素で、一括設定したい環境変数が格納されている環境変数格納ファイルを定義する。

注※3 環境変数格納ファイルを定義した DD 要素の DD 名を指定する。

3.21.6 環境変数の優先順位

バッチジョブ実行時に設定される環境変数の優先順位を次に示す（優先順位は 1 が最も高い）。

1. STEP 要素内の SETENV 要素で定義した環境変数
2. SETENVFILE 要素で指定した環境変数格納ファイル中の環境変数
3. JOB 要素内の SETENV 要素で定義した環境変数
4. ローカルログインスクリプトで定義した環境変数
5. システムログインスクリプトで定義した環境変数

4

運用

この章では、バッチジョブ実行システムの運用について説明する。

4.1 概要

定型業務を実行するシステムでは、特に運用面では固定化した方式より柔軟性のある方式が採れることが望まれる。バッチジョブ実行システムによって、OS上で柔軟なバッチ運用を実現できる。

バッチジョブ実行システムを効果的に運用するための具体的な方法を中心に説明する。

4.2 利用者の役割

4.2.1 システム管理者

システム管理者は、バッチジョブ実行システムの運営上の責任者で、スーパーユーザ権限をあらかじめ与えられている。システム管理者は、バッチジョブの実行できる環境を整備・管理して、バッチジョブ実行システムを実行できる一般ユーザをシステムに登録する。

4.2.2 一般ユーザ

JP1/AJS からジョブコントローラを起動した要求はジョブとして受け付けられる。ジョブを利用する一般ユーザは指示をまとめて記述したジョブ定義 XML ファイルをジョブコントローラに与える。

ジョブを利用する一般ユーザが指定した指示は、ジョブコントローラによって何が要求されているかが分析され、システム資源が効率良く利用されるようにジョブを実行する。

(1) ジョブ

ジョブとは一般に、一般ユーザが用意する 1 つのまとまった仕事をシステムに要求する単位である。要求する仕事は互いに独立したものと考えられる。

ジョブは一連の処理プログラムから構成されている。これらの処理プログラムを実行するためには、その実行の順序、実行の条件、および処理プログラムの実行に必要なファイルを定義する必要がある。

(2) ジョブステップ

ジョブステップとは、ジョブを構成する実行の単位で、1 つの処理プログラムの実行をいう。ジョブは 1 つのまたはそれ以上のジョブステップから構成されている。ジョブステップは資源を割り当てる単位でもある。

幾つかのジョブステップは互いに関連を持っており、各ジョブステップは定義された順序で処理される。

ジョブステップは定義された順序に処理されるが、前のジョブステップが正しく処理されないと次のジョブステップの実行が意味を持たない。この場合、ジョブステップの実行条件を指定して、処理のスキップもできる。

(3) ジョブとジョブステップの概念

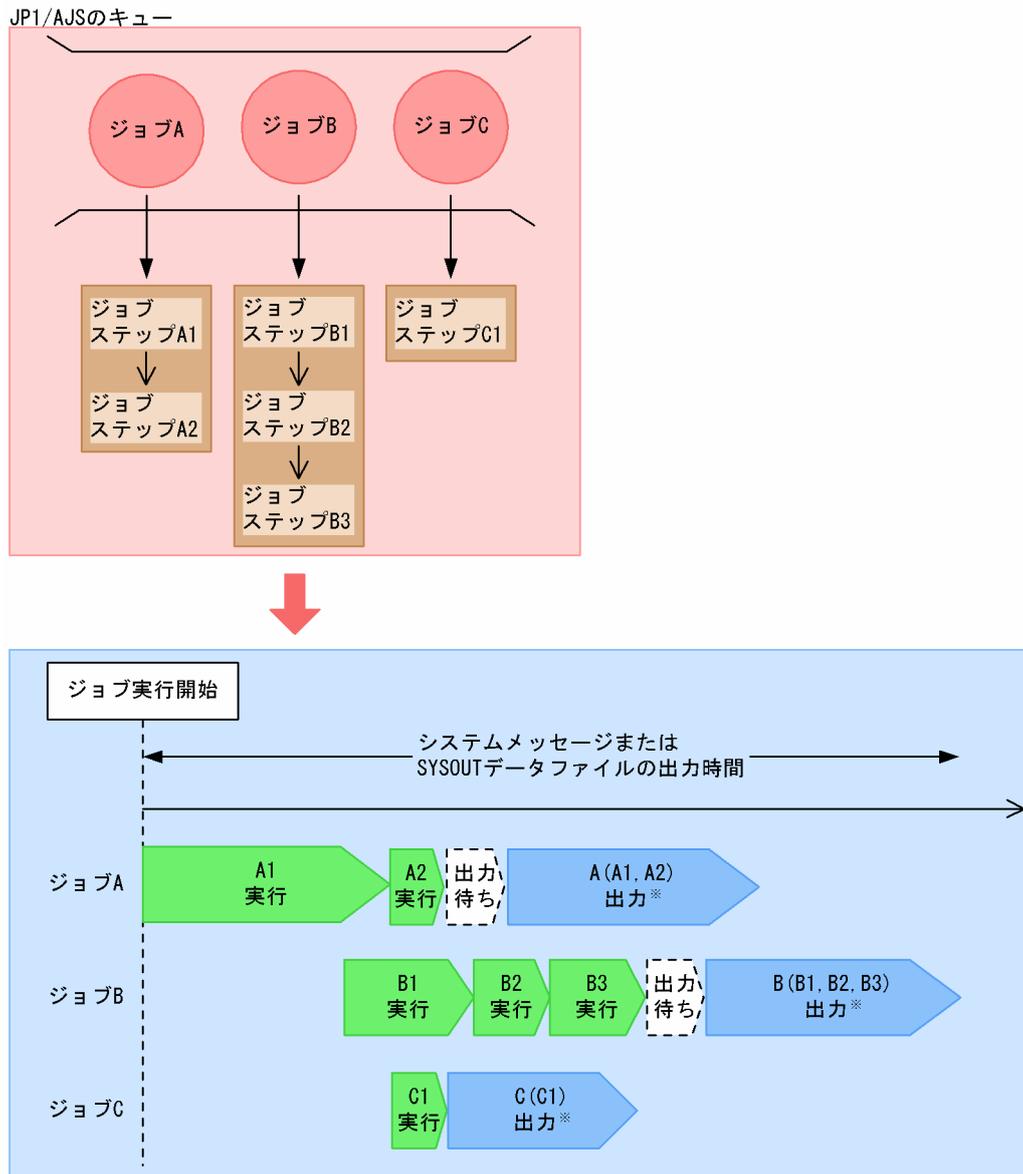
ジョブとジョブとは互いに独立である。したがって、同時に処理するジョブ同士、または先に実行したジョブがあとに実行するジョブに影響を及ぼすことはない。さらに、ファイル上の情報を除けば、ジョブからジョブに情報を引き継がれることもない。ただし、JP1/AJS によるスケジュールによっては、ジョブ相互に実行順序関係ができる。

ジョブ内ではジョブステップ間に前後関係があり、システムに入力された順番で連続的に処理される。

例えば、ある COBOL 言語で書かれたソースがある。このソースをプログラムとして実行するためには、まず COBOL コンパイラでこのソースをコンパイルし、リンクをとり、そのあとで実行する必要がある。このコンパイル、リンクエディット、および実行の各段階が、ジョブの3つのジョブステップとなる。

ジョブとジョブステップの実行順序の関係を次の図に示す。この図では、ジョブ多重度を3に設定した JP1/AJS から3つのジョブを実行する場合を仮定する。

図 4-1 ジョブとジョブステップの実行順序の関係



(凡例)

○ ○ ○ ○ : キュー

➡ : ジョブステップの実行, 出力時間

⋯ : ジョブステップの出力待ち時間

注※ スプールジョブ転送サービスによる出力を示す。

各ジョブ内のジョブステップは、それぞれのジョブ内に定義された順序で処理される。例えば、ジョブ A の場合はジョブステップ A1 から A2 という順番でジョブステップが実行される。一方、ジョブ多重度が 3 であるため、3 つのジョブは並列に動作する。そのため、ジョブの完了は必ずしもキューに蓄積された順番どおりとはならない。

(4) ジョブ実行のスケジュール

JP1/AJS でジョブ実行のスケジュールを立てる場合、JP1/AJS によって選択されたジョブをプロセス管理下（ジョブコントローラの元）で実行する。

ジョブコントローラの詳細については、「[3.1.3 ジョブコントローラ](#)」を参照のこと。

4.3 バッチジョブ実行システムのシステム設計

4.3.1 動作環境の確認

(1) ユーザプログラム

ジョブステップは「4.4.2 バッチジョブの実行環境」で説明する環境下で動作するため、この環境でユーザプログラムが実行できるようにユーザプログラムを配置する必要がある。プログラムへの PATH 環境変数が設定されていない場合は、JOBLIB DD 要素、STEPLIB DD 要素を利用してジョブ定義 XML ファイルで定義する必要がある。この 2 点に注意してユーザプログラムの配置を決定する。

JOBLIB DD 要素、STEPLIB DD 要素については、「3.3 プログラムのシステムへの登録と使用」を参照のこと。

(2) 定義ファイル

- 設定ファイル (bjex.conf)
バッチジョブを実行する静的な環境を定義する。
- ジョブ定義 XML ファイル
任意のファイルとして用意できるが、業務別にディレクトリに分けて管理することが望ましい。

(3) 一時ファイル

一時ファイルを作成するディレクトリのデフォルトの設定は/tmp である。バッチジョブの一時ファイルとほかのアプリケーションの一時ファイルは、分けて管理することが望ましい。バッチジョブ実行システムで一時ファイルを格納するディレクトリパスは TEMP_FILE_DIR パラメータに指定する。一時ファイルの最大容量は、同時に実行するジョブの最大出力ファイル容量である。障害で一時ファイルが残ることもあり、定期的に削除する必要がある。

(4) 恒久ファイル

バッチジョブが利用する恒久ファイルは、任意のディレクトリに配置ができる。

恒久ファイルは特定のディレクトリ配下にまとめておくことが望ましい。バッチジョブ実行システムで恒久ファイルを格納するディレクトリパスは PERMANENT_FILE_DIR パラメータに指定する。この指定をした場合、ジョブ定義 XML ファイルに記述した恒久ファイル名は、prefix としてこのパラメータが付加され、このパラメータ配下に限定できる。

(5) スプール

プリンタ印刷のために利用するファイルの容量の合計を見積もり、専用の論理ボリュームを作成して利用することを推奨する。プリンタ印刷については、出力後保存属性の指定に従って出力後保存されるため、保存する期間も考慮すること。

(6) システムリソース

バッチジョブ実行システムの実行に必要なシステムリソースについては、リリースノートを参照してください。

4.3.2 スプールジョブ転送サービス

(1) スプールジョブ転送デーモン

スプールジョブ転送サービスがスプールの管理をしている。スプールジョブ転送デーモンを止めると、スプールの管理が正しく行われず。バッチジョブの実行およびスプール管理コマンドを使用する場合には、スプールジョブ転送デーモンを起動しておくこと。システムの起動時および終了時に自動的に実行させることを推奨する。

(2) 並列転送機能

スプールジョブは、伝送カード単位に並列に転送する。転送多重度と転送されるスプールジョブのデータ量を考慮して出力クラスおよびネットワークの設計をする必要がある。

(3) リトライ間隔とリトライ回数の指定

スプールジョブ転送時に転送エラーが発生した場合に、再転送するまでのリトライ間隔を設定ファイルの `JP1FTP_RETRY_INTERVAL` パラメータで指定できる。また、再転送のリトライ回数を `JP1FTP_RETRY_TIMES` パラメータで指定できる。

- リトライ回数に 0 を指定した場合、あるいは指定しない場合、無限にリトライを繰り返す。転送エラーが発生している場合は、障害が回復するまで当該スプールジョブの再転送を続ける。
- リトライ回数に 1 以上を指定した場合は、指定回数分のリトライを実行して転送処理を終了し、次のスプールジョブの転送処理に移る。

転送エラー発生時はリトライ間隔で指定した時間の経過後に再度転送が実行され、このときに障害が回復していればスプールジョブが転送される。再転送はリトライ回数分実行されるが、障害が回復せずに再転送もすべて転送エラーとなった場合は、そのスプールジョブの転送を中止し、次のスプールジョブの転送処理に移る。

転送を中止されたジョブは、`bjextrjob` コマンドで転送ディレクトリから再転送することができる。

なお、リトライする契機は次に示すエラーが発生した場合である。

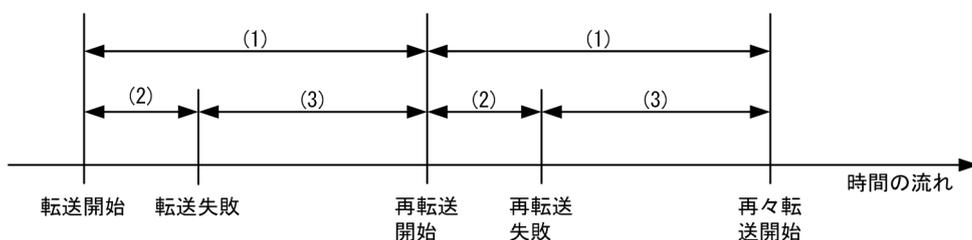
- JP1/FTP との接続確立時 [fts_ftp_open()] の異常
- 伝送要求の登録時 [fts_ftp_syn_request_ex()] の異常
- 伝送の異常終了

JP1/FTP の起動が遅れた場合などで、JP1/FTP との接続確立時に異常を検出してリトライ待ちになっているときは、bjexpdctl コマンドで待ち状態を確認したり解除したりすることができる。

- bjexpdctl コマンドの status オプションで JP1/FTP との接続確立時の異常でリトライ待ちになっているかを確認できる。
- bjexpdctl コマンドの ftp_connect オプションでリトライ待ち状態を解除し、JP1/FTP との接続の確立を再実行できる。

リトライ間隔は、次に示す図の(3)に示す転送エラー検出後の時間を指定する。

図 4-2 リトライ間隔で指定する時間



- (1) : 転送が行われる間隔
- (2) : 転送処理 (JP1/FTPの処理) で経過する時間*
- (3) : 転送エラー検出後、次回転送を開始するまでの待ち時間 (JP1FTP_RETRY_INTERVALパラメータで指定した時間)

注※ JP1/FTPの状態や設定、エラー事象によって異なる。

転送間隔を長く設定すると、転送エラー検出後の再転送開始までの待ち時間が長くなるため注意すること。例えば、JP1/FTP の最大同時伝送数を超えた場合などの一時的な状態の変化によるエラーが発生した場合でも、指定した待ち時間を経過しないと再転送は行われず。このため、JP1/FTP の伝送数については、最大同時伝送数を超えないような運用設計をする必要がある。

4.3.3 出力クラスの設定

バッチジョブを印刷する場合、印刷形式に対応した出力クラスを設定ファイルに設定する必要がある。印刷形式とは、転送先のプリンタの属性「Prinfina MANAGER SP」、[PDE]、および「プリンタ装置」の構成を基に分類し、分類数だけの定義を行う必要がある。

4.3.4 セキュリティ情報の設定

(1) 実行時のユーザ ID

JP1/AJS でジョブの定義およびジョブ実行結果を参照する場合は、JP1 ユーザ名による認証（ログイン）が必要である。JP1 ユーザ ID と各計算機の OS ユーザ ID とのマッピングを JP1/Base のマッピング定義ファイルにあらかじめ定義しておき、ログイン後は、JP1 ユーザ ID に対応する OS ユーザ ID の権限で各計算機のリソースにアクセスする。

ジョブコントローラを起動する場合は、JP1/AJS のジョブ定義上の JP1 ユーザ ID に（マッピング定義ファイルで）対応づけた Agent 上の OS ユーザ名の権限で起動する（ジョブコントローラの ruid（実ユーザ ID）および euid（実効ユーザ ID）は OS ユーザ名のものが設定される）。

ジョブステッププロセスの umask 値はジョブコントローラ起動時の実行ユーザの値が設定される。

(2) ファイルのセキュリティ

ファイルおよびディレクトリへのアクセス権限は、UNIX のパーミッションの設定に従う。ジョブコントローラによって作成されるファイルは、ファイルを作成した OS ユーザだけが参照・更新の権限を持つ。パーミッションを変更したい場合は、設定ファイルで PERMISSION_FILE パラメータおよび PERMISSION_DIR パラメータを指定するか、ジョブ定義 XML ファイルで PERMISSION 要素または DD 要素の FPERM 属性を指定する。

世代管理ファイルは bjexgdg コマンドで作成する。このとき、bjexgdg コマンドの起動時の umask の指定に従いファイルパーミッションが設定される。

(3) 作成ファイルの権限

(a) ファイルのアクセス権限

ジョブコントローラで作成するファイルのアクセス権限を次の表に示す。ただし、恒久ファイルおよび世代ファイルは、ファイル作成時のアクセス権限設定機能によって権限を変更できる。

表 4-1 ジョブコントローラで作成するファイルのアクセス権限

パーミッション	意味
0600	ファイルの所有者（user）だけ read/write を許可する。group および other には read/write を許可しない。

(b) ディレクトリのアクセス権限

ディレクトリおよびスプールジョブディレクトリのアクセス権限を次の表に示す。

表 4-2 ディレクトリおよびスプールジョブディレクトリのディレクトリのアクセス権限

パーミッション	意味
0700	ディレクトリの所有者 (user) だけディレクトリ下での read/write を許可する。group および other にはディレクトリ下での read/write を許可しない。

ディレクトリのアクセス権限は、ファイル作成時のアクセス権限設定機能によって権限を変更できる。

4.3.5 バッチジョブ業務の自動化の検討

バッチジョブ実行システムで業務を自動的に実行する場合、次に示す内容を決めておく必要がある。

(1) バッチジョブ業務に必要なファイルとプログラムの決定

バッチジョブ業務を自動的に実行する場合に必要なファイルとプログラムを決定する必要がある。

(2) バッチジョブ業務の処理の定義

バッチジョブ業務を実現するために、ジョブ定義 XML ファイルで処理を定義する。

(3) バッチジョブ業務のスケジュール

バッチジョブ業務を自動化するために、システム全体として実行するスケジュールを決める。スケジュールの登録は、JP1/AJS の機能を利用すること。

4.3.6 排他/共用制御機能の設定

排他/共用制御機能は、設定ファイルの LOCK_OPTION パラメータで ENABLE を指定することで使用可能となる。LOCK_OPTION パラメータで指定を省略、または DISABLE を指定した場合は、排他/共用制御機能は使用できない。

排他/共用制御をするためには、排他するジョブの実行多重度数と同時期に確保する排他ファイルおよびディレクトリの最大値を見積もっておく必要がある。

バッチジョブ実行システムでは、次に示す内容でデフォルト値を設定している。

ジョブ多重度数：5

同時期に確保する排他ファイルおよびディレクトリの最大値：1275

これらの設定値を変更する場合、設定ファイルで次に示すパラメータを指定する。

- ジョブ多重度数の変更

[LOCK_JOBMPLEX ジョブ多重度数]

- 同時期に確保する排他ファイルおよびディレクトリの最大値の変更

[LOCK_MAXFILE ファイル数]

指定できる値の範囲については、「8. 設定ファイル」の「LOCK_JOBMPLEX（同時に実行できるジョブ数の定義）」および「LOCK_MAXFILE（排他/共用制御で使用する排他資源管理テーブル数の定義）」を参照のこと。

4.3.7 メッセージ制御機能の設定

(1) JP1IM_MANAGER_HOST パラメータの指定

メッセージ制御機能を使用する場合は、JP1IM_MANAGER_HOST パラメータによって、どのホストの JP1/IM - Manager と連携するかを指定する必要がある。指定したホストの JP1/IM - Manager に接続した JP1/IM - View からだけ、応答を入力できる。

JP1IM_MANAGER_HOST パラメータの詳細については、「8. 設定ファイル」の「JP1IM_MANAGER_HOST (JP1/IM - Manager が稼働するホスト名定義)」を参照のこと。

JP1IM_MANAGER_HOST パラメータに存在しないホストを誤って指定した場合、メッセージ制御機能を使用するプログラムまたはコマンドが終了しなくなる。JP1/Base のイベントサーバ設定ファイルの remote-server パラメータを指定しておくことによって、JP1IM_MANAGER_HOST パラメータの指定が誤っていた場合に、プログラムまたはコマンドをエラーで終了させることができるため、remote-server パラメータを指定する運用を推奨する。

JP1/Base のイベントサーバ設定ファイルの remote-server パラメータについては、JP1 のマニュアルを参照のこと。

(2) MAX_RESPONSE_REQUEST パラメータの指定

MAX_RESPONSE_REQUEST パラメータによって、JP1/IM - Manager に同時期に大量のメッセージを送付しないようにするため、サーバ内のメッセージ送付数を指定する。これによって、JP1/IM - Manager に送付するメッセージ数を制限できる。

システム内の応答要求メッセージの数が指定値に達した状態で、アプリケーションが応答要求メッセージを出力しようとした場合、そのアプリケーションは、応答要求メッセージを出力済みのほかのアプリケーションに応答が入力されるまで待ち続ける。このような状態が発生した場合、実行ログに KAKC5000-E メッセージが出力されているため、JP1/IM - Manager 側の運用に問題ないか、確認する必要がある。

MAX_RESPONSE_REQUEST パラメータの詳細については、「8. 設定ファイル」の「MAX_RESPONSE_REQUEST (JP1/IM - Manager に同時に受け付けられる応答待ちイベント数定義)」を参照のこと。

(3) JP1/IM - Manager の設定方法

あらかじめ JP1IM_MANAGER_HOST パラメータに指定したホストで、JP1/IM - Manager の応答待ちイベント管理機能を有効にしておく必要がある。有効にしていない場合、JP1/IM - View による応答の入力ができない。

JP1/IM - Manager の設定およびバッチジョブ実行システムと JP1/IM - Manager との通信の設定については、JP1 のマニュアルを参照のこと。

4.4 JP1/AJS によるジョブスケジュール

4.4.1 バッチジョブ業務の自動化

バッチジョブ実行システムのバッチジョブ業務は JP1/AJS から起動することを前提としている。

JP1/AJS でのバッチジョブ業務の自動化の詳細については、JP1 のマニュアルを参照のこと。

バッチジョブ業務を自動化することによって、コストを削減できるだけでなく、少ない人員で確実にシステムを運用できる。JP1/AJS は、このような定型的なバッチジョブ業務を自動化するための製品である。JP1/AJS は、複雑なバッチジョブ業務の組み合わせの自動化にも対応できる。

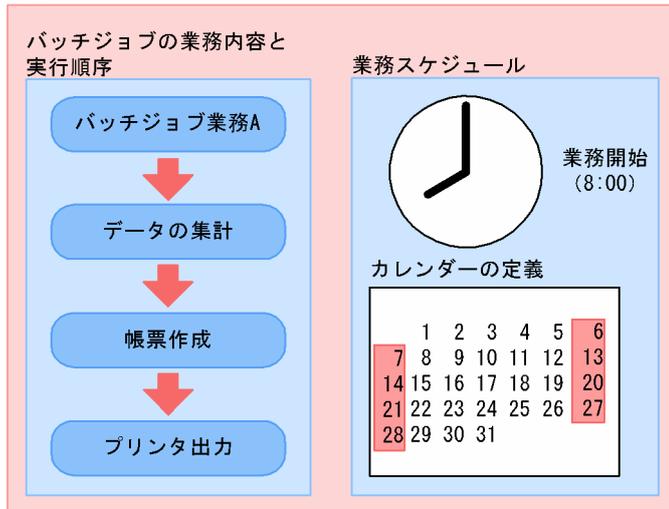
JP1/AJS を使用して、バッチジョブ業務を自動的に実行する場合、次に示す内容を定義する必要がある。

- バッチジョブ業務内容と順序
- バッチジョブ業務を実行するスケジュールまたはバッチジョブ業務の契機となる事象の登録

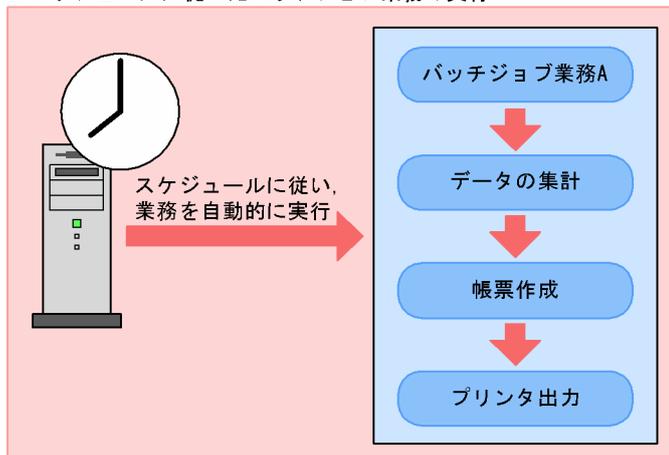
JP1/AJS を使用したバッチジョブ業務の自動化の概要を次の図に示す（図中の番号は、次に示す説明の項番と対応している）。

図 4-3 JP1/AJS を使用したバッチジョブ業務の自動化の概要

1. バッチジョブの業務内容と実行順序、および業務スケジュールを登録



2. スケジュールに従ったバッチジョブ業務の実行



1. バッチジョブの業務内容と実行順序、および業務スケジュールを登録する。

2. 登録されたスケジュールに従って、バッチジョブ業務が自動的に実行される。

また、バッチジョブ実行システムを使用することによって、JP1/AJS の運用時に得られるメリットを次に示す。

- 世代ファイル機能によって、以前の結果ファイルを容易に利用できる。
- 一時ファイル機能によって、不要なファイルをジョブ終了時に削除できる。
- プロシジャ機能によって、ジョブ定義を業務間で共用できる。
- ジョブ定義 XML ファイルに対する変更、追加、または削除によって、柔軟なジョブ定義ができる。
- バッチジョブ業務と実行順序の定義の帳票印刷をスプールとクラス管理することによって、容易に集中管理できる。

(1) バッチジョブ業務と実行順序の定義

多くの業務は、決まった時間に順序に従って実行される。

(例) 売上傳票の集計の順序を次に示す。

1. データベースからのデータの抽出
2. データのソート
3. 帳票作成
4. プリンタ出力

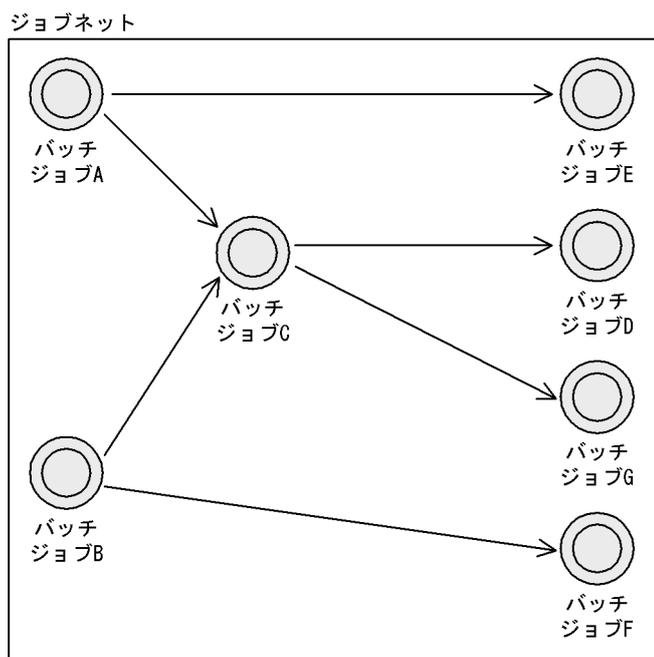
これらの流れをジョブコントローラのジョブステップとして、1~4をジョブ定義XMLファイルに定義することで自動化を実現できるが、12:00にデータベースからデータを抽出するという作業の自動化は行えない。バッチジョブ実行システムとJP1/AJSでバッチジョブ業務と実行順序の定義を実行するには、業務を構成する一連の流れをジョブコントローラで定義し、それぞれのバッチジョブ業務と実行順序の定義の関係をJP1/AJSの実行順序、または実行時間として定義する。

JP1/AJSだけでもコマンド、アプリケーションプログラム、またはシェルスクリプトなどのそれぞれの作業単位に分解すれば、バッチジョブ実行システム相当のジョブを実現できる。これらをJP1/AJSでもジョブと呼ぶ。

バッチジョブ実行システムとJP1/AJSでバッチジョブ業務と実行順序の定義を行う場合には、JP1/AJSのジョブネット相当の提携業務を簡易化できる。

バッチジョブ実行システムとJP1/AJSでバッチジョブ業務と実行順序の定義を行う場合のジョブネットを次の図に示す。

図 4-4 バッチジョブ実行システムと JP1/AJS でバッチジョブ業務と実行順序の定義を行う場合のジョブネット



(2) バッチジョブ業務と実行順序の定義スケジュールの定義

複数のバッチジョブ業務と実行順序の定義スケジュールの定義を自動化するには、バッチジョブ業務と実行順序の定義スケジュールの定義をいつ実行するかを決めるスケジュールの定義が必要である。

JP1/AJS のスケジュールの定義では、会社の営業日・休業日を設定したカレンダー、実行を開始する日時や実行間隔などを定義する。この定義に基づいて、JP1/AJS が実行予定を決め、その日時が来ると自動的にバッチジョブ実行システムのジョブ実行を始める。

(3) バッチジョブ業務と実行順序での定義スケジュールの定義開始の契機登録

ファイルの作成またはイベントの発生などをバッチジョブ業務開始のきっかけの条件として登録しておくことで、決まった時刻にバッチジョブ業務を開始するだけでなく、ファイルの作成またはイベントの発生など何らかの事象が起こった場合にも、バッチジョブ業務を開始できる。

4.4.2 バッチジョブの実行環境

バッチジョブが実行されるシステムの実行環境は JP1/AJS のエージェントを設定することで、バッチジョブの多重度、実行リソース環境を変更する。JP1/AJS のエージェントホスト、デフォルトキュー、バッチジョブ実行時に必要な環境を考えてエージェントの設定をすること。バッチジョブ実行システムを使ってバッチジョブを実行する環境を作成する場合、次に示す内容を考慮する必要がある。

(1) 文字コード

JP1/AJS の文字コードはバッチジョブ実行システムと同じ文字コードに使用すること。バッチジョブ実行システムで使用できる文字コードについては、「[2.3.1\(1\)\(a\) 言語種別の設定](#)」を参照のこと。

(2) JP1/AJS のジョブの種類

バッチジョブ実行システムのジョブを実行するために使用できる JP1/AJS のジョブの種類を次に示す。

- UNIX ジョブ (ただし、キューレスジョブは対象外)
- サブミットジョブ

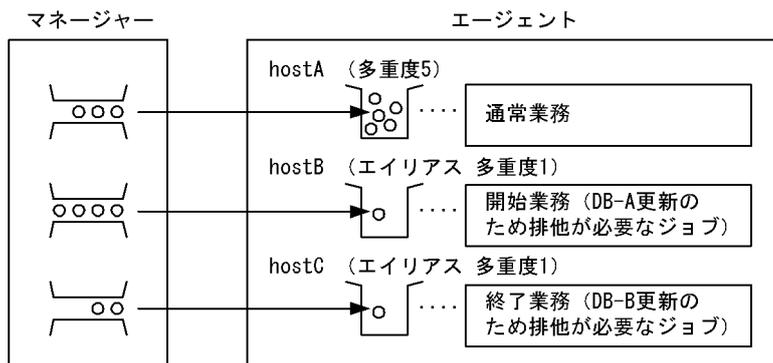
上記以外の種類のジョブで、バッチジョブ実行システムのジョブを実行した場合、動作は保証されない。

(3) バッチジョブの実行多重度の設定

JP1/AJS で管理するエージェントは、エージェントごとにバッチジョブの実行多重度を設定できる。バッチジョブの利用するリソースが大きく異なる場合、同一マシン (1つの IP アドレス) に対して複数のエージェントホストをエイリアスとして定義し、特性の異なるバッチジョブごとにエージェントホストに登録することで、1つのマシン上で多重度を制御できる。例えば、実ホスト (hostA) に対して hostB, hostC というエイリアスを定義し、1つのマシン上で次の図のように業務を振り分けることができる。

バッチジョブ業務の振り分け例を次の図に示す。

図 4-5 バッチジョブ業務の振り分け例



(凡例)

○ ○ ○ ○ : キュー

○ ○ ○ : エージェント

注意事項

長時間掛かるバッチジョブが複数実行されている場合などに、エージェントのジョブ実行多重度に達してしまう場合がある。

実行中のジョブ数がジョブ実行多重度に達している間、後続のジョブはエージェントで実行中のジョブが終了するまでキューイングの状態で待機する。このような場合、予想以上にジョブの実行時間が長く掛かることになるため、ジョブの実行時間、および単位時間当たりのジョブ実行数を考慮したジョブ実行多重度を設定する必要がある。

4.4.3 バッチジョブ定義の記述方法

バッチジョブ定義情報の形式、および必須パラメータについて、UNIX ジョブの UNIT 定義ファイルを基に説明する。

(1) 形式

```
[te="コマンドテキスト";]  
[sc="スクリプトファイル名";]  
[prm="パラメータ";]  
[wkp="作業用パス名";]  
[ev="環境変数ファイル名";]  
[env="環境変数";]...  
[si="標準入力ファイル名";]  
[so="標準出力ファイル名";]  
[se="標準エラー出力ファイル名";]  
[soa={new|add};]  
[sea={new|add};]  
[etm=n];  
[pr="ジョブの実行優先順位";]
```

```
[ex="実行ホスト名";]  
[un="実行ユーザ名";]  
[jd={nm|ab|cod|mdf|exf};]  
[wth=n];  
[tho=n];  
[jdf="終了判定ファイル名";]  
[ts1="転送元ファイル名1";]  
[td1="転送先ファイル名1";]  
[top1={sav|del};]  
[ts2="転送元ファイル名2";]  
[td2="転送先ファイル名2";]  
[top2={sav|del};]  
[ts3="転送元ファイル名3";]  
[td3="転送先ファイル名3";]  
[top3={sav|del};]  
[ts4="転送元ファイル名4";]  
[td4="転送先ファイル名4";]  
[top4={sav|del};]  
[ha={y|n};]  
[eu={ent|def};]  
[jty={q|n};]
```

(2) 必須パラメータ

必須となるパラメータを次に示す。これ以外のパラメータの設定は任意である。

- **sc="スクリプトファイル名"**

スクリプトファイル名として `bjexec` を定義する。バッチジョブ実行システムのバッチジョブ起動コマンドを指定する。

指定例

```
sc="/opt/hitachi/bjex/bin/bjexec"
```

- **prm="パラメータ"**

パラメータとして実行するジョブ定義 XML ファイルを指定する。

指定例

```
prm="/home/bjex/batch/xml/sample.xml"
```

- **env="環境変数"**

環境変数を「環境変数名=値」の形式で指定する。AJS_BJEX_STOP=TERM を必ず指定すること。指定できる文字数は 1~20,479 (単位: バイト) である。

指定例

```
env="AJS_BJEX_STOP=TERM"
```

- **se="標準エラー出力ファイル名"**

標準エラー出力ファイル名を指定する。ジョブコントローラでは、ジョブの終了時にエージェントホストからマネージャーストに対して標準エラー出力ファイルとしてジョブログを転送している。ジョブログが数 MB を超えるような大量データを出力するジョブを実行すると、ファイルのデータ解析処理、

またはファイルのデータ転送で負荷が掛かり、CPU 使用率の増加や、メモリ使用量の増加などジョブの実行が遅延するだけでなく、システム全体の処理に影響を与えるおそれがある。

ジョブログを参照する場合は、指定しないこと。ジョブログを出力しない場合は/dev/null を指定する。また、追加書きオプションを設定すると、ジョブ実行のたびに出力データが蓄積され、転送するファイルのサイズが増加する。この場合も、ファイルのデータ解析処理、またはファイルのデータ転送で負荷が掛かり、ジョブが異常終了したり、JP1/AJS のエージェントホストからマネージャーホストに対するデータ転送が遅延することがある。ジョブが正常に動作しない場合は、次に示す注意事項を確認し、対処方法を実施すること。

注意事項

- 標準エラー出力データが大量に出力される。

"/dev/null"を記述すること。

"/dev/null"を指定した場合、標準エラー出力への出力内容は JP1/AJS - View などから参照できなくなるので、ジョブ定義の「標準エラー出力ファイル名」にファイル名を明示的に指定しているときは、バッチファイルやシェルスクリプト内で、標準エラー出力をリダイレクトすること。

設定方法の詳細については、JP1 のマニュアルを参照のこと。

- pr=ジョブの実行優先順位

ジョブの実行優先順位を指定する。指定できる実行優先順位は「なし」および「1」～「5」である。ジョブの実行優先順位の指定がない場合は「なし」が仮定値となる（「なし」の場合は「1」を仮定する）。JP1/AJS を介さないでサービスからプロセスを直接起動した場合、またはコンソールからプロセスを起動した場合には、nice 値は「20」で実行される。JP1/AJS から実行するジョブをこれらのプロセスと同等の実行優先順位で実行するには、ジョブの実行優先順位に「3」を設定すること。

4.4.4 バッチジョブ実行システム実行ユーザ管理

ジョブネットを実行時、ジョブネットに定義されているジョブの実行ユーザを設定したものを実行ユーザ種別という。ジョブの実行ユーザ種別には、登録ユーザと所有ユーザがあり、バッチジョブ実行システムを実行する場合のユーザ権限を配慮する必要がある。それらを実行する実行先ホスト（エージェントホスト）の OS ユーザを指定する。

4.4.5 注意事項

- ジョブコントローラはバックグラウンドで実行されるため、標準入力を使用するプログラムをジョブで使用している場合、バッチジョブ実行システムがジョブ実行時にエラーとなることがある。標準入力を使用するジョブステッププログラムを実行する場合は、「<」によるリダイレクションによって、テキストファイルなどの内容を標準入力として実行ファイルに与えるなどの措置を取ること。
- ジョブコントローラは、ジョブステップが出力した標準出力を、ジョブコントローラの標準出力、標準エラー出力に出力しない。ジョブステップが出力した標準出力は、JP1/AJS から参照できない。必要に応じて、スプール内を確認すること。

- 標準エラー出力ファイルに、ジョブログが出力される。大量にログが出力される場合、/dev/null の指定を検討すること。
- ジョブの優先順位を指定すること。指定しない場合、ほかのプロセスより実行優先順位が低くなる。
- /etc/profile に TZ 環境変数を指定すること。バッチジョブ実行システムが出力するローカルタイムを合わせないと、ログから実行時間が一意に求まらなくなる。
- AJS_BJEX_STOP=TERM を必ず指定すること。
- **【AIX 限定】**
実行環境はシフト JIS または EUC にする必要がある。

4.5 スプールジョブ転送サービスの運用

スプールジョブ転送サービスは、スプールの管理やスプールジョブ（帳票などの SYSOUT データファイル）をプリントサーバに転送するサービスであり、スプールジョブ転送デーモン (bjexpd)、スプールジョブ管理コマンド (bjexchjob, bjexrmjob など) から成る。

スプールジョブの遷移については、「[3.1.4\(1\) スプールの制御](#)」を参照のこと。

4.5.1 スプールジョブ転送デーモンの運用

スプールジョブ転送デーモンの運用の留意点を次に示す。

スプールジョブ転送デーモンはスプールジョブのプリントサーバへの転送だけではなく、スプールの管理も実施するため、バッチジョブ実行システムの運用中は、スプールジョブ転送デーモンを停止しないこと。

(1) 起動

- バッチジョブ実行システムの運用開始、終了に起動停止するように、マシン起動停止時に自動的に起動、停止する運用を推奨する。
- スプールジョブ転送デーモンは root で起動すること。
- スプールジョブ転送デーモン起動中は、設定ファイルを変更しないこと。

(2) POSIX メッセージキュー

ジョブコントローラからスプールジョブ転送デーモンへジョブの終了などを通知するために、POSIX メッセージキューを使用する。

バッチジョブ実行システムでは、次に示す内容をメッセージキュー名称として使用しているため、この名称をほかの用途で使用しないこと（POSIX メッセージキューはスプールジョブ転送デーモン起動時に生成される）。

- AIX の場合
スプールディレクトリの絶対パス名
- Linux の場合
 - 論理ホストを指定するコマンドオプションを指定した場合
/bjex_mqueue_論理ホスト名
 - 論理ホストを指定するコマンドオプションを指定しなかった場合
/bjex_mqueue

(3) 設定ファイルの変更手順

設定ファイルの変更は次に示す手順で実施する。

1. ジョブコントローラを利用するプログラムとスプールジョブ転送サービスをすべて終了する
2. 設定ファイル変更後、必ず `bjexpdctl conftest` を実行してエラーのないことを確認する
3. エラーがなかった場合、`bjexpdctl start` でスプールジョブ転送デーモンを起動する

4.5.2 帳票処理の概要

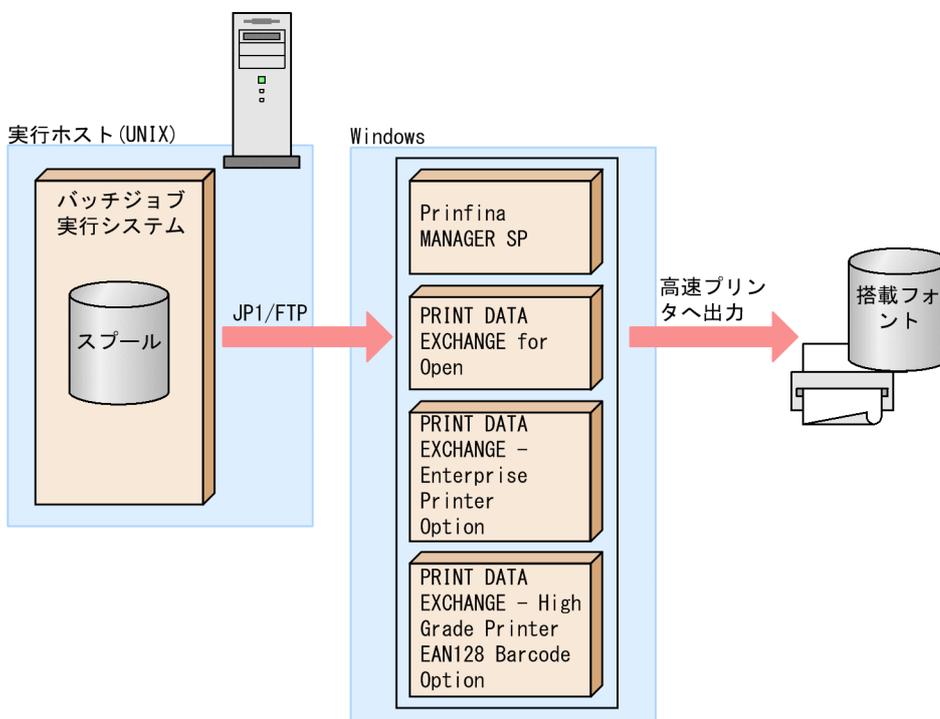
バッチジョブ実行システムは、PDE や Prinfina MANAGER SP と連携して帳票データの印刷や PDF 変換を実行する。バッチジョブの実行終了後、スプールに出力された SYSOUT データファイルをスプールジョブ転送サービスによって出力クラスに従い、PDE が動作する印刷ホストに転送することで実行する。

(1) 高速プリンタへの印刷

Prinfina MANAGER SP と連携して高速プリンタへ印刷を実行する。

Prinfina MANAGER SP との連携による高速プリンタへの印刷の流れを次の図に示す。

図 4-6 Prinfina MANAGER SP との連携による高速プリンタへの印刷の流れ

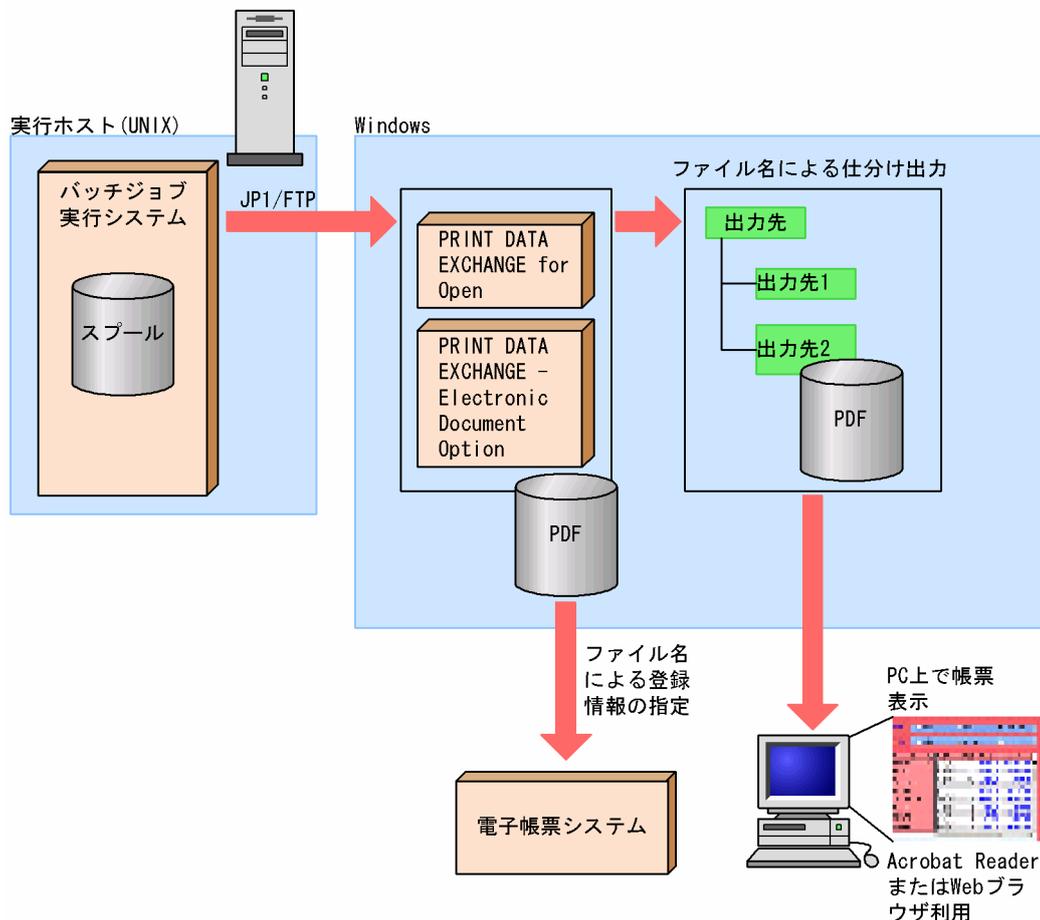


(2) 電子帳票として出力

印刷データを PDF ファイルに変換できる。PDF ファイルは電子帳票システム（HOPSS3/AS REV/OSE）と連携ができる。

電子帳票システムとの連携による PDF ファイル変換の流れを次の図に示す。

図 4-7 電子帳票システムとの連携による PDF ファイル変換の流れ

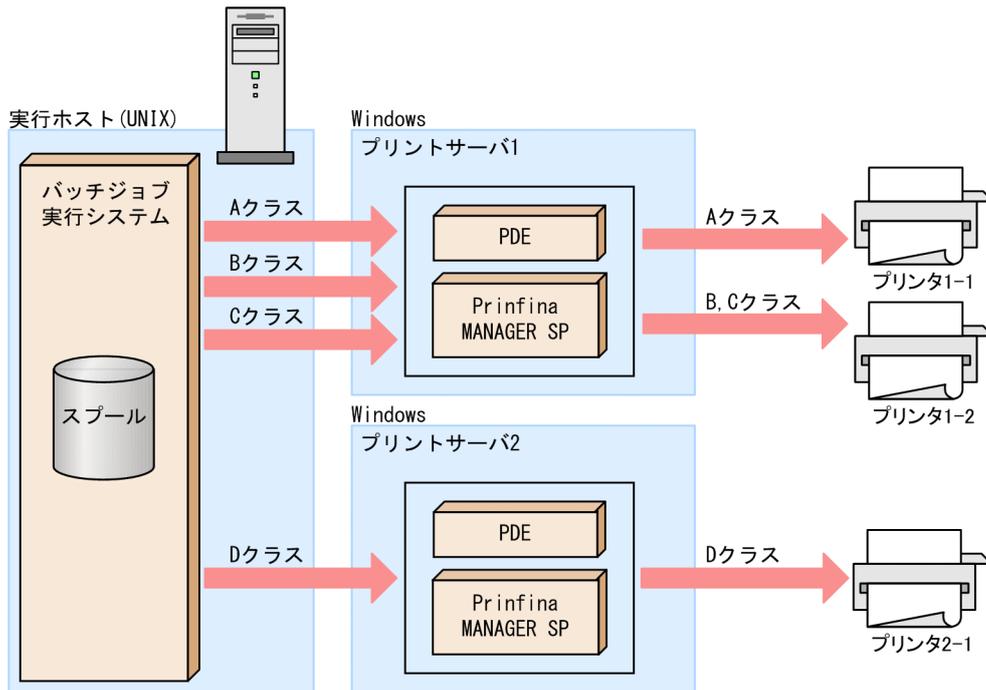


4.5.3 出力クラスによる制御

出力クラスはジョブ定義 XML ファイルと設定ファイルによって帳票の属性として指定される。各クラスは設定ファイルに帳票の属性定義と転送先の定義があり、ジョブ終了後、この出力クラスに振り分けられたスプール内のファイルの印刷を行う。これらの印刷の設定はバッチジョブ実行システムと Prinfina MANAGER SP および PDE が連携して印刷する。

バッチジョブ実行システムと Prinfina MANAGER SP および PDE の連携による印刷の流れを次の図に示す。

図 4-8 バッチジョブ実行システムと Prinфина MANAGER SP および PDE の連携による印刷の流れ



- 設定ファイルに従い、スプールジョブ転送サービスは A, B, および C クラスの印刷データをプリントサーバ 1 に転送し、D クラスの印刷データをプリントサーバ 2 に転送する。
- プリントサーバ 1 では PDE と Prinфина MANAGER SP が連携し、A クラスの印刷データはプリンタ 1-1 へ、B および C クラスの印刷データはプリンタ 1-2 に印刷する。同様に、プリントサーバ 2 では D クラスの印刷データをプリンタ 2-1 に印刷する。

4.5.4 出力保留属性および出力非保留属性

印刷データ (SYSOUT データファイル) は SysoutClass パラメータおよびジョブ定義 XML ファイルの DD 要素の指定によって、出力保留属性 (HOLD 状態) と出力非保留属性 (NOHOLD 状態) になる。

出力非保留属性の SYSOUT データファイルはジョブが終了すると、プリントサーバに転送される。

出力保留属性の SYSOUT データファイルは、ジョブが終了してもプリントサーバには転送しないでスプールに保持する。出力保留属性の SYSOUT データファイルをプリントサーバに転送するには `bjexchjob` コマンドを使用する。

SysoutClass パラメータおよびジョブ定義 XML ファイルの DD 要素の指定によるジョブ実行直後の印刷データの扱いを次の表に示す。

表 4-3 SysoutClass パラメータおよびジョブ定義 XML ファイルの DD 要素の指定によるジョブ実行直後の印刷データの扱い

SysoutClass パラメータの HOLD 値	DD 要素の HOLD 値		
	YES	NO	省略
YES	HOLD	NOHOLD	HOLD
NO または省略	HOLD	NOHOLD	NOHOLD

HOLD 状態の印刷データをプリントサーバに転送する場合、bjexchjob コマンドを入力する。

注

次に示す SYSOUT データファイルはプリントサーバへ転送されない。

- サイズ 0 の SYSOUT データファイル
- 設定ファイルで JP1FTPCARD パラメータが指定されていない出力クラスの SYSOUT データファイル

4.5.5 スプールジョブの削除

出力後保存属性の SYSOUT データファイルがあるスプールジョブは、プリントサーバへの転送後も、出力保留状態のスプールジョブとして残るので、bjexrmjob コマンドを使用して定期的に削除すること。

使用例

7 日以上前のスプールジョブを削除する場合

```
bjexrmjob -T 7
```

4.5.6 印刷ジョブの削除

スプールジョブ転送サービスは印刷ジョブをプリントサーバに転送後、転送ディレクトリ下の印刷ジョブを削除する。ただし、スプールジョブ転送サービスが起動されていない、転送が終了していない状態でスプールジョブ転送サービスを終了した、または回線、転送先サーバの障害などの場合に、転送ディレクトリ下に印刷対象ファイルが削除されないで残るときがある。

bjextrjob コマンドで定期的に転送ディレクトリ下のファイルを削除すること。

使用例

259,200 秒 (3 日) 以上前に属性変更を行った印刷ジョブを削除する。

```
bjextrjob -d -t 259200
```

4.5.7 スプールジョブの回復

運用中はスプールジョブ転送サービスを停止しないこと。誤って停止した状態で実行したスプールジョブは、印刷対象にはならない。

このようなスプールジョブは、bjexchjob コマンドの-S オプション指定で、印刷対象のスプールジョブとして回復できる。

使用例

- スプールジョブの状態を確認する。

```
$ bjexlsjob -f 523
JOBID S JOBNAME
000523 - SAMPLE1
```

- スプールジョブの状態を回復する。

```
$ bjexchjob -S 523
$ bjexlsjob 523
JOBID S JOBNAME
000523 H SAMPLE1
```

- 回復したスプールジョブを印刷する。

```
$ bjexchjob -R 523
```

注 上記の手順はスプールジョブ転送サービスを起動した状態で実施すること。

4.5.8 スプールジョブを転送しない運用

帳票印刷に、Prinfina MANAGER SP および PDE を使用しないで、EUR を使用する場合は、スプールジョブを転送しない運用ができる。スプールジョブを転送しない運用をする場合は、設定ファイルで SPOOL_JOB_TRANSFER パラメータに NO を指定する。この場合、JP1/FTP のセットアップは不要である。

ただし、スプールジョブを転送しない運用でもジョブの管理をするために、スプールジョブ転送サービスの起動およびスプールディレクトリは必要である。

4.6 JP1/IM による実行状態の監視と実行結果の確認

スケジュール定義したジョブネットを実行登録したあとは、各ジョブネットの実行状況の監視、および実行結果の確認を行う。

バッチジョブ実行システムは JP1/IM と連携してバッチジョブの監視ができる。JP1/IM の統合コンソールによって、バッチジョブの運用状況を集中的に管理できる。

4.6.1 統合コンソールによる集中監視

バッチジョブ実行システムは、JP1/Base の提供するイベントサービスを使用している。

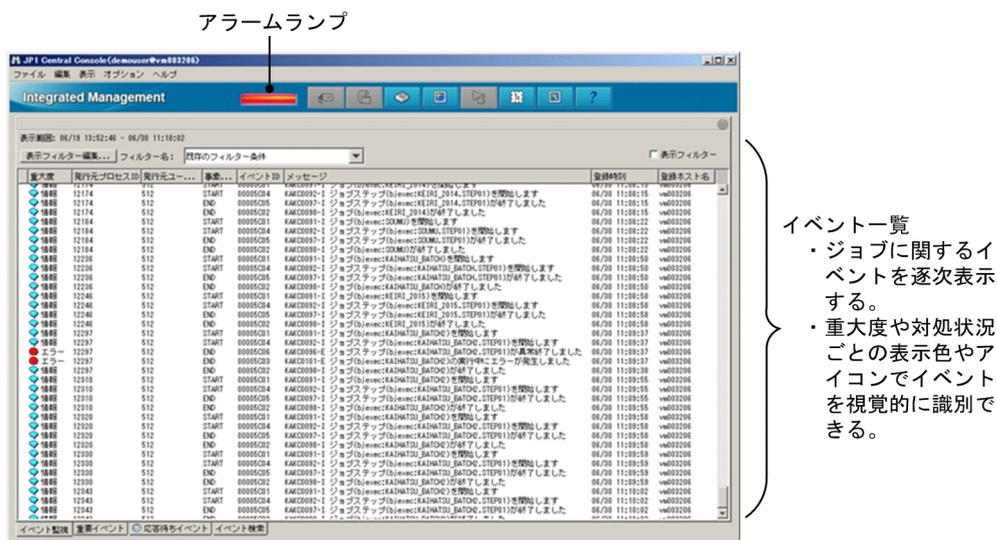
統合コンソールは、バッチジョブで発生した事象を JP1 イベントとして集め、時系列に表示することによって、システムを集中的に監視する。

バッチジョブ実行システムが発行する JP1 イベントについては、「10.6 JP1 イベント一覧」を参照のこと。

システムで発生した管理を必要とする事象は、JP1/IM のマネージャーホストに集められ、一元管理している。[イベントコンソール] 画面の [イベント監視] ページでは、この一元管理している JP1 イベントを逐次表示し、システム全体で発生した事象を集中的に監視できる。

[イベントコンソール] 画面 ([イベント監視] ページ) を次の図に示す。

図 4-9 [イベントコンソール] 画面 ([イベント監視] ページ)



[イベント監視] タブを選択して、
[イベント監視] ページを表示する。

4.6.2 ジョブ名などの情報をイベント一覧に表示

バッチジョブ実行システムのジョブ名やジョブ通し番号などをイベント一覧の項目に表示できる。

[イベントコンソール] 画面で表示されるイベント一覧には、固有の拡張属性をマッピングできる。バッチジョブ実行システムが発行する JP1 イベントは、固有の拡張属性としてジョブ通し番号やジョブ名を設定している。詳細については、JP1 のマニュアルを参照のこと。また、バッチジョブ実行システムが発行する JP1 イベントについては、「10.6 JP1 イベント一覧」を参照のこと。

バッチジョブ実行システムのジョブ通し番号、ジョブ名、およびステップ名を表示させた [イベントコンソール] 画面を次の図に示す。

図 4-10 ジョブ名などの情報を表示させた [イベントコンソール] 画面

重大度	発行元プロセスID	ユーザー名	発行元ユー...	事象...	イベントID	メッセージ
情報	# 010064	# SAMPLE	# STEP01	START	00005C04	KAKC0092-I ジョブステップ(bjexec:...
情報	# 010064	# SAMPLE	# STEP01	END	00005C05	KAKC0097-I ジョブステップ(bjexec:...
情報	10320	root	0	END	00005C02	KAKC0098-I ジョブ(bjexec:SAMPLE)?
情報	14128	root	0	END	00005C12	KAKC3001-I bjexpdが終了しました。
情報	# 010065	# SAMPLE	512	START	00005C10	KAKC3000-I bjexpdが起動しました。
情報	# 010065	# SAMPLE	# STEP01	START	00005C01	KAKC0091-I ジョブ(bjexec:SAMPLE)?
情報	# 010065	# SAMPLE	# STEP01	START	00005C04	KAKC0092-I ジョブステップ(bjexec:...
情報	# 010065	# SAMPLE	# STEP01	END	00005C05	KAKC0097-I ジョブステップ(bjexec:...
情報	# 010066	# SAMPLE	512	END	00005C02	KAKC0098-I ジョブ(bjexec:SAMPLE)?
情報	# 010066	# SAMPLE	512	START	00005C01	KAKC0091-I ジョブ(bjexec:SAMPLE)?
情報	# 010066	# SAMPLE	# STEP01	START	00005C04	KAKC0092-I ジョブステップ(bjexec:...
情報	# 010066	# SAMPLE	# STEP01	END	00005C05	KAKC0097-I ジョブステップ(bjexec:...
情報	# 010066	# SAMPLE	512	END	00005C02	KAKC0098-I ジョブ(bjexec:SAMPLE)?
情報	16682	root	0	START	00005C10	KAKC3000-I bjexpdが起動しました。
情報	# 010067	# SAMPLE	512	START	00005C01	KAKC0091-I ジョブ(bjexec:SAMPLE)?
情報	# 010067	# SAMPLE	# STEP01	START	00005C04	KAKC0092-I ジョブステップ(bjexec:...

注 #が付与されているものがバッチジョブ実行システムの情報

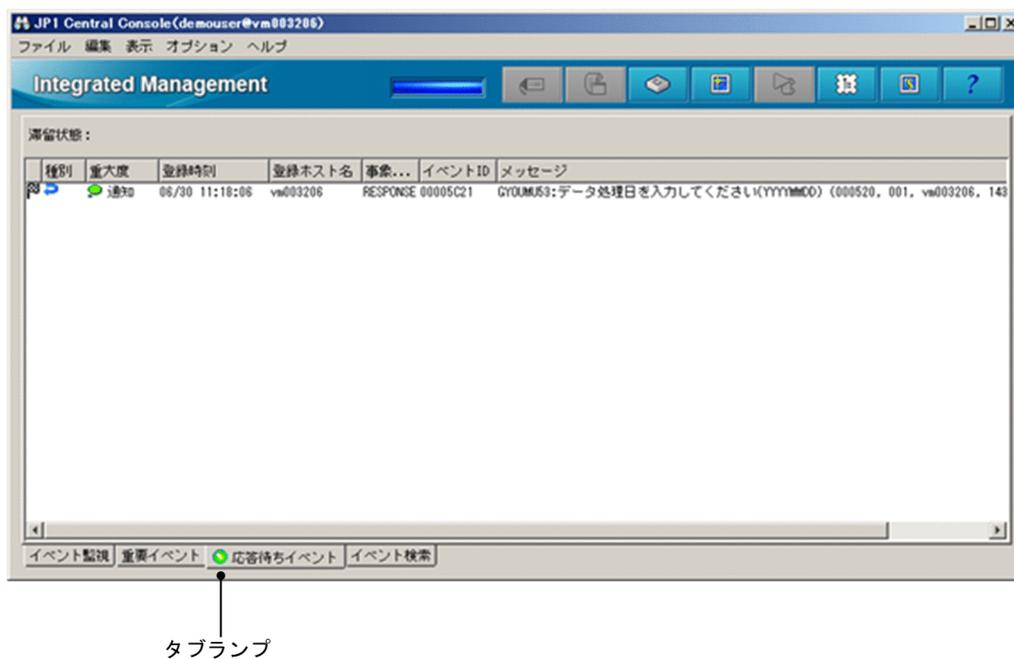
4.6.3 応答要求メッセージの監視

COBOL アプリケーションが ACCEPT 文、STOP 文または CBLTYPE サービスルーチンで応答要求メッセージを出力した場合、または bjexm_message コマンドまたはメッセージ出力用カタログドプロシジャで応答要求メッセージを出力した場合、そのメッセージは応答待ちイベント（イベント ID：00005C21）として JP1/IM - View に表示される。

応答していない応答待ちイベントが存在する場合は、[応答待ちイベント] 画面のタブランプが緑色に点灯する。

[応答待ちイベント] 画面を次の図に示す。

図 4-11 「応答待ちイベント」画面



4.7 EUR を使用した帳票印刷

ここでは、EUR (EUR Print Service または EUR Print Manager) を使用した帳票印刷について説明する。

4.7.1 機能概要

CSV 形式の SYSOUT データファイルを EUR 帳票として印刷できる。

CSV 形式のファイルは、ジョブ定義 XML ファイルの DD 要素の EURFORM 属性で指定した EUR 定義ファイルに従って、ジョブ終了時に EUR の eurps コマンドまたは eurpmc コマンドでプリンタに印刷する。

EUR のコマンドによる印刷は、SysoutClass の HOLD 属性に関係なく行う。

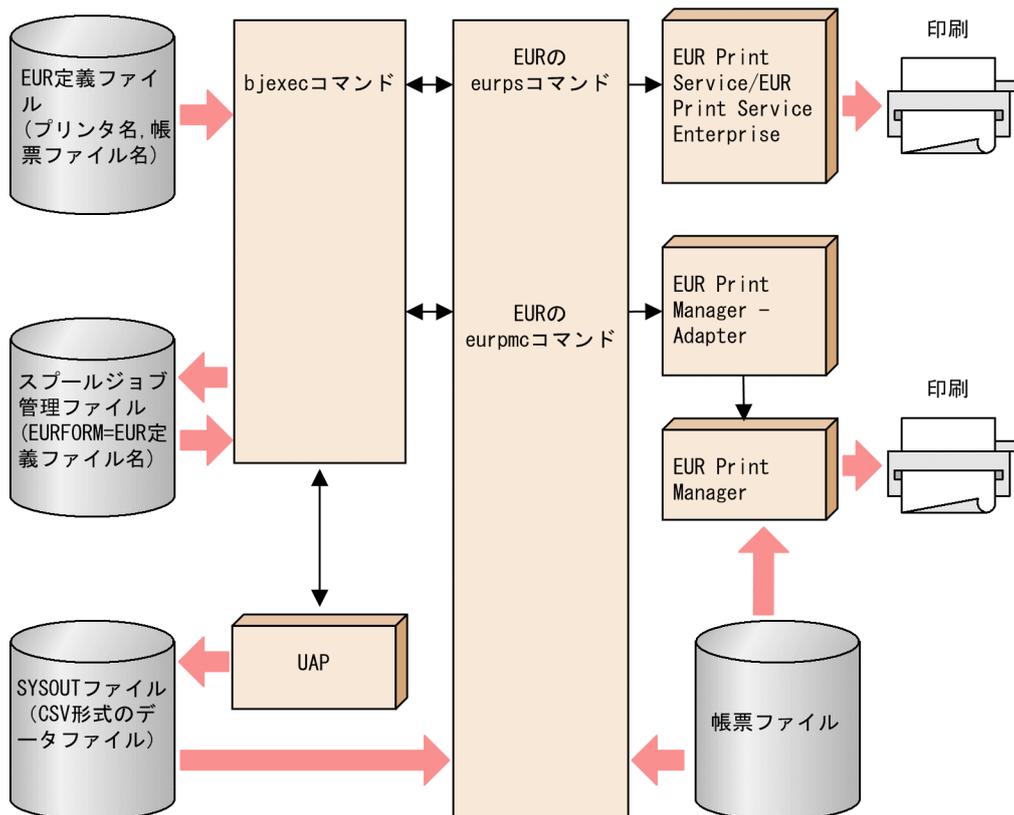
EUR 定義ファイルには、EUR のコマンドに指定するプリンタまたは使用する帳票ファイル名を定義しておく。

EUR で印刷する SYSOUT データファイルはジョブ終了時に印刷するため、PDE などのプリンタサーバに転送する必要がなく、転送しない SysoutClass を使用する必要がある。また、SYSOUT データファイルの HOLD 属性に関係なく印刷する。

EUR を使用した帳票印刷の流れを次の図に示す。

なお、EUR Print Manager を使用した印刷では「印刷管理サーバを使用した直接印刷」の方法で印刷する。

図 4-12 EUR を使用した帳票印刷の流れ



4.7.2 EUR 定義ファイルの準備

CSV 形式の SYSOUT データファイルを EUR で印刷するには、事前に EUR 定義ファイルで印刷先のプリンタや帳票ファイル名を定義しておく必要がある。EUR 定義ファイルには、次に示す内容で作成・定義する。作成した EUR 定義ファイルは、設定ファイルで定義した EUR 定義ファイル格納ディレクトリに格納すること。

記述形式

```
[プリンタ名];[帳票ファイル名];[ユーザ定義データファイル名];[/messageオプション|/nomessageオプション];[追加のオプション]
```

説明

プリンタ名

出力するプリンタ名を定義する。

EUR Print Manager では OS のプリンタ名を指定する。EUR Print Service ではプリンタ名を指定する。

EUR Print Manager では eurpmc コマンドに次のオプションが付加される。

- -pm pm -m print -sm printer -lsp プリンタ名

EUR Print Service では、eurps コマンドに次のオプションが付加される。

- -p プリント名

プリント名を省略した場合、上記のオプションは自動的に付加されないため、追加のオプションにコマンドの形式で必要なものを個別に指定する必要がある。

帳票ファイル名

帳票ファイルのファイル名を指定する。帳票ファイル名を省略した場合、EUR のコマンドに帳票ファイル名が付加されないため通常は省略しないこと。フルパスまたはファイル名を指定できるが、ファイル名だけを指定した場合、EUR の仕様に従ってパスが決定される。

帳票ファイル名を指定した場合、EUR Print Manager では eurpmc コマンドに次のオプションが付加される。

- -rf 帳票ファイル名

帳票ファイル名を指定した場合、EUR Print Service では eurps コマンドに次のパラメータが付加される。

- 帳票ファイル名,SYSOUT データファイル名[,ユーザ定義データファイル名]

ユーザ定義データファイル名

ユーザ定義データファイル名を指定する。ユーザ定義データファイル名を使用しない場合は省略できる。省略した場合、EUR のコマンドにユーザ定義データファイル名は付加されない。

フルパスまたはファイル名が指定できるが、ファイル名だけを指定した場合、EUR の仕様に従ってパスが決定される。また、帳票ファイル名を省略した場合、ユーザ定義データファイル名の指定は無効となる。

ユーザ定義データファイル名を指定した場合、EUR Print Manager では eurpmc コマンドに次のオプションが付加される。

- -usr ユーザ定義データファイル名

ユーザ定義データファイル名を指定した場合、EUR Print Service では eurps コマンドに次のパラメータが付加される。

- 帳票ファイル名,SYSOUT データファイル名,ユーザ定義データファイル名

/message オプション, /nomessage オプション

メッセージを出力するかどうかを次のオプションで指定する。

- message : メッセージを出力する。
- nomessage : メッセージを出力しない。
- 上記以外 : メッセージを出力する (message を仮定する)。

省略した場合、EUR コマンドに -message または -nomessage オプションは付加されないため、EUR コマンドのデフォルト値に従って処理する。

このオプションを指定した場合、EUR Print Manager では次のオプションが eurpmc コマンドに付加される。

-message または -nomessage

このオプションを指定した場合、EUR Print Service でも次のオプションが eurps コマンドに付加される。

-message または -nomessage

追加のオプション

追加のオプションを EUR のコマンドの形式で指定する。指定はコマンドの指定形式のままであり、各オプションは空白で区切って指定する。使用できる文字は ASCII 文字の範囲とする。また、ファイル名などに空白を含むことはできない。追加のオプションにそれ以外の指定項目と同様のオプション、または背反のオプションを指定した場合、EUR のコマンドはエラーになるので指定する場合は注意すること。追加オプションには、空白で区切られた文字列として最大 126 個指定でき、それ以上指定した場合はエラー (KAKC0213-E メッセージ出力) で終了する。

指定できるオプションについてはマニュアル「uCosminexus EUR Print Manager 帳票出力」の eurpmc コマンドの説明を参照のこと。または、マニュアル「uCosminexus EUR サーバ帳票出力」の eurps コマンドの説明を参照のこと。

なお、使用できないオプションを指定してもバッチジョブ実行システムはエラーチェックはしないで EUR のコマンドの処理に従う。

注意事項

- EUR 定義ファイルには、ジョブを実行するユーザに参照権限を設定しておく必要がある。
- EUR 定義ファイル名は、記号名称の範囲で記述する。それ以外の文字は使用できない。
- 定義は 1,024 バイト以内で記述する。
- 各項目は「;」または「空白」で区切る。

項目を省略する場合は、「;」を使用する必要がある。使用できる文字は ASCII の範囲であり、ファイル名などに空白を含めることはできない。「"」で囲んでも空白は区切り記号として扱う。

上記指定は、次のように EUR のコマンドに変換されて EUR のコマンドが実行される。

eurpmc コマンドの場合 (EUR Print Manager)

```
eurpmc [追加のオプション] {-message|-nomessage}  
[-pm pm -m print -sm printer -lsp 印刷先プリンタサーバ名]  
[-rf 帳票ファイル名] [-map マッピングデータファイル名]  
[-usr ユーザ定義データファイル名]
```

- -message | -nomessage : メッセージ出力オプション
- -pm pm : 印刷モードの指定 (印刷管理サーバで直接印刷する)
- -m print : 機能の指定。プロパティで指定した帳票名およびデータファイルに従って、帳票を印刷する。
- -sm printer : プリンタ指定方法 (OS 上のプリンタ名)
- -lsp 印刷先プリンタサーバ名 : プリンタ名 (EUR 定義ファイルのプリンタ名)
- -rf 帳票ファイル名 : 帳票ファイル名の指定
- -map マッピングデータファイル名 : SYSOUT データファイル名をフルパスで指定

- -usr ユーザ定義データファイル名：ユーザ定義データファイル名を指定

eurps コマンドの場合 (EUR Print Service)

```
eurps [追加のオプション] {-message|-nomessage}  
[-p プリンタ名] [帳票ファイル名, SYSOUTファイル名  
[, ユーザ定義データファイル名]]
```

- -message | -nomessage：メッセージ出力オプション
- -p プリンタ名：印刷先プリンタ名
- 帳票ファイル名：印刷時に使用する帳票ファイル名
- SYSOUT ファイル名：マッピングデータファイルである SYSOUT データファイル名をフルパスで指定
- ユーザ定義データファイル名：ユーザ定義データファイル名を指定

注意事項

- 1,025 バイト以上で記述した場合は、KAKC0213-E メッセージを出力して EUR 帳票の印刷処理を中断する。
- フォームズフィード、改行、復帰、タブは空白として扱う。
- 帳票ファイル名を省略した場合は、ユーザ定義データファイル名は無視する。

4.7.3 設定ファイルの定義

(1) EUR 定義ファイル格納ディレクトリの定義

EUR 定義ファイル格納ディレクトリは、設定ファイルの EUR_DEFINITION_FILE_DIR パラメータで定義する。また、ジョブを実行するすべてのユーザに参照権限を設定する必要がある。

(2) EUR 連携するプログラムの選択

設定ファイルの EUR_PROGRAM パラメータに EUR Print Service または EUR Print Manager のどちらと連携するかを EUR のコマンドのパス名で指定する。

(3) CSV 形式の SYSOUT データファイルを転送しない設定

EUR で印刷する CSV 形式のファイルは、PDE などのプリンタサーバに転送しない SysoutClass を使用すること。プリンタサーバに転送しない設定は、設定ファイルの SysoutClass パラメータで、JP1FTPCARD パラメータを指定しないことで設定できる。なお、EUR で印刷する CSV 形式のファイルをプリンタサーバに転送した場合、プリンタサーバの動作は保証しない。

(4) EUR のエンコード種別の定義

ジョブ定義 XML ファイルで指定する EUR のデータファイルのエンコードに対応する LANG 環境変数値を定義する。通常は定義する必要はなく、バッチジョブ実行システムの LANG 環境変数と UAP の LANG 環境変数が異なるときに定義し、DD 要素の EURENCODE 属性で EUR のマッピングデータごとにエンコード種別を指定する。

EUR で使用できるデータのエンコードとバッチジョブ実行システムで許される組み合わせを次の表に示す。なお、この表の組み合わせ以外は保証しない。詳細については、EUR のマニュアルを参照のこと。

表 4-4 EUR で使用できるデータのエンコードとバッチジョブ実行システムで許される組み合わせ

OS	バッチジョブ実行システムの LANG	UAP の LANG	EURENCODE の指定	マッピングデータのエンコード
AIX	SJIS	SJIS	なし	SJIS, UTF-16 (BOM で自動判定)
	EUC	EUC	なし	EUC, UTF-16 (BOM で自動判定)
Linux	C	SJIS	SJIS	SJIS, UTF-16 (BOM で自動判定)

EURENCODE 属性で指定できる属性値は、設定ファイルの EUR_LANG パラメータで定義する。

EUR_LANG パラメータの定義例を次に示す。

```
EUR_LANG SJIS ja_JP.SJIS
```

注意事項

バッチジョブ実行システムの LANG 環境変数と異なるエンコードを EURENCODE 属性で指定した場合、EUR のコマンド実行時のメッセージが文字化けすることがある。

4.7.4 ジョブ定義 XML ファイルでの指定

CSV 形式の SYSOUT データファイルを EUR で印刷するには、ジョブ定義 XML ファイルであらかじめ作成した EUR 定義ファイル名を指定する。

```
<DD NAME="EURDATA" SYSOUT="A" EURFORM="EUR定義ファイル名" />
```

UAP 実行時の LANG 環境変数とバッチジョブ実行システムの LANG 環境変数が異なる場合、SYSOUT データファイルのエンコード種別を明示指定する。EUR エンコード種別は、EUR によるデータのエンコードの仕様を理解した上で指定すること。

```
<DD NAME="EURDATA" SYSOUT="A" EURFORM="EUR定義ファイル名" EURENCODE="EURエンコード種別" />
```

UAP 実行時の LANG 環境変数とバッチジョブ実行システムの LANG 環境変数が異なる場合の定義例を次に示す。

```
<DD NAME="EURDATA" SYSOUT="A" EURFORM="EUR001" EURENCODE="SJIS" />
```

注意事項

バッチジョブ実行システムの設定ファイルで EUR_DEFINITION_FILE_DIR パラメータの指定をしない場合、ジョブ定義 XML ファイルで指定した EURFORM および EURENCODE 属性は無効となる。

4.8 EUC 言語環境でのジョブ実行【AIX 限定】

ここでは、EUC 言語環境でのジョブ実行について説明する。

4.8.1 機能概要

EUC 言語環境でバッチジョブを実行して、UAP で文字コードが EUC-HJ であるデータを扱うことができる。

EUC-HJ の文字コードデータを格納できるのは、SYSOUT データファイルと UAP が使用する一般のファイルだけである。それ以外のファイルは、すべて内字の範囲で使用する必要がある。

EUC の文字を含んだ SYSOUT データファイルは、PDE ではサポートしていないため、PDE を使用して印刷や PDF 変換する場合には、UAP で UTF-8 に変換して SYSOUT データファイルを作成する必要がある。

4.8.2 言語環境の設定

バッチジョブを EUC 言語環境で実行する場合、次の設定が必要である。

文字コード種別：EUC コード

LANG 環境変数値：ja_JP

4.8.3 ジョブ定義 XML ファイルの指定

ジョブ定義 XML ファイルのパス名は ASCII の範囲とする必要がある。ジョブ定義 XML ファイルを指定する際の注意事項を次に示す。

(1) prolog

prolog に記述する encode 属性は EUC_JP, Shift_JIS, または ASCII にする必要がある。encode 属性を ASCII にした場合、ジョブ定義 XML ファイルは ASCII の範囲で記述する必要がある。encode 属性を Shift_JIS にした場合、ジョブ定義 XML ファイルはコメント以外 ASCII の範囲で記述する必要がある。

(2) DD 要素の ENCODE 属性

PDE で EUC のデータはサポートしていないため、PDE で印刷または PDF 変換する場合、UAP で UTF-8 にコード変換する必要がある。また、UTF-8 の印刷データを印刷または PDF 変換するためには、次に示す DD 要素の ENCODE 属性でエンコード種別を指定する必要がある。

```
ENCODE="{SJIS|UTF8}"
```

TYPE=SYSOUT を指定した場合にだけ指定でき、SYSOUT データファイルのエンコード種別を指定する。

SJIS : シフト JIS

UTF8 : UTF-8

4.8.4 EUC 言語環境でのメッセージ制御機能の注意事項

メッセージ制御機能を使用した COBOL アプリケーションのメッセージと応答は、ASCII の範囲だけで使用できる。

なお、メッセージの日本語部分は文字化けすることがある。

4.8.5 設定ファイル

EUC 言語環境でジョブを実行する場合、設定ファイルは ASCII のコード範囲で記述する必要がある。また、SysoutClass パラメータで UTF-8 の SYSOUT データファイルを PDE で印刷または PDF 変換する場合、次に示すエンコード種別を SysoutClass パラメータの属性として定義できる。

```
ENCODE {SJIS|UTF8}
```

ジョブ定義 XML ファイルで指定がないときに仮定する SYSOUT データファイルのエンコード種別を指定する。

SJIS : シフト JIS

UTF8 : UTF-8

4.9 バッチジョブ実行システムの環境を複数構築する場合の設定

物理的なサーバの中に複数のバッチジョブ実行システムの環境を構築できる。ここでは、複数のバッチジョブ実行システム環境を構築する設定について説明する。

4.9.1 複数のバッチジョブ実行システムの運用の前提条件

複数のバッチジョブ実行システムの環境を構築する場合、JP1 の論理HOST環境を複数構築し、JP1 の論理HOST環境単位で起動する必要がある。

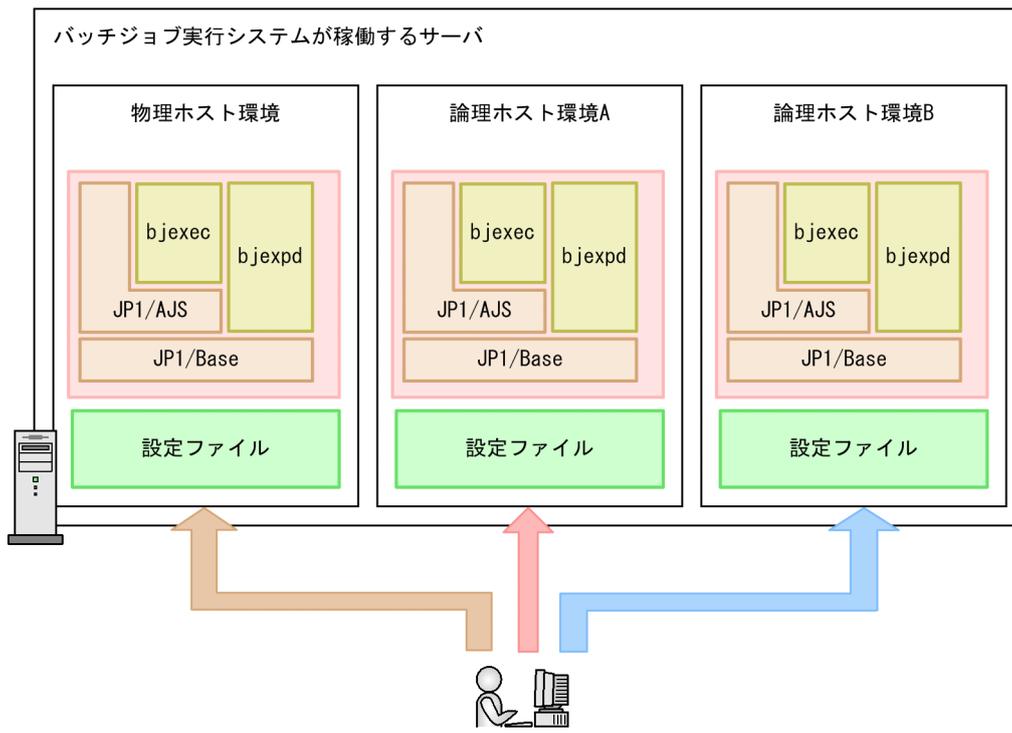
(1) バッチジョブ実行システムの物理HOSTと論理HOST

バッチジョブ実行システムには物理HOST環境と論理HOST環境の2種類が存在する。

- バッチジョブ実行システムの物理HOST環境
設定ファイルとして `bjex.conf` を参照し、物理HOST環境のプールジョブ転送デーモンと連携する。また、物理HOST環境のJP1 と論理HOST環境のJP1 のどちらとも連携することができる。論理HOSTのJP1 と連携するには設定ファイルの `JP1_HOSTNAME` パラメータにJP1 論理HOST名を指定する。
- バッチジョブ実行システムの論理HOST環境
設定ファイルとして `bjex_論理HOST名.conf` を参照し、論理HOST環境のプールジョブ転送デーモンと連携する。ここで、`bjex_論理HOST名.conf` の論理HOST名は、連携するJP1 の論理HOST名である。なお、論理HOST環境のJP1 とだけ連携できる。

バッチジョブ実行システムの物理HOST環境は1つであるが、バッチジョブ実行システムの論理HOST環境は複数構築できる。複数のバッチジョブ実行システムの環境の構成例を次の図に示す。

図 4-13 複数のバッチジョブ実行システム環境



(凡例)

-  : 物理ホスト環境へのアクセス
-  : 論理ホスト環境Aへのアクセス
-  : 論理ホスト環境Bへのアクセス

(2) バッチジョブ実行システムの論理ホストでのコマンド指定方法

バッチジョブ実行システムの論理ホストでコマンドを実行する場合、対応する JP1 の論理ホスト名を指定してコマンドを実行する。

各コマンドの実行方法を次に示す。

- bjexec コマンドの場合
 - JP1/AJS から実行する場合
コマンド -h
 - コマンドラインから実行する場合
コマンド -h -n 論理ホスト名
- bjexec コマンド以外の場合
コマンド -h 論理ホスト名

bjexec コマンドを JP1/AJS から実行する場合、論理ホスト環境で稼働している JP1/AJS から実行すること。論理ホストで稼働している JP1/AJS から実行した場合、環境変数 JP1_HOSTNAME に論理ホスト名が設定される。bjexec コマンドは、環境変数 JP1_HOSTNAME から論理ホスト名を取得する。

4.9.2 論理ホストのバッチジョブ実行システムの環境設定

バッチジョブ実行システムの論理ホスト環境を作成する際の環境設定について説明する。

バッチジョブ実行システムの環境ごとに次に示す準備が必要である。

(1) ディレクトリの作成

次のディレクトリを作成する。

- カタログドプロシジャ用
- 一時ファイル用
- スプール用
- 恒久ファイル用
- 実行ログ用
- トレース用

ただし、同じ物理ホスト上に複数のバッチジョブ実行システムを運用する場合、次のディレクトリをバッチジョブ実行システム間で共有してはならない。個々の環境ごとに独自のディレクトリを指定すること。

- 実行ログ出力ディレクトリ（設定ファイルの LOG_DIR パラメータで指定できる）
- スプールディレクトリ（設定ファイルの SPOOL_DIR パラメータで指定できる）
- 一時ファイルディレクトリ（設定ファイルの TEMP_FILE_DIR パラメータで指定できる）
- トレースを出力するディレクトリ（設定ファイルの TRACE_DIR パラメータで指定できる）

(2) JP1 のセットアップ

JP1 の論理ホスト環境をセットアップしたあとにバッチジョブ実行システムの論理ホストのセットアップが必要である。JP1 のセットアップについては、JP1 のマニュアルを参照すること。

(3) バッチジョブ実行システムの設定ファイル

バッチジョブ実行システムの論理ホスト環境ごとに設定ファイルを作成する必要がある。次の形式で設定ファイルを作成すること。

論理ホストで動作するバッチジョブ実行システムの設定ファイル

```
/opt/hitachi/bjex/conf/bjex_論理ホスト名.conf
```

複数のバッチジョブ実行システムを運用する場合の設定ファイルに記述する内容は、物理ホストの設定ファイル (bjex.conf) に記述する内容と同様である。詳細については、「8. 設定ファイル」を参照すること。

(4) スプールジョブ転送デーモンの起動

論理ホストのバッチジョブ実行システムを運用する場合、論理ホスト環境ごとにスプールジョブ転送デーモンを起動する必要がある。起動方法については、「7. コマンド」の「bjexpdctl (スプールジョブ転送デーモンの起動, 停止 (管理者用))」を参照のこと。

(5) 論理ホスト用スプールジョブ転送デーモンの自動起動と自動停止

論理ホストのスプールジョブ転送デーモンを、システムの起動時および終了時に自動的に起動および停止させる設定方法を次に示す。

(a) AIX の場合

システム起動時の自動起動機能

システム起動時の自動起動機能を設定するには、mkitab コマンドを使用して、自動起動の設定をする。

```
mkitab "論理ホストのスプールジョブ転送デーモンのレコード:2:wait:/opt/hitachi/bjex/sbin/bjexpdctl -h 論理ホスト名 start"
```

論理ホストのスプールジョブ転送デーモンは、連携する JP1 シリーズの論理ホスト用デーモンのあとに起動するよう設定すること。JP1/Base, JP1/IM, JP1/AJS, JP1/FTP の順に自動起動を設定する場合、次に示す指定をしてコマンドを実行する。JP1 製品の自動起動設定の詳細については、各 JP1 製品のマニュアルを参照のこと。

```
mkitab -i hnr2mon "論理ホスト用JP1/Baseのレコード:2:Wait:/etc/opt/jp1base/jbs_start.cluster  
論理ホスト名"  
mkitab -i 論理ホスト用JP1/Baseのレコード "論理ホスト用JP1/IM - Managerのレコード:2:wait:/etc/  
opt/jp1cons/jco_start.cluster 論理ホスト名"  
mkitab -i 論理ホスト用JP1/IM - Managerのレコード "論理ホスト用JP1/AJSのレコード:2:wait:/etc/  
opt/jp1ajs2/jajs_start.cluster 論理ホスト名"  
mkitab -i 論理ホスト用JP1/AJSのレコード "jp1_fts:2:wait:/etc/opt/jp1_fts/startup "  
mkitab -i jp1_fts "論理ホスト用スプールジョブ転送デーモンのレコード:2:Wait:/opt/jp1as/sbin/  
bjexpdctl -h 論理ホスト名 start"
```

設定後、lsitab コマンドを使用して、設定内容を確認すること。lsitab コマンドは、次に示す指定をして実行する。

```
lsitab -a
```

次に出力例を示す。

```

init:2:initdefault:
brc::sysinit:/sbin/rc.boot 3 >/dev/console 2>&1 # Phase 3 of system boot:
hntr2mon:2:once:/opt/hitachi/HNTRLib2/etc/D002start
論理ホスト用JP1/Baseのレコード:2:wait:/etc/opt/jp1base/jbs_start.cluster 論理ホスト名
論理ホスト用JP1/IM - Managerのレコード:2:wait:/etc/opt/jp1cons/jco_start.cluster 論理ホスト名
論理ホスト用JP1/AJSのレコード:2:wait:/etc/opt/jp1ajs2/jajs_start.cluster 論理ホスト名
jp1_fts:2:wait:/etc/opt/jp1_fts/startup
論理ホスト用スプールジョブ転送デーモンのレコード:2:wait:/opt/hitachi/bjex/sbin/bjexpdctl -h
論理ホスト名 start

```

システム終了時の自動停止機能

システム終了時の自動停止機能を設定するには、`/etc/rc.shutdown` を編集して、連携する JP1 シリーズの論理ホスト用デーモンより先に停止させるように次に示す記述を追加のこと。

```

test -x /opt/hitachi/bjex/sbin/bjexpdctl && /opt/hitachi/bjex/sbin/bjexpdctl -h 論理ホスト名 stop
:
連携するJP1シリーズ製品の論理ホスト用デーモンの停止処理
:

```

(b) Linux の場合

`/usr/lib/systemd/system` ディレクトリへの追加

`/opt/hitachi/bjex/sample` ディレクトリに格納されている `bjex.service` を、ファイル名を変更して `/usr/lib/systemd/system` に追加する。`bjex.service` を、ファイル名を変更して追加する例を次に示す。

```

cp /opt/hitachi/bjex/sample/bjex.service /usr/lib/systemd/system/bjex_論理ホスト名.service
chmod u=rw,go=r /usr/lib/systemd/system/bjex_論理ホスト名.service
chown root:root /usr/lib/systemd/system/bjex_論理ホスト名.service

```

`bjex_論理ホスト名.service` ファイルの編集

`bjex_論理ホスト名.service` ファイルは、物理ホスト用の設定になっているため、論理ホスト用の設定に修正する必要がある。次の項目を修正すること。

表 4-5 `bjex_論理ホスト名.service` ファイルの変更内容

項目	変更前	変更後
After	<code>jp1_base.service</code>	JP1/Baseの論理ホストのUnit名
ExecStart	<code>/opt/hitachi/bjex/sbin/bjexpdctl start</code>	<code>/opt/hitachi/bjex/sbin/bjexpdctl -h 論理ホスト名 start</code>
ExecStop	<code>/opt/hitachi/bjex/sbin/bjexpdctl stop</code>	<code>/opt/hitachi/bjex/sbin/bjexpdctl -h 論理ホスト名 stop</code>

また、必要に応じて他の各項目を編集すること。

自動起動の有効化

`bjex.service` に対して自動起動を有効化する。自動起動を有効化するには次のように指定する。

```
systemctl --system enable bjex_論理ホスト名.service
```

4.9.3 複数のバッチジョブ実行システムの運用の注意事項

- バッチジョブ実行システムの論理ホスト環境ごとに、設定ファイル/opt/hitachi/bjex/conf/bjex_論理ホスト名.conf を作成すること。
- ジョブ識別子はバッチジョブ実行システムの環境ごとに管理する。
- 物理ホストのバッチジョブ実行システムで、JP1 の論理ホストと連携する場合、JP1_HOSTNAME パラメータに JP1 の論理ホスト名を指定する必要がある。詳細については、「8. 設定ファイル」を参照のこと。
- 複数のバッチジョブ実行システムを運用している場合、排他/共用制御および実行中のジョブ情報表示はバッチジョブ実行システムごとに行う。あるバッチジョブ実行システムから別のバッチジョブ実行システムの排他/共用制御および実行中のジョブの情報を参照することはできない。
- 同じ物理ホスト上に複数のバッチジョブ実行システムを運用する場合に、共有できないディレクトリがある。詳細については、「4.9.2 論理ホストのバッチジョブ実行システムの環境設定」を参照のこと。
- 設定ファイルの LOCK_KEY_ID パラメータおよび MSG_KEY_ID パラメータは、論理ホスト環境ごとに異なる識別情報を指定することが望ましい。

なお、バッチジョブ実行システムの運用全体の注意事項については、「4.11 バッチジョブ実行システムの運用時の注意事項」を参照のこと。

4.10 クラスタシステムで運用する場合の設定

ここでは、クラスタシステムで運用する場合の設定について説明する。

4.10.1 クラスタシステムおよびクラスタ運用の概要

ここでは、クラスタシステムおよびバッチジョブ実行システムのクラスタ運用の概要について説明する。

(1) クラスタシステムの概要

クラスタシステムとは、複数のサーバシステムを連携して1つのシステムとして運用するシステムで、1つのサーバで障害が発生しても、別のサーバで業務を継続できるようにすることを目的としている。

クラスタシステムは、処理を実行するホストと、障害が発生したときに処理を引き継げるように待機しているホストで構成されている。業務の実行中のサーバを実行系サーバ、実行系の障害時に業務を引き継げるよう待機しているサーバを待機系サーバと呼ぶ。障害発生時は、実行系サーバから待機系サーバに処理を引き継いで業務の停止を防ぐ。この障害時に処理を引き継ぐことをフェールオーバーという。

フェールオーバーはJP1の論理ホスト単位で実施される。JP1と連携するアプリケーションは、フェールオーバーして業務を継続するために、JP1の論理ホスト環境で動作させる必要がある。

クラスタシステム構成のJP1の論理ホストは、サービスとして動作するアプリケーション、共有ディスク、および論理IPアドレスの3つの要素で構成されている。サービスとして動作するJP1やバッチジョブ実行システムなどのアプリケーションは、共有ディスクにデータを格納し、論理IPアドレスで通信を行う。

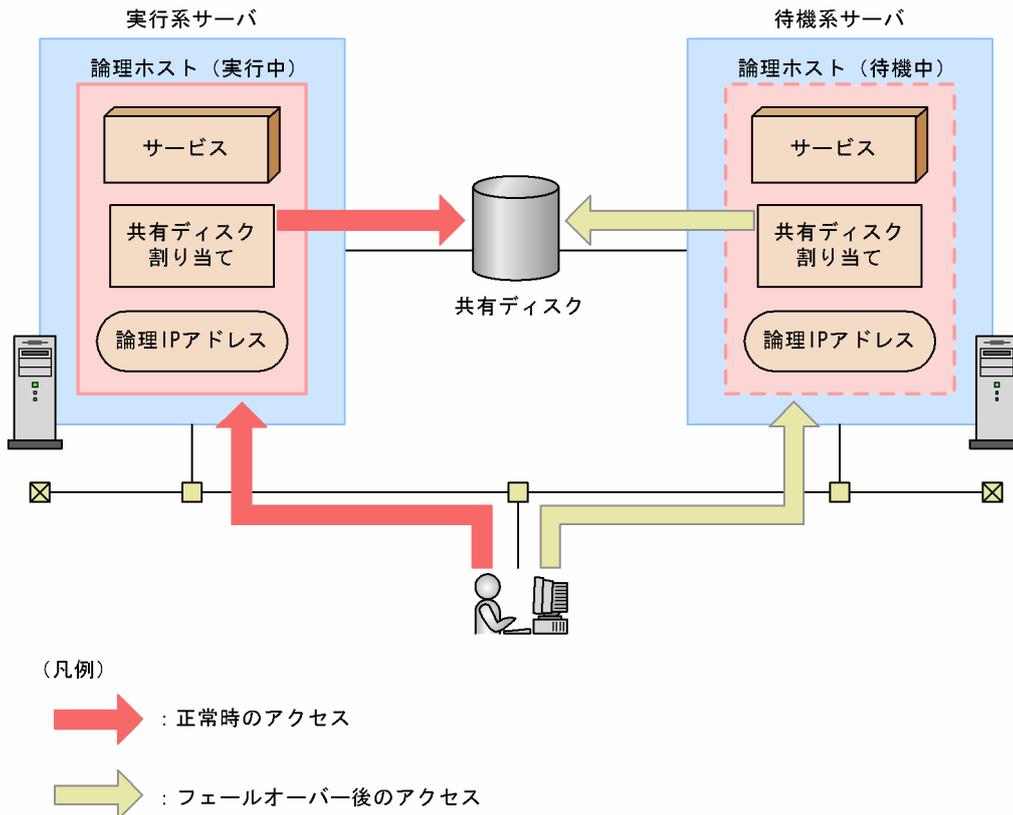
クラスタシステム構成のJP1の論理ホストを構成する各要素を次の表に示す。

表 4-6 クラスタシステム構成のJP1の論理ホストの構成要素

クラスタシステム構成のJP1の論理ホストの構成要素	構成要素の説明
サービス	クラスタシステムで実行するJP1やバッチジョブ実行システムなどのアプリケーションである。実行系サーバで障害が発生すると、待機系サーバで同じ名称のサービスを起動し、処理を引き継ぐ。
共有ディスク	実行系と待機系の両方に接続されたディスク装置である。フェールオーバー時に引き継ぐ情報(定義情報、実行状況など)を保存すると、実行系サーバで障害が発生した場合、待機系サーバが共有ディスクへの接続を引き継ぐ。
論理IPアドレス	実行系サーバの動作中に割り当てられるIPアドレスである。実行系サーバで障害が発生したときは、同じ論理IPアドレスの割り当てを待機系サーバが引き継ぐ。そのため、クライアントからは同じIPアドレスでアクセスでき、1つのサーバが常に動作しているように見える。

正常時およびフェールオーバー後のアクセスを次の図に示す。

図 4-14 正常時およびフェールオーバー後のアクセス



実行系サーバが稼働している場合は、実行系サーバで共有ディスクや論理 IP アドレスが割り当てられてサービスが動作する。実行系サーバで障害が発生した場合、待機系サーバが共有ディスクと論理 IP アドレスを引き継ぎ、実行系サーバと同じサービスを起動する。フェールオーバーによって物理的なサーバが変わった場合、待機系サーバが共有ディスクと論理 IP アドレスを引き継ぐため、クライアントには同じ IP アドレスのサーバが動作しているように見える。

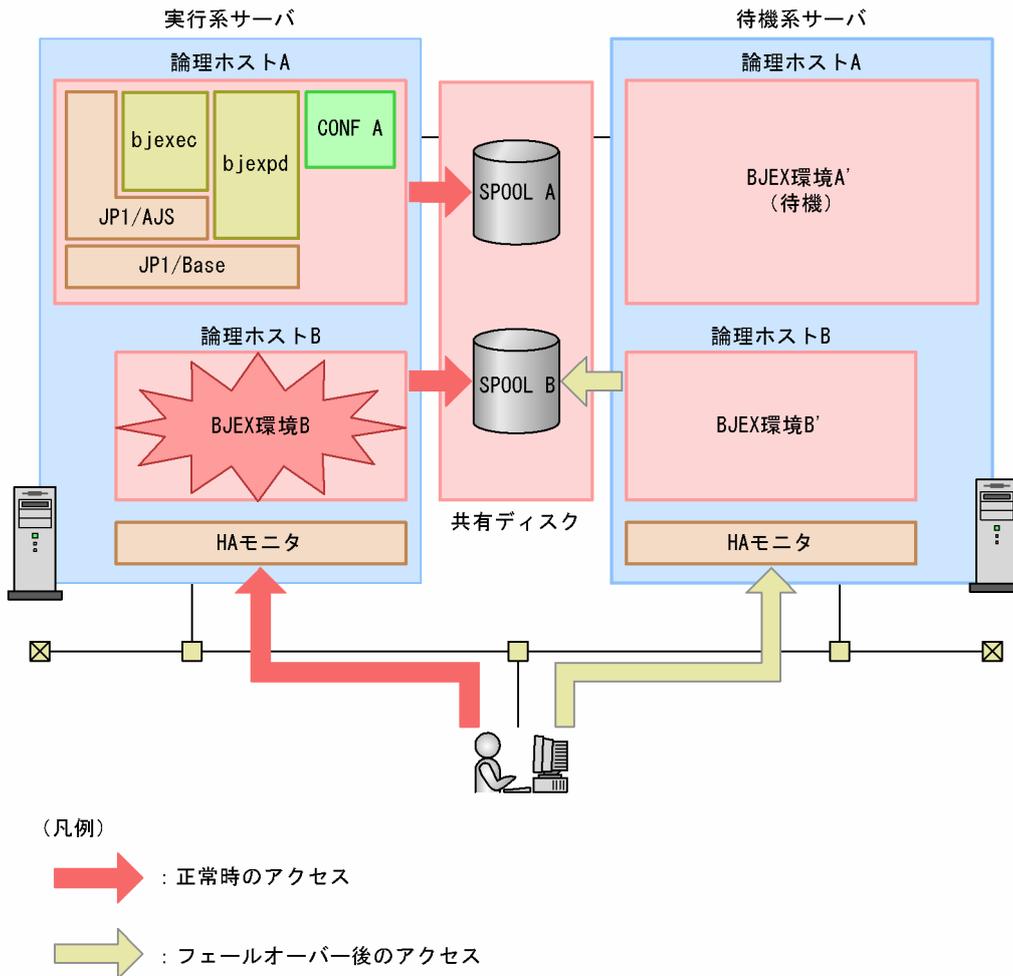
(2) バッチジョブ実行システムのクラスタ運用の概要

ここでは、クラスタシステムでバッチジョブ実行システムを運用する概要について説明する。バッチジョブ実行システムはクラスタシステムに対応している。

クラスタシステムでバッチジョブ実行システムを運用するためには、論理ホスト環境の JP1 と連携する必要がある。論理ホスト環境の JP1 と連携していれば、バッチジョブ実行システムの物理ホスト環境でも論理ホスト環境でもどちらともクラスタシステムに対応できる。また、フェールオーバー時に引き継ぎが必要なデータを格納するための共有ディスク、および論理 IP アドレスが必要となる。JP1 のセットアップについては、JP1 のマニュアルを参照すること。クラスタシステムで実行されるバッチジョブ実行システムは、共有ディスクに格納したデータを使用し、フェールオーバー時に実行系から待機系に引き継いで処理を続けることができる。バッチジョブ実行システムが共有ディスクに格納する必要のあるデータはスプールである。

バッチジョブ実行システムでのクラスタ運用の概要を次の図に示す。

図 4-15 バッチジョブ実行システムでのクラスタ運用の概要



4.10.2 クラスタ運用の前提条件とサポート範囲

バッチジョブ実行システムは、クラスタシステムでは論理ホスト環境で動作する JP1 と連携し、フェールオーバーに対応する。クラスタシステムで運用する場合のバッチジョブ実行システムの前提条件は、前提プログラムの JP1 と同様に、共有ディスクや論理 IP アドレスの割り当て、削除、および動作監視がクラスタソフトによって正常に制御されていることである。

(1) バッチジョブ実行システムの論理ホスト環境の前提条件

JP1 の論理ホスト名を次の表に示す文字数にする必要がある。

表 4-7 JP1 の論理ホスト名の文字数

バッチジョブ実行システムの環境	JP1 の論理ホスト名の文字数
物理ホスト	1～255
論理ホスト	1～244

また、使用するクラスタソフトに上限がある場合、上限を超えないように JP1 の論理ホスト名を設定すること。JP1 の論理ホスト環境自体の前提条件については、JP1 のマニュアルを参照すること。

(2) システム環境の前提条件

クラスタ構成にするためのシステム環境の前提条件は、JP1 でクラスタシステムを構築する場合と同様である。JP1 のマニュアルを参照すること。

(3) バッチジョブ実行システムがサポートする範囲

クラスタシステムでバッチジョブ実行システムを運用する場合、バッチジョブ実行システムがサポートする範囲はバッチジョブ実行システム自体の動作だけである。共有ディスクや論理 IP アドレスの制御はクラスタソフトの制御に依存する。

また、バッチジョブ実行システムの環境およびシステム環境の前提条件が満たされていない、またはクラスタシステム環境の制御に問題がある場合は、バッチジョブ実行システムの動作で発生した問題もサポートの対象外となる。この場合は、クラスタソフトや OS で問題に対処すること。

4.10.3 クラスタ運用の環境設定

クラスタ運用に対応するためのバッチジョブ実行システムの環境設定について説明する。

(1) 環境設定で設定する項目

環境設定で設定する項目を次に示す。

(a) 共有ディスク上への共有ディレクトリの作成

実行系・待機系の切り替え時に情報を共有するために、次の表に示すディレクトリを共有ディスク上に作成する。

表 4-8 情報を共有するために作られる共有ディレクトリの種別

共有ディレクトリの種別	作成基準
カタログドプロシジャ用	△
一時ファイル用	△
スプール用	○
恒久ファイル用	△
実行ログ用	△
トレース用	△

(凡例)

- ：共有ディスク上に作成することが必須である。
- △：システムの運用方針に従い設定する。

(b) クラスタソフトへの登録

クラスタソフトに登録する機能と各機能で使用するコマンドを次の表に示す。

表 4-9 クラスタソフトに登録する機能と各機能で使用するコマンド

登録する機能	説明	使用するコマンド
起動	スプールジョブ転送デーモンを起動する。	bjexpdctl start ^{※1} bjexpdctl -h 論理ホスト名 start ^{※2}
停止	スプールジョブ転送デーモンを停止する。	bjexpdctl stop ^{※1} bjexpdctl -h 論理ホスト名 stop ^{※2}
動作監視	スプールジョブ転送デーモンが正常に動作していることを監視する。	bjexpdctl status ^{※1} bjexpdctl -h 論理ホスト名 status ^{※2}

注※1

物理ホストのバッチジョブ実行システムの場合

注※2

論理ホストのバッチジョブ実行システムの場合

(2) インストール

実行系、待機系それぞれのローカルディスク上にバッチジョブ実行システムをインストールすること。また、共有ディスク上にはバッチジョブ実行システムをインストールしないこと。

(3) セットアップ

バッチジョブ実行システムをクラスタシステムで運用するには、JP1 をセットアップしたあとに実行系および待機系のセットアップが必要である。JP1 のセットアップについては、JP1 のマニュアルを参照すること。

バッチジョブ実行システムのセットアップを次に示す。

(a) バッチジョブ実行システムの設定ファイル

バッチジョブ実行システムをクラスタシステムで動作させるためには、JP1 の論理ホストと連携するバッチジョブ実行システムの設定ファイルを実行系・待機系それぞれに作成する必要がある。

クラスタシステムで運用する場合の設定ファイルに記述する内容は、物理ホストの設定ファイル (bjex.conf) に記述する内容と同様である。詳細については、「8. 設定ファイル」を参照すること。

(b) 実行系・待機系のファイル構成

バッチジョブ実行システムは、実行系サーバと待機系サーバで次のファイルを同じ構成にする必要がある。

- ジョブ定義 XML ファイル
- プロシジャ XML ファイル
- バッチジョブ実行システムの設定ファイル
- ジョブ定義 XML ファイルの EXEC 要素に指定する実行プログラム

4.10.4 クラスタ運用に関する注意事項

- 物理ホストのバッチジョブ実行システムをクラスタシステムで運用する場合、必ず実行系・待機系それぞれの設定ファイルの JP1_HOSTNAME パラメータに JP1 の論理ホスト名を指定すること。
- バッチジョブ実行システムをクラスタシステムで運用する場合、必ず JP1 の論理ホスト環境で bjexec コマンドを実行すること。
- 共有ディスク上のファイルは、実行系サーバと待機系サーバのどちらからでもアクセスできるようにアクセス権限を設定すること。
- フェールオーバー後、ジョブを再実行するかどうかは JP1/AJS の設定による。詳細については、JP1/AJS のマニュアルを参照のこと。
- 共有するディスクは、実行系サーバと待機系サーバとで同じアクセス権限を設定すること。

なお、バッチジョブ実行システムの運用全体の注意事項については、「[4.11 バッチジョブ実行システムの運用時の注意事項](#)」を参照のこと。

4.11 バッチジョブ実行システムの運用時の注意事項

4.11.1 システム停止時の注意事項

ジョブの実行中に OS やハードウェア障害などでシステムが停止した場合、停止直前に実行中だったジョブの実行結果や、一時ファイルの削除や世代ファイルの更新などのファイルの後処理の完了は保証されない。このため、OS のリブート後やクラスタ構成の系切り替え後にジョブを再実行する前に、次の点を確認する必要がある。

- 障害発生時に実行されていたジョブが世代ファイルを使用している場合は、bjexgdg コマンドで世代ファイル群の状況を確認する。ジョブ実行前の状態になっていない場合は、bjexhkgdg コマンドで不要な世代を削除する。
- 削除できなかった一時ファイルは自動的に削除されないため、設定ファイルの TEMP_FILE_DIR パラメータで指定したディレクトリの内容を確認し、手動で削除する。また、設定ファイルの TEMP_FILE_DIR_XML_USE パラメータで YES を指定している場合、各ジョブの一時ファイルを定義する DD 要素の DSN 属性で指定したディレクトリ内の一時ファイルも削除する。

4.11.2 ウイルス対策ソフト実行時の注意事項

バッチジョブ実行中にウイルスチェックを実行した場合、ファイルのオープンエラーや、ジョブの遅延が発生するおそれがある。このため、バッチジョブ実行中にウイルスチェックを実施する場合は、次に示すファイルおよびディレクトリをウイルスチェックの対象から外すこと。

- /opt/hitachi/bjex ディレクトリ以下すべて
- /var/opt/bjex ディレクトリ以下のすべて
- 設定ファイルに指定されたディレクトリおよび配下のファイル（指定を省略した場合に仮定されるディレクトリとその配下のファイルを含む）
- ジョブ定義 XML ファイル
- ジョブ定義 XML ファイルで指定したディレクトリおよびファイル

4.12 PREST 機能を使用するジョブの定義

- PREST 機能を使用するジョブは、PREST ファイルでペアとなるジョブまたはジョブステップの障害で長時間待ち状態となることを考慮して、設定ファイルにJOBPARM_JSPECAN パラメータを指定すること、および JP1/AJS のジョブ定義で適切な打ち切り時間を設定することを推奨する。
- ジョブ間 PREST ファイルは、バッチジョブ実行システムの実行ホスト内でのグローバルなファイル名を使用する。このため、各ジョブで使用する PREST ファイル名は、ホスト内でユニークになるようにシステム管理者が管理して割り当てる必要がある。

ジョブ間 PREST ファイルを使用するジョブは、入力側と出力側のジョブを同時に実行する必要がある。このため、JP1/AJS で同時に実行されるようにジョブを定義すること。

4.13 COBOL プログラムで使用する文字コードを変更する場合の設定

ここでは、COBOL プログラムで使用する文字コードを変更する場合の設定について説明する。

4.13.1 機能概要

バッチジョブ実行システムから COBOL プログラムを呼び出す際、次の 2 点を設定することによって自動で文字コードを変更できる。

- ジョブ定義 XML ファイルの EXEC 要素に LANG="COBOL" を指定
- バッチジョブ実行システムの設定ファイルに COBOL_LANG を設定

オープンシステムでの COBOL プログラムはシフト JIS で動作させるのが一般的である。

ところが、バッチジョブ実行システムを Linux 環境で使用する場合、言語種別の設定では環境変数 LANG の値に C または ja_JP.UTF-8 を指定する必要があるため、COBOL プログラムを実行する場合、シフト JIS に変更する必要がある。このとき、ジョブ定義 XML ファイルには EXEC 要素の PGM パラメータに COBOL プログラムを直接指定できず、次のように文字コードを変更する記述が必要である。

```
<EXEC PGM="*" LANG="COBOL" >  
  <![CDATA[  
  !  
    LANG=ja_JP.SJIS COBOLプログラム  
  !  
  ]]>  
</EXEC>
```

この機能を使用した場合、EXEC 要素の PGM パラメータに COBOL プログラムを直接指定して、COBOL プログラムを任意の文字コードで実行できる。

4.13.2 ジョブ定義 XML ファイルの指定

ジョブ定義 XML ファイルで、EXEC 要素に LANG="COBOL" と設定する。これによって、このステップを設定ファイルの LANG_COBOL で指定した文字コードで実行できる。

```
<EXEC PGM="プログラム名" LANG="COBOL"/>
```

4.13.3 設定ファイル

設定ファイルの COBOL_LANG パラメータで定義する。

COBOL_LANG パラメータの定義例を次に示す。

注意事項

オペランドに文字コードを 2 個以上指定することはできない。

4.14 バッチジョブ実行システムにおける SystemV セマフォ，共有メモリの管理

4.14.1 使用する共有メモリ

バッチジョブ実行システムでは，次の機能で SystemV セマフォ，共有メモリを使用する。

- 排他制御機能
- 実行中のジョブ情報表示機能
- メッセージ制御機能

4.14.2 管理方法

セマフォ，共有メモリは，ftok 関数で生成する IPC (InterProcessCommunication：プロセス間通信) キーを使用して，バッチジョブ実行システムが生成，削除を行う。生成，削除するセマフォ，共有メモリは，IPC キーによって区別され，同一のセマフォ，共有メモリを使用するには，同一の IPC キーを使用する。

バッチジョブ実行システムが IPC キーを作成するときには，次の識別子とファイル情報を指定している。

表 4-10 IPC キー作成時の識別子とファイル情報

機能	識別子 (仮定値)	ファイル情報
排他制御機能および実行中のジョブ情報表示機能	1	設定ファイルの SPOOL_DIR パラメータで指定したスプールディレクトリにある排他制御管理ファイルの情報
メッセージ制御機能	77	設定ファイルの SPOOL_DIR パラメータで指定したスプールディレクトリの情報

ftok 関数で生成する IPC キーは，完全に一意にならないで，場合によってはシステム内で重複することがある。IPC キーが重複すると，本来別のセマフォ，共有メモリを使用しなければならないが，同じセマフォ，共有メモリを重複して使用することがある。

4.14.3 セマフォ，共有メモリ重複時のメッセージ

バッチジョブ実行システムでは使用するセマフォを基に共有メモリを割り当て，共有メモリの内容で重複しているかを判断している。共有メモリが重複していると判断した場合，次のメッセージを出力する。

(1) 排他制御機能および実行中のジョブ情報表示機能

ジョブ実行時に重複を検知した場合、KAKC1731-E, KAKC1732-E, KAKC1733-E のどれかのメッセージを出力し、ジョブは異常終了する。

また、スプールジョブ転送サービスデーモンの起動、停止時に重複を検知した場合も同様のメッセージを出力する。

(2) メッセージ制御機能

メッセージ出力時に重複を検知した場合、KAKC5006-E メッセージを出力し、メッセージを出力したジョブは異常終了する。

また、スプールジョブ転送サービスデーモンの起動、停止時に重複を検知した場合も同様のメッセージを出力する。

4.14.4 セマフォ、共有メモリ重複時の対処方法

セマフォ、共有メモリの重複が検知された場合、バッチジョブ実行システムの設定ファイルの次の指定を行う。

(1) 排他制御機能および実行中のジョブ情報表示機能用のセマフォ、共有メモリが重複した場合

LOCK_KEY_ID パラメータで、任意の値を指定する。

指定された値を識別子として、重複している IPC キーとは異なる IPC キーを生成し、セマフォ、共有メモリを生成、削除する。

指定する値は、ほかの論理ホストのバッチジョブ実行システムと重複しないようにすることが望ましい。

(2) メッセージ制御機能用のセマフォ、共有メモリが重複した場合

MSG_KEY_ID パラメータで、任意の値を指定する。

指定された値を識別子として、重複している IPC キーとは異なる IPC キーを生成し、セマフォ、共有メモリを生成、削除する。

指定する値は、ほかの論理ホストのバッチジョブ実行システムと重複しないようにすることが望ましい。

5

操作

この章では、バッチジョブ実行システムと JP1 プログラムや Prinfina MANAGER SP および PDE を連携して使用するための設定および操作方法について説明する。

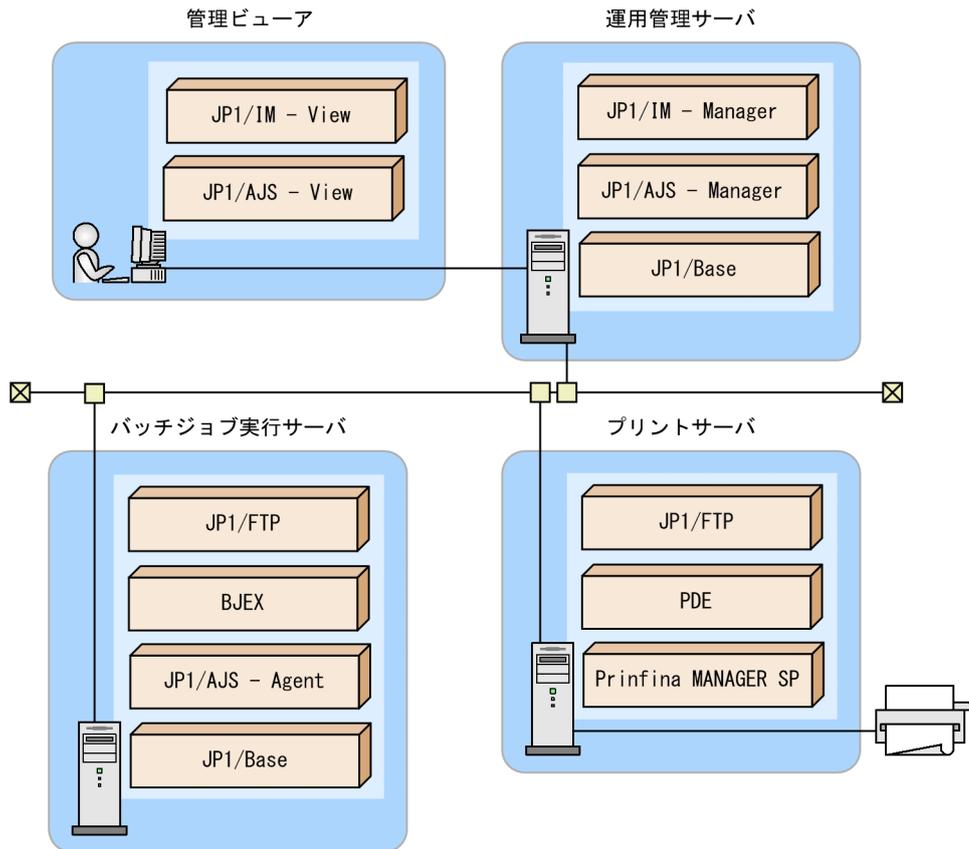
5.1 バッチジョブ実行システムの使用概要

ここでは、バッチジョブ実行システムでのバッチジョブ実行操作の概要について説明する。

5.1.1 バッチジョブ実行システムを使用するためのシステム構成

バッチジョブ実行システムを使用するためのシステム構成例を次の図に示す。

図 5-1 バッチジョブ実行システムを使用するためのシステム構成例



5.2 バッチジョブの実行

ここでは、バッチジョブの実行について説明する。JP1/AJS のジョブネットワークの定義方法などの操作については、JP1 のマニュアルを参照のこと。

5.2.1 バッチジョブの起動

バッチジョブ実行システムのバッチジョブは、JP1/AJS からシェルを経由して起動することやコマンドラインから起動することもできる。バッチジョブ業務で使用する場合は、JP1/AJS から直接またはシェルを経由して起動することを前提としている。コマンドラインからの実行は動作確認などで使用すること。

(1) JP1/AJS から起動する場合

JP1/AJS からバッチジョブ実行システムのバッチジョブを起動するには、次のどちらかのジョブを使う。

- UNIX ジョブ
実行するジョブを UNIX ジョブに定義することで、ジョブネットからバッチジョブ実行サーバにジョブを入力できる。バッチジョブ実行システムのバッチジョブは、UNIX ジョブのうちキュー属性だけに対応しており、キューレス属性には対応していない。
- サブミットジョブ
ジョブ実行コマンド (jpbjobs sub コマンド) を使ってジョブをキューに直接登録できる。

JP1/AJS から OS 標準のシェル*を経由して起動する場合、JP1/AJS からバッチジョブ実行システムを強制終了したときジョブログが途中までしか表示されないことがある。これはシェルがバッチジョブ実行システムのジョブログ出力を待たずに終了してしまうためである。ジョブログをすべて表示させるためには、シェルで SIGTERM シグナルをトラップする必要がある。以下にシェルで SIGTERM シグナルをトラップする例を示す。

注※ AIX の場合 ksh, Linux の場合 bash

シェルで SIGTERM シグナルをトラップする例

```
trap 'exit 255' TERM          . . . SIGTERMを受けると終了コード255で終了する
bjexec /xml/gyomu1.xml
```

バッチジョブの起動方法による強制終了の方法と制限事項を以下にまとめる。

表 5-1 JP1/AJS からバッチジョブを起動した場合の強制終了の方法と制限事項

項目	バッチジョブの起動		
	直接実行する場合	OS 標準のシェルを經由して起動	Advanced Shell を經由して起動
制限事項	なし		<ul style="list-style-type: none"> 環境変数 AJS_BJEX_STOP を設定しても KAKC0002-W が出力される※¹ ジョブ起動時に環境変数 SHELL に OS 標準のシェルを設定すること (JP1/AJS が環境変数 SHELL に Advanced Shell を設定することがある) ※³
ジョブ強制終了方法	JP1/AJS の強制終了	JP1/AJS の強制終了※ ²	JP1/AJS の強制終了

注※1 動作には影響ない。環境変数 AJS_BJEX_STOP の設定は必要である。

注※2 ジョブログを表示させるには trap コマンドを定義する必要がある。

注※3 UNIX ジョブの [スクリプトファイル名] に Advanced Shell のジョブ定義スクリプトファイルを指定すると、JP1/AJS が環境変数 SHELL に Advanced Shell を設定する。EXEC 要素内の PGM 属性値に "*" の指定と EXEC 要素の内容にシェルの指定をしても、シェルスクリプトとして実行できなくなる。環境変数 SHELL に OS 標準のシェルを設定する例を次に示す。次のどちらかの設定をする必要がある。

(1)Advanced Shell のスクリプトによる設定方法

Advanced Shell のジョブ定義スクリプトファイルに次の指定をすることで、環境変数 SHELL に OS 標準のシェルを設定してジョブを起動できる。

```
export SHELL=/bin/sh
bjexec ジョブ定義xmlファイル
```

(2)JP1/AJS のジョブ定義による設定方法

JP1/AJS の UNIX ジョブによるジョブの定義を次のようにすることで、環境変数 SHELL に OS 標準のシェルを設定してジョブを起動できる。

[詳細定義 - [UNIX Job]] 画面の [定義] タブ

- スクリプトファイル名：/opt/jplas/bin/adshexec
- パラメータ：ジョブ定義スクリプトファイル名、および実行時パラメータ
サブミットジョブの場合、それぞれ -sc, -arg オプションとなる。

(2) コマンドラインから起動する場合

JP1/AJS を使用せず起動することもできる。ただし、前提プログラムである JP1/AJS および JP1/BASE の組み込みが必要である。コマンドラインからバッチジョブの起動方法による強制終了の方法と制限事項を以下にまとめる。

表 5-2 コマンドラインからバッチジョブを起動した場合の強制終了の方法と制限事項

項目	バッチジョブの起動		
	直接実行する場合	OS 標準のシェルを経由して起動	Advanced Shell を経由して起動
制限事項	<ul style="list-style-type: none"> 日付世代番号指定機能が使用できない ジョブログヘッダおよび SYSOUT 管理ファイルに JP1 の情報が出力されない 標準入力を使用するプログラムを実行すると停止することがある 		
ジョブ強制終了方法	bjexec プロセスに SIGTERM を送る		

5.2.2 JP1/AJS からバッチジョブを実行する場合のジョブ定義

JP1/AJS からバッチジョブを実行するには、UNIX ジョブまたはサブミットジョブを使用する。

(1) UNIX ジョブの実行

[詳細定義－ [UNIX Job]] 画面で、バッチジョブ実行システムのバッチジョブの詳細を定義する。定義方法と内容は、UNIX ジョブに従う。バッチジョブ実行システムのバッチジョブとして指定必須項目について示す。

(a) [定義] タブ

- [スクリプトファイル名]

バッチジョブ起動コマンド名として次を指定する。

```
/opt/hitachi/bjex/bin/bjexec
```

- [パラメータ]

ジョブ定義 XML ファイルのパス名を指定する。

- [環境変数]

環境変数に次を指定する。

```
AJS_BJEX_STOP=TERM
```

AJS_BJEX_STOP=TERM を指定すると、JP1/AJS の強制終了機能使用時プロセスグループに対して、SIGKILL ではなく SIGTERM が送付される。

- [実行優先順位]

実行優先順位に 3 を指定する。

バッチジョブ実行システムのバッチジョブとしての指定例を次の図に示す。

図 5-2 [詳細定義－ [UNIX Job]] 画面の [定義] タブの指定例

ユニット名	job-sample1
コメント	
実行ホスト	
定義 転送ファイル 属性	
コマンド文	
スクリプトファイル名	/opt/hitachi/bjex/bin/bjexec
パラメーター	/home/bjex/batch/xml/sample1.xml
環境変数	AJS_BJEX_STOP=TERM
環境変数ファイル名	
ワークパス	
実行優先順位	3
標準入力ファイル名	
標準出力ファイル名	<input type="checkbox"/> 追加書き
標準エラー出力ファイル名	<input type="checkbox"/> 追加書き
終了判定	判定結果 しきい値による判定 警告しきい値 <input type="text"/> 異常しきい値 <input type="text" value="0"/>
実行時のユーザー	
OK キャンセル ヘルプ	

(b) [属性] タブ

- [実行先サービス]

実行先のサービスは [標準] を選択すること。それ以外のオプションを選択すると、強制終了機能使用時、環境変数に AJS_BJEX_STOP=TERM を指定した場合でも SIGKILL が送付される。

バッチジョブ実行システムのバッチジョブとしての指定例を次の図に示す。

図 5-3 [詳細定義－ [UNIX Job]] 画面の [属性] タブの指定例

詳細定義-[UNIX Job]

ユニット名: job-sample1

コメント:

実行エージェント:

定義 | 転送ファイル | **属性**

保留: しない する

種別: 通常 リカバリ

実行先サービス: 標準

打ち切り時間指定: しない する

打ち切り時間: [] 分

終了遅延監視: 実行所要時間 [] 分

所有者: bjex

JPI資源グループ:

実行ユーザー種別: 登録ユーザー 所有ユーザー

OK | キャンセル | ヘルプ

(2) サブミットジョブの実行

実行するジョブやキューを指定して、jqjobs sub コマンドを実行する。バッチジョブ実行システムのバッチジョブの指定必須項目を次に示す。

- -sc 実行ファイル名
バッチジョブ起動コマンド名を指定する。
- -arg "引数..."
ジョブ定義 XML ファイルのパス名を指定する。
- -env 環境変数名=値
付け加える環境変数に AJS_BJEX_STOP=TERM を指定する。

- -pr 実行優先順位
サブミットジョブの実行優先順位に 3 を指定する。

バッチジョブ実行システムのバッチジョブの設定例を次に示す。

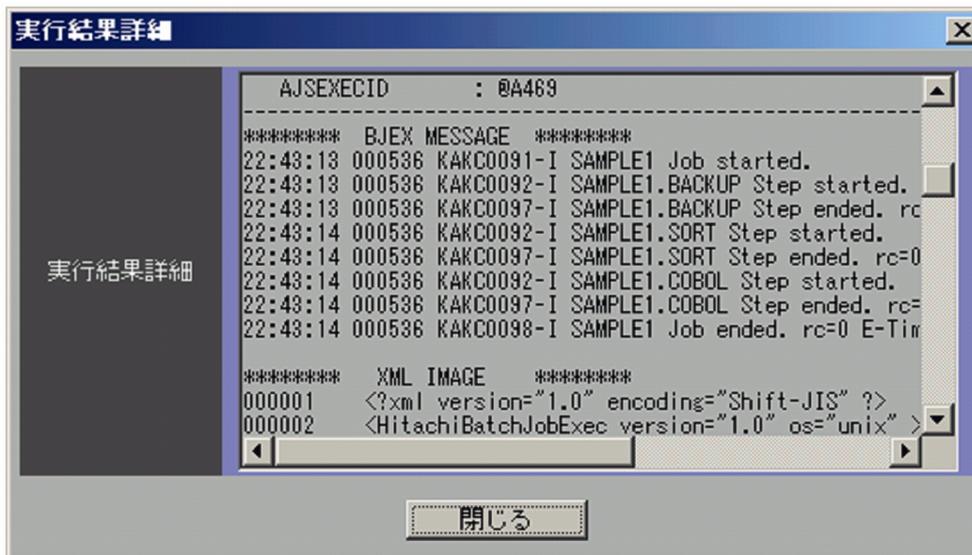
```
-sc /opt/hitachi/bjex/bin/bjexec
-arg "/home/bjex/batch/xml/sample1.xml"
-env AJS_BJEX_STOP=TERM
-pr 3
```

5.2.3 実行結果の参照

バッチジョブ実行システムのバッチジョブの実行結果は、ジョブログとして標準エラー出力ファイルに出力される。ジョブログは、JP1/AJS - View の [実行結果詳細] 画面から参照できる。ジョブログのフォーマットについては、[3.1.3(2) ジョブログ出力機能] を参照のこと。

[実行結果詳細] 画面の表示例を次の図に示す。

図 5-4 [実行結果詳細] 画面の表示例



5.2.4 バッチジョブの強制終了

バッチジョブの強制終了には次の 2 とおりの方法がある。

- JP1/AJS からジョブを起動している場合、JP1/AJS の強制終了操作を実行する。*
- bjexec プロセスに SIGTERM シグナルを送付する。

注※ AJS_BJEX_STOP 環境変数に TERM の指定が必要である。

実行中のバッチジョブを強制終了すると、ジョブステップのプロセスグループに対して SIGKILL シグナルが送信し、ジョブステップのプロセスを強制終了する。強制終了の指示を複数回受けた場合、ジョブコントローラ自身が強制終了する。

(1) JP1/AJS から起動したジョブの強制終了操作

JP1/AJS の操作によって実行中のバッチジョブおよびジョブネットを強制終了できる。

また、強制終了の指示を受けたジョブネットは、新たなジョブを起動しなくなり、ジョブネット内の実行中のジョブすべてを強制終了する。この操作は、次に示すウィンドウ、または ajskill コマンド、もしくは jpqjobcan コマンドで操作できる。

- [デイリースケジュール (階層表示)] ウィンドウ
- [JP1/AJS3 - View] ウィンドウ
- [マンスリースケジュール] ウィンドウ
- [ジョブネットモニタ] ウィンドウ

ウィンドウによる強制終了、ajskill コマンド、および jpqjobcan コマンドの詳細については、JP1 のマニュアルを参照のこと。

上記の操作で強制終了をした場合、JP1/AJS で確認できるジョブの状態は、「異常検出終了」となる。

(2) bjexec プロセスに SIGTERM シグナルを送付

bjexec プロセスに SIGTERM シグナルを送付することで実行中のバッチジョブを強制終了できる。

bjexec プロセスのプロセス ID は bjexlslock コマンドの PID の列に出力される。bjexlslock コマンドで bjexec プロセスのプロセス ID を参照し、SIGTERM シグナルを送付する。排他/共用制御機能を使用していない場合は、ps コマンドなどで調べること。

(3) 注意事項

- 次に示すジョブステップ中のプロセスは強制終了されない。このようなプロセスは、kill コマンドなどのほかの手段で終了させる必要がある。
 - setpgrp システムコールなどを実行し、別プロセスグループとなっているプロセス
 - 常駐プロセスなどから起動され、ジョブコントローラの子孫プロセスとなっていないプロセス
- 次に示す方法などでジョブコントローラ自身が強制終了させた場合、bjexlsjob コマンドは、実行中のジョブの終了時刻を出力しない。
 - ジョブコントローラに強制終了を複数回指示する。
 - ジョブコントローラに SIGKILL シグナルを送る。

5.3 Prinфина MANAGER SP および PDE との連携による帳票印刷

ここでは、Prinфина MANAGER SP および PDE との連携による帳票印刷について説明する。

5.3.1 帳票印刷

帳票印刷は、Prinфина MANAGER SP、PDE と連携して実施する。バッチジョブの実行終了後、スプールに出力された SYSOUT データファイルをスプールジョブ転送サービスによって出力クラスに従い、Prinфина MANAGER SP、PDE が動作するプリントサーバに転送することで実施する。

5.3.2 JP1/FTP を使った印刷データ転送

スプールジョブ転送サービスは、JP1/FTP を使用してバッチジョブの実行結果である印刷データ (SYSOUT データファイル) をプリントサーバに転送する。ここでは、印刷データの転送方法について説明する。

(1) バッチジョブ実行システムの設定

バッチジョブ実行システムでの設定について次に示す。

(a) 設定ファイル (bjex.conf)

出力クラスごとの転送先についての情報を設定する。

設定ファイルの設定例を次に示す。

```
<SysoutClass クラス名>
:
JP1FTPCARD  伝送カード名
FTPCNTL    スプールジョブ管理ファイル転送先ディレクトリ名
FTPDATA    SYSOUTデータファイル転送先ディレクトリ名
:
</SysoutClass>
```

設定ファイルの指定例を次に示す。

```
<SysoutClass A>
:
JP1FTPCARD  cardA
FTPCNTL    A/CNTL
FTPDATA    A/DATA
:
</SysoutClass>
```

(b) ジョブ定義 XML ファイル

DD 要素の SYSOUT 属性で、出力クラスを設定する。ジョブ定義 XML ファイルの設定例を次に示す。

```
：  
<DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" />  
：
```

(2) JP1/FTP の設定

転送ホスト、ログインユーザ名などの FTP 転送に関するパラメータは、設定ファイルの出力クラスで指定する伝送カード名で定義する。

伝送カードの詳細については、JP1 のマニュアルを参照のこと。

バッチジョブ実行システムで使用する伝送カードに設定する内容について次の表に示す。

表 5-3 バッチジョブ実行システムで使用する伝送カード設定項目

設定パラメータ	ユーザ指定	設定時の注意事項
カード名	○	クラスに対応するカード名を使用する。
接続先ホスト名	○	—
接続先ポート番号	○	—
ログインユーザ名	○	—
ログインパスワード	○	—
送受信種別	○※1	SEND を指定する。 転送時に設定を上書きして使用する。
伝送モード	○※1	BINARY を指定する。 転送時に設定を上書きして使用する。
圧縮モード	△	—
サイズ確認	△	CHECK を指定することを推奨する。
単/複伝送	△※1	転送時に設定を上書きして使用する。
ローカルファイル名	○※1	カードにはダミーで登録する (必須パラメータのため)。 転送時に上書きして使用する。
リモートファイル名	○※1	カードにはダミーで登録する (必須パラメータのため)。 転送時に上書きして使用する。
正常時に起動するプログラム名	×	—
異常時に起動するプログラム名	×	—
FTP コマンド	×	—

設定パラメータ	ユーザ指定	設定時の注意事項
コメント	△	—

(凡例)

- ：バッチジョブ実行システムを使用するための必須パラメータである。
- △：任意に指定できるパラメータである。
- ×：指定できないパラメータ※²である。
- ：注意事項なし。

注※1

スプールジョブ転送サービスが指定を上書きするため、カードの登録時はダミーでよい。

注※2

指定されてもバッチジョブ実行システムではエラーとしない。なお、この項目を指定した場合は、PDEサーバ上でファイルアクセスの競合が発生し、PDEの動作に悪影響を与えるおそれがあるので注意が必要である。

JP1/FTP の設定例を次に示す。

```
ftsregc -N -TC cardA -TH prtserve -TP 20125 -TU prtuser -TW prtpasswd -TY BINARY
-TL dummy -TR dummy -TT SEND
**伝送カード情報**
伝送カード名：cardA
相手ホスト名：prtserve
ポート番号：20125
ログインユーザ名：prtuser
パスワード：*****
送受信種別：送信（上書き）
伝送モード：BINARY
圧縮モード：非圧縮
サイズチェック：なし
単／複伝送：自動切り替え
ローカルファイル名：dummy
リモートファイル名：dummy
終了プロシジャ名（正常）：
終了プロシジャ名（異常）：
コマンド：コメント：
```

(3) Prinfinia MANAGER SP および PDE の設定

Prinfinia MANAGER SP および PDE の設定について次に示す。

(a) バッチジョブ実行システムと Prinfinia MANAGER SP および PDE との連携

バッチジョブ実行システムから出力クラス情報に従った転送先にスプールジョブ管理ファイルと SYSOUT データファイルを転送する。

Prinfina MANAGER SP および PDE は、スプールジョブ管理ファイルの受信を監視することで、印刷ジョブの受信を完了したと判断し、印刷処理を開始する。また、Prinfina MANAGER SP および PDE の監視ディレクトリの設定で、バッチジョブ実行システムの設定ファイルの FTPCNTL で指定したスプールジョブ管理ファイル転送先ディレクトリを設定しておく必要がある。

(b) 転送のための準備

バッチジョブ実行システムから SYSOUT データファイルを転送するために、Prinfina MANAGER SP および PDE では次に示す準備が必要である。

詳細については、マニュアル「PRINT DATA EXCHANGE for Open」、[PRINT DATA EXCHANGE - Form Designer]、または Prinfina MANAGER SP および PDE のドキュメントを参照のこと。

• ディレクトリの作成

バッチジョブ実行システムの設定ファイル (bjex.conf) の出力クラスごとに指定する、スプールジョブ管理ファイルと SYSOUT データファイルの転送先ディレクトリをそれぞれ作成する。

• ディレクトリアクセス権の設定

転送で使用する JP1/FTP には、転送先のユーザ (JP1/FTP で定義するユーザ) のホームディレクトリ以下にだけ転送可能とするオプションがある。このオプション指定時には、転送先ディレクトリについて Prinfina MANAGER SP、または PDE の監視サービスから参照可能になっている必要がある。

• プリンタ制御情報の準備

PDE-Form Designer を使用してプリンタ制御情報を作成する。なお、PRINT DATA EXCHANGE - Form Designer は Windows で動作する。

PC での制御イメージ作成の作業を次に示す。

- 書式を作成および修正して書式オーバーレイイメージを生成
- FCB イメージを作成および修正
- デフォルト機能キャラクタイメージを作成および修正
- プリントサーバへ転送

(4) 注意事項

スプールジョブ転送サービスは、FTP 転送処理中にエラーが発生した場合、エラー要因が取り除かれるまで、設定ファイルのパラメータ JP1FTP_RETRY_INTERVAL および JP1FTP_RETRY_TIMES で指定した待ち時間の間隔と回数でリトライを繰り返す。このとき、スプールジョブ転送サービスは次のエラーメッセージをリトライするたびに出力する。

メッセージ	出力先
KAKC3004-E	実行ログ, JP1/IM - View
KAKC3123-E	実行ログ

上記のメッセージが出力された場合は、メッセージの出力情報からエラー要因を取り除くこと。ただし、リソース不足や論理矛盾など回復困難なエラーの場合は、スプールジョブ転送サービスを再起動したあとで、次のどちらかを実施すること。

- バッチジョブを再実行する。
- bjextrjob コマンドで明示的に印刷データを再転送する。

詳細は「6.6 JP1/FTP を使った印刷データ転送での障害対応」を参照すること。

5.3.3 SYSOUT 属性の PDE および Prinfina MANAGER SP での扱い

バッチジョブ実行システムの設定する SYSOUT 属性について、PDE,Prinfina MANAGER SP での扱いについて示す。なお、詳細については、マニュアル「PRINT DATA EXCHANGE for Open」、または Prinfina MANAGER SP および PDE のドキュメントを参照のこと。

書式、FCB などの主に帳票出力プログラムを記載する上で、プログラマが留意すべき属性について次の表に示す。

表 5-4 プログラマが留意すべき SYSOUT 属性の PDE,Prinfina MANAGER SP での扱い

分類	項目	ジョブ定義 XML ファイル/SysoutClass パラメータで指定できる項目			PDE での扱い	Prinfina MANAGER SP での扱い
		属性名/パラメータ名	ジョブ定義 XML ファイル	SysoutClass パラメータ		
D D	文字配列 テーブル名 または文字 セットイ メージ名	CHARS	○	○	表 5-5 の項番 1 へ	
	FCB 名称	FCB	○	○	表 5-5 の項番 2 へ	
	書式番号	FORMS	○	○	表 5-5 の項番 3 へ	
	漢字モード	MODE	○	○	表 5-5 の項番 4 へ	
	ANSI 制御 コード	RECFM	○	×	表 5-5 の項番 5 へ	
	文字配列 テーブル順 序番号属性	OPTCDJ	○	×	表 5-5 の項番 6 へ	
	複写回数	COPIES	○	○	表 5-5 の項番 7 へ	
	グループ複 写回数	COPYG	○	○	表 5-5 の項番 8 へ	
	オーバレ イ名	OVERLAY	○	○	表 5-5 の項番 9 へ	

分類	項目	ジョブ定義 XML ファイル/SysoutClass パラメータで指定できる項目			PDE での扱い	Prinfina MANAGER SP での扱い
		属性名/パラメータ名	ジョブ定義 XML ファイル	SysoutClass パラメータ		
D D	適用枚数	OVERC	○	○	表 5-5 の項番 10 へ	
	デフォルト機能キャラクタータイメージ名	FCDFLT	○	○	表 5-5 の項番 11 へ	
	出力先識別子	DEST	○	○	表 5-5 の項番 12 へ	
	レベル 1 外字名	GAIJ1	○	○	表 5-5 の項番 13 へ	
	エンコード種別	ENCODE	○	○	表 5-5 の項番 14 へ	
	PDE ファイル形式 2	FTYPE	○	○	表 5-5 の項番 15 へ	

(凡例)

○：指定できる。

×：指定できない。

表 5-5 プログラムが留意すべき SYSOUT 属性の PDE,Prinfina MANAGER SP での扱い (扱いについての詳細)

項番	PDE での扱い	Prinfina MANAGER SP での扱い
1	CHARS イメージ名 (書体情報の適用)	—
2	FCB イメージ名 (FCB イメージの適用など)	FCB 名の属性値として使用
3	用紙形式コード (用紙名称の表示など)	用紙コードの属性値として使用
4	プリンタモード	—
5	プリンタ制御文字	—
6	順序番号の有無	—
7	部数複写	部数(部複写回数)の属性値として使用
8	ページ複写	部数(ページ複写回数)の属性値として使用
9	書式オーバーレイイメージ名 (書式オーバーレイイメージの適用)	—
10	書式適用枚数 (適用する書式オーバーレイイメージの枚数)	—
11	デフォルト機能キャラクタータイメージ名 (デフォルト機能キャラクタータイメージの適用)	—
12	出力先コード (対象ジョブ条件に使用)	—

項番	PDE での扱い	Prinfina MANAGER SP での扱い
13	外字フォント名	—
14	エンコード種別	—
15	PDE ファイル形式 2	—

(凡例)

—：使用しない。

書式、FCB などの主に帳票出力プログラムを記載する上で、出力クラスによる出力先の振り分けなどの運用管理者が留意すべき属性について次の表に示す。

表 5-6 運用管理者が留意すべき SYSOUT 属性の PDE,Prinfina MANAGER SP での扱い

分類	項目	ジョブ定義 XML ファイル/SysoutClass パラメータで指定できる項目			ジョブ定義 XML ファイル/SysoutClass パラメータで指定できない項目	PDE での扱い	Prinfina MANAGER SP での扱い
		属性名/パラメータ名	ジョブ定義 XML ファイル	SysoutClass パラメータ			
ジョブ	ジョブ識別子 (10 進数)	×	×	×	ジョブコントローラで割り当てるジョブ識別子	表 5-7 の項番 1 へ	
	ジョブ名	NAME	○	×	×	表 5-7 の項番 2 へ	
	ユーザ名	×	×	×	環境変数 JP1_USERNAME または OS ユーザ名	表 5-7 の項番 3 へ	
	仕分け番号	ROOM	○	×	×	表 5-7 の項番 4 へ	
	ホスト名	×	×	×	次に示す優先順位のホスト名となる (優先順位は 1 が最も高い) 1. -h オプションで指定 2. 設定ファイルの JP1_HOSTNAME パラメータで指定 3. gethostname() で取得	表 5-7 の項番 5 へ	
DD	DD 名称	NAME	○	×	×	表 5-7 の項番 6 へ	
	出力クラス名	SYSOUT	○	○	×	表 5-7 の項番 7 へ	
	注釈情報	CMT	○	×	×	表 5-7 の項番 8 へ	
ジョブステップ	ジョブステップ名	NAME	○	×	×	表 5-7 の項番 9 へ	
	ステップ番号	×	×	×	ステップ通し番号*	表 5-7 の項番 10 へ	

(凡例)

○：指定できる。

×：指定できない。

注※ ジョブ定義 XML ファイルで定義したジョブステップの番号ではなく、ジョブ実行時にバッチジョブ実行システムで割り当てる番号。

表 5-7 運用管理者が留意すべき SYSOUT 属性の PDE, Prinfina MANAGER SP での扱い (扱いについての詳細)

項番	PDE での扱い	Prinfina MANAGER SP での扱い
1	ジョブ識別子 (出力ファイル名/ドキュメント名に使用)	ジョブ識別子の属性値として使用
2	ジョブ名 (対象ジョブ条件, 出力ファイル名/ドキュメント名などに使用)	ジョブ名として使用
3	ユーザ名 (対象ジョブ条件, 出力ファイル名/ドキュメント名などに使用)	ユーザ名の属性値として使用
4	—	仕分コードの属性値として使用
5	ホスト名 (対象ジョブ条件, 出力ファイル名/ドキュメント名などに使用)	ホスト ID の属性値として使用
6	—	DD 名の属性値として使用
7	出力クラス名 (対象ジョブ条件, 出力ファイル名/ドキュメント名などに使用)	出力クラス名の属性値として使用
8	ジョブコメント (出力ファイル名/ドキュメント名, FAX 送信先などに使用)	コメントの属性値として使用
9	ジョブステップ名 (出力ファイル名/ドキュメント名に使用)	—
10	ステップ通し番号 (出力ファイル名/ドキュメント名に使用)	—

(凡例)

—：使用しない。

5.3.4 印刷データの転送契機

出力非保留属性の SYSOUT データファイルは、ジョブ実行終了時にプリントサーバへ転送する。

出力保留属性の SYSOUT データファイルはジョブ終了時にはプリントサーバへ転送しない。出力保留属性の SYSOUT データファイルをプリントサーバに転送するには、bjexchjob コマンドによってスプールジョブ単位で印刷指示をする。

コマンドによる印刷データの転送の設定例を次に示す。

```
:  
<DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" HOLD="YES"/>  
:
```

bjexlsjob コマンドでスプールジョブ一覧を表示する。

```
$ bjexlsjob  
JOBID  S  JOBNAME  
000009 H  SAMPLE1  
000010 H  SAMPLE2_JCL  
000011 H  SAMPLE3_JCL  
000012 H  SAMPLE1  
000013 H  SAMPLE2_JCL  
000014 H  SAMPLE3_JCL  
000015 H  SAMPLE1  
000016 H  SAMPLE2_JCL  
000017 H  SAMPLE3_JCL  
000018 H  SAMPLE1  
000019 H  SAMPLE1
```

スプールジョブ一覧から選択したジョブ識別子 15 のスプールジョブの保留状態を解除し、プリントサーバへ転送する。

```
bjexchjob -R 15
```

5.4 JP1/IM - Manager との連携による応答要求メッセージへの応答入力

ここでは、JP1/IM - Manager との連携による応答要求メッセージへの応答入力について説明する。

5.4.1 応答の入力

メッセージ制御機能を使用して応答要求メッセージを出力した場合、そのメッセージは応答待ちイベント（イベント ID：00005C21）として JP1/IM - View に表示される。

応答を入力する場合は、JP1/IM - View の応答待ちイベントに対して応答を入力する。プログラムまたはコマンドには、JP1/IM - View から入力された応答が入力される。

5.4.2 JP1/IM - Manager との連携

バッチジョブ実行システムは、設定ファイルの JP1IM_MANAGER_HOST パラメータに指定したホストで稼働する JP1/IM - Manager と連携する。連携する JP1/IM - Manager では、応答待ちイベント管理機能を有効にしておく必要がある。

応答を入力する JP1/IM - View は、バッチジョブ実行システムと連携する JP1/IM - Manager に接続している必要がある。連携していない JP1/IM - Manager に接続した JP1/IM - View では、応答の入力ができない。応答を入力する JP1/IM - View では、応答待ちイベントの応答・操作を有効にしておく必要がある。

JP1/IM - Manager および JP1/IM - View の設定については、JP1 のマニュアルを参照のこと。

5.4.3 応答要求メッセージの同時出力数

バッチジョブ実行システムは、応答要求メッセージを応答待ちイベントとして、連携する JP1/IM - Manager のホストに送付する。JP1/IM - Manager に滞留できる応答待ちイベントは 2,000 件であるため、同時期に送付できる応答待ちイベントの数を MAX_RESPONSE_REQUEST パラメータで制限する。応答待ちイベントを出力するホストでは、次に示す内容となるように MAX_RESPONSE_REQUEST パラメータを指定する。

全ホスト※の MAX_RESPONSE_REQUEST パラメータ値の総和 ≤ 2,000

注※ JP1IM_MANAGER_HOST パラメータが同一のホストを示す。

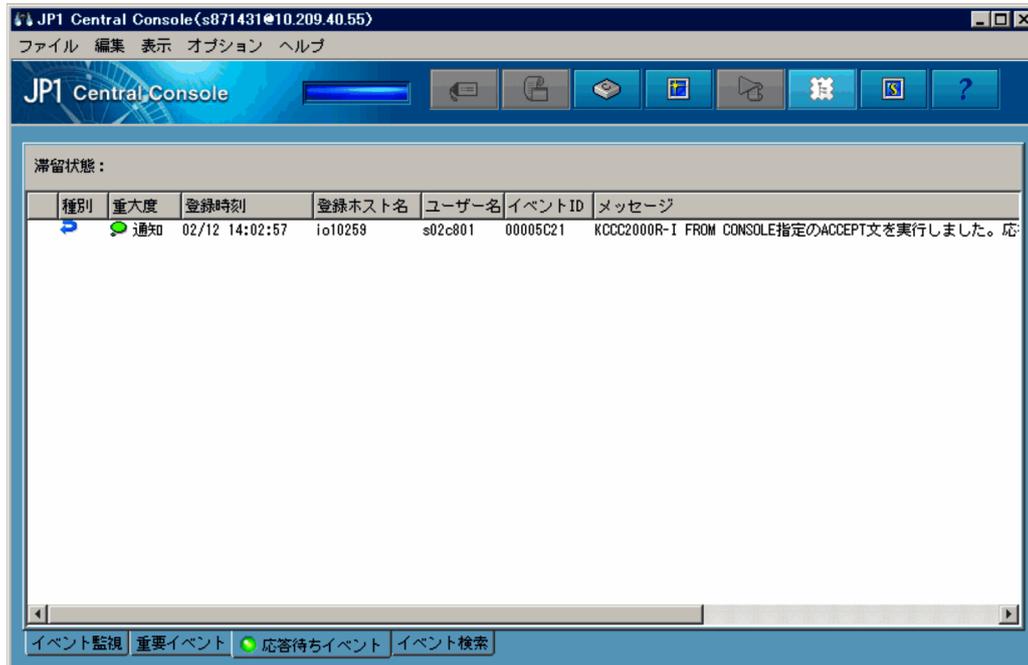
5.4.4 応答入力

JP1/IM - View から応答を入力する一例として、[応答待ちイベント] 画面を使用する例を示す。

JP1/IM - View の [応答待ちイベント] 画面には、応答待ちイベントが表示される。

COBOL アプリケーションでの [応答待ちイベント] 画面の例を次の図に示す。

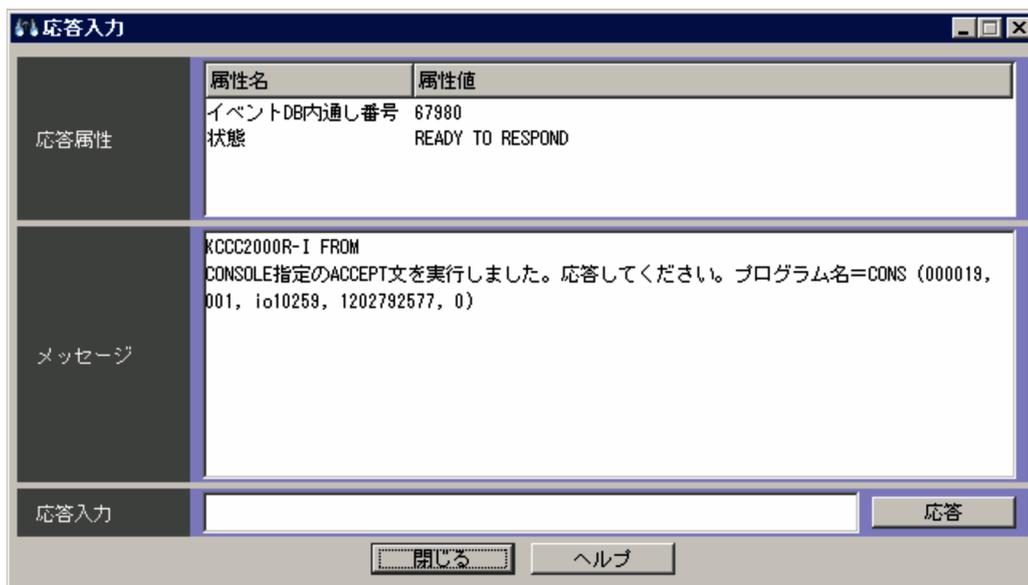
図 5-5 [応答待ちイベント] 画面の例



[応答待ちイベント] 画面に表示された応答待ちイベントから [応答入力] 画面を表示し、[応答入力] 欄に応答を入力して [応答] ボタンをクリックする。

[応答入力] 画面を次の図に示す。

図 5-6 [応答入力] 画面



[応答] ボタンをクリックしたあとに表示される画面で [はい] ボタンをクリックすると、COBOL アプリケーションに応答が入力される。

6

障害対策

この章では、バッチジョブ実行システムに必要な障害対策について説明する。

6.1 バックアップとリカバリー

バッチジョブ実行システムのバックアップとリカバリーについて説明する。

6.1.1 バックアップ

環境をバックアップする場合は、次に示す環境の設定情報をバックアップする。

- バッチジョブ実行システムの定義ファイル (bjex.conf) のバックアップ
- ジョブ定義 XML ファイルのバックアップ
- 恒久ファイルのバックアップ

「8. 設定ファイル」の「PERMANENT_FILE_DIR (恒久ファイル用ディレクトリのパス名定義)」を参考にする。

恒久ファイルの中で世代ファイルについては、個々のファイルをバックアップしないで、「[世代ディレクトリ名] 世代データ群名」が同じファイルを同時にバックアップすること。

- カタログドプロシジャ格納ディレクトリのバックアップ

「8. 設定ファイル」の「PROCEDURE_DIR (プロシジャライブラリのパス名定義)」を参考にする。

- スプールディレクトリのバックアップ (任意)

6.1.2 リカバリー

ディスク障害などで環境が壊れた場合、JP1 のインストール、セットアップを実施してから、バッチジョブ実行システムをリカバリーする。バッチジョブ実行システムのリカバリーでは、環境をバックアップしたファイルを、元のディレクトリ構成に復元することで完了する。

6.2 トラブルシューティング

6.2.1 対処の方法

バッチジョブ実行システムでトラブルが起きた場合は次に示す手順で対処する。

(1) 現象の確認

トラブルが発生したときの現象を確認する。メッセージが出力されている場合は、メッセージの内容を確認する。各メッセージの要因と対処方法については、「10. メッセージと終了コード」を参照のこと。また、バッチジョブ実行システムが出力するログ情報については、「6.2.2 ログ情報の種類」を参照のこと。

(2) 資料の採取

トラブルの要因を調べるために資料の採取が必要である。「6.2.3 トラブル発生時に採取が必要な資料」を参照して、必要な資料を採取すること。

(3) 問題の調査

採取した資料を基に問題の要因を調査し、問題が発生している部分と問題の範囲を切り分ける。

6.2.2 ログ情報の種類

バッチジョブ実行システムを運用しているときに出力されるログ情報について、次の表に示す。

表 6-1 バッチジョブ実行システム運用時に出力されるログ情報

ログ情報	ログ情報の格納先
ジョブログ (バッチジョブのログ)	スプールディレクトリの配下
実行ログ (バッチジョブ実行システムの実行ログ)	設定ファイルの LOG_DIR パラメータで指定したディレクトリ
トレースログ (バッチジョブ実行システムの内部トレース)	設定ファイルの TRACE_DIR パラメータで指定したディレクトリ
起動ログ (bjexpd の起動ログ)	物理ホストの bjexpd の場合 • /opt/hitachi/bjex/system/bjexpd.log 論理ホストの bjexpd の場合 • /opt/hitachi/bjex/system/bjexpd_論理ホスト名.log

(1) ジョブログ

ジョブログとは、ジョブの実行結果を通知する利用者向けのログ情報のことである。このログ情報は、JP1/AJS - View などによって確認ができる。

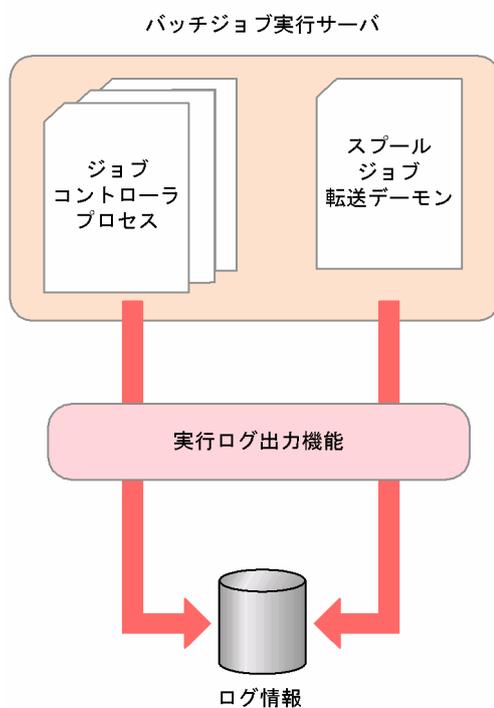
(2) 実行ログ

実行ログとは、バッチジョブの実行履歴を統括的に参照するためのシステム管理者向けのログ情報である。このログ情報は、設定ファイルの LOG_DIR パラメータで指定したディレクトリ下の BjexLog.log に出力される。

(a) 機能

実行ログには、各ジョブコントローラプロセスで実行しているジョブの情報のほかに、スプールジョブ転送デーモンのログを集めることができる。

図 6-1 ログ情報の出力の流れ



実行ログは次のように作成する。

- 実行ログに出力するメッセージを集め、CSV 形式で出力する。
出力するメッセージについては、「[10.2 メッセージの出力先一覧](#)」を参照のこと。
- ローテーションを行い、バックアップを作成する。
 - 設定ファイルの LOG_FILE_SIZE パラメータに指定されたファイルサイズを超える直前に、実行ログのファイル名を変更してバックアップを作成し、新たに実行ログを作成して出力を継続する。

- バックアップのファイル名は、BjexLog_N.log (Nは整数) となる。Nには新しいバックアップから昇順に1から番号を割り当てる。
- 設定ファイルの LOG_FILE_CNT パラメータに指定された面数のバックアップを作成し、面数を超えた場合は古いバックアップを削除する。

(b) 形式

メッセージは次のよう出力される。

```
seqnum=1, date=2007-02-26T02:52:04.670Z, pgmid=Exec_c, jobid=86, pid=104276,
msgid=KAKC0004-l, msg="Jobid=000086, JP1NBQQueueName=¥¥io10288¥io10288, JP1JobID=956"

seqnum=2, date=2007-02-26T02:52:04.674Z, pgmid=Exec_c, jobid=86, pid=104276,
msgid=KAKC0091-l, msg="TESTJOB Job started."
seqnum=3, date=2007-02-26T02:52:04.760Z, pgmid=Exec_c, jobid=86, pid=104276,
msgid=KAKC0092-l, msg="TESTJOB.PROCSTEP.PSTEP00 Step started."

seqnum=4, date=2007-02-26T02:52:04.974Z, pgmid=Exec_c, jobid=86, pid=104276,
msgid=KAKC0097-l, msg="TESTJOB.PROCSTEP.PSTEP00 Step ended. rc=0 E-Time=0.214s
C-Time=0.000s"

seqnum=5, date=2007-02-26T02:52:05.160Z, pgmid=Exec_c, jobid=86, pid=104276,
msgid=KAKC0092-l, msg="TESTJOB.PROCSTEP.PSTEP01 Step started."

seqnum=6, date=2007-02-26T02:52:05.381Z, pgmid=Exec_c, jobid=86, pid=104276,
msgid=KAKC0097-l, msg="TESTJOB.PROCSTEP.PSTEP01 Step ended. rc=0 E-Time=0.220s
C-Time=0.010s"

seqnum=7, date=2007-02-26T02:52:05.561Z, pgmid=Exec_c, jobid=86, pid=104276,
msgid=KAKC0092-l, msg="TESTJOB.step00 Step started."

seqnum=8, date=2007-02-26T02:52:05.776Z, pgmid=Exec_c, jobid=86, pid=104276,
msgid=KAKC0097-l, msg="TESTJOB.step00 Step ended. rc=0 E-Time=0.214s C-Time=0.000s"

seqnum=9, date=2007-02-26T02:52:05.960Z, pgmid=Exec_c, jobid=86, pid=104276,
msgid=KAKC0092-l, msg="TESTJOB.step01 Step started."

seqnum=10, date=2007-02-26T02:52:06.189Z, pgmid=Exec_c, jobid=86, pid=104276,
msgid=KAKC0097-l, msg="TESTJOB.step01 Step ended. rc=0 E-Time=0.230s C-Time=0.010s"

seqnum=11, date=2007-02-26T02:52:06.360Z, pgmid=Exec_c, jobid=86, pid=104276,
msgid=KAKC0098-l, msg="TESTJOB Job ended. rc=0 E-Time=0.878s C-Time=0.020s MaxRSS=2548KB"
```

各データの意味を次に示す。

seqnum=プロセスごとの通し番号

date=出力した日時

pgmid=メッセージ出力元

メッセージ出力元には次のどれかが出力される。

Exec_c : ジョブコントローラ

Pd_d : スプールジョブ転送デーモン

Msg_lib : メッセージ制御機能を使用した COBOL アプリケーション, bjexm_message コマンド

Adapter : メッセージ制御機能の応答受信コマンド

jobid=ジョブ識別子

pid=プロセス ID

msgid=メッセージ番号とタイプコード

msg=メッセージ

(c) 注意事項

実行ログのファイルはローテーション時に新しく作成するため、ファイルの所有者はローテーション時のユーザとなる。

(3) トレースログと起動ログ

トレースログと起動ログは、トラブルが発生した場合にトラブル発生の経緯を調査、および各処理の処理時間を測定したりするために採取するログ情報である。

• トレース※

設定ファイルの TRACE_DIR パラメータで指定したディレクトリ下の BjexTrace_N.log (N は整数) へ出力される。ファイル面数は TRACE_FILE_CNT パラメータ、ファイルサイズは TRACE_FILE_SIZE パラメータで指定する。

• 起動ログ

bjexpd の起動時のログで固定ファイルへ出力される。

注※

- トレースログファイルは、TRACE_FILE_CNT パラメータで指定した面数のファイルを順番にラップアラウンドして使用する。デフォルトは、ファイルサイズ 2MB、ファイル面数 4 面である。
- ファイルサイズを小さくする場合、およびファイル面数を少なくする場合は、TRACE_DIR パラメータで指定した出力先にあるトレースログファイルとディレクトリ mmap を削除しなければならない。削除していない場合は、変更前の設定値が有効になる。
- トレースログファイルやディレクトリ mmap を削除する場合は、バッチジョブ実行システムのプロセスをすべて終了させておくこと。
- 通常運用時は、トレース用の各パラメータはデフォルトの設定とすることを推奨する。

6.2.3 トラブル発生時に採取が必要な資料

トラブルが発生したときに採取が必要な資料を次に示す。

なお、バッチジョブ実行システムでは、採取が必要な資料をバッチジョブ実行システムごと一括採取するための「資料採取ツール」がある。資料採取ツールは、シェルスクリプト (bjexcollect) で提供している。

る。資料採取ツールの詳細については、「付録 C bjexcollect コマンド（障害調査のための資料の一括収集）」を参照のこと。

(1) OS のログ情報

採取が必要な OS のログ情報を次の表に示す。これらの情報は資料採取ツールで採取できる。

表 6-2 採取が必要な OS のログ情報【AIX の場合】

情報の種類		採取資料
種類	内容	
基礎情報	日付	[date] の実行結果
	システム名称	[uname -x] の実行結果
	機種名	[uname -M] の実行結果
	CPU 数	[bindprocessor -q] の実行結果
	物理メモリ量	[bootinfo -r] の実行結果
	ランレベル	[who -r] の実行結果
	OS レベル	[oslevel] の実行結果
	保守レベル	[oslevel -rq] の実行結果
	システム定義	[genkex] の実行結果
	パッケージの一覧	[lspp -al] の実行結果
	インストール済みの日立製品	[/etc/.hitachi/pplistd/pplistd] ファイル
	システムパラメータ	[pstat -T] の実行結果
	カーネルのモード	[bootinfo -K] の実行結果
	環境変数	[/etc/environment] ファイル
	共有メモリ情報	[/usr/bin/ipcs -a] の実行結果
ホスト名	[/usr/bin/hostname] の実行結果	
プロセス情報	プロセス一覧	[ps -efl] の実行結果
	制限値	[ulimit -a] の実行結果
	マシンの環境変数	[set] の実行結果
メモリ使用情報	システム・ページング・スペース情報	[lsps -a] の実行結果
ネットワーク使用状況	ネットワークステータスの表示	<ul style="list-style-type: none"> • [netstat -an] の実行結果 • [netstat -in] の実行結果
	ネットワークインタフェースのパラメータの表示	[ifconfig -a] の実行結果

情報の種類		採取資料
種類	内容	
エラーログ	エラーログ情報	[errpt -a] の実行結果
syslog	syslog ファイル (/var/adm/messages など)	ツールのデフォルトの定義ファイルでは採取しない。ツールの FILES に対象となる syslog ファイルを指定することで採取できる。

表 6-3 採取が必要な OS のログ情報【Linux の場合】

情報の種類		採取資料
種類	内容	
基礎情報	日付	[date] の実行結果
	システム名称	[uname -a] の実行結果
	機種名	—
	CPU 数	[/proc/cpuinfo] ファイル
	物理メモリ量	[/proc/meminfo] ファイル
	ランレベル	[who -r] の実行結果
	OS レベル	[/etc/redhat-release] ファイル
	保守レベル	[/etc/redhat-release] ファイル
	システム定義	[/sbin/lsmmod] の実行結果
	パッケージの一覧	[rpm -qa] の実行結果
	インストール済みの日立製品	[/etc/.hitachi/pplistd/pplistd] ファイル
	システムパラメータ	[/sbin/sysctl -a] の実行結果
	カーネルのモード	—
	環境変数	[/etc/environment] ファイル
	プロセス情報	共有メモリ情報
プロセス一覧		[ps -efl] の実行結果
制限値		[ulimit -a] の実行結果
メモリ使用情報	マシンの環境変数	[set] の実行結果
メモリ使用情報	システム・ページング・スペース情報	[/proc/swaps] ファイル
ネットワーク使用状況	ネットワークステータスの表示	<ul style="list-style-type: none"> • [netstat -an] の実行結果 • [netstat -in] の実行結果
	ネットワークインタフェースのパラメータの表示	• [/sbin/ifconfig -a] の実行結果

情報の種類		採取資料
種類	内容	
ネットワーク使用状況	ネットワークインタフェースのパラメータの表示	• 「/sbin/ip addr show」の実行結果
エラーログ	エラーログ情報	—
syslog	syslog ファイル (Linux の場合は/var/log/message*など)	ツールのデフォルトの定義ファイルでは採取しない。ツールの FILES に対象となる syslog ファイルを指定することで採取できる。

(凡例)

— : Linux では採取されない。

(2) バッチジョブ実行システムの情報

採取が必要なバッチジョブ実行システムの情報を次の表に示す。これらの情報は資料採取ツールで採取できる。また、ネットワーク接続でのトラブルの場合、接続先マシン上のファイルの採取も必要である。

表 6-4 採取が必要なバッチジョブ実行システムの情報

情報の種類	採取資料
バッチジョブ実行システムのファイル情報	[/opt/hitachi/bjex] 下のディレクトリとファイル一覧
バッチジョブ実行システムの稼働に関する情報 (ログ以外)	<ul style="list-style-type: none"> • [/opt/hitachi/bjex/system] 下のファイルすべて PID ファイルなど • [/opt/hitachi/bjex/conf] 下のファイルすべて 設定ファイルなど
実行ログ	ツールの定義ファイルの LOG_DIR パラメータで指定されたディレクトリ下のファイルすべて
トレースログ	ツールの定義ファイルの TRACE_DIR パラメータで指定されたディレクトリ下のファイルすべて
起動ログ	[/opt/hitachi/bjex/system/bjexpd.log] ファイル
スプールを管理する情報	<ul style="list-style-type: none"> • ツールの定義ファイルの SPOOL_DIR パラメータで指定されたディレクトリ下のスプール管理ファイル • ツールの定義ファイルの SPOOL_DIR パラメータで指定されたディレクトリ下のディレクトリとファイル一覧 • POSIX メッセージキューの状態
JP1/FTP に関する情報※1	次のコマンドの実行結果 /opt/jp1_fts/bin/ftshistory /opt/jp1_fts/bin/ftsutil -L /opt/jp1_fts/bin/ftsregcv -v
core ファイル※2	各プロセスの異常終了時にシステムによって次に出力されるファイル <ul style="list-style-type: none"> • bjexpd [/opt/hitachi/bjex/system] 下, またはスプールディレクトリ下 (転送ディレクトリ下を含む)。 ファイル格納先は処理状態によって異なる。

情報の種類	採取資料
core ファイル※2	<ul style="list-style-type: none"> • bjexec bjexec 起動時のカレントディレクトリ※3

注※1 SYSOUT データのプリントサーバへの転送時の障害には、ツールで採取する情報のほかに、JP1/FTP のトラブルシューティング情報も必要である。JP1/FTP のトラブルシューティング情報については、JP1 のマニュアルを参照のこと。

注※2 Linux の標準の設定では core ファイルは出力されない。トラブル要因の初期調査が迅速に行えるように適切な CORE 出力設定をすること。

注※3 bjexec 起動時のカレントディレクトリは、デフォルトではジョブ実行ユーザが OS にログインしたときのホームディレクトリとなる。ただし、core ファイルの出力先は、OS の設定によって異なるディレクトリに出力するように設定されている場合があるため、ジョブを実行する前に core ファイルの出力先の設定を確認しておくこと。

また、bjexec の core ファイルは、ツールのデフォルトの定義ファイル指定では採取しないため、ツールの定義ファイルで CORE キーワードに対象ファイルを指定して採取すること。

(3) バッチジョブ実行システムのプロセス一覧

ps コマンドでプロセスの動作状態を確認する。バッチジョブ実行システムのプロセス名は、ジョブコントローラプロセスは bjexec、スプールジョブ転送デーモンプロセスは bjexpd である。

(4) オペレーション内容

トラブル発生時のオペレーションとジョブ定義 XML ファイルが必要である。

- バッチ定義 XML ファイル
- オペレーション内容の詳細
- トラブル発生時刻
- マシン構成
各 OS のバージョン、ホスト名、JP1/AJS - Manager, JP1/AJS - Agent, PDE, および JP1/FTP の構成など。
- 再現性の有無
- その他の情報
コマンド実行時にトラブルが発生した場合は、コマンドに指定した引数。

6.2.4 資料の採取方法

資料採取ツールの `bjexcollect` コマンドの定義ファイルに、収集対象の情報を指定して資料を採取する。`bjexcollect` コマンドの使用方法については、「付録 C `bjexcollect` コマンド (障害調査のための資料の一括収集)」を参照のこと。

【Linux 限定】

トラブル要因の初期調査が迅速に行えるように適切な `core` 出力設定をすること。

6.3 PREST ファイルが存在して、新たに PREST ファイルを使用するジョブを実行できない場合の対処方法

使用する PREST ファイルは、ジョブ間 PREST ファイルとジョブステップ間 PREST ファイルとでは対処方法が異なる。このため、使用している PREST ファイルの種類をジョブ定義 XML ファイルとジョブログのエラーメッセージなどで調査する。

6.3.1 ジョブ間 PREST ファイルの場合

- bjexlslock コマンドを使用して、エラーになった PREST ファイルがほかのジョブで使用中でないことを確認する。
- ほかのジョブで使用中でない場合、OS が提供する名前付きパイプの削除手段でジョブ間 PREST ファイルを削除する。PREST ファイルを削除後に、ジョブを再実行する。
- ほかのジョブで使用中的の場合、そのジョブが終了するのを待ってジョブを再実行する。またはジョブ定義 XML ファイルの PREST のファイル名を変更して、ジョブを再実行する。

6.3.2 ジョブステップ間 PREST ファイルの場合

ジョブを再実行すると別の PREST ファイル名を割り当てるため、ジョブを実行することができる。ただし、不当な PREST ファイルが残っている可能性があるため、次に示す手順で不当に残ったジョブステップ間 PREST ファイルを削除することを推奨する。

1. ジョブ間ステップ PREST ファイルが名前付きパイプであることを確認する。名前付きパイプでない場合、ほかのジョブで使用しているか判断できないため、ファイルの削除はできない（運用上、作成日付などから使用していないことが明確であれば削除できる）。
2. 名前付きパイプの場合、ジョブステップ間 PREST ファイルのファイル名からジョブ識別子を調査する。
3. bjexlsjob コマンドでジョブ識別子のジョブが実行中でないことを確認する。
4. ジョブが実行中でなければ、OS が提供する名前付きパイプの削除手段でジョブステップ間 PREST ファイルを削除する。

6.4 ジョブが長時間待ち状態となった場合の対処方法

- bjexlslock コマンドでジョブの排他状況を表示し、排他資源管理テーブルの空き待ち (KAKC1654-W メッセージを出力) およびファイルのロック待ち (KAKC1650-W メッセージを出力) で長時間待ち状態になっていないか確認する。
- 排他資源管理テーブルの空き待ちおよびファイルのロック待ちになっていない場合は、プログラムが異常である可能性があるため、JP1/AJS からジョブを強制終了する。
- 排他資源管理テーブルの空き待ちおよびファイルのロック待ちになっている場合、bjexlslock コマンドで該当ファイルの排他状況を表示して、ファイルをロック確保しているジョブを特定し、そのジョブに異常がないか確認する。異常がない場合は、そのジョブの終了を待つ。異常がある場合は、そのジョブを JP1/AJS から強制終了する。
- JP1/IM - View または bjexlsmmsg コマンドで、プログラムがメッセージの応答待ちになっていないか確認する。メッセージの応答待ちの場合、速やかにメッセージに対して応答する。
- PREST ファイルを使用するジョブが長時間待ち状態になっている場合、PREST ファイルの使用方法が誤っている可能性がある。「[3.12.2\(4\)\(a\) プログラムが長時間待ち状態となる例](#)」, および「[3.12.2\(4\)\(b\) プログラムがデッドロックになる例](#)」を参照し、PREST ファイルの使用方法が誤っている場合は、両方のジョブを JP1/AJS から強制終了する。

6.5 メッセージ制御機能での障害対応

バッチジョブ実行システムでは、ネットワーク障害が発生した場合でも業務を終わらせることができるように、応答要求メッセージを適切に処理するコマンドを提供する。障害発生時は、OSにログインしてコマンドで応答を入力することで処理を続行できる。

なお、リトライ可能と判断しているエラーについては、COBOL2002で作成したCOBOLアプリケーション、bjexm_message コマンドまたはメッセージ出力用カタログドプロシジャを異常終了させないでリトライを繰り返す。通常メッセージ出力については、3回まで動作を実施する。その後のリトライは行わないで、かつ異常終了させないでジョブを続行させる。応答要求メッセージについては、MESSAGE_RETRY_TIMEOUT パラメータの指定によって、リトライ時間を指定できる。通信処理などで異常が発生したことについては、バッチジョブ実行システムの実行ログに記録を残して、あとから障害解析ができるようにする。

また、バッチジョブの仕掛かり中に障害などによって処理系の切り替えが発生した場合、バッチジョブ実行システムは該当するジョブを継続処理しない。このような場合には、中断されたバッチジョブを切り替え後の処理系で再実行すること。

障害が発生した場合の対処方法を次に示す。

(1)JP1/IM - View にメッセージが表示されない場合

JP1 イベントの送信処理でリトライが発生し、JP1/IM - View にメッセージが表示されないで、COBOLアプリケーション、bjexm_message コマンドまたはメッセージ出力用カタログドプロシジャが停止したままの状態になる場合がある。この場合に考えられる原因と対処方法を次に示す。

- バッチジョブ実行システムが稼働するホストで、JP1/Base のイベントサービスが稼働していない。
JP1/Base のイベントサービスを起動する。起動方法については、JP1 のマニュアルを参照のこと。
- JP1_HOSTNAME パラメータの指定が誤っている。
すべてのジョブを終了したあと、設定ファイルを修正して bjexpd を再起動し、ジョブを再実行する。
- JP1IM_MANAGER_HOST パラメータの指定が誤っている。
すべてのジョブを終了したあと、設定ファイルを修正して bjexpd を再起動し、ジョブを再実行する。

なお、JP1/IM - View にメッセージが表示されない場合でも、bjexlsmmsg および bjexchmsg コマンドを使用して応答を入力できる。入力手順については、「(2) JP1/IM - View から応答が入力できない場合」を参照のこと。

(2)JP1/IM - View から応答が入力できない場合

バッチジョブ実行システムと JP1/IM - Manager 間で、ネットワーク障害の発生または JP1/IM - Manager で応答待ちイベントの滞留が解除された場合などによって、JP1/IM - View から応答が入力できない場合、bjexlsmmsg および bjexchmsg コマンドを使用し、次に示す手順で応答を入力する。

1.bjexlsmmsg コマンドで応答要求メッセージ一覧を表示

ジョブ識別子が 000001 のジョブで、応答要求メッセージを出力済みの場合の表示例を次に示す。

```
# /opt/hitachi/bjex/sbin/bjexlsmg
MESSAGE-NO STATUS JOBID STEP DATE/TIME MESSAGE or RESPONSE
[0000010001] [Wait] 000001 001 08/01/15 19:06:08 msg=KCCC2000R-I FROM
CONSOLE指定のACCEPT文を実行しました。応答してください。プログラム名=CONS
```

2.bjexchmsg コマンドで応答を入力

「応答してください。」のメッセージに対して「y」と応答する場合、bjexchmsg コマンドで次に示すように指定する。

- -n オプションに bjexlsmg コマンドで表示される MESSAGE-NO を指定
- -r オプションに応答として"y"を指定

```
# /opt/hitachi/bjex/sbin/bjexchmsg -n 10001 -r y
```

応答を入力しないで応答要求メッセージを解除して、COBOL2002 で作成した COBOL アプリケーション、bjexm_message コマンドまたはメッセージ出力用カタログドプロシジャを終了する場合は、bjexchmsg コマンドで次に示すように指定する。

- -n オプションに bjexlsmg コマンドで表示される MESSAGE-NO を指定
- -d オプションを指定

```
# /opt/hitachi/bjex/sbin/bjexchmsg -n 10001 -d
```

(3)JP1/IM - View の [応答入力] 画面に表示される状態が、「NO LONGER MANAGED BY BJEX」となる場合

応答要求メッセージが出力済みの状態で次に示す内容が発生した場合、バッチジョブ実行システムは、出力済みの応答待ちイベントに対してキャンセルイベントを送信する。JP1/IM - View で滞留している応答待ちイベントは、キャンセルイベントによって滞留が解除される。

- JP1/AJS からの強制終了で bjexec を終了した。
- bjexchmsg コマンドで応答入力または解除を行った。
- bjexpdctl コマンドで bjexpd を停止した。

ネットワーク障害でキャンセルイベントが JP1/IM - Manager に到達しなかった場合は、JP1/IM - View に応答待ちイベントが滞留したままになる。これらの応答待ちイベントに対して JP1/IM - View で [応答入力] 画面を開くと、表示される状態が「NO LONGER MANAGED BY BJEX」となる。このような応答待ちイベントが滞留している場合は、JP1/IM - View から手動で滞留を解除する。手動で解除する手順については、JP1 のマニュアルを参照のこと。

なお、bjexec の終了時は、EXEC 要素に指定したアプリケーションが出力した応答待ちイベントに対してだけキャンセルイベントを出力する。EXEC 要素に指定したアプリケーションが生成する子プロセスが出力した応答待ちイベントに対しては、キャンセルイベントを出力しない。

6.6 JP1/FTP を使った印刷データ転送での障害対応

印刷データ (SYSOUT データファイル) に出力非保留属性 (NOHOLD 状態) が設定されている場合、スプールジョブ転送サービスは、ジョブが完了した順番で印刷データをプリントサーバに FTP 転送する。印刷順番を極力保証するために、転送処理中にエラーが発生した場合、デフォルトの設定では、スプールジョブ転送サービスは、エラー要因が取り除かれるまで該当する印刷データの転送をリトライし続ける。

ただし、設定ファイルの JP1FTP_RETRY_TIMES パラメータによって、エラー発生時に転送をリトライする回数を指定できる。

JP1FTP_RETRY_TIMES パラメータを指定している場合は、転送処理中にエラーが発生した場合にパラメータで指定した回数分、転送のリトライを繰り返す。このとき、指定回数分リトライしてもエラーが発生している場合は当該ジョブの転送処理を中断する。

プログラムの内部矛盾やリソース不足などの理由によって、スプールジョブ転送サービスのリトライ処理中にエラー要因を取り除くことが困難な場合、スプールジョブ転送サービスを一度停止・再起動してから、次のどちらかを実施する必要がある。

(1) バッチジョブを再実行する。

(2) bjextrjob コマンドで明示的に印刷データを再転送する。

(1) のとおりバッチジョブを再実行する場合には、印刷データの転送はジョブの完了順に行われることがバッチジョブ実行システムによって自動的に保証されるが、(2) のとおり bjextrjob コマンドで明示的に印刷データを再転送する場合は、バッチジョブ実行システムによる印刷データの転送順序の自動的な保証は行われない。そのため、印刷の順序性を重視する場合は、(1) のとおりバッチジョブを再実行することを推奨する。その場合、スプールジョブ転送サービスを再起動したあとで、「bjextrjob -d -t0」コマンドを実行して印刷ジョブをすべて削除してから、バッチジョブを再実行すること。

6.7 実行ログ出力用ディスクの容量不足の対処方法

スプールジョブを転送する運用をしている場合に、スプールジョブ転送デーモンの実行ログ出力でエラーが発生した場合や、実行ログ出力が停止している場合は、次の手順で回復作業を実施する。

なお、各コマンドの詳細については、「7. コマンド」を参照のこと。

1. スプールジョブ転送サービス(bjexpd)が稼働中の場合は停止する。

次のコマンドを実行して、稼働中の bjexpd を停止する。

```
# /opt/hitachi/bjex/sbin/bjexpdctl stop
```

2. 実行ログファイルのあるディスクの状態を確認する。

実行ログファイルのあるディスクの状態を確認し、容量不足が発生していれば不要なファイルを削除して空き領域を確保する。

3. スプールジョブ転送サービス(bjexpd)を起動する。

次のコマンドを実行して、bjexpd を起動する。

```
# /opt/hitachi/bjex/sbin/bjexpdctl start
```

4. 印刷ジョブが滞留している場合

実行ログ出力中のエラー発生によって、転送処理が中断され、印刷ジョブが滞留することがある。次の bjextrjob コマンドを実行して転送されなかった印刷ジョブを再転送する。

```
# /opt/hitachi/bjex/sbin/bjextrjob -r -t 秒数
```

-t 秒数 ～< 1 ～ 8 けたの 10 進数> ((0-99999999)) 《259200》

再転送する印刷ジョブの対象範囲をコマンド投入時刻から差し引いた秒数の値で指定する。

コマンドはこの秒数以前の印刷ジョブについて再転送を実行する。

エラー回復手順として指定する値は、このコマンド入力時点の時刻から 3 で投入したコマンドの時刻を引いた秒数を指定する。

3 で投入したコマンドの時刻が 15:00:00 で、このコマンドを入力する時刻が 15:00:10 のときに指定する例を次に示す。

```
# /opt/hitachi/bjex/sbin/bjextrjob -r -t 10
```

6.8 スプールジョブ転送デーモンの障害対策

スプールジョブ転送デーモンは、スプールジョブの転送、スプールの管理、共有メモリ・セマフォ・メッセージキューの作成・削除を実施する。

運用中にスプールジョブ転送デーモンを停止した場合、次に示す弊害がある。

- 印刷対象の帳票がプリントサーバへ転送されない。
- ジョブの状態表示が不正となる。
- ジョブ、スプールジョブ管理コマンドが終了しない場合がある。
- 排他/共用制御機能および実行中のジョブ情報表示機能を使用している場合は、ジョブコントローラがエラーで終了する。
- メッセージ制御機能で応答要求メッセージを出力済みの場合は、COBOL アプリケーション、bjexm_message コマンドおよびメッセージ出力用カタログドプロシジャのジョブステップがエラーで終了する。

スプールジョブ転送デーモンが停止したときに実行中だったジョブ、停止中に実行されたジョブの動作結果は保証しない。

障害などで運用中にスプールジョブ転送デーモンを停止した場合の対処方法を次に示す。

(1)障害の要因を取り除いた後、スプールジョブ転送デーモンを再起動する。

(2)次を確認する。

- スプールジョブ転送デーモンが停止したときに実行中だったジョブ、停止中に実行されたジョブの動作結果
- 転送ディレクトリの印刷ジョブ
- プリントサーバに転送されたファイル
- スプールの状態

(3)必要に応じて次のことを実施する。

- ジョブの再実行
- スプールジョブの再転送・削除
- 印刷ジョブの再転送・削除
- スプールジョブの回復

7

コマンド

この章では、バッチジョブ実行システムのコマンドについて説明する。

コマンドの記述形式

コマンドの記述形式を次に示す。

Δ_0 コマンド名 [Δ_1 -オプション名 [Δ_1 値]] ... [Δ_1 -オプション名 [Δ_1 値]]
[任意名]

(凡例)

Δ_0 : 0 バイト以上の空白文字を指定する。空白文字は省略できる。

Δ_1 : 1 バイト以上の空白文字を必ず指定する。空白文字は省略できない。

- 最初にオプションを指定し、次に任意名を指定する。
- 値のないオプションは連続して指定できる (例: 「-a-b-c」 と 「-abc」 は同じ)。
- オプションを連続して指定する場合、最後のオプションには値を指定できる (例: 「-abc xyz」 の 「xyz」 は、オプション-c の値となる)。
- 不当なオプション、または指定できる範囲外の値を指定した場合、エラーとなる。

コマンド一覧

バッチジョブ実行システムのコマンド一覧（管理者用）、バッチジョブ実行システムのコマンド一覧（一般ユーザ用）、およびバッチジョブ実行システムの設定で使用するコマンド一覧を次の表に示す。

表 7-1 バッチジョブ実行システムのコマンド一覧（管理者用）

コマンド名	機能概要	コマンドの格納場所
bjexchmsg	応答待ちメッセージに応答を入力する。	/opt/hitachi/bjex/sbin
bjexlsmg	応答待ちメッセージおよび受信待ちメッセージの一覧を表示する。	
bjexpdctl, bjexpd	スプールジョブ転送デーモンを起動, 停止する。	
bjextrjob	転送ディレクトリ下の印刷ジョブ一覧を表示, 削除, および再転送する。	

表 7-2 バッチジョブ実行システムのコマンド一覧（一般ユーザ用）

コマンド名	機能概要	コマンドの格納場所
bjexchjob	出力保留状態のスプールジョブの属性を変更したり, 出力保留状態のスプールジョブを保管したりする。	/opt/hitachi/bjex/bin
bjexec	バッチジョブを実行する。	
bjexgdg	世代データ群を生成, 更新, および登録情報を表示する。	
bjexhkgdg	世代データ群に登録されている世代ファイルの最終更新時間から保存日数に指定した日数を超過した世代ファイルを一括削除する。	
bjexlsexjob	実行中のジョブ情報を表示する。	
bjexlsjob	操作できるスプールジョブの一覧を表示する。	
bjexlslock	ファイルの排他状態を表示する。	
bjexrmjob	HOLD 状態またはエラー状態のスプールジョブを削除する。	
bjexm_message	ジョブステップとして起動し, メッセージまたは応答要求メッセージを出力し, 応答メッセージを取得する。	

表 7-3 バッチジョブ実行システムの設定で使用するコマンド一覧

コマンド名	機能概要	コマンドの格納場所
bjex_msg_setup	メッセージ制御機能の初期設定をする。	/opt/hitachi/bjex/sbin

bjex_msg_setup (メッセージ制御機能の初期設定 (バッチジョブ実行システムの設定用))

形式

```
bjex_msg_setup
```

機能

メッセージ制御機能の初期設定をする。

このコマンドを実行すると、/opt/hitachi/bjex/system/Adapter_HITACHI_BJEX_MESSAGE.conf が JP1/Base のディレクトリ (/opt/jp1base/plugin/conf) にコピーされ、JP1/IM - View に表示される応答待ちイベントから応答が入力できるようになる。

このコマンドは、バッチジョブ実行システムのインストール後に一度だけ実行する。JP1/Base を再インストールした場合は、このコマンドを再度実行する。

オプション

なし。

戻り値

戻り値	意味
0	正常終了
0 以外	エラー終了

注意事項

このコマンドは、引数を指定してもすべて無視する。

bjexchjob (スプールジョブの属性変更 (一般ユーザ用))

形式

日数およびジョブ識別子で指定する場合

```
bjexchjob [-h 論理ホスト名]
           {-R [-A | -K | -F] | -S [-a 秒数] [-i] | {-K | -F} | -C}
           -T 日数
           [ジョブ識別子… | ジョブ識別子:ジョブ識別子]
```

ジョブ識別子で指定する場合

```
bjexchjob [-h 論理ホスト名]
           {-R [-A | -K | -F] | -S [-a 秒数] [-i] | {-K | -F} | -C}
           {ジョブ識別子… | ジョブ識別子:ジョブ識別子}
```

日数およびジョブ名で指定する場合

```
bjexchjob [-h 論理ホスト名]
           {-R [-A | -K | -F] | -S [-a 秒数] [-i] | {-K | -F} | -C}
           -T 日数 -J
           [ジョブ名… | ジョブ名ワイルドカード]
```

ジョブ名で指定する場合

```
bjexchjob [-h 論理ホスト名]
           {-R [-A | -K | -F] | -S [-a 秒数] [-i] | {-K | -F} | -C}
           -J
           {ジョブ名… | ジョブ名ワイルドカード}
```

機能

出力保留状態のスプールジョブの属性を変更する。

出力保留状態のスプールジョブを保管する。

ただし、-S オプションを指定した場合は、障害などが原因で不当に実行中状態やジョブが存在しないなどの出力保留状態以外になっているスプールジョブを回復する。

オプション

-h 論理ホスト名 ~<英字と数字と- (ハイフン) >((1~244))

バッチジョブ実行システムの論理ホストで運用している場合に指定する。設定ファイルとして/opt/hitachi/bjex/conf/bjex_論理ホスト名.conf を読み込み実行する。

-R

印刷指示をする (プリントサーバに転送する)。

転送できない SYSOUT データファイルが含まれている場合、その SYSOUT データファイルは転送しない。

転送処理後の SYSOUT データファイルは、出力後保存属性の指定の有無または-K, -F オプションの指定に従って、削除または保存される。

-A

-R オプションを指定した場合に指定できる。この場合、SYSOUT データファイルの出力後保存属性指定の有無に従って出力後の処理をする（出力後保存属性の有無は変更しない）。

-K

-R オプションの指定がない場合、SYSOUT データファイルを出力後保存属性に変更する。

-R オプションの指定がある場合、転送後、ジョブを削除しない（出力後保存属性の有無は変更しない）。この場合、SYSOUT データファイルは出力後保存属性指定の有無に関係なく、転送後も保存する。

-F

-R オプションの指定がない場合、SYSOUT データファイルの出力後保存属性をなくす。

-R オプションの指定がある場合、転送後に SYSOUT データファイルを削除する。この場合、SYSOUT データファイルは出力後保存属性の有無に関係なく、転送後は削除する。

-S

スプールジョブの回復処理を要求する。

スプールジョブ転送デーモンが起動されていないなどで、管理から漏れたスプールジョブおよび出力保留状態以外のスプールジョブの管理状態を保留状態に回復処理する。

回復処理した結果として SYSOUT データファイルの状態に応じて出力保留状態にできないで、その他の状態への遷移または削除されたりすることがある。

また、サイズが 0 のスプールジョブ管理ファイルは作成処理中の可能性があり、回復処理の対象外となる。

-a 秒数 ~< 1~3 けたの 10 進数>((0~999)) 《30》

-S オプションを指定した場合に有効であり、回復処理でスプールジョブ管理ファイルの最終更新時刻が、現在時刻からこの指定秒数以内であれば、処理中のスプールジョブと判断して回復処理を行わない。

-i

-S オプションと指定した場合に有効であり、回復するスプールジョブに出力非保留属性の SYSOUT データファイルが含まれているとき、プリントサーバに転送処理する。

指定しない場合は、転送処理を行わないで回復処理をする。

-C

スプールジョブを保管する。

スプールジョブを保管するには、設定ファイルの指定で、スプールジョブ保管機能を有効にしている必要がある。また、スプールジョブを保管するには、ジョブを実行したユーザと同一ユーザ、または特権ユーザで実行する必要がある。

保管後のスプールジョブのファイルのパーミッションや所有者属性と保管機能の詳細については、「3.16 スプールジョブ保管機能」を参照のこと。

-T 日数 ~< 1~2 けたの 10 進数>((0~99))

指定した日数以前の日を処理対象とする。

属性変更処理ではジョブ実行開始時刻が指定日数以前の日であるスプールジョブを対象とする。

スプールジョブの回復処理ではスプールジョブ管理ファイルの最終更新時刻が指定日数以前の日であるスプールジョブを対象とする。

スプールジョブ管理ファイルの最終更新時刻はジョブ実行時とは異なり、スプールジョブの操作を行うと更新される。例えば、bjexchjob コマンドで属性変更すると更新される。2月2日に-T1を指定した場合、2月1日 23:59:59以前が対象となる。

-J

処理対象をジョブ名で指定する場合に指定する。この指定がない場合、ジョブ識別子オプションで指定する。

ジョブ識別子 ~<10進数>((1~999999))

処理対象とするスプールジョブのジョブ識別子を指定する。

-T オプションを指定した場合にだけ省略ができる。

このオプションの指定形式には次に示す2つの方法がある。なお、両方の指定形式は混在できない。

- ジョブ識別子△※

空白で区切って複数指定できる。

注※ △は空白を示す。

- ジョブ識別子：ジョブ識別子

1回だけ指定できる。

"ジョブ識別子：ジョブ識別子"の形式で指定した場合、左に指定したジョブ識別子から右に指定したジョブ識別子までの範囲を処理対象とする。

また、左のジョブ識別子は右のジョブ識別子よりも等しいかまたは小さくする必要がある。

ジョブ名 ~<記号名称>、ジョブ名ワイルドカード ~<任意文字列>

処理対象をジョブ名またはジョブ名ワイルドカードで指定する。この場合、-J オプションの指定が必要である。

このオプションの指定形式には次に示す2つの方法がある。なお、両方の指定形式は混在できない。

- ジョブ名△※

空白で区切って最大32個指定できる。

注※ △は空白を示す。

- 文字列*

1回だけ指定できる。

ジョブ名ワイルドカードは*を指定したワイルドカード指定で、先頭の文字列が一致するジョブ名を一括で指定できる。ワイルドカード指定をする場合は、シェルが自動的に処理しないように""で囲む必要がある。

ジョブ名またはジョブ名ワイルドカードの指定は、-T オプションを指定した場合にだけ省略ができ、すべてのジョブが処理対象となる。

戻り値

戻り値	意味
0	正常終了
0 以外	エラー終了

注意事項

- コマンドを割り込みなどで強制終了させないこと。コマンドを強制終了させた場合、スプールジョブ管理ファイルが破壊されることがある。
- このコマンドは共用資源の排他をしているため、処理途中で止まるような使用をした場合（pg コマンドなどでメッセージ表示の途中で止める場合など）、ほかのコマンドやスプールジョブ転送デーモンが待ち状態になることがあるので、そのような使用はしないこと。
- このコマンド実行にはスプールジョブ転送デーモンが起動している必要がある。スプールジョブ転送デーモンが起動していないとき、またはスプールジョブ転送デーモンが処理途中で終了したとき、コマンドの動作は保証できないため、コマンドが待ち状態になり、終了しないことがある。また、処理途中のスプールジョブが処理されないことがある。このような場合、コマンドを割り込みなどで停止する。また、処理されなかったスプールジョブは再度コマンドを入力する。
- ジョブ識別子の範囲指定をした場合、その範囲内の操作できるジョブが操作対象となる。操作対象外のジョブはエラーメッセージを出力しない。処理結果は、bjexlsjob または bjextrjob コマンドで確認する。ジョブ識別子を指定したコマンドを使用した場合はエラー情報を出力する。
- ジョブ名またはジョブ名ワイルドカード指定をした場合、操作できるジョブから該当するジョブ名のジョブを操作対象とする。操作対象外のジョブはエラーメッセージを出力しない。処理結果は、bjexlsjob または bjextrjob コマンドで確認する。ジョブ識別子を指定したコマンドを使用した場合はエラー情報を出力する。
- 保管先に同一ディレクトリやファイルがすでに存在する場合、上書きで保管する。

使用例

- ジョブ識別子 227 のスプールジョブを印刷する。

```
bjexchjob -R 227
```

- スプールジョブ転送デーモン未起動時に実行されたジョブ識別子 672 のスプールジョブを出力保留状態に回復する。スプールジョブはカレント時刻よりも 30 秒以前に終了している場合に出力保留状態に回復する。

```
bjexchjob -S -a 30 672
```

- ジョブ名が JOB001, JOB002 のジョブを印刷する。

```
bjexchjob -RJ JOB001 JOB002
```

- ジョブ識別子 10 から 100 までを印刷する。

```
bjexchjob -R 10:100
```

- ジョブ名の先頭が KEIRI のジョブを印刷する。

```
bjexchjob -RJ "KEIRI*"
```

bjexchmsg (応答待ちメッセージへの応答入力 (管理者用))

形式

```
bjexchmsg [-h 論理ホスト名] -n メッセージ番号 { -r 応答 | -d }
```

機能

応答待ちメッセージに応答を入力する。

-n オプションで指定した応答待ちメッセージに、-r オプションで指定した応答を入力する。応答待ちメッセージをキャンセルする場合は、-d オプションを指定する。

オプション

-h 論理ホスト名 ~<英字と数字と- (ハイフン) >((1~244))

バッチジョブ実行システムの論理ホストで運用している場合に指定する。設定ファイルとして/opt/hitachi/bjex/conf/bjex_論理ホスト名.conf を読み込み実行する。

-n メッセージ番号 ~<数字>((1~2147483647))

bjexlsmmsg コマンドで表示されるメッセージ番号を指定する。

-r または-d オプションの指定がない場合はエラーとする。

-r 応答 ~<ASCII 文字列>((1~512))

応答待ちメッセージへの応答を指定する。また、JP1/IM - Manager にキャンセルイベントを送信し、応答待ちイベントの滞留を解除する。

-n オプションの指定がない場合は、エラーとする。

-d

応答待ちメッセージおよび受信待ちメッセージ一覧からメッセージを削除する。また、JP1/IM - Manager にキャンセルイベントを送信し、応答待ちイベントの滞留を解除する。

-n オプションの指定がない場合は、エラーとする。

応答要求メッセージを出力したユーザアプリケーションは異常終了する。このとき該当するジョブのJOBLOG および実行ログに、次に示すメッセージを出力する。

```
KAKC5005-E An error occurred during shared memory processing. (errno=404,function=shm_clear)
```

戻り値

戻り値	意味
0	正常終了
0 以外	エラー終了

注意事項

-r オプションと-d オプションを同時に指定した場合、後に指定したオプションが有効になる。

使用例

メッセージ番号「2000000000」の応答待ちメッセージを、応答待ちメッセージおよび受信待ちメッセージ一覧から削除する。

```
bjexchmsg -n 2000000000 -d
```

bjexec (バッチジョブの実行 (一般ユーザ用))

形式

```
bjexec {[-r リスタートステップ名] [-h [-n 論理ホスト名] ] [-c]ジョブ定義XMLファイルのパス名  
| -v }
```

機能

ジョブコントローラを起動して、引数に指定したジョブ定義 XML ファイルのバッチジョブを実行する。

オプション

-r リスタートステップ名

リスタートするジョブステップ名を指定する。指定したジョブステップから実行を開始する。リスタートステップ名は次の形式で指定する。

- ジョブステップ名[.プロシジャステップ名]

ジョブステップ名 ~<記号名称>((1~31)) : リスタートするジョブステップ名を記述する。

プロシジャステップ名 ~<記号名称>((1~31)) : リスタートするジョブステップがカタログドプロシジャの場合、ピリオド (.) に続けてプロシジャ内のジョブステップ名を記述する。

指定したジョブステップが存在しないときは、ジョブを実行しないで終了する。

-r ""などで空白だけの文字列や¥0 だけの空文字列を指定すると通常ジョブとして先頭から実行する。同一ジョブステップ名のジョブステップが複数存在する場合、先頭から探して最初に見つかったジョブステップから実行する。ただし、JSPEC 機能で並列実行するジョブステップ、または IF 要素の THEN 節や ELSE 節内のジョブステップは検索の対象にならない。

-h

バッチジョブ実行システムの論理ホストで運用している場合に指定する。

JP1/AJS から起動する場合

JP1/AJS の論理ホストから実行すること。JP1/AJS の論理ホストから実行すると、環境変数 JP1_HOSTNAME に JP1/AJS の論理ホスト名が設定される。bjexec コマンドは環境変数 JP1_HOSTNAME から論理ホスト名を取得し、設定ファイル (bjex_論理ホスト.conf) を読み込み実行する。

コマンドラインから起動する場合

-n オプションで指定したバッチジョブ実行システムの論理ホスト環境で設定ファイル (bjex_論理ホスト.conf) を読み込み実行する。JP1_HOSTNAME が設定されている場合、無視する。

-n 論理ホスト名

コマンドラインからバッチジョブ実行システムの論理ホスト環境でバッチジョブを起動する場合に JP1 の論理ホスト名を指定する。

このオプションを指定した場合、-h オプションを指定しなければならない。

このオプションを指定した場合、指定した論理ホスト名のバッチジョブ実行システムの論理ホスト環境で設定ファイル (bjex_論理ホスト.conf) を読み込み実行する。

-C

ジョブ定義 XML ファイルの文法チェックをする。

このオプションを指定した場合、ジョブ定義 XML ファイルの文法チェックだけを行い、バッチジョブの実行はしない。

ジョブ定義 XML ファイルのパス名 ~<パス名>

ジョブ定義 XML ファイルのパス名を指定する。

-V

このオプションを指定した場合、バージョン情報を表示する。バッチジョブの実行はしない。

戻り値

戻り値については、「表 10-39 ジョブコントローラの終了コード一覧」を参照のこと。

注意事項

- bjexec コマンドは、環境変数 TZ に従って情報を出力する。JP1/AJS から起動したジョブは、次に示す方法で TZ を指定すること。
 - JP1/AJS のジョブ定義、環境変数の定義で指定する。
 - システムプロファイル (/etc/profile) に指定する。
 - ユーザプロファイル (~/.profile) に指定する。

なお、JP1/AJS のジョブとして起動した場合、環境変数 TZ については次に示す注意事項がある。

- AIX の場合は、/etc/environment の情報は引き継がれない。
- ジョブ定義 XML ファイル、ジョブ内のシェルスクリプトで環境変数 TZ を指定しても、ジョブコントローラには有効にならない。
- 論理ホストの JP1/AJS からコマンドを実行して-h オプションを省略した場合、設定ファイル (bjex.conf) を読み込み実行する。なお、この場合、物理ホスト環境の排他/共用制御の情報を参照する。
- 論理ホストの JP1/AJS からコマンドを実行した場合、環境変数 JP1_HOSTNAME に論理ホスト名が設定される。このとき、-n オプションを指定しても-n オプションに指定した論理ホスト名を無視して環境変数 JP1_HOSTNAME に設定されたバッチジョブ実行システムの論理ホストの設定ファイル (bjex_論理ホスト.conf) を読み込み実行する。

使用例

ジョブ定義 XML ファイルの文法チェックをする。

```
bjexec -c /home/bjex/xml/sample5.xml
```

bjexgdg (世代データ群の生成および更新 (一般ユーザ用))

形式

```
bjexgdg [-m [-d] [-l エントリ数][世代ディレクトリ名/]世代データ群名] |  
[-u -l エントリ数 [世代ディレクトリ名/]世代データ群名] |  
[-q [世代ディレクトリ名/]世代データ群名] |  
[-i [世代ディレクトリ名/]世代データ群名[(相対世代番号)]] |  
[-e [世代ディレクトリ名/]世代データ群名[(相対世代番号)]] |  
[-v]
```

機能

世代データ群の生成, 更新, および登録情報を表示する。

オプション

-m

世代データ群を生成することを示す。世代ディレクトリ名で示されるディレクトリ (指定のない場合は, カレントディレクトリ) 内に世代データ群名.gdg という名称の世代管理ファイルを生成する。

-d

日付世代番号指定機能を選択することを示す。

-l エントリ数 ~<数字> ((相対世代番号指定機能: 1 ~ 255), (日付世代番号指定機能: 1~511)) 《相対世代番号指定機能: 255, 日付世代番号指定機能: 省略不可》

世代データ群に属する世代ファイルの最大数を指定する。世代ファイルが最大数を超えた場合, 相対世代番号指定機能では, 最も古い (相対世代番号が最も小さい) 世代ファイルから順に削除し, 日付世代番号指定機能では, 実行開始日 (EXD に格納された日付または Dyyyymmdd) が最も古い世代ファイルから順に削除する。これを自動消去機能 (SCRATCH) と呼ぶ。

相対世代番号指定機能で, このオプションを省略した場合, 世代データ群で管理される世代ファイル数は, 255 まで利用できる。世代ファイルの数が 255 を超えた場合, 最も古い世代ファイルは削除しないため, 世代管理対象からは外れた世代ファイルは増加し続ける。これをデータ保存機能 (NOSCRATCH) と呼ぶ。日付世代番号指定機能の場合, -l オプションを省略することはできない。相対世代番号指定機能の場合, 世代ファイルに与える 4 けたの通し番号がなくなった時点で世代ファイル機能自体が利用できなくなるため, 世代データ群を作る場合は必ずエントリ数を指定して作成すること。

[世代ディレクトリ名/] 世代データ群名 ~<パス名> ((1~1023 文字))

世代ディレクトリの名称と, 世代データ群名を指定する。世代管理ファイルのパス名から ".gdg" を除いた名称を指定する。

-u

指定された既存世代データ群のエントリ数を変更する。

自動消去機能 (SCRATCH) が設定されている世代データ群の場合、設定されている世代データファイルのエントリ数がこの指定によって小さくなる時、登録済みの世代データファイル数が世代データファイルの最大数を超過した世代データファイルの最も古いものから順に消去される。

-q

世代管理ファイルを削除する。

この指定では、世代管理ファイルで1つでも管理情報が残っている場合は削除できない。

-i

指定した世代データ群の世代管理ファイルの登録情報および登録されている世代ファイル名とその世代ファイルの作成日時を表示する。

相対世代番号指定機能の場合、相対世代番号を指定したとき、特定の世代ファイル名と作成日時だけを表示する。

日付世代番号指定機能の場合、相対世代番号を指定して特定の世代ファイル名の情報を表示することはできない。

相対世代番号 ~<整数>((-254~0))

指定した世代データ群の世代管理ファイルに登録されている世代ファイルの相対世代番号を指定する。

相対世代番号を指定する場合、メタキャラクタである左括弧 "(" と右括弧 ")" を使用するため、一般的な文字列として入力するためには、引用符 (バックスラッシュまたは円記号、単一引用符、二重引用符) で囲む必要がある。

-e

指定した世代データ群の世代管理ファイルの登録情報、登録されている世代ファイル名および最終更新時間から経過した日数を表示する。

相対世代番号指定機能の場合、相対世代番号を指定したとき、特定の世代ファイル名とその世代ファイルの作成日時とその世代ファイルの最終更新時間から経過した日数を表示する。日付世代番号指定機能の場合、相対世代番号を指定して特定の世代ファイルの情報を表示することはできない。

-v

バージョン情報を表示する。

戻り値

戻り値については、「表 10-39 ジョブコントローラの終了コード一覧」を参照のこと。

注意事項

- 世代管理ファイルは `bjexgdg` コマンドの起動時の `umask` の指定に従い、ファイルパーミッションが設定される。世代管理ファイルには、世代ファイルを使用するすべてのユーザに対して書き込み権限・読み込み権限を設定する必要がある。世代ファイルを参照するだけでも世代管理ファイルには書き込み権限が必要である。
- 世代データ群の削除は、`bjexgdg` コマンドの `-q` オプションで行うこと。`rm` コマンドなどで削除が必要となった場合、次に示すファイルを削除すること。
 - "世代データ群名.gdg"ファイル

- ".世代データ群名.gdg"ファイル
- bjexgdg コマンドの-e オプションで表示される経過日数は、該当するコマンドを実行した時間*と表示対象とした世代ファイルの最終更新時間*の差が、24 時間で除算した商（24 時間未満は切り捨て）を経過日数とする。
注※ 紀元（1970 年 1 月 1 日 00:00:00 UTC）からの経過秒

使用例

- 相対世代番号指定機能で、世代ファイル最大数 10 の世代データ群 bjgdg を生成する。

```
bjexgdg -m -l 10 bjgdg
```

- 日付世代番号指定機能で、世代ファイル最大数 10 の世代データ群 bjgdg を生成する。

```
bjexgdg -m -d -l 10 bjgdg
```

- 世代データ群 bjgdg の世代ファイル最大数を 20 に変更する。

```
bjexgdg -u -l 20 bjgdg
```

- 世代データ群 bjgdg を削除する。

```
bjexgdg -q bjgdg
```

- bjexgdg コマンドのバージョン情報を表示する。

```
bjexgdg -v
```

- 世代データ群 bjgdg の管理情報および登録されているすべての世代ファイル名を表示する。

```
bjexgdg -i bjgdg
```

- 世代データ群 bjgdg の管理情報および相対世代番号 (-1) の世代ファイル名を表示する。

```
bjexgdg -i "bjgdg(-1)"
```

- 世代データ群 bjgdg の管理情報および登録されているすべての世代ファイル名と、世代ファイルの最終更新時間から経過した日数を表示する。

```
bjexgdg -e bjgdg
```

- 世代データ群 bjgdg の管理情報および相対世代番号 (-1) の世代ファイル名と、その世代ファイルの最終更新時間から経過した日数を表示する。

```
bjexgdg -e "bjgdg(-1)"
```

出力例

- 世代データ群 bjgdg の管理情報および登録されているすべての世代ファイル名を表示する。

```
bjexgdg -i bjgdg
```

- 相対世代番号指定機能

```

GDG BASE ----- bjpgdg                               ... (1)
ATTRIBUTES
LIMIT ----- 10  SCRATCH                             ... (2)
INFILE ----- 4                                       ... (3)
ASSOCIATIONS
( 0) -- bjpgdg.G0004 20130615 11:22                   ... (4)
(-1) -- bjpgdg.G0003 20130615 11:22                   ... (5)
(-2) -- bjpgdg.G0002 20130615 11:22                   ... (6)
(-3) -- bjpgdg.G0001 20130615 11:22                   ... (7)

```

- (1)bjexgdg コマンドで指定した世代管理ファイルに登録されている世代データ群名を示す。
- (2)世代管理ファイルで登録できる世代ファイル数を"LIMIT- - -"に続いて表示する (1~255)。世代データ群に設定された属性 (SCRATCH/NOSCRATCH) を表示する。
- (3)登録されている世代ファイル数を"INFILE- - -"に続いて表示する。
- (4)登録されている世代ファイル名を" (相対世代番号) - -"に続いて表示し、ファイルの作成日時を表示する。
- (5)登録されている世代ファイル名を" (相対世代番号) - -"に続いて表示し、ファイルの作成日時を表示する。
- (6)登録されている世代ファイル名を" (相対世代番号) - -"に続いて表示し、ファイルの作成日時を表示する。
- (7)登録されている世代ファイル名を" (相対世代番号) - -"に続いて表示し、ファイルの作成日時を表示する。

注 (4)~(7)は相対世代番号 0 から-254 の順に登録されている世代ファイル名とその世代ファイルの作成日時だけ表示する。

• 日付世代番号指定機能

```

GDG BASE ----- bjpgdg                               ... (1)
ATTRIBUTES
LIMIT ----- 10  SCRATCH                             ... (2)
INFILE ----- 4                                       ... (3)
OPTION -----EXDATE                                   ... (4)
ASSOCIATIONS
( 1) -- bjpgdg.D20130615 20130615 11:22               ... (5)
( 2) -- bjpgdg.D20130614 20130615 11:22               ... (6)
( 3) -- bjpgdg.D20130613 20130615 11:22               ... (7)
( 4) -- bjpgdg.D20130612 20130615 11:22               ... (8)

```

- (1)bjexgdg コマンドで指定した世代管理ファイルに登録されている世代データ群名を示す。
- (2)世代管理ファイルで登録できる世代ファイル数を"LIMIT- - -"に続いて表示する (1~511)。世代データ群に設定された属性 (SCRATCH) を表示する。
- (3)登録されている世代ファイル数を"INFILE- - -"に続いて表示する。
- (4)日付世代番号指定機能の世代データ群であることを示す。
- (5)登録されている世代ファイル名を" (番号) - -"に続いて表示し、ファイルの作成日時を表示する。
- (6)登録されている世代ファイル名を" (番号) - -"に続いて表示し、ファイルの作成日時を表示する。
- (7)登録されている世代ファイル名を" (番号) - -"に続いて表示し、ファイルの作成日時を表示する。
- (8)登録されている世代ファイル名を" (番号) - -"に続いて表示し、ファイルの作成日時を表示する。

注 (5)~(8)は相対世代番号 1 から 511 の順に登録されている世代ファイル名だけ表示する。新しい世代ファイルほど小さい番号となる。

- 世代データ群 bjgdg の管理情報と相対世代番号 (-1) の世代ファイル名を表示する。この指定は、相対世代番号指定機能の世代データ群に対してだけ使用できる。

```
bjexgdg -i "bjgdg(-1)"
```

```
GDG BASE ----- bjgdg                ... (1)
ATTRIBUTES
LIMIT ----- 10  SCRATCH              ... (2)
INFILE ----- 4                       ... (3)
ASSOCIATIONS
( -1) -- bjgdg.G0003 20130615 11:22    ... (4)
```

- (1)bjexgdg コマンドで指定した世代管理ファイルに登録されている世代データ群名を示す。
- (2)世代管理ファイルで登録できる世代ファイル数を"LIMIT- - -"に続いて表示する (1~255)。世代データ群に設定された属性 (SCRATCH/NOSCRATCH) を表示する。
- (3)登録されている世代ファイル数を"INFILE- - -"に続いて表示する。
- (4)bjexgdg -i "bjgdg(-1)"で指定した相対世代番号に対応した、世代ファイル名とその世代ファイルの作成日時を" (相対世代番号) - -"に続いて表示する。

- 世代データ群 bjgdg の管理情報と登録されているすべての世代ファイル名および最終更新時間からコマンド入力時点までの経過日数を表示する。

```
bjexgdg -e bjgdg
```

• 相対世代番号指定機能

```
GDG BASE ----- bjgdg                ... (1)
ATTRIBUTES
LIMIT ----- 10  SCRATCH              ... (2)
INFILE ----- 4                       ... (3)
ASSOCIATIONS
(  0) -- bjgdg.G0004 0005              ... (4)
( -1) -- bjgdg.G0003 0005              ... (5)
( -2) -- bjgdg.G0002 0005              ... (6)
( -3) -- bjgdg.G0001 0005              ... (7)
```

- (1)bjexgdg コマンドで指定した世代管理ファイルに登録されている世代データ群名を示す。
- (2)世代管理ファイルで登録できる世代ファイル数を"LIMIT- - -"に続いて表示する (1~255)。世代データ群に設定された属性 (SCRATCH/NOSCRATCH) を表示する。
- (3)登録されている世代ファイル数を"INFILE- - -"に続いて表示する。
- (4)登録されている世代ファイル名を" (相対世代番号) - -"に続いて表示し、最終更新時間からの経過日数を表示する。
- (5)登録されている世代ファイル名を" (相対世代番号) - -"に続いて表示し、最終更新時間からの経過日数を表示する。
- (6)登録されている世代ファイル名を" (相対世代番号) - -"に続いて表示し、最終更新時間からの経過日数を表示する。
- (7)登録されている世代ファイル名を" (相対世代番号) - -"に続いて表示し、最終更新時間からの経過日数を表示する。

• 日付世代番号指定機能

```

GDG BASE ----- bjpgdg                ... (1)
ATTRIBUTES
LIMIT ----- 10  SCRATCH                ... (2)
INFILE ----- 4                          ... (3)
OPTION ----- EXDATE                      ... (4)
ASSOCIATIONS
( 1) -- bjpgdg.D20130615 0005            ... (5)
( 2) -- bjpgdg.D20130614 0006            ... (6)
( 3) -- bjpgdg.D20130613 0007            ... (7)
( 4) -- bjpgdg.D20130612 0008            ... (8)

```

(1)bjexgdg コマンドで指定した世代管理ファイルに登録されている世代データ群名を示す。

(2)世代管理ファイルで登録できる世代ファイル数を"LIMIT- - -"に続いて表示する (1~511)。
世代データ群に設定された属性 (SCRATCH) を表示する。

(3)登録されている世代ファイル数を"INFILE- - -" に続いて表示する。

(4)日付世代番号指定機能の世代データ群であることを示す。

(5)登録されている世代ファイル名を" (番号) - -"に続いて表示し、最終更新時間からの経過日数を表示する。

(6)登録されている世代ファイル名を" (番号) - -"に続いて表示し、最終更新時間からの経過日数を表示する。

(7)登録されている世代ファイル名を" (番号) - -"に続いて表示し、最終更新時間からの経過日数を表示する。

(8)登録されている世代ファイル名を" (番号) - -"に続いて表示し、最終更新時間からの経過日数を表示する。

- 世代データ群 bjpgdg の管理情報と相対世代番号 (-1) の世代ファイル名および最終更新時間からコマンド入力時点までの経過日数を表示する。この指定は、相対世代番号指定機能の世代データ群に対してだけ使用できる。

bjexgdg -e "bjpgdg(-1)"

```

GDG BASE ----- bjpgdg                ... (1)
ATTRIBUTES
LIMIT ----- 10  SCRATCH                ... (2)
INFILE ----- 4                          ... (3)
ASSOCIATIONS
( -1) -- bjpgdg.G0003 0005                ... (4)

```

(1)bjexgdg コマンドで指定した世代管理ファイルに登録されている世代データ群名を示す。

(2)世代管理ファイルで登録できる世代ファイル数を"LIMIT- - -"に続いて表示する (1~255)。
世代データ群に設定された属性 (SCRATCH/NOSCRATCH) を表示する。

(3)登録されている世代ファイル数を"INFILE- - -"に続いて表示する。

(4)登録されている世代ファイル名を" (相対世代番号) - -"に続いて表示し、最終更新時間からの経過日数を表示する。

bjexhkgdg (保存日数に指定した日数を超過した世代ファイルの一括削除を行う (一般ユーザ用))

形式

```
bjexhkgdg [-d 保存日数 [世代ディレクトリ名/]世代データ群名] |  
          [-v]
```

機能

世代データ群に登録されている世代ファイルの最終更新時間から保存日数に指定した日数を超過した世代ファイルを一括削除する。この機能は、相対世代番号指定機能の世代データ群に対してだけ使用できる。

オプション

-d

世代データ群で管理する世代ファイルの削除と管理情報を削除する。

保存日数 ~<数字>((0~9999))

世代ファイルの最終更新時間からの保存日数を指定する。ここで指定した日数+1日を経過している世代ファイルを削除する。

[世代ディレクトリ名/] 世代データ群名 ~<パス名>((1~1023文字))

世代ディレクトリの名称と、世代データ群名を指定する。世代管理ファイルのパス名から".gdg"を除いた名称を指定する。

-v

バージョン情報を表示する。

戻り値

戻り値については、「表 10-39 ジョブコントローラの終了コード一覧」を参照のこと。

注意事項

- 保存日数で指定した日数は、bjexhkgdg コマンドを入力した時刻から算出し、24 時間経過ごとに 1 日、2 日と計算する。24 時間を経過していないものは、1 日経過とは見なさない。経過日数は、bjexgdg コマンドで-e オプションを指定して確認する必要がある。
- bjexhkgdg コマンドでは、世代データ群は削除できない。
- bjexhkgdg コマンドでは、世代データ群の管理から外れた世代ファイルは削除できない。
- bjexhkgdg コマンドを日付世代番号指定機能の世代データ群に対して使用することはできない。

使用例

- 世代データ群 bjgdg に登録されている世代ファイルで、保存日数 10 日を超過したもの（11 日以上経過したもの）を削除対象とする。

```
bjexhkgdg -d 10 bjgdg
```

- bjexhkgdg コマンドのバージョン情報を表示する。

```
bjexhkgdg -v
```

bjexlsexjob (実行中のジョブ情報の表示 (一般ユーザ用))

形式

```
bjexlsexjob [-h 論理ホスト名]
             [-s | -c | -j ジョブ識別子]
```

機能

実行中のジョブの情報を表示する。

実行中のジョブの表示形式には、簡易表示 (ステップに関する情報を表示するものと CPU 使用時間に関する情報を表示するもの) と詳細表示がある。

簡易表示でジョブの一覧を表示する場合、ジョブ識別子の昇順に表示される。

オプション

-h 論理ホスト名 ～<英字と数字と- (ハイフン) >((1～244))

バッチジョブ実行システムの論理ホストで運用している場合に、このコマンドを実行させる論理ホスト名を指定する。設定ファイルとして/opt/hitachi/bjex/conf/bjex_論理ホスト名.conf を読み込み実行する。

-s

実行中のジョブの一覧を表示する。

-c

実行中のジョブの CPU 使用時間を表示する。

-j ジョブ識別子 ～<数字>((1～999999))

実行中のステップ情報・CPU 情報を含む詳細情報を表示するジョブのジョブ識別子を指定する。

戻り値

戻り値	意味
0	正常終了
0 以外	エラー終了

使用例

- 実行中のジョブ一覧のステップに関する情報を表示する

```
bjexlsexjob
```

出力例

KAKC4600-I EXECUTION JOB INFORMATION DATE=2015/03/11 TIME=11:22:13									
JOBID	JP1JOBID	JOBSTART	STEPSTRT	JOBNAME	STEPNAME	PROCSTEP	PPID	PID	STEP PID
001770	1229	11:17:56	11:21:57	SAMPLE_JOB01	STEP01	PROC01_STEP3	9562	9562	9690
001771	1230	11:17:55	11:21:08	SAMPLE_JOB02		PROC01_STEP1	9564	9564	9684
001772	1231	11:17:55	11:17:56	BATCH001	STEP1		9563	9563	9630
001773		11:19:10	11:22:11	JOB0091a	SORT3		9649	9649	9696
001774		11:19:15	11:22:10	PRINT_JOB	print_a3	PROC01_PRINT	9653	9653	9694

出力項目の説明

- JOBID
ジョブ識別子 (6 けた)
- JP1JOBID
JP1 ジョブ識別子 (10 けた。JP1 から投入されたジョブでない場合は空白)
- JOBSTART
ジョブの実行開始時刻
- STEPSTRT
ジョブステップの実行開始時刻
- JOBNAME
実行中のジョブ名称 (ジョブ名称の先頭から 12 文字を表示する)
- STEPNAME
実行中のジョブステップ名称 (ジョブステップ名称の先頭から 12 文字を表示する。ステップ名の指定がない場合は空白)
- PROCSTEP
実行中のプロシジャステップ名称 (プロシジャステップ名称の先頭から 12 文字を表示する。プロシジャステップ名の指定がない場合は空白)
- PPID
ジョブコントローラプロセス (bjexec) のプロセス番号 (10 進数, 10 けた)
- PID
ジョブコントローラプロセス (bjexec) のプロセス番号 (10 進数, 10 けた)
PPID と同じ内容である。
- STEP PID
ジョブステッププロセスのプロセス番号 (10 進数, 10 けた)
- 実行中のジョブ一覧の CPU 使用時間に関する情報を表示する

```
bjexlsexjob -c
```

出力例

KAKC4600-I EXECUTION JOB INFORMATION DATE=2015/03/11 TIME=11:22:15									
JOBID	JP1JOBID	JOBSTART	STEPSTRT	JOBNAME	STEPNAME	C-TIME	PRI	NICE	JP1PRI
001770	1229	11:17:56	11:21:57	SAMPLE_JOB01	STEP01	102.440	35	10	2
001771	1230	11:17:55	11:21:08	SAMPLE_JOB02		171.160	25	0	3
001772	1231	11:17:55	11:17:56	BATCH001	STEP1	5.510	39	19	1
001773		11:19:10	11:22:11	JOB0091a	SORT3	100.190	21	0	
001774		11:19:15	11:22:10	PRINT_JOB	print_a3	100.350	23	0	

出力項目の説明

- JOBID
ジョブ識別子 (6 けた)
- JP1JOBID
JP1 ジョブ識別子 (10 けた。JP1 から投入されたジョブでない場合は空白)
- JOBSTART
ジョブの実行開始時刻
- STEPSTRT
ジョブステップの実行開始時刻
- JOBNAME
実行中のジョブ名称 (ジョブ名称の先頭から 12 文字を表示する)
- STEPNAME
実行中のジョブステップ名称 (ジョブステップ名称の先頭から 12 文字を表示する。ステップ名の指定がない場合は空白)
- C-TIME
実行中のジョブが使用した CPU 時間。6 けたの秒 (約 277 時間) と 3 けたのミリ秒で表示する。最大値を超えた場合 999999.999 となる。実行終了したジョブステップで使用した CPU 時間と実行中のジョブステップで使用している CPU 時間の和で求めた参考値である。
実行終了したジョブステップの CPU 時間は KAKC0097-I メッセージで表示されるものの和であり、実行中のジョブステップの CPU 時間は、次のように求める。
 - Linux の場合
CPU 使用時間は、プロセス番号を基に /proc/プロセス番号/stat ファイル中の utime と stime の和で求めた参考値である。
 - AIX の場合
CPU 使用時間は、プロセス番号を基に /proc/プロセス番号/psinfo ファイル中の pr_time から求めた参考値である。
- PRI
ジョブステッププロセスの優先度であり、次のように求める。
 - Linux の場合

PRI 値は、プロセス番号を基に/proc/プロセス番号/stat ファイル中の priority から求めた参考値である。

- AIX の場合

PRI 値は、プロセス番号を基に/proc/プロセス番号/psinfo ファイル中の lwpsinfo 構造体中の pr_nice から求めた参考値である。

- NICE

ジョブステッププロセスの NICE 値であり、次のように求める。

- Linux の場合

NICE 値は、プロセス番号を基に/proc/プロセス番号/stat ファイル中の nice から求めた参考値である。

- AIX の場合

NICE 値は、プロセス番号を基に/proc/プロセス番号/psinfo ファイル中の lwpsinfo 構造体中の pr_nice から 40 を引いて求めた参考値である (NICE 値はマイナス表示になることがある)。

- JP1PRI

JP1 実行優先順位 (JP1 から投入されたジョブでない場合は空白)

- 実行中のジョブの詳細情報を表示する

```
bjexlsexjob -j ジョブ識別子
```

出力例

```
KAKC4600-I EXECUTION JOB INFORMATION DATE=2015/03/11 TIME=11:22:21
JOBID          : 001770
JOB NAME       : SAMPLE_JOB01
JOB START TIME : 2015/03/11 11:17:56
JOB C-TIME     : 103.390
STEP NAME      : STEP01
PROC-STEP NAME : PROC01_STEP3
STEP START TIME : 2015/03/11 11:21:57
STEP C-TIME    : 3.570
PGM            : /home/bjexuser/bin/loopm
XML FILE       : /home/bjexuser/etc/sample/xml/sample1.xml
PPID           : 9562
PID            : 9562
STEP PID      : 9690
JP1JOBID      : 1229
JP1PRI        : 2
PRI           : 35
NICE          : 10
```

出力項目の説明

- JOBID
ジョブ識別子 (6 けた)
- JOBNAME
実行中のジョブ名称 (ジョブ名称 31 文字を表示する)
- JOB START TIME
ジョブの実行開始日付・時刻
- JOB C-TIME

実行中のジョブが使用した CPU 時間。6 けたの秒（約 277 時間）と 3 けたのミリ秒で表示する。最大値を超えた場合 999999.999 となる。

実行終了したジョブステップで使用した CPU 時間と実行中のジョブステップで使用している CPU 時間の和で求めた参考値である。

実行終了したジョブステップの CPU 時間は KAKC0097-I メッセージで表示されるものの和であり、実行中のジョブステップの CPU 時間は、次のように求める。

- **Linux の場合**

CPU 使用時間は、プロセス番号を基に /proc/プロセス番号/stat ファイル中の utime と stime の和で求めた参考値である。

- **AIX の場合**

CPU 使用時間は、プロセス番号を基に /proc/プロセス番号/psinfo ファイル中の pr_time から求めた参考値である。

- **STEP NAME**

実行中のジョブステップ名称（ジョブステップ名称 31 文字を表示する。ステップ名の指定がない場合は空白）

- **PROC-STEP NAME**

実行中のプロシジャステップ名称（プロシジャステップ名称 31 文字を表示する。プロシジャステップ名の指定がない場合は空白）

- **STEP START TIME**

ジョブステップの実行開始日付・時刻

- **STEP C-TIME**

実行中のジョブステップで使用した CPU 時間。6 けたの秒（約 277 時間）と 3 けたのミリ秒で表示する。最大値を超えた場合 999999.999 となる。

- **Linux の場合**

CPU 使用時間は、プロセス番号を基に /proc/プロセス番号/stat ファイル中の utime と stime の和で求めた参考値である。

- **AIX の場合**

CPU 使用時間は、プロセス番号を基に /proc/プロセス番号/psinfo ファイル中の pr_time から求めた参考値である。

- **PGM**

実行プログラム名称（最大 1023 けた）

- **XML FILE**

ジョブ定義 XML ファイル名称（パスの先頭から 1023 文字を表示する）

- **PPID**

ジョブコントローラプロセス (bjexec) のプロセス番号（10 進数, 10 けた）

- **PID**

ジョブコントローラプロセス (bjexec) のプロセス番号（10 進数, 10 けた）

PPID と同じ内容である。

- STEP PID

ジョブステッププロセスのプロセス番号 (10 進数, 10 けた)

- JP1JOBID

JP1 ジョブ識別子 (10 けた。JP1 から投入されたジョブでない場合は空白)

- JP1PRI

JP1 実行優先順位 (JP1 から投入されたジョブでない場合は空白)

- PRI

ジョブステッププロセスの優先度であり, 次のように求める。

- Linux の場合

PRI 値は, プロセス番号を基に /proc/プロセス番号/stat ファイル中の priority から求めた参考値である。

- AIX の場合

PRI 値は, プロセス番号を基に /proc/プロセス番号/psinfo ファイル中の lwpsinfo 構造体中の pr_nice から求めた参考値である。

- NICE

ジョブステッププロセスの NICE 値であり, 次のように求める。

- Linux の場合

NICE 値は, プロセス番号を基に /proc/プロセス番号/stat ファイル中の nice から求めた参考値である。

- AIX の場合

NICE 値は, プロセス番号を基に /proc/プロセス番号/psinfo ファイル中の lwpsinfo 構造体中の pr_nice から 40 を引いて求めた参考値である (NICE 値はマイナス表示になることがある)。

注意事項

- ジョブステップの前後処理中のジョブステッププロセスが実行中でない場合は, ジョブステップの情報は表示されないでジョブの情報だけが表示される。
- JSPEC ジョブの場合, サブステップの情報は表示されないで, メインステップの情報だけ表示される。
- 実行中のジョブステップで使用した CPU 時間は, ジョブコントローラが起動したプロセスだけの情報となる。ジョブコントローラが起動したプロセスから新たに起動したプロセス分は含まれない。
また, 実行中のジョブステップで使用した CPU 時間は KAKC0097-I メッセージで表示する C-Time とは精度が異なるため, 一致しない場合がある。
- このコマンドでジョブ情報を表示するには, 設定ファイルの JOB_STATUS_DISPLAY パラメータで ENABLE の指定が必要である。

bjexlsjob (スプールジョブの一覧表示 (一般ユーザ用))

形式

ジョブ識別子で指定する場合

```
bjexlsjob [-h 論理ホスト名]
           [{ -l | -s }]
           [-n]
           [-f]
           [{ -t | -T [-D { UTC | LT }] }]
           [ジョブ識別子… | ジョブ識別子:ジョブ識別子]
```

ジョブ名で指定する場合

```
bjexlsjob [-h 論理ホスト名]
           [{ -l | -s }]
           [-n]
           [-f]
           [{ -t | -T [-D { UTC | LT }] }]
           -J
           [ジョブ名… | ジョブ名ワイルドカード]
```

機能

スプールジョブの一覧を表示する。

スプールジョブの表示形式には、簡易表示と詳細表示がある。

オプション

-h 論理ホスト名 ~<英字と数字と- (ハイフン) >((1~244))

バッチジョブ実行システムの論理ホストで運用している場合に指定する。設定ファイルとして/opt/hitachi/bjex/conf/bjex_論理ホスト名.confを読み込み実行する。

-l

操作できるスプールジョブの一覧を詳細表示する。

詳細表示では出力保留状態のジョブの情報だけを表示する。

出力保留状態であっても、削除処理中などで詳細情報が求まらない場合は表示しない。

-s

スプールジョブの一覧を簡易表示する。

ジョブ名、ジョブ名ワイルドカード指定では、-f オプションの指定を除いて出力保留状態のジョブだけを表示する。また、削除処理中などでジョブ名が求まらない場合は表示しない。

-n

ヘッダ表示を抑止する。

-f

スプール管理ファイルに異常があっても表示する。

スプールジョブ転送デーモンが停止するなどの理由でスプールジョブ転送サービスの管理から漏れたスプールジョブ、またはその他の予期しない異常がある場合、通常の実行ではスプールジョブは表示しない。これらのジョブを、スプール管理ファイルに障害があった場合の回復支援として一覧表示する場合に使用する。

ただし、障害の状態によっては表示できない場合もある。

-t

1970年1月1日からの通算秒でジョブ実行開始、および終了時刻をUTC (Coordinated Universal Time) で10けた固定で表示する。

出力保留状態の場合だけ表示する。表示できない状態の場合は、 "-" (ハイフン) 10けたを表示する。

-T

ジョブ実行の開始時刻および終了時刻を表示する。

出力保留状態の場合だけ表示する。表示できない状態の場合は、 "-" (ハイフン) で表示する。

-D UTC

-T オプションを指定した場合に、時刻の形式をUTCで表示する。

-D LT

-T オプションを指定した場合に、時刻の形式をタイムゾーンの設定に従ったローカルタイムで表示する。

-J

処理対象をジョブ名で指定する場合に指定する。この指定がない場合、ジョブ識別子オプションで指定する。

ジョブ識別子 ~<10進数>((1~999999))

表示するジョブ識別子を指定する。省略した場合、操作できるすべてのスプールジョブを対象とする。このオプションの指定形式には次に示す2つの方法がある。なお、両方の指定形式は混在できない。

- **ジョブ識別子△※**

空白で区切って複数指定できる。

注※ △は空白を示す。

- **ジョブ識別子:ジョブ識別子**

1回だけ指定できる。

"ジョブ識別子:ジョブ識別子"の形式で指定した場合、左に指定したジョブ識別子から右に指定したジョブ識別子までの範囲を処理対象とする。

また、左のジョブ識別子は右のジョブ識別子よりも等しいかまたは小さくする必要がある。

ジョブ名 ~<記号名称>, **ジョブ名ワイルドカード** ~<任意文字列>

処理対象をジョブ名またはジョブ名ワイルドカードで指定する。この場合、-Jオプションの指定が必要である。

このオプションの指定形式には次に示す2つの方法がある。なお、両方の指定形式は混在できない。

- ジョブ名△※

空白で区切って最大 32 個指定できる。

注※ △は空白を示す。

- 文字列*

1 回だけ指定できる。

ジョブ名ワイルドカードは*を指定したワイルドカード指定で、先頭の文字列が一致するジョブ名を一括で指定できる。ワイルドカード指定をする場合は、シェルが自動的に処理しないように""で囲む必要がある。

ジョブ名を指定した場合、-f オプション指定以外では出力保留状態のジョブだけが表示対象となる。

ジョブ名を省略した場合、ユーザが操作できるスプールジョブを対象とする。

戻り値

戻り値	意味
0	正常終了
0 以外	エラー終了

注意事項

- このコマンドは共用資源の排他を行っており、処理途中で止まるような使用をした場合 (pg コマンドなどでメッセージ表示の途中で止める場合など)、ほかのコマンドやスプールジョブ転送デーモンが待ち状態になることがあるので、そのような使用をしないこと。
- ジョブ識別子の範囲指定をした場合、その範囲内の操作できるジョブを表示する。操作できないジョブはエラーメッセージを出力しない。ジョブ識別子を指定したコマンドを使用した場合は、エラー情報を出力する。
- ジョブ名またはジョブ名ワイルドカード指定をした場合、操作できるジョブの中から指定したジョブ名に該当するジョブを表示する。操作できないジョブは、エラーメッセージを出力しない。ジョブ識別子を指定したコマンドを使用する場合はエラー情報を出力する。

使用例

- スプールジョブ一覧を 1970 年 1 月 1 日からの通算秒での時刻指定で簡易表示する。

```
bjexlsjob -t
```

出力例

```
JOBID S START TIME END TIME JOBNAME
000003 H 1165291250 1165291320 JOB0030
000004 E -----
000005 H 1165295250 1165295320 JOB0005
000008 H 1165290250 1165300320 JOB0005
000009 H 1165490250 1165490320 JOB0005
000029 H 1170437793 1170437794 JOB0001
000123 H 1170442872 1170442872 JOB001A
000125 H 1170442899 1170442899 JOB001A001
000227 H 1170442999 1170442999 JOB001A001
000231 E -----
000300 X -----
000301 P -----
```

出力項目の説明

- **JOBID**
ジョブ識別子。
- **S**
スプールジョブの状態。

状態	説明
B	スプールジョブはほかで処理中であり、正しく表示できていないことを示す。
H	出力保留状態であることを示す。
E	エラー状態（スプールジョブの削除に失敗して、異常状態でファイルが残っている状態）であることを示す。
X	実行中であることを示す。
P	スプールジョブ転送デーモンでスプールジョブを処理中であることを示す。通常は一定時間後に出力保留状態に移行する。運用中に設定ファイルの出力クラスの設定を変更したり、スプールジョブ転送デーモンの処理中に障害が発生した場合、この状態のままとなることがある。この場合、bjexchjob コマンドの-S オプションで回復はできるが、上記の要因が解決しないかぎり、プリントサーバには転送できない。
-	スプールジョブ転送デーモンが停止したなどの原因でスプールジョブ転送サービスの管理から漏れている状態であることを示す。この場合、bjexchjob コマンドの-S オプションでスプールジョブの回復処理をする必要がある。
?	不当な状態であることを示す。スプール管理ファイルが破壊されている可能性がある。通常は運用を停止し、スプール管理ファイルを削除後にスプールジョブ転送デーモンを起動して bjexchjob コマンドの-S オプションでスプールジョブの回復処理をする必要がある。特定ジョブだけでそのスプールジョブが存在する場合には、bjexchjob コマンドの-S オプションでそのスプールジョブだけを回復処理できる場合もある。

- **START TIME, END TIME**
ジョブの実行開始時刻および終了時刻。
- **JOBNAME**
バッチジョブ実行システムのジョブ定義 XML ファイルで指定したジョブ名称を表示する。削除処理中などのタイミングによってジョブ名が求まらない場合は、 "-" (ハイフン) 8 けたで表示する。
- スプールジョブ一覧をローカルタイムでの時刻指定で簡易表示する。

```
bjexlsjob -T
```

出力例

```
JOBID S START TIME      END TIME      JOBNAME
000003 H 2006/12/04 22:00:50 2006/12/04 22:02:00 JOB0030
000004 E ----/--/-- --:--:-- ----/--/-- --:--:-- -----
000005 H 2006/12/04 23:07:30 2006/12/04 23:08:40 JOB0005
000008 H 2006/12/04 21:44:10 2006/12/05 00:32:00 JOB0005
000009 H 2006/12/07 05:17:30 2006/12/07 05:18:40 JOB0005
000029 H 2007/02/02 11:36:33 2007/02/02 11:36:34 JOB0001
000123 H 2007/02/02 13:01:12 2007/02/02 13:01:12 JOB001A
000125 H 2007/02/02 13:01:39 2007/02/02 13:01:39 JOB001A001
000227 H 2007/02/02 13:03:19 2007/02/02 13:03:19 JOB001A001
000231 E ----/--/-- --:--:-- ----/--/-- --:--:-- -----
000300 X ----/--/-- --:--:-- ----/--/-- --:~:~:~ -----
000301 P ----/--/-- --:~:~:~ ----/--/-- --:~:~:~ -----
```

- スプールジョブ一覧を時刻指定なしで簡易表示する。

```
bjexlsjob
```

出力例

```
JOBID S JOBNAME
000003 H JOB0030
000004 E -----
000005 H JOB0005
000008 H JOB0005
000009 H JOB0005
000029 H JOB0001
000123 H JOB001A
000125 H JOB001A001
000227 H JOB001A001
000231 E -----
000300 X -----
000301 P -----
```

- スプールジョブ一覧をローカルタイムでの時刻指定で詳細表示する。

```
bjexlsjob -lT
```

出力例

```
JOBNAME=JOB0030                TIME=2006/12/04 22:00:50 2006/12/04 22:02:00
JOBID  ssshk DD NAME            SYSOUT CLASS
000003 ---HK JOBLOG             A
000003 ---HK XML                A
000003 ---HK SYMSG              A
000003 001HK STDOUT             A
000003 001HK STDERR             A
000003 001HK SYSUT2             A
000003 001HK SYSUT1             A
000003 001HK SYSPRINT           A
000003 002HK STDOUT             A
000003 002HK STDERR             A
000003 002HK SYSUT2             A
000003 002HK SYSPRINT           A
JOBNAME=JOB0005                TIME=2006/12/04 23:07:30 2006/12/04 23:08:40
JOBID  ssshk DD NAME            SYSOUT CLASS
000005 ---HK JOBLOG             A
000005 ---HK XML                A
000005 ---HK SYMSG              A
000005 001HK STDOUT             A
```

出力項目の説明

- **JOBNAME**

ジョブ名。

- **TIME**

ジョブ実行開始時刻，終了時刻。

- **JOBID**

ジョブ識別子。

- **SSS**

ジョブステップ番号を3けたの数字で表示する。ジョブステップに依存しないSYSOUTの場合は"-"（ハイフン）を表示する。ジョブステップ番号はジョブステップが終了した順番に与えた番号であり，ジョブ定義XMLファイルで定義した順番とは一致しない。並列実行するジョブステップの場合，ジョブ定義XMLファイルの定義順とは一致しないことがある。

- **h**

出力保留属性の表示をする。

属性	説明
H	出力保留属性であることを示す。
N	出力非保留属性であることを示す。

- **k**

出力後保存属性を表示する。

属性	説明
K	出力後保存属性であることを示す。
N	出力後保存属性でないことを示す。

属性	説明
-	属性が存在しないことを示す。

- DD NAME
DD 名。
- SYSOUT CLASS
出力クラス名。
- スプールジョブ一覧の詳細をヘッダ情報を抑止して表示する。

```
bjexlsjob -ln
```

出力例

```
000003 ---HK JOBL0G          A
000003 ---HK XML            A
000003 ---HK SYSMSG         A
000003 001HK STD0UT         A
000003 001HK STDERR         A
000003 001HK SYSUT2         A
000003 001HK SYSUT1         A
000003 001HK SYSPRINT       A
000003 002HK STD0UT         A
000003 002HK STDERR         A
000003 002HK SYSUT2         A
000003 002HK SYSPRINT       A
000005 ---HK JOBL0G          A
000005 ---HK XML            A
000005 ---HK SYSMSG         A
000005 001HK STD0UT         A
```

- ジョブ識別子 10 から 100 までのスプールジョブ一覧を表示する。

```
bjexlsjob 10:100
```

出力例

```
JOBID S JOBNAME
000025 H JOB002500A
000042 E -----
000088 H JOB00A00B00C
000100 H JOB100A
```

- ジョブ名が JOB001, JOB002 のジョブを表示する。

```
bjexlsjob -J JOB001 JOB002
```

出力例

```
JOBID S JOBNAME
000015 H JOB002
000142 H JOB001
000181 H JOB001
000200 H JOB001
```

- ジョブ名の先頭が KEIRI のジョブを表示する。

```
bjexlsjob -J "KEIRI*"
```

出力例

```
JOBID S JOBNAME  
000034 H KEIRI  
000128 H KEIRI001  
000141 H KEIRI20559  
000262 H KEIRI01A
```

bjexlslock (ファイルの排他状態の表示 (一般ユーザ用))

形式

```
bjexlslock [-h 論理ホスト名]
            [-l | -w]
            [-j ジョブ識別子 | -t タイプ | パス名...]
```

機能

ジョブコントローラから実行されたジョブの排他/共用情報を表示する。

指定できるオプションおよびパス名のどちらも指定しない場合、ジョブの排他状態を表示する。

オプション

-h 論理ホスト名 ~<英字と数字と- (ハイフン) >((1~244))

バッチジョブ実行システムの論理ホストで運用している場合に指定する。設定ファイルとして/opt/hitachi/bjex/conf/bjex_論理ホスト名.conf を読み込み実行する。

-l

確保している資源の一覧を表示する。

-w

待ち状態になっている資源の一覧を表示する。

-j ジョブ識別子 ~<数字>((1~999999))

表示対象とするジョブのジョブ識別子を指定する。

-t タイプ

表示対象とするロックのタイプを指定する。指定できるタイプを次に示す。

- DIR (LIB も DIR で指定する)
- FILE
- GDG
- PREST

パス名 ~<パス名>

表示対象とするロックのパス名を指定する。

戻り値

戻り値については、「表 10-39 ジョブコントローラの終了コード一覧」を参照のこと。

注意事項

- パス名に相対パス名を指定した場合は、コマンド実行時のカレントディレクトリからの相対パスであると見なす。コマンド実行時のカレントディレクトリがジョブ実行時のカレントディレクトリと異なる場合、ジョブ定義 XML ファイルの DSN 属性値に記述された相対パスを指定しても同じファイルとは見なされないので注意すること。
- パス名は、絶対パス名に変換してから比較する。このため、パス名を相対パスで指定した場合でも、ジョブ定義 XML ファイルの DSN 属性値のパス名と絶対パスが一致していれば、同じファイルと見なす。
- 絶対パス名が同じであるパス名を複数指定した場合は、同じファイルの排他状態が複数回出力される。
- パス名に指定したパスの各ディレクトリは、すべて存在している必要がある。ファイルは存在していなくてもよい。
- ジョブ識別子で指定したジョブが実行中でない場合は、メッセージを出力して、排他情報は出力しないで終了する。
- パス名を指定した場合に、パス名がディレクトリであるとき、ディレクトリ配下のすべての排他情報を出力する。
- このコマンドで排他状態を表示させる場合、排他/共用制御のための共用メモリ使用量定義で見積もった共用メモリサイズを確保できるだけのデータ領域でのリミット値の設定が必要となる。

使用例

現在待ち状態になっている資源の一覧を表示する。

```
bjexlslock -w
```

出力例

- ジョブ識別子もパス名も指定しない場合

```
>bjexlslock
JOBID STATUS ELAPSED PID JP1JobID JOBNAME
-----
000001 LOCK 00:01:00 1001 401 JOB1
000002 WAIT 00:02:00 1002 402 JOB2
000003 LOCK 00:03:00 1003 403 JOB3
```

出力項目の説明

- JOBID**
ジョブ識別子 (6 けた 10 進数)。
- STATUS**
ジョブの排他状態。

項目	説明
LOCK	ロック獲得状態であることを示す。

項目	説明
WAIT	ロック待ち状態であることを示す。

- **ELAPSED**

ロック獲得またはロック待ち開始からの経過時間。

STATUS が"LOCK"の場合は、ロック獲得からの経過時間を表示する。

STATUS が"WAIT"の場合は、ロック待ち開始からの経過時間を表示する。

24 時間を超える場合は、"over24"を表示する。

- **PID**

ジョブコントローラプロセスのプロセス番号 (10 進数)。

- **JP1JobID**

JP1 ジョブ識別子。

- **JOBNAME**

ジョブ名。

- ジョブ識別子 (-j オプション) を指定した場合

```

> bjexlslck -j 1
(1) JOBID STATUS ELAPSED PID JP1JobID JOBNAME
-----
000001 LOCK 00:01:00 1001 401 JOB1

(2) MODE STATUS LOCKJOB TYPE FILEPATH
-----
EXC LOCK SELF FILE /home/userid/exclusiveLockedFile
SHR LOCK SELF.. FILE /home/userid/sharedLockedFile

> bjexlslck -j 2
(1) JOBID STATUS ELAPSED PID JP1JobID JOBNAME
-----
000002 WAIT 00:02:00 1002 402 JOB2

(2) MODE STATUS LOCKJOB TYPE FILEPATH
-----
EXC WAIT 000001 FILE /home/userid/exclusiveLockedFile
EXC WAIT 000001.. FILE /home/userid/sharedLockedFile
SHR UNLK NONE FILE /home/userid/allocateWaitingFile

```

前半 (上記出力例の (1)) は、オプションに指定したジョブの排他状態を表示する。形式は「ジョブ識別子もパス名も指定しない場合」と同じである。

後半 (上記出力例 (2)) は、オプションに指定したジョブが割り当て中または割り当てようとしているファイルの排他状態を表示する。

出力項目の説明

- **MODE**

このファイルまたはディレクトリを割り当てた場合の排他モード。

項目	説明
EXC	排他割り当て要求であることを示す。

項目	説明
SHR	共用割り当て要求であることを示す。

- **STATUS**

このファイルまたはディレクトリのロック状態。

項目	説明
LOCK	ロック獲得状態であることを示す。
WAIT	ロック待ち状態であることを示す。
UNLK	複数ファイルの排他要求をし、そのうちの1つでロック待ちとなった場合に、そのファイル以外のほかのファイルがロック待ちをしている状態であることを示す。

- **LOCKJOB**

このファイルまたはディレクトリをロックしたジョブ識別子およびロック状態。

項目	説明
NONE	どのジョブもこのファイルに対するロックを獲得していないことを示す。
SELF	オプションに指定したジョブがこのファイルに対するロックを獲得したことを示す。
DIRLOCK	上位ディレクトリが排他割り当てまたは共用割り当て中であることを示す。
SHRLOCK	共用割り当て中のジョブが少なくとも1ジョブ以上存在することを示す。
OTHERLK	TYPE="DIR"の指定の場合、ディレクトリ配下のファイルまたはディレクトリが排他割り当てもしくは共用割り当て中、または上位ディレクトリが排他割り当てもしくは共用割り当て中であることを示す。
6けた10進数	このファイルを排他割り当てまたは共用割り当てしたジョブのジョブ識別子であることを示す。

ジョブ識別子または SELF に".."が付く場合は、このファイルを共用割り当て中のジョブがほかにあることを示す。

- **TYPE**

ファイル種別。

項目	説明
FILE	ファイルを示す。
GDG	世代データ群を示す。
PREST	名前付きパイプを示す。
DIR	ディレクトリを示す。

- **FILEPATH**

ファイルまたはディレクトリの絶対パス。

TYPE が"GDG"の場合、世代データ群の絶対パス名および世代データ群名を表示する。

(例) <DD NAME="DD1" DSN="/home/user1/SEDAI(-1)" TYPE="GDG" DISP="SHR"/>

この場合、パス名は"/home/user1/SEDAI"と表示される。

TYPE が PREST の場合、DD 要素の DSN 属性値で指定した PREST 識別名から生成した名前付きパイプの絶対パス名および名前付きパイプ名を表示する。

(例) <DD NAME="DD1" DSN="SHRFILE1" TYPE="PREST" DISP="SHR"/>

この場合、パス名は"一時ファイルディレクトリパス/PREST_SHRFILE1"と表示される。

- パス名を指定した場合

```
> cd /home/userid
> bjexlslock exclusiveLockedFile sharedLockedFile allocateWaitingFile
WAITCT EXCJOB SHRCT TYPE FILEPATH
  1 000001 0 FILE /home/userid/exclusiveLockedFile
  JOBID STATUS MODE
  000001 LOCK EXC
  000002 WAIT SHR
WAITCT EXCJOB SHRCT TYPE FILEPATH
  1 ----- 2 FILE /home/userid/sharedLockedFile
  JOBID STATUS MODE
  000001 LOCK SHR
  000002 WAIT EXC
  000003 LOCK SHR
WAITCT EXCJOB SHRCT TYPE FILEPATH
  0 ----- 0 FILE /home/userid/allocateWaitingFile
  JOBID STATUS MODE
  000002 UNLK SHR
```

出力項目の説明

- **WAITCT**
このファイルまたはディレクトリのロック解除を待っているジョブ数 (10 進表示)。
- **EXCJOB**
このファイルまたはディレクトリを排他割り当てしているジョブのジョブ識別子。
- **SHRCT**
このファイルまたはディレクトリを共用割り当てしているジョブ数 (10 進表示)。
- **TYPE**
ファイル種別。

項目	説明
FILE	ファイルを示す。
GDG	世代データ群を示す。
PREST	名前付きパイプを示す。
DIR	ディレクトリを示す。

- **FILEPATH**
ファイルまたはディレクトリの絶対パス。
TYPE が "GDG" の場合、世代データ群の絶対パス名および世代データ群名を表示する。
(例) <DD NAME="DD1" DSN="/home/user1/SEDAI(-1)" TYPE="GDG" DISP="SHR"/>
この場合、パス名は"/home/user1/SEDAI"と表示される。

TYPE が PREST の場合、DD 要素の DSN 属性値で指定した PREST 識別名から生成した名前付きパイプの絶対パス名および名前付きパイプ名を表示する。

(例) <DD NAME="DD1" DSN="SHRFILE1" TYPE="PREST" DISP="SHR"/>

この場合、パス名は"一時ファイルディレクトリパス/PREST_SHRFILE1"と表示される。

- **JOBID**

このファイルまたはディレクトリを、排他割り当てまたは共用割り当てしているジョブのジョブ識別子。

- **STATUS**

このファイルまたはディレクトリのロック状態。

項目	説明
LOCK	ロック獲得中であることを示す。
WAIT	ロック獲得待ち中であることを示す。
UNLK	複数ファイルの排他要求をし、そのうちの1つでロック待ちとなった場合に、そのファイル以外のほかのファイルがロック待ちをしている状態であることを示す。

- **MODE**

このファイルまたはディレクトリを割り当てた場合の排他モード。

項目	説明
EXC	排他割り当て要求であることを示す。
SHR	共用割り当て要求であることを示す。

- **タイプ (-t オプションで FILE) を指定した場合**

```

> bjexlslock -t FILE
WAITCT  EXCJOB  SHRCT  TYPE  FILEPATH
  1 000001    0 FILE  /home/userid/exclusiveLockedFile
      JOBID  STATUS  MODE
      000001 LOCK   EXC
      000002 WAIT   SHR
WAITCT  EXCJOB  SHRCT  TYPE  FILEPATH
  1 -----    2 FILE  /home/userid/sharedLockedFile
      JOBID  STATUS  MODE
      000001 LOCK   SHR
      000002 WAIT   EXC
      000003 LOCK   SHR
WAITCT  EXCJOB  SHRCT  TYPE  FILEPATH
  0 -----    0 FILE  /home/userid/allocateWaitingFile
      JOBID  STATUS  MODE
      000002 UNLK   SHR

```

出力項目の説明

- **WAITCT**

このファイルまたはディレクトリのロック解除を待っているジョブ数 (10 進表示)。

- **EXCJOB**

このファイルまたはディレクトリを排他割り当てしているジョブのジョブ識別子。

- **SHRCT**

このファイルまたはディレクトリを共用割り当てしているジョブ数（10進表示）。

- **TYPE**

ファイル種別。

項目	説明
FILE	ファイルを示す。
GDG	世代データ群を示す。
PREST	名前付きパイプを示す。
DIR	ディレクトリを示す。

- **FILEPATH**

ファイルまたはディレクトリの絶対パス。

TYPE が"GDG"の場合、世代データ群の絶対パス名および世代データ群名を表示する。

(例) <DD NAME="DD1" DSN="/home/user1/SEDAI(-1)" TYPE="GDG" DISP="SHR"/>

この場合、パス名は"/home/user1/SEDAI"と表示される。

TYPE が PREST の場合、DD 要素の DSN 属性値で指定した PREST 識別名から生成した名前付きパイプの絶対パス名および名前付きパイプ名を表示する。

(例) <DD NAME="DD1" DSN="SHRFILE1" TYPE="PREST" DISP="SHR"/>

この場合、パス名は"一時ファイルディレクトリパス/PREST_SHRFILE1"と表示される。

- **JOBID**

このファイルまたはディレクトリを、排他割り当てまたは共用割り当てしているジョブのジョブ識別子。

- **STATUS**

このファイルまたはディレクトリのロック状態。

項目	説明
LOCK	ロック獲得中であることを示す。
WAIT	ロック獲得待ち中であることを示す。
UNLK	複数ファイルの排他要求をし、そのうちの1つでロック待ちとなった場合に、そのファイル以外のほかのファイルがロック待ちをしている状態であることを示す。

- **MODE**

このファイルまたはディレクトリを割り当てた場合の排他モード。

項目	説明
EXC	排他割り当て要求であることを示す。
SHR	共用割り当て要求であることを示す。

- **タイプ (-t オプションで PREST) を指定した場合**

```

> bjexlslock -t PREST
WAITCT EXCJOB SHRCT PRESTCT TYPE FILEPATH
  0 ----- 2      2 PREST /home/userid/prestFile/PREST_B
      JOBID STATUS MODE
      000001 LOCK  SHR
      000002 LOCK  SHR

```

出力項目の説明

- **WAITCT**
このファイルまたはディレクトリのロック解除を待っているジョブ数（10進表示）。
- **EXCJOB**
このファイルまたはディレクトリを排他割り当てしているジョブのジョブ識別子。
- **SHRCT**
このファイルまたはディレクトリを共用割り当てしているジョブ数（10進表示）。
- **PRESTCT**
この PREST ファイルを共用割り当てしているジョブ数（10進表示）。
- **TYPE**
ファイル種別。

項目	説明
FILE	ファイルを示す。
GDG	世代データ群を示す。
PREST	名前付きパイプを示す。
DIR	ディレクトリを示す。

- **FILEPATH**
ファイルまたはディレクトリの絶対パス。
TYPE が"GDG"の場合、世代データ群の絶対パス名および世代データ群名を表示する。
(例) <DD NAME="DD1" DSN="/home/user1/SEDAI(-1)" TYPE="GDG" DISP="SHR"/>
この場合、パス名は"/home/user1/SEDAI"と表示される。
TYPE が PREST の場合、DD 要素の DSN 属性値で指定した PREST 識別名から生成した名前付きパイプの絶対パス名および名前付きパイプ名を表示する。
(例) <DD NAME="DD1" DSN="SHRFILE1" TYPE="PREST" DISP="SHR"/>
この場合、パス名は"一時ファイルディレクトリパス/PREST_SHRFILE1"と表示される。
- **JOBID**
このファイルまたはディレクトリを、排他割り当てまたは共用割り当てしているジョブのジョブ識別子。
- **STATUS**

このファイルまたはディレクトリのロック状態。

項目	説明
LOCK	ロック獲得中であることを示す。
WAIT	ロック獲得待ち中であることを示す。
UNLK	複数ファイルの排他要求をし、そのうちの1つでロック待ちとなった場合に、そのファイル以外のほかのファイルがロック待ちをしている状態であることを示す。

- **MODE**

このファイルまたはディレクトリを割り当てた場合の排他モード。

項目	説明
EXC	排他割り当て要求であることを示す。
SHR	共用割り当て要求であることを示す。

bjexlsmg (応答待ちメッセージおよび受信待ちメッセージ一覧の表示 (管理者用))

形式

```
bjexlsmg [-h 論理ホスト名] [-n メッセージ番号]
```

機能

応答待ちメッセージおよび受信待ちメッセージの一覧を表示する。

オプション

-h 論理ホスト名 ~<英字と数字と- (ハイフン) >((1~244))

バッチジョブ実行システムの論理ホストで運用している場合に指定する。設定ファイルとして/opt/hitachi/bjex/conf/bjex_論理ホスト名.confを読み込み実行する。

-n メッセージ番号 ~<数字>((1~2147483647))

指定したメッセージ番号の応答待ちイベントを表示する。

戻り値

戻り値	意味
0	正常終了
0以外	エラー終了

使用例

- 応答待ちメッセージおよび受信待ちメッセージの一覧ならびにメッセージ番号を詳細表示する。

```
bjexlsmg
```

- メッセージ番号「2000000000」の応答待ちメッセージおよび受信待ちメッセージの一覧を表示する。

```
bjexlsmg -n 2000000000
```

出力例

応答待ちメッセージおよび受信待ちメッセージの一覧ならびにメッセージ番号を詳細表示する場合の出力例を次に示す。

```

$ bjexlsmg
MESSAGE-NO STATUS JOBID STEP DATE/TIME MESSAGE or RESPONSE
[0001994844] [Wait] 001673 001 07/12/22 00:01:29 msg=KCCG2003R-I STOP (001673, 001, host1,
1198043673, 0)
[0002298002] [Set ] 001674 001 07/12/22 00:07:03 msg=KCCG2000R-I FROM CONSOLE指定のACCEPT文を実行
しました。応答してください。プログラム名=CONS (001674, 001, host1, 1198044152, 0)
[0002298002] [Set ] 001674 001 07/12/22 00:07:03 res=Y

```

出力項目の説明

- **MESSAGE-NO**

10 けたの 10 進数値のメッセージ番号。bjexlsmg コマンドの -n オプションおよび bjexchmsg コマンドで指定する値である。

- **STATUS**

メッセージの状態。

Wait：応答待ちメッセージ

Set：受信待ちメッセージ

- **JOBID**

6 けたのジョブ識別子。

- **STEP**

3 けたのステップ番号。

- **DATE/TIME**

応答要求メッセージを出力した時刻（ローカルタイム）。

- **MESSAGE or RESPONSE**

メッセージ本体または応答されたレスポンス。

msg=：メッセージ本体

res=：応答されたレスポンス（受信待ちメッセージの場合だけこの行が表示される）

bjexm_message (メッセージまたは応答要求メッセージの出力 (一般ユーザ用))

形式

```
bjexm_message -p メッセージ  
bjexm_message -m メッセージ  
bjexm_message {-r | -c} 応答要求メッセージ 応答0 [応答1 [応答n] ... ]
```

機能

bjexm_message はジョブステップとして起動することで、メッセージ制御機能を使用して JP1/IM - View の画面にメッセージや応答要求メッセージを出力し、応答メッセージを取得する。また、応答メッセージに従って戻り値を設定して終了する。

戻り値を判定して実行するジョブステップを制御することで、オペレータからの応答に応じて実行するジョブステップを選択できる。

オプション

-p : PAUSE 要求

メッセージを出力し、コマンドは応答入力待ちになる。処理を続行するには何らかのメッセージを応答する。応答メッセージを入力しないで、応答だけでもよい。

応答入力後、コマンドは戻り値 0 で終了する。

-m : WRITE 要求

指定したメッセージを出力する。

コマンドは戻り値 0 で終了する。

-r : WRITER 要求 (応答の再入力なし)

応答要求メッセージを出力し、応答 n で指定したメッセージの応答を待つ。処理を続行するには応答 n で指定した応答メッセージを応答する。

コマンドは応答メッセージに対応した戻り値で終了する。指定した応答メッセージに該当しない応答メッセージを応答した場合、コマンドは戻り値 127 で終了する。

-c : WRITERX 要求 (応答の再入力あり)

応答要求メッセージを出力し、応答 n で指定されたメッセージの応答を待つ。処理を続行するには応答 n で指定した応答メッセージを応答する。

コマンドは応答メッセージに対応した戻り値で終了する。

指定した応答メッセージに該当しない応答メッセージを応答した場合、コマンドは KAKC5208-E メッセージを出力したあと、再び応答要求メッセージを出力して、応答待ちになる。このメッセージは LANG 環境変数が次に示す値のとき日本語メッセージを出力する。

次に示す LANG 環境変数以外の場合、英語のメッセージ (エンコードは ASCII) を出力する。

AIX : Ja_JP (メッセージのエンコードはシフト JIS となる)

ja_JP (メッセージのエンコードは EUC となる)

RHEL : ja_JP.UTF-8 (メッセージのエンコードは UTF-8 となる)

メッセージ ~((1~512 バイト))

JP1/IM - View の画面に出力するメッセージを指定する。

メッセージ出力後コマンドは戻り値 0 で終了する。

メッセージ長の上限は 512 バイトであり、513 バイト以上を指定すると切り捨てられる。

また、日本語で指定する場合は LANG 環境変数で示される言語環境と一致するエンコードで指定する必要がある。また、複数バイトで表現される文字の途中で打ち切られた場合の動作は保証しないので注意が必要である。

メッセージ長が 0 の場合やメッセージの指定がない場合は、KAKC0311-E メッセージを出力して終了する。

応答要求メッセージ ~((1~512 バイト))

JP1/IM - View の画面に出力する応答要求メッセージを指定する。応答メッセージ出力後は応答入力待つ。

メッセージ長の上限は 512 バイトであり、513 バイト以上を指定すると切り捨てられる。

また、日本語で指定する場合は LANG 環境変数で示される言語環境と一致するエンコードで指定する必要がある。また、複数バイトで表現される文字の途中で打ち切られた場合の動作は保証しないので注意が必要である。

メッセージ長が 0 の場合やメッセージの指定がない場合は、KAKC0311-E メッセージを出力して終了する。

応答 n ~((1~512 バイト))

応答メッセージを指定する。n は 0 から 31 までであり、応答はスペースで区切って最大 32 個指定できる。

応答要求メッセージに対して応答できるメッセージを定義し、応答されたメッセージに対応した戻り値でこのコマンドは終了する。

例えば、応答 0 を応答した場合は戻り値 0 となり、応答 1 を応答した場合には戻り値 1 となる。つまり、指定順に対応した戻り値となる。

応答 n に指定されたすべてが、Null 文字列の場合、応答メッセージの内容に関係なく、戻り値 0 で終了する。

応答を 33 個以上指定した場合、33 個以上は無視する。

応答メッセージのメッセージ長の上限は 512 バイトであり、513 バイト以上を指定すると無視される。

また、ASCII の範囲の文字だけ動作を保証し、日本語は使用できない。ASCII 以外の文字を指定してもエラーにはならないが、ASCII の文字だけが応答できる。

同一の応答メッセージを複数回指定してもエラーにはならないが、最初に記述した応答メッセージの戻り値で終了する。

戻り値

戻り値	意味
0	メッセージ出力が完了した。または、応答要求メッセージに対して応答 0 の応答があった。 PAUSE 機能の応答操作があった。
1~31	応答要求メッセージに、戻り値に対応する応答があった。
127	予期しない障害が発生した。標準エラー出力に事象を示すエラーメッセージを出力する。 -r を指定してコマンドを実行した場合に定義した応答メッセージ以外の応答があった。 コマンドの指定が正しくない。

注意事項

- このコマンドは uCosminexus Batch Job Execution Server の配下だけで使用できる。その他の場合はエラーで終了する。(KAKC5205-E)
- このコマンドが出力するメッセージの後ろにはメッセージ制御機能の仕様に従って次の情報が付加される。ただし、標準エラー出力に出力するエラーメッセージなどの場合は、情報は付加されない。
(ジョブ識別子、ステップ番号、ホスト名、メッセージ識別子、保守情報)
- このコマンドが出力するメッセージはバッチジョブ実行システムの MESSAGE_REPLACE_SPACE 初期設定パラメータの置き換えの対象になる。ただし、標準エラー出力に出力するエラーメッセージなどの場合は、置き換えの対象にならない。
- ジョブ定義 XML の EXEC 要素の PARM 属性には 958 文字までの制限があり、その範囲で指定すること。
また、空白を含むメッセージや応答は PARM では指定できない。このため、空白を含むメッセージや応答を使用したい場合は PGM="*" でスクリプトに記述する必要がある。

使用例

KEY_IN のメッセージを出力して、Yes または No の応答を受け取る。

ジョブステップは Yes を応答すると rc=0 で終了し、No を応答すると rc=1 で終了する。また、応答内容に応じたメッセージを出力する。

コマンドの戻り値をジョブの戻り値に反映したくない場合は SETMAXRC 要素で戻り値をリセットする必要がある。

ジョブ定義 XML ファイルの例を次に示す。

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version="1.4" os="unix" >
  <JOB NAME="Sample">
    <STEP NAME="STEP01" >
      <EXEC PGM="/opt/hitachi/bjex/bin/bjexm_message"
          PARM="-c KEY_IN Yes No"
          PARMSPLIT="YES" >
      </EXEC>
```

```

</STEP>
<STEP NAME="STEP02" >
  <EXEC PGM="/opt/hitachi/bjex/bin/bjexm_message"
    PARM="-m PROGRAM_IS_RC=0[Yes]" COND="0,NE,STEP01"
    PARMSPLIT="YES" >
  </EXEC>
</STEP>
<STEP NAME="STEP03" >
  <EXEC PGM="/opt/hitachi/bjex/bin/bjexm_message"
    PARM="-m PROGRAM_IS_RC_not_0[no]" COND="0,EQ,STEP01"
    PARMSPLIT="YES" >
  </EXEC>
</STEP>
</JOB>
</HitachiBatchJobExec>

```

実行結果

上記の使用例の実行結果を次に示す。

```

***** BJEX MESSAGE *****
10:41:44 000129 KAKC0091-I Sample Job started.
10:41:44 000129 KAKC0092-I Sample.STEP01 Step started.
10:41:45 000129 KAKC5013-I KEY_IN (000129, 001, io10288, 1288143705, 0)
10:42:12 000129 KAKC5001-I XX (000129, 001, io10288, user01)
10:42:12 000129 KAKC5012-I KAKC5208-E 応答されたメッセージは未定義です。再度応答してください。
(000129, 001, io10288, 1288143732, 0)
10:42:12 000129 KAKC5013-I KEY_IN (000129, 001, io10288, 1288143732, 0)
10:42:42 000129 KAKC5001-I No (000129, 001, io10288, user01)
10:42:42 000129 KAKC0097-I Sample.STEP01 Step ended. rc=1 E-Time=57.787s C-Time=0.010s
10:42:42 000129 KAKC1907-I Sample.STEP02 Step was skipped because the value of COND attribute
of EXEC element was met.
10:42:42 000129 KAKC0092-I Sample.STEP03 Step started.
10:42:43 000129 KAKC5012-I PROGRAM_IS_RC_not_0[no] (000129, 003, io10288, 1288143763, 0)
10:42:43 000129 KAKC0097-I Sample.STEP03 Step ended. rc=0 E-Time=0.398s C-Time=0.030s
10:42:43 000129 KAKC0098-I Sample Job ended. rc=1 E-Time=58.185s C-Time=0.040s MaxRSS=2820KB

```

```

***** XML IMAGE *****
000001 <?xml version="1.0" encoding="Shift-JIS" ?>
000002 <HitachiBatchJobExec version="1.4" os="unix" >
000003 <JOB NAME="Sample">
000004 <STEP NAME="STEP01" >
000005 <EXEC PGM="/opt/hitachi/bjex/bin/bjexm_message"
    PARM="-c KEY_IN Yes No"
    PARMSPLIT="YES" >
000006 </EXEC>
000007 </STEP>
000008 <STEP NAME="STEP02" >
000009 <EXEC PGM="/opt/hitachi/bjex/bin/bjexm_message"
    PARM="-m PROGRAM_IS_RC=0[Yes]" COND="0,NE,STEP01"
    PARMSPLIT="YES" >
000010 </EXEC>
000011 </STEP>
000012 <STEP NAME="STEP03" >
000013 <EXEC PGM="/opt/hitachi/bjex/bin/bjexm_message"
    PARM="-m PROGRAM_IS_RC_not_0[no]" COND="0,EQ,STEP01"
    PARMSPLIT="YES" >
000014 </EXEC>
000015 </STEP>
000016 </JOB>
000017 </HitachiBatchJobExec>

```

***** SYSTEM MESSAGE *****

```
10:41:44 000129 KAKC0900-I Parsing of xml file "/home/user01/sample_msg.xml" started.
10:41:44 000129 KAKC0909-I Parsing of xml file "/home/user01/sample_msg.xml" ended. code=0
10:41:44 000129 KAKC0707-I MSGCLASS is "A".
10:41:44 000129 KAKC1600-I Sample Job allocated.
10:41:45 000129 KAKC1601-I Sample.STEP01 Step allocated.
10:41:45 000129 KAKC5013-I KEY_IN (000129, 001, io10288, 1288143705, 0)
10:42:12 000129 KAKC5012-I KAKC5208-E 応答されたメッセージは未定義です。再度応答してください。
(000129, 001, io10288, 1288143732, 0)
10:42:12 000129 KAKC5013-I KEY_IN (000129, 001, io10288, 1288143732, 0)
10:42:43 000129 KAKC1601-I Sample.STEP03 Step allocated.
10:42:43 000129 KAKC5012-I PROCGRAM_IS_RC_not_0[no] (000129, 003, io10288, 1288143763, 0)
```

bjexpd (スプールジョブ転送デーモン (管理者用))

形式

```
bjexpd { [-h 論理ホスト名] [-c [-p 設定ファイル名]] | -v }
```

機能

bjexpdctl コマンドから呼び出されるスプールジョブ転送サービスのメインプロセスである。スプールジョブ転送デーモンを起動する場合は、bjexpdctl コマンドを使用する。

オプション

-h 論理ホスト名 ~<英字と数字と- (ハイフン) >((1~244))

バッチジョブ実行システムの論理ホストで運用している場合に指定する。設定ファイルとして/opt/hitachi/bjex/conf/bjex_論理ホスト名.conf を読み込み実行する。

-c

設定ファイルのパラメータの文法チェックをする。結果は標準出力に出力する。

このオプションを指定した場合、設定ファイルのパラメータの文法チェックだけを行い、スプールジョブ転送デーモンは起動しない。

-p 設定ファイル名 ~<パス名>

文法チェックする初期設定パラメータファイル名を指定する。スプールジョブ転送デーモンは起動しない。

このオプションは、-c オプションとともに指定する。

-v

スプールジョブ転送デーモンのバージョンを表示する。

このオプションを指定した場合、スプールジョブ転送デーモンのバージョンの表示だけを行い、起動はしない。

戻り値

戻り値	意味
0	正常終了
0 以外	エラー終了

注意事項

- スプールジョブ転送デーモンは root で起動すること。
- スプールジョブ転送サービス起動中は、設定ファイルを変更しないこと。

- スプールジョブ転送デーモンは処理内容や結果を bjex.conf の LOG_DIR ディレクトリ内のログファイルに保存している。ファイルが存在するファイルシステムの利用率が 100%になるなどの要因でログファイルへの書き込み失敗が発生すると、スプールジョブ転送デーモンはスプール転送処理を停止する。スプールの転送処理を再開させるには、bjexpdctl コマンドを使ってスプールジョブ転送デーモンを再起動（スプールジョブ転送デーモンを停止してから起動）すること。
- 論理ホストの設定ファイルを文法チェックする場合、-c オプションより前に-h オプションを指定すること。
- -h オプションと-p オプションの両方が指定された場合、-p オプションで指定した設定ファイルを文法チェックする。
- **【AIX 限定】**
SIGTERM および SIGDANGER 受信時に終了する。
- **【Linux 限定】**
SIGTERM 受信時に終了する。

bjexpdctl (スプールジョブ転送デーモンの起動, 停止 (管理者用))

形式

```
bjexpdctl [-h 論理ホスト名] { start | stop [-f] | status | status_log | conftest | ftp_connect | help }
```

機能

スプールジョブ転送デーモンを起動, 停止する。

オプション

-h 論理ホスト名 ~<英字と数字と- (ハイフン) >((1~244))

バッチジョブ実行システムの論理ホストで運用している場合に指定する。設定ファイルとして/opt/hitachi/bjex/conf/bjex_論理ホスト名.conf を読み込み実行する。

start

スプールジョブ転送デーモンを起動する。

stop

スプールジョブ転送デーモンを停止する。転送中のプロセスが終了してから, スプールジョブ転送デーモンを停止する。ただし, 実行中のジョブがある場合にはスプールジョブ転送デーモンを停止しない。

-f

スプールジョブ転送デーモンを強制停止する。実行中のジョブがある場合でも強制的に停止する場合に使用する。

status

スプールジョブ転送デーモンの動作状態を表示する。

スプールジョブ転送デーモンが動作している場合は, 終了コードに 0, スプールジョブ転送デーモンが動作していない場合は, 終了コードに 1 を応答する。

スプールジョブ転送デーモンが動作しており, スプールジョブ転送プロセスが JPI/FTP とのコネクションを確立できずに待ち状態にある場合, 終了コードに 0 を応答する。

status_log

スプールジョブ転送デーモンの実行ログ出力状態を表示する。

スプールジョブ転送デーモンの実行ログ出力が動作している場合は, 終了コードに 0, スプールジョブ転送デーモンが動作していない場合は, 終了コードに 1, スプールジョブ転送デーモンは動作していて, かつ実行ログ出力が停止している場合は, 終了コードに 2 を応答する。

conftest

設定ファイルのパラメータをチェックする。結果は標準出力に出力する。

ftp_connect

スプールジョブ転送デーモンが、JP1/FTP との接続を確立できずにリトライ待ち状態にあるとき、待ち状態を解除する。待ち状態が解除されると、接続を確立する処理を再実行する。スプールジョブ転送デーモンが JP1/FTP との接続生成待ち状態でない場合は、bjexpd が動作中である旨のメッセージを出力する。

help

bjexpdctl コマンドのヘルプを表示する。

戻り値

戻り値	意味
0	正常終了
0 以外	エラー終了

注意事項

- スプールジョブ転送デーモンは root で起動すること。
- スプールジョブ転送サービス起動中は、設定ファイルを変更しないこと。
- -h オプションを指定する場合は、最初に指定すること。
- stop -f オプションでスプールジョブ転送デーモンを強制終了した場合、その間に実行中であったジョブは異常終了することがある。

使用例

- スプールジョブ転送デーモンを起動する。

```
bjexpdctl start
```

- スプールジョブ転送デーモンを停止する。

```
bjexpdctl stop
```

- バッチジョブ実行システムの設定ファイルをチェックする。

```
bjexpdctl conftest
```

bjexrmjob (HOLD 状態またはエラー状態のスプールジョブの削除 (一般ユーザ用))

形式

日数または時間数およびジョブ識別子で指定する場合

```
bjexrmjob [-h 論理ホスト名]
           {-T 日数 | -H 時間数} [-f[-a 秒数]]
           [ジョブ識別子… | ジョブ識別子:ジョブ識別子]
```

ジョブ識別子で指定する場合

```
bjexrmjob [-h 論理ホスト名]
           [-f[-a 秒数]]
           { ジョブ識別子… | ジョブ識別子:ジョブ識別子 }
```

日数または時間数およびジョブ名で指定する場合

```
bjexrmjob [-h 論理ホスト名]
           {-T 日数 | -H 時間数} [-f[-a 秒数]]
           -J
           [ジョブ名… | ジョブ名ワイルドカード]
```

ジョブ名で指定する場合

```
bjexrmjob [-h 論理ホスト名]
           [-f[-a 秒数]]
           -J
           { ジョブ名… | ジョブ名ワイルドカード }
```

機能

スプールジョブを削除する。

強制削除ではスプールジョブを削除し、そのディレクトリとともに削除する。

強制削除を指定しない場合、出力保留状態のスプールジョブを削除する。スプールジョブのディレクトリ下に SYSOUT データ、スプールジョブ管理ファイル以外を不当に作成している場合、そのファイルは削除しない。このようなスプールジョブは削除できないため、エラー状態に移行する。強制削除を指定した場合、不当なファイルがあっても削除する。

エラー状態のスプールジョブを削除する場合は、強制削除の指定をする。この場合、スプールジョブのディレクトリ下のファイルはすべて削除する。

オプション

-h 論理ホスト名 ~<英字と数字と- (ハイフン) >((1~244))

バッチジョブ実行システムの論理ホストで運用している場合に指定する。設定ファイルとして/opt/hitachi/bjex/conf/bjex_論理ホスト名.conf を読み込み実行する。

-T 日数 ～< 1～2 けたの 10 進数>((0-99))

指定日数以前の日に実行したジョブのプールジョブを削除する。実行した日の判定はジョブ実行開始時刻で判断する。

強制削除でジョブ実行開始時刻が求まらない場合は、プールジョブファイルまたはディレクトリの最終更新時刻で判定する。2月2日に-T1を指定した場合、2月1日 23:59:29 以前が対象となる。

-H 時間数 ～< 1～2 けたの 10 進数>((0-96))

指定時間数以前の時刻に実行したジョブのプールジョブを削除する。実行した時刻の判定はジョブ実行開始時刻で判断する。

2月2日 10:15:00 に-H1を指定した場合、2月2日 09:14:59 以前が対象となる。

-f

このオプションを指定した場合、プールジョブの状態に関係なくプールジョブディレクトリがあればプールジョブを削除する。実行中のジョブに対して指定しないように注意が必要であり、一般的には障害などで管理から漏れたプールジョブまたはエラー状態のプールジョブの削除に使用する。

エラー状態のプールジョブはSYSOUT データファイル以外のファイルを作成し、プールジョブのディレクトリが削除できない状態のプールジョブである。強制終了時にはこれらのSYSOUT データファイル以外のファイルも消去してプールジョブを削除する。

このオプションを省略した場合、出力保留状態のプールジョブを削除する。この場合、SYSOUT データファイル以外のファイルがプールディレクトリに存在すると、そのファイルは削除しないで、プールジョブはエラー状態となる。

-a 秒数 ～< 1～3 けたの 10 進数>((0-999)) 《30》

ジョブの強制削除時に有効な値であり、プールディレクトリ、プールジョブ管理ファイルの最終更新時刻が現在から指定秒数前の場合、ジョブ実行中として削除対象としない。

プールジョブファイル、ディレクトリの最終更新時刻はジョブ実行時とは異なりプールジョブの操作を行うと更新される。例えば、bjexchjob コマンドで属性変更すると更新される。

-J

処理対象をジョブ名で指定する場合に指定する。この指定がない場合、ジョブ識別子オプションで指定する。

ジョブ識別子 ～< 10 進数>((1～999999))

削除するジョブ識別子を指定する。

-T オプションや-H オプションを指定しない場合、ジョブ識別子、ジョブ名、またはジョブ名ワイルドカードのどれかの指定が必要である。

このオプションの指定形式には次に示す 2 つの方法がある。なお、両方の指定形式は混在できない。

- ジョブ識別子△※

空白で区切って複数指定できる。

注※ △は空白を示す。

- ジョブ識別子：ジョブ識別子

1 回だけ指定できる。

"ジョブ識別子：ジョブ識別子"の形式で指定した場合、左に指定したジョブ識別子から右に指定したジョブ識別子までの範囲を処理対象とする。

また、左のジョブ識別子は右のジョブ識別子よりも等しいかまたは小さくする必要がある。

ジョブ名 ～<記号名称>，ジョブ名ワイルドカード ～<任意文字列>

処理対象をジョブ名またはジョブ名ワイルドカードで指定する。この場合、-J オプションの指定が必要である。

このオプションの指定形式には次に示す 2 つの方法がある。なお、両方の指定形式は混在できない。

- ジョブ名△※

空白で区切って最大 32 個指定できる。

注※ △は空白を示す。

- 文字列*

1 回だけ指定できる。

ジョブ名ワイルドカードは*を指定したワイルドカード指定で、先頭の文字列が一致するジョブ名を一括で指定できる。ワイルドカード指定をする場合は、シェルが自動的に処理しないように""で囲む必要がある。

ジョブ名またはジョブ名ワイルドカードを省略した場合は、すべてのユーザが操作できるスプールジョブを対象とする。

戻り値

戻り値	意味
0	正常終了
0 以外	エラー終了

注意事項

- 特定ユーザのスプールジョブだけを削除する場合は、削除する対象のユーザで実行すること。
- このコマンドは共用資源の排他を行っており、処理途中で止まるような使用をした場合 (pg コマンドなどでメッセージ表示の途中で止める場合など)、ほかのコマンドやスプールジョブ転送デーモンが待ち状態になることがあるので、そのような使用をしないこと。
- コマンドを割り込みなどで強制終了させないこと。コマンドを強制終了させた場合、スプールジョブ管理ファイルが破壊されることがある。
- このコマンド実行にはスプールジョブ転送デーモンを起動している必要がある。スプールジョブ転送デーモンが起動していないとき、またはスプールジョブ転送デーモンが処理途中で終了したとき、コマンドの動作は保証できないため、コマンドが待ち状態になり、終了しないことがある。また、処理途中のスプールジョブが完全には削除されないことがある。このような場合、コマンドを割り込みなどで停止する。また、削除されなかったジョブは bjexrmjob コマンドの -f オプションで削除する。

- ジョブ識別子の範囲指定をした場合、その範囲内の操作できるジョブを操作対象とする。操作できないジョブはエラーメッセージを出力しない。処理結果は、bjexlsjob コマンドで確認する。ジョブ識別子を指定したコマンドを使用した場合、エラー情報を出力する。
- ジョブ名またはジョブ名ワイルドカード指定をした場合、操作できるジョブの中から指定したジョブ名に該当するジョブを操作対象とする。対象外のジョブは、エラーメッセージを出力しない。処理結果は、bjexlsjob コマンドで確認する。ジョブ識別子を指定したコマンドを使用する場合はエラー情報を出力する。

使用例

- ジョブ識別子 125 のスプールジョブを削除する。

```
bjexrmjob 125
```

- ジョブ識別子 231 のエラー状態のスプールジョブを強制削除する。

```
bjexrmjob -f 231
```

2月2日 10:15:40 にこの指定をした場合、2月2日 10:15:09 以前のものが削除対象となる。この例では-a オプションを省略しており、-a オプションのデフォルト値である 30(秒)が有効となるため、このコマンドを投入した時刻から 30 秒以前が対象となる。

- ジョブ実行開始時刻が 30 日以前の出力保留状態のスプールジョブを削除する。

```
bjexrmjob -T 30
```

2月2日にこの指定をした場合、1月3日 23:59:59 以前に開始されたスプールジョブが削除の対象となる。

- ジョブ識別子 123 のスプールジョブについて、スプールジョブディレクトリまたはスプールジョブ管理ファイルの最終更新時刻がカレント時刻より 40 秒以上前の場合に削除する。

```
bjexrmjob -f -a 40 123
```

2月2日 10:15:30 にこの指定をした場合、2月2日 10:14:49 以前が対象となる。

- ジョブ識別子 10 から 100 までのスプールジョブを削除する。

```
bjexrmjob 10:100
```

- ジョブ名が JOB001, JOB002 のジョブを削除する。

```
bjexrmjob -J JOB001 JOB002
```

- ジョブ名の先頭が KEIRI のジョブを削除する。

```
bjexrmjob -J "KEIRI*"
```

- ジョブ実行開始時刻が 10 時間以前の出力保留状態のスプールジョブを削除する。

```
bjexrmjob -H 10
```

- ジョブ実行開始時刻が 36 時間以前の出力保留状態のスプールジョブのうち、ジョブ名の先頭が SOUMU のジョブを削除する。

```
bjexrmjob -H 36 -J "SOUMU*"
```

bjextrjob (転送ディレクトリ下の印刷ジョブ一覧の表示, 削除, および再転送 (管理者用))

形式

```
bjextrjob [-h 論理ホスト名] [-l | -e | -d | -r] [-t 秒数] [伝送カード]
```

機能

転送ディレクトリ下の印刷ジョブ一覧を表示, 削除, および再転送する。コマンド入力時点の時刻から-t オプションで指定した秒を引いた時刻以前に属性変更が行われた印刷ジョブ, またはこのコマンドのオプションで指定した伝送カードに該当する印刷ジョブを表示, 削除, および再転送する。

オプション

-h 論理ホスト名 ~<英字と数字と- (ハイフン) >((1~244))

バッチジョブ実行システムの論理ホストで運用している場合に指定する。設定ファイルとして/opt/hitachi/bjex/conf/bjex_論理ホスト名.conf を読み込み実行する。

-l

印刷ジョブの一覧を表示する。

-e

印刷ジョブの一覧を拡張形式で表示する。

-d

印刷ジョブを削除する。

-r

印刷ジョブを再転送する。

-t 秒数 ~<1~8 けたの 10 進数>((0-99999999)) 《259200》

表示, 削除, および再転送対象とする印刷ジョブの属性変更時刻を指定する。コマンド入力時点の時刻から指定した秒を引いた時刻以前に, 属性変更が行われた印刷ジョブを表示, 削除, および再転送する。

伝送カード ~<任意文字列>

-e オプションによって出力される, 表示, 削除, および再転送対象とする印刷ジョブの伝送カードを指定する。該当する印刷ジョブが複数存在する場合, すべてが対象となる。

使用例

- 3日 (259,200 秒) 以上前に属性変更を行った印刷ジョブ一覧を表示する。

```
bjextrjob -l -t 259200
```

出力例

```
JOBID S START TIME END TIME CHG TIME COUNT USERNAME JOBNAME
002000 T ----- 1171958742 ----- bjex -----
002009 N 1170932144 1170932234 1171958892 000019 bjex SAMPLE1
002009 E 1170932144 ----- 1172054691 000020 bjex SAMPLE1
002010 N 1170933374 1170933400 1171958892 000019 bjex SAMPLE1
002020 P ----- 1174000000 000005 bjex -----
002050 N 1184227363 1184227364 1184227413 000001 username JOB1NAME0000000ZZZZZZZ1234567
```

出力項目の説明

- JOBID

ジョブ識別子。同一ジョブを複数回印刷することもできるため、複数回表示されることもある。

- S

ジョブの状態。

項目	概要	説明
N	NOHOLD 状態	スプールジョブ転送デーモンによる転送が開始されていない状態、またはスプールジョブ転送デーモンが起動していないなどの要因で、処理されなかった状態である。
E	エラー状態	転送処理中にエラーが発生した状態である。
P	処理中	スプールジョブ転送サービスが処理中の状態である。
T	転送中	プリントサーバに転送中の状態である。

- START TIME

ジョブの開始時刻。1970/1/1 00:00:00 (UTC) からの通算秒である。

- END TIME

ジョブの終了時刻。1970/1/1 00:00:00 (UTC) からの通算秒である。

- CHG TIME

ジョブの属性変更時刻。1970/1/1 00:00:00 (UTC) からの通算秒である。

属性変更時刻とは、ジョブの終了時刻または `bjexchjob -R` コマンド入力時刻である。

- COUNT

属性変更が行われた回数。

- USERNAME

ユーザ名。

- JOBNAME

ジョブ名。

注意事項

値を求めることができない場合は、ハイフンまたは空白を表示する。

- すべての印刷ジョブ一覧を拡張形式で表示する。

```
bjextrjob -e -t0
```

出力例

```
JOBID S START TIME      END TIME      CHG TIME      COUNT USERNAME JOBNAME      CARDNAME
002000 T ----- 2007/02/20 17:05:42 ----- bjex -----
002009 N 2007/02/08 19:55:44 2007/02/08 19:57:14 2007/02/20 17:08:12 000019 bjex  SAMPLE1  FTPCARD00000000000001
002009 E 2007/02/08 19:55:44 ----- 2007/02/21 19:44:51 000020 bjex  SAMPLE1  FTPCARD00000000000001
002010 N 2007/02/08 20:16:14 2007/02/08 20:16:40 2007/02/20 17:08:12 000019 bjex  SAMPLE1  FTPCARD00000000000001
002020 P ----- 2007/03/16 08:06:40 000005 bjex -----
002050 N 2007/07/12 17:02:43 2007/07/12 17:02:44 2007/07/12 17:03:33 000001 username JOB1NAME00000000ZZZZZZZZ1234567 FTPCARD00000000000002
```

出力項目の説明

- **JOBID**

ジョブ識別子。同一ジョブを複数回印刷することもできるため、複数回表示されることもある。

- **S**

ジョブの状態。

項目	概要	説明
N	NOHOLD 状態	スプールジョブ転送デーモンによる転送が開始されていない状態、またはスプールジョブ転送デーモンが起動していないなどの要因で、処理されなかった状態である。
E	エラー状態	転送処理中にエラーが発生した状態である。
P	処理中	スプールジョブ転送サービスが処理中の状態である。
T	転送中	プリントサーバに転送中の状態である。

- **START TIME**

ジョブの開始時刻。コマンドを入力したユーザのローカル時刻で表示する。

- **END TIME**

ジョブの終了時刻。コマンドを入力したユーザのローカル時刻で表示する。

- **CHG TIME**

ジョブの属性変更時刻。コマンドを入力したユーザのローカル時刻で表示する。

属性変更時刻とは、ジョブの終了時刻または bjexchjob -R コマンド入力時刻である。

- **COUNT**

属性変更が行われた回数。

- **USERNAME**

ユーザ名。

- **JOBNAME**

ジョブ名。

- **CARDNAME**

転送に使用する伝送カード名を表示する。

- 3日（259,200 秒）以上前に属性変更を行った印刷ジョブを削除する。

```
bjextrjob -d -t 259200
```

戻り値

戻り値	意味
0	正常終了
0以外	エラー終了

注意事項

- このコマンドは root で実行する。
- -l オプション, -e オプション, -d オプション, および -r オプションのいずれかを同時に指定した場合, 最後に指定したオプションが有効になる。
- このコマンドは共用資源の排他を行っており, 処理途中で止まるような使用をした場合 (pg コマンドなどでメッセージ表示の途中で止める場合など), ほかの bjextrjob コマンドやスプールジョブ転送デーモンが待ち状態になることがあるので, そのような使用をしないこと。
- このコマンドは, 運用終了後に何らかのエラーによって転送されなかった印刷ジョブを削除または再転送するために使用する。

8

設定ファイル

バッチジョブ実行システムの設定ファイルについて説明する。

設定ファイルの記述形式

設定ファイル (bjex.conf) の記述形式を次に示す。

- 通常のパラメータ

```
△0パラメータ名△1パラメータのオペランド名
```

- 出力クラスのパラメータ

```
△0<SysoutClass△1クラス名>  
△0オペランド名△1オペランドの指定値  
△0</SysoutClass>
```

- 注釈

```
△0#△0注釈
```

(凡例)

△₀ : 0 バイト以上の空白文字を指定する。空白文字は省略できる。

△₁ : 1 バイト以上の空白文字を必ず指定する。空白文字は省略できない。

- 改行コードだけの行、空白と改行コードだけの行、および行の始まりが"#"または空白と"#"である行は、行全体がコメントであると見なし、読み飛ばす。
- 空白はパラメータの区切り文字として使用するため、パラメータのオペランド名には使用できない。
- コメントには日本語を使用できる。
- 文法エラーを検出した場合に、解析処理を中断しないでファイルの最後までエラーチェックをする。
- < SysoutClass△クラス名 >などの、タグの開始で文法エラーを検出した場合は、タグの終了が現れるまで文法チェックをしない。
- パス名には絶対パスを指定する。相対パスや存在しないパスの場合はエラーとする。
- 1 行の長さが 4,092 文字を超える場合はエラーとする。
- 同一ファイル中の、異なるパラメータ名の指定順序は任意とする。
- 設定ファイル内の大文字と小文字は区別する。
- 複数指定できないパラメータを複数記述した場合はエラーとする。
- 複数指定できるパラメータで記述できる最大数を超えた場合はエラーとする。

設定ファイルのパラメーター一覧

バッチジョブ実行システムの設定ファイルのパラメーター一覧を次の表に示す。

表 8-1 バッチジョブ実行システムの設定ファイルのパラメーター一覧

パラメータ名	定義内容	パラメータ指定	指定できる個数
BJEX_DIR_任意名	パス名置換の対象となる識別子とディレクトリのパス名を定義する。	任意	100
COBOL_LANG	COBOL プログラムを実行するときの LANG 環境変数値を指定する。		1
DD_DEALLOC_ON_ALLOCERR	ファイル割り当て処理中にエラーが発生した場合の後処理の方法を定義する。		
DD_EXISTCHECK	DD 要素に指定したファイルが存在しない場合にエラー終了するかどうかを定義する (EXEC 要素で LANG="COBOL" を指定していないジョブステップに有効)。		
DD_EXISTCHECK_COBOL	DD 要素に指定したファイルが存在しない場合にエラー終了するかどうかを定義する (EXEC 要素で LANG="COBOL" を指定したジョブステップに有効)。		
DD_SAMF	ジョブ定義 XML ファイルの DD 要素の SAMF 属性での省略値を定義する。		
DISPLAY_REPLY_MESSAGE	応答要求メッセージに対する応答メッセージを JP1/IM - View の画面に出力するかどうかを指定する。		
DISPMOD_NOFILE	DD 要素の DISP 属性に MOD を指定したファイルが存在しない場合の処理を定義する。		
DUMMY_FILE_TYPE	DD 要素に TYPE="DUMMY" を指定した場合に割り当てるファイルを定義する。		
EUR_DEFINITION_FILE_DIR	EUR 定義ファイル格納ディレクトリを定義する。		
EUR_LANG	EUR のエンコード種別を定義する。		31
EUR_PROGRAM	EUR プログラムのパスを定義する。		1
EXEC_COND_NOSTEPNAME	EXEC 要素の COND 属性にステップ名を指定しなかった場合、演算対象とするジョブステップを先行ジョブステップすべてか実行済みの最終ジョブステップかを定義する。		
EXEC_COND_STEPSEARCH 【AIX 限定】	COND 属性の検索範囲および逆方向参照の検索範囲を定義する。		

パラメータ名	定義内容	パラメータ指定	指定できる個数	
EXEC_PARMSPPLIT	ジョブ定義 XML ファイルの EXEC 要素の PARMSPPLIT 属性での省略値を定義する。	任意	1	
GDG_REGISTER_JOBRC_THRESHOLD	世代ファイルの世代管理ファイルへの登録タイミングの変更および登録させられる条件を定義する。			
ISAMLIB_MODE 【AIX 限定】	一時 ISAM ファイルを割り当てる場合に使用する ISAM ライブラリ（32 ビット版または 64 ビット版）を定義する。			
JOB_RCDEFINE	ジョブコントローラの終了コードの決定方法を定義する。			
JOB_STATUS_DISPLAY	実行中のジョブ情報表示機能を使用するかどうかを定義する。			
JOBLOG_CONVERT	ジョブログを出力する際に文字コードを変換するかどうかを定義する。			
JOBPARM_JSPECAN	並列実行しているジョブステップの 1 つが異常終了した場合、実行が終了していないジョブステップを打ち切るかどうかを指定する。			
JP1_HOSTNAME	JP1 が稼働する論理ホスト名を定義する。			
JP1EVENT_JOBINFO	バッチジョブ実行システムが発行する JP1 イベントに設定されるジョブの情報を指定する。			
JP1EVENT_SUPPRESS_ID	出力抑止する JP1 イベントのイベント ID を定義する。			無制限
JP1FTP_RETRY_INTERVAL	印刷ジョブ転送時に転送エラーが発生した場合に、再転送するまでのリトライ間隔を定義する。			1
JP1IM_MANAGER_CHKDELAY	メッセージ制御機能でイベント到達確認を行うまでの待ち時間を指定する。			
JP1IM_MANAGER_HOST	JP1 イベントを出力する固定のマネージャホスト名を定義する。			
LOCK_DIR_RECURSIVE	排他制御機能使用時、再帰的にロックを確保するかどうかを定義する。			
LOCK_JOBMPLEX	ジョブ多重度（同時に実行できるジョブ数）を定義する。			
LOCK_KEY_ID	排他制御機能および実行中のジョブ情報表示機能で使用するセマフォ、共有メモリのキーの識別情報を定義する。			
LOCK_MAXFILE	ロック獲得中またはロック待ち中のファイル合計数の最大値を定義する。			
LOCK_OPTION	排他制御機能とジョブ間 PREST ファイルを使用するかどうかを定義する。			

パラメータ名	定義内容	パラメータ指定	指定できる個数	
LOCK_RELEASE	ファイルおよびディレクトリの排他/共用制御で、排他/共用制御の解除契機を定義する。	任意	1	
LOG_DIR	実行ログを出力するディレクトリのパス名を定義する。			
LOG_FILE_CNT	実行ログをバックアップする面数を定義する。			
LOG_FILE_SIZE	実行ログを出力するファイルサイズを定義する。			
MAX_RESPONSE_REQUEST	JPl/IM - Manager に同時に受け付けられる応答待ちイベント数を定義する。			
MESSAGE_REPLACE_SPACE	メッセージ制御機能で半角空白に置き換える 1 バイトコードを定義する。		無制限	
MESSAGE_RETRY_TIMEOUT	メッセージ制御機能で、指定時間内にイベント送信処理が完了しなかった場合に、リトライを中断するかどうかを定義する。		1	
MSG_KEY_ID	メッセージ制御機能で使用するセマフォ、共有メモリのキーの識別情報を定義する。	必須	1	
MSGCLASS_DEFAULT	ジョブ定義 XML ファイルの JOB 要素の MSGCLASS 属性での省略値を定義する。			
PERMANENT_FILE_DIR	恒久ファイルを格納するディレクトリのパス名を定義する。	任意		
PERMISSION_DIR	ジョブ定義 XML ファイルの PERMISSION 要素の DIR 属性での省略値を定義する。			
PERMISSION_FILE	ジョブ定義 XML ファイルの PERMISSION 要素の FILE 属性での省略値を定義する。			
PERMISSION_SPOOL_STORE_JOB	保管するスプールジョブのパーミッションを変更する場合に定義する。			
PGM_RCABEND	ジョブステップを異常終了とするユーザプログラムの終了コードを定義する。			256
PROCEDURE_DIR	カタログドプロシジャのディレクトリのパス名を定義する。			255
PROCGROUP_QUEUE_NAME	JPl/AJS のキュー名をプロシジャグループ名として使用する機能を定義する。			1
RNW_FILE_METHOD	RNW を指定してファイルを再割り当てする場合のファイル再作成の方法を定義する。			
SPOOL_DIR	スプールディレクトリのパス名を定義する。			
SPOOL_JOB_STORE	ジョブ終了時にスプールジョブを保管する機能の使用を指定する。			
SPOOL_JOB_TRANSFER	スプールジョブを転送するかどうかを定義する。			

パラメータ名	定義内容	パラメータ指定	指定できる個数
SPOOL_STORE_DIR	スプールジョブを保管するときの保管先のディレクトリのパス名を指定する。	任意	1
SPOOL_STORE_DIRNAME	スプールジョブを保管するときの保管先ディレクトリ名称を定義する。		
SPOOL_STORE_SUBDIR	スプールジョブを保管するときの保管先のサブディレクトリを定義する。		
STDERRCLASS_DEFAULT	ジョブ定義 XML ファイルの JOB 要素の STDERRCLASS 属性での省略値を定義する。		
STDOUTCLASS_DEFAULT	ジョブ定義 XML ファイルの JOB 要素の STDOUTCLASS 属性での省略値を定義する。		
SYSOUT_APPEND_WRITE	SYSOUT データファイルへの書き込みモードを定義する。		
SysoutClass	各出カクラスの属性値を定義する。	1 個以上 必須	256
TEMP_FILE_DIR	次に示すファイルを格納するディレクトリのパス名を定義する。 <ul style="list-style-type: none"> • 一時ファイル • PREST ファイル 	任意	1
TEMP_FILE_DIR_XML_USE	一時ファイル格納ディレクトリパス名をジョブ定義 XML ファイルの DD 要素の DSN 属性で指定可能とするかどうかを定義する。		
TRACE_DIR	トレースを出力するディレクトリのパス名を定義する。		
TRACE_FILE_CNT	トレースを出力する面数を定義する。		
TRACE_FILE_SIZE	トレースを出力するファイルサイズを定義する。		
TRACE_LEVEL	トレースを出力するレベルを定義する。		

BJEX_DIR_任意名 (パス名置換識別子とディレクトリのパス名定義)

形式

```
[BJEX_DIR_任意名※ パス名]
```

注※ 環境変数を指定する。任意名が 1～22 文字で、それを含めて最大 31 文字で指定する。

機能

ジョブ定義 XML ファイルで、ジョブが使用するファイルに"絶対パス"で記述するのではなく、識別子を付加して記述することで、実行時にファイル名を置換する機能である。このパラメータでは、XML 中のパス名の変数を設定ファイルで置換する。パス名置換識別子はパラメータ名 (BJEX_DIR_任意名) となる。

このパラメータは、次に示すように識別子を用いて DD 要素の DSN 属性に指定する。

```
<DD NAME="REPDD" DSN="{BJEX_DIR_任意名}FILE" DISP="NEW,KEEP"/>
```

このパラメータを指定した場合、ジョブステップのプロセスに識別名の環境変数を作り、指定されたパス名を格納する。

詳細については、「[3.4.7 プログラムへのファイル割り当て機能](#)」を参照のこと。

オペランド

パス名 ～<パス名>((1～512 文字))

識別子に対応するディレクトリのパス名を指定する。

指定するパス名は、絶対パス名とすること。絶対パス名でない場合、設定ファイルのエラーとなる。このパラメータのオペランドにかぎり、存在しないパス名であってもエラーとならない。

注意事項

このパラメータで指定するパス名の末尾に「/」が付与されていない場合、そのパス名に対して自動的に末尾に「/」が付与される。

COBOL_LANG (COBOL プログラムを実行するときの LANG 環境変数値の指定)

形式

```
[COBOL_LANG LANG環境変数値]
```

機能

COBOL プログラムを実行するときの LANG 環境変数値を指定する。バッチジョブ実行システムの LANG 環境変数値と異なる値にしたいときに定義する。指定しない場合は、バッチジョブ実行システムの LANG 環境変数のままとする。

存在しない文字コードを指定した場合は、エラーとしないで、指定した文字列を LANG 環境変数に設定する。

オペランド

LANG 環境変数値 ~<任意文字列>((1~31 文字))

COBOL プログラムを実行するときの LANG 環境変数値を指定する。

注意事項

- オペランドに文字コードを 2 個以上指定することはできない。
- このパラメータが適用されるのは、ジョブ定義 XML ファイルの EXEC 要素に LANG="COBOL" と指定したジョブステップだけである。また、PGM="*" を指定して複数のプログラムを起動する場合、LANG="COBOL" の指定があれば、ジョブステップ内のすべてのプログラムが指定したコードで動作する。

DD_DEALLOC_ON_ALLOCERR (ファイル割り当て処理中にエラーが発生した場合の後処理の方法の定義)

形式

```
[DD_DEALLOC_ON_ALLOCERR {OPT1|OPT2}]
```

機能

ジョブステップ実行開始時のファイル割り当て処理でエラーとなった場合の、ファイルの後処理方法を定義する。

オペランド

DD_DEALLOC_ON_ALLOCERR ~{OPT1 | OPT2}

ジョブステップ実行開始時のファイル割り当て処理でエラーが発生した場合に、そのジョブステップで割り当て済みのファイルに対する後処理方法を定義する。

このパラメータが省略されている場合、オペランドに OPT1 が指定されたものとする。

OPT1 : 「3.4.2(2)(d) ファイル割り当て処理中にエラーが発生した場合の後処理の指定」の表 3-7 に従って処理する。

OPT2 : 「3.4.2(2)(d) ファイル割り当て処理中にエラーが発生した場合の後処理の指定」の表 3-8 に従って処理する。

DD_EXISTCHECK (DD 要素に指定したファイルパスの存在確認要否を実施するための定義)

形式

```
[DD_EXISTCHECK {CONCAT|NO}]
```

機能

DD 要素に指定したファイルの連結を行う際、どれかのファイルが存在しない場合、ファイル割り当てエラーとするときに指定する (KAKC1660-E メッセージが表示される)。

このパラメータの指定に関係なく、JOB LIB/STEPLIB、一時ファイル、一時 ISAM ファイルのファイルパスに対しては存在確認を行う。

TYPE="GDG"かつ世代データ群指定の場合、世代群に対する存在確認を行うのではなく、世代データ群で管理されている世代ファイルに対してファイルの存在確認を行う。また、世代データ群指定の場合、DD 連結の有無に関係なく、世代データ群で管理されている世代ファイルの存在確認を行う。

このパラメータは、EXEC 要素の LANG 属性に"COBOL"を指定していないジョブステップに有効となる。

オペランド

DD_EXISTCHECK ~ {CONCAT | NO}

EXEC 要素の LANG 属性に"COBOL"以外を指定した場合に、DD 要素でファイル連結として指定した恒久ファイルと世代ファイルのファイルパスの存在確認をするかどうかを指定する。

CONCAT : DD 要素でファイル連結を指定した場合に存在確認を行う。

NO : 存在確認を行わない。

このパラメータの指定が省略されている場合、オペランドに NO を仮定する。

DD_EXISTCHECK の指定値によるファイルパスの存在確認の対象となる TYPE 属性一覧を次の表に示す。

表 8-2 DD_EXISTCHECK の指定値によるファイルパスの存在確認の対象となる TYPE 属性一覧

オペランド	DD 要素の TYPE 属性										
	LIB	FILE	DIR	GDG	DAT A	DUMM Y	TEMP	TEMPISAM	SYSOUT	PREST	TEMPPREST
NO	◎	×	×	×	—	—	○	◎	—	×	×
CONCAT	◎	○	○	○	—	—	○	◎	—	×	×

(凡例)

◎ : ファイル連結の指定に関係なくファイルパスの存在確認を行う。

○：ファイル連結を指定した場合、ファイルパスの存在確認対象である。

×：ファイルパスの存在確認対象でない。

－：バッチジョブ実行システムが用意するファイルのため、ファイルパスの存在確認を行う必要がない。

注意事項

- ファイルパスの存在確認を行う DD 要素は、DISP 属性で OLD または SHR が指定されている必要がある。
- ディレクトリの連結はできないため、TYPE="DIR"の DD 要素はファイルパスの存在確認対象でない。ただし、ファイル連結の指定をしてジョブを実行した場合は、TYPE="FILE"や TYPE="DIR"の DD 要素と同様にファイルパスの存在確認を行う。
- DD 連結を指定した際に 1 つでも存在しないファイルがある場合、そのジョブステップは KAKC1660-E メッセージを出力して異常終了する。

DD_EXISTCHECK_COBOL (DD 要素に指定したファイルパスの存在確認要 否を実施するための定義)

形式

```
[DD_EXISTCHECK_COBOL {CONCAT|NO}]
```

機能

DD 要素に指定したファイルの連結を行う際、どれかのファイルが存在しない場合、ファイル割り当てエラーとするときに指定する (KAKC1660-E メッセージが表示される)。

このパラメータの指定に関係なく、JOB LIB/STEPLIB、一時ファイル、一時 ISAM ファイルのファイルパスに対しては存在確認を行う。

TYPE="GDG"かつ世代データ群指定の場合、世代群に対する存在確認を行うのではなく、世代データ群で管理されている世代ファイルに対してファイルの存在確認を行う。また、世代データ群指定の場合、DD 連結の有無に関係なく、世代データ群で管理されている世代ファイルの存在確認を行う。

このパラメータは、EXEC 要素の LANG 属性に"COBOL"を指定したジョブステップだけ有効となる。

オペランド

DD_EXISTCHECK_COBOL ~ {CONCAT | NO}

EXEC 要素の LANG 属性に"COBOL"を指定した場合に、DD 要素でファイル連結として指定した恒久ファイルと世代ファイルのファイルパスの存在確認をするかどうかを指定する。

CONCAT: DD 要素でファイル連結を指定した場合に存在確認を行う。

NO: 存在確認を行わない。

このパラメータの指定が省略されている場合、オペランドに NO を仮定する。

DD_EXISTCHECK_COBOL の指定値によるファイルパスの存在確認の対象となる TYPE 属性一覧を次の表に示す。

表 8-3 DD_EXISTCHECK_COBOL の指定値によるファイルパスの存在確認の対象となる TYPE 属性一覧

オペラ ンド	DD 要素の TYPE 属性										
	LIB	FILE	DI R	GD G	DAT A	DUMM Y	TEM P	TEMPISAM	SYSOUT	PREST	TEMPPREST
NO	◎	×	×	×	-	-	◎	◎	-	×	×
CONCA T	◎	○	○	○	-	-	◎	◎	-	×	×

(凡例)

- ◎：ファイル連結の指定に関係なくファイルパスの存在確認を行う。
- ：ファイル連結を指定した場合、ファイルパスの存在確認対象である。
- ×：ファイルパスの存在確認対象でない。
- －：バッチジョブ実行システムが用意するファイルのため、ファイルパスの存在確認を行う必要がない。

注意事項

- ファイルパスの存在確認を行う DD 要素は、DISP 属性で OLD または SHR が指定されている必要がある。
- ディレクトリの連結はできないため、TYPE="DIR"の DD 要素はファイルパスの存在確認対象でない。ただし、ファイル連結の指定をしてジョブを実行した場合は、TYPE="FILE"や TYPE="DIR"の DD 要素と同様にファイルパスの存在確認を行う。
- DD 連結を指定した際に 1 つでも存在しないファイルがある場合、そのジョブステップは KAKC1660-E メッセージを出力して異常終了する。

DD_SAMF (DD 要素の SAMF 属性での省略値定義)

形式

```
[DD_SAMF {YES|NO}]
```

機能

ジョブ定義 XML ファイルの DD 要素の SAMF 属性での省略値を定義する。

オペランド

DD_SAMF ~{YES | NO}

ジョブ定義 XML ファイルに記述した SYSIN データを、ジョブコントローラが COBOL プログラムに渡す場合のデータの形式を指定する。

YES : COBOL が順編成固定長ファイルとして SYSIN データを入力する場合に YES を指定する。ジョブコントローラは、ジョブ定義 XML ファイルに記述した SYSIN データを 80 バイトの順編成固定長ファイルに変換し、COBOL プログラムに渡す。

NO : COBOL がテキスト編成ファイルとして SYSIN データを入力する場合に NO を指定する。ジョブ定義 XML ファイルに記述した SYSIN データを変更しないで、COBOL プログラムに渡す。

注意事項

ジョブ定義 XML ファイルの DD 要素の SAMF 属性を省略した場合でも、TYPE 属性値が"DATA"でない、または同一ジョブステップでの EXEC 要素の LANG 属性値が"COBOL"でない場合は、このパラメータの指定は無効となり、"NO"を指定したときと同じ動作になる。

DISPLAY_REPLY_MESSAGE (応答メッセージを JP1/IM - View の画面に出力するための定義)

形式

```
[DISPLAY_REPLY_MESSAGE {YES|NO}]
```

機能

応答要求メッセージに対する応答メッセージを JP1/IM - View の画面に出力するかどうかを指定する。

オペランド

DISPLAY_REPLY_MESSAGE ~{YES | NO}

応答要求メッセージに対する応答メッセージを JP1/IM - View の画面に出力するかどうかを指定する。
このパラメータの指定が省略されている場合、オペランドに NO を仮定する。

YES：応答メッセージを JP1/IM - View の画面に出力する。

NO：応答メッセージを JP1/IM - View の画面に出力しない。

注意事項

このパラメータは環境変数 BJEX_DISPLAY_REPLY に何も設定されていない場合に有効となる。

DISPMOD_NOFILE (DD 要素の DISP 属性に MOD を指定したファイルが存在しない場合の処理の定義)

形式

```
[DISPMOD_NOFILE {NEW|ERROR}]
```

機能

DD 要素の DISP 属性に MOD を指定して恒久ファイル、一時ファイル、一時 ISAM ファイルのどれかを割り当てる際、ファイルが存在しなかった場合、NEW を仮定して処理を続行するか、エラー終了するかを定義する。

オペランド

DISPMOD_NOFILE ~ {NEW | ERROR}

DD 要素の DISP 属性に MOD を指定して恒久ファイル、一時ファイル、一時 ISAM ファイルのどれかを割り当てる際、ファイルが存在しなかった場合の動作を指定する。

このパラメータの指定が省略されている場合、オペランドに NEW が指定されたものとする。

NEW：ファイルが存在しない場合、NEW を仮定して処理を続行する。

ERROR：ファイルが存在しない場合、エラー終了する (KAKC1660-E または KAKC1614-E を出力する)。

DISPMOD_NOFILE, DISP 属性に MOD を指定してファイルが存在しない場合にエラー終了となる TYPE 属性一覧を次の表に示す。

表 8-4 DISP 属性に MOD を指定してファイルが存在しない場合の扱い

オペランド	DD 要素の TYPE 属性										
	LIB	FILE	DIR	GD G	DAT A	DUMM Y	TEMP	TEMPISAM	SYSOUT	PREST	TEMPPREST
NEW	-	○	-	×	-	-	○	○	-	-	-
ERRO R	-	×	-	×	-	-	×	×	-	-	-

(凡例)

○：エラー終了とならないで、DISP 属性に NEW を仮定する。

×：エラー終了する。

-：DISP="MOD"を指定できない、または無視する。

DUMMY_FILE_TYPE (DD 要素に TYPE="DUMMY"を指定した場合の割り当てファイルの定義)

形式

```
[DUMMY_FILE_TYPE {DEVNULL|TEMP}]
```

機能

DD 要素に TYPE="DUMMY"を指定した場合に割り当てるファイルを定義する。

ファイルを排他してオープンするプログラムの場合、DD 要素に TYPE="DUMMY"を指定したときに/dev/null を割り当てると、/dev/null で排他エラーとなることがある。

このような問題を回避する場合、このパラメータに TEMP を指定し、一時ファイルを割り当てる。

オペランド

DUMMY_FILE_TYPE ~{DEVNULL | TEMP}

DD 要素に TYPE="DUMMY"を指定した場合に割り当てるファイルを指定する。

このパラメータの指定が省略されている場合、オペランドに DEVNULL が指定されたものとする。

DEVNULL：/dev/null ファイルを割り当てる。

TEMP：一時ファイルを割り当てる。DD 要素で TYPE="DUMMY"の代わりに、TYPE="TEMP" (DISP 属性, DSN 属性の指定なし) が指定されたものとしてジョブを実行する。

注意事項

- このパラメータは2つ以上指定しないこと。
- 割り当てファイルタイプ="TEMP"を指定した場合、DD 要素に TYPE="DUMMY"を指定した場合のファイルとして、実体を持つ一時ファイルを割り当て、入出力が行われる。
設定ファイルの TEMP_FILE_DIR パラメータで指定する一時ファイルの割り当て先ファイルシステムには、実行プログラムが DD 要素に TYPE="DUMMY"を指定した場合の一時ファイルに出力する量に応じた空き容量が必要である。
- DD 要素に TYPE="DUMMY"を指定した場合に割り当てたファイルに対して入出力をするときの入出力データを次の表に示す。

表 8-5 DD 要素に TYPE="DUMMY"を指定した場合に割り当てたファイルに対して入出力をするときの入出力データ

割り当てファイルタイプオペランド	入力	出力
DEVNULL	常にファイルの終わり (EOF) が報告される。	出力データは破棄される。

割り当てファイルタイプオペランド	入力	出力
TEMP	<ul style="list-style-type: none"> • 入力前に実行プログラムでデータを一時ファイルに出力していない場合、ファイルの終わり (EOF) が報告される。 • 入力前に実行プログラムで一時ファイルに出力したデータがあれば、出力されたデータが読み込まれる場合がある。 	割り当てた一時ファイルにデータを出力する。

- COBOL2002 で作成したアプリケーションおよび SORT EE は、通常、/dev/null に対して排他をする。ただし、次の製品を使用する場合には、このパラメータに DEVNULL を指定しても問題ない。
 - バッチジョブ実行システムから起動した場合に /dev/null を排他しない機能をサポートした SORT EE
 - バッチジョブ実行システムのダミーファイルに対応した COBOL2002
- これらの製品および機能の詳細については、各製品のドキュメントを参照のこと。

EUR_DEFINITION_FILE_DIR (EUR 定義ファイル格納ディレクトリの定義)

形式

[EUR_DEFINITION_FILE_DIR パス名]

機能

CSV 形式の SYSOUT データファイルを EUR で印刷するときのプリンタ名や帳票ファイル名を定義した EUR 定義ファイルの格納ディレクトリを定義する。

オペランド

パス名 ~<パス名>((1~128 文字))

EUR 定義ファイルを格納するディレクトリを指定する。

EUR_LANG (EUR のエンコード種別の定義)

形式

[EUR_LANG エンコード種別 LANG環境変数値]

機能

ジョブ定義 XML ファイルで指定する EUR のデータファイルのエンコードに対応する LANG 環境変数値を定義する。通常は定義する必要はなく、バッチジョブ実行システムの LANG 環境変数と UAP の LANG 環境変数が異なるときに定義する。また、ジョブ定義 XML ファイルの DD 要素の EURENCODE 属性で EUR のマッピングデータごとにエンコード種別を指定する。

オペランド

エンコード種別 ～<任意文字列>((1～31 文字))

エンコード種別を任意の文字で定義する。

この値は、ジョブ定義 XML ファイルの DD 要素の EURENCODE 属性で指定する。

LANG 環境変数値 ～<任意文字列>((1～31 文字))

エンコードに対応した LANG 環境変数値を指定する。

ジョブ定義 XML ファイルの DD 要素の EURENCODE 属性で EUR_LANG パラメータのエンコード種別を指定した場合、指定した DD 要素の SYSOUT データファイルを EUR のコマンドで処理するときに、この LANG 環境変数値が設定される。

注意事項

- EUR_LANG パラメータは最大 31 枚指定できる。
- 同一エンコード種別を二重に定義した場合、設定ファイル解析エラーとなる。

EUR_PROGRAM (EUR プログラムのパス定義)

形式

```
[EUR_PROGRAM {/opt/eurps/bin/eurps | /opt/eurpm_adp/bin/eurpmc}]
```

機能

EUR 連携で使用するコマンドのパスを指定することで、連携する EUR プログラムを指定する。この指定によって EUR Print Service と連携するかまたは EUR Print Manager と連携するかを選択できる。両方の指定を混在させることはできない。

オペランド

/opt/eurps/bin/eurps

EUR Print Service と連携した帳票印刷の場合に指定する。

/opt/eurpm_adp/bin/eurpmc

EUR Print Manager と連携した帳票印刷の場合に指定する。

上記以外の指定でもエラーにはならない。プログラムのファイル名が eurps の場合 EUR Print Service との連携処理を行い、eurpmc の場合 EUR Print Manager との連携処理を行う。また、eurps でも eurpmc でもないコマンドを指定した場合、そのコマンドを起動するが、コマンドの引数の作成処理などは EUR Print Service の処理を実行する。

EXEC_COND_NOSTEPNAME (COND 属性ステップ名未指定時の演算対象定義)

形式

```
[EXEC_COND_NOSTEPNAME {STEPALL|STEPLAST}]
```

機能

EXEC 要素の COND 属性にステップ名を指定しない場合、比較するジョブステップを定義する。

オペランド

EXEC_COND_NOSTEPNAME ~{STEPALL | STEPLAST}

STEPALL：ステップ名を指定しなかった場合、先行ジョブステップすべての終了コードと論理演算をする。

STEPLAST：ステップ名を指定しなかった場合、実行済みの最終ジョブステップの終了コードと論理演算をする。

注意事項

- STEPLAST 指定時に直前のジョブステップが COND 属性や IF 要素で実行されなかった場合、演算対象にはならないでその前のジョブステップが対象となる。
- STEPLAST 指定時に直前のジョブステップが JSPEC 機能のメインステップであった場合、メインステップが演算対象となる。
- STEPLAST 指定時に JSPEC 機能のメインステップおよびサブステップで EXEC 要素の COND 属性にステップ名を指定しなかった場合、そのメインステップおよびサブステップの直前のジョブステップが演算対象となる。

EXEC_COND_STEPSEARCH (COND 属性および逆方向参照の検索範囲の定義) 【AIX 限定】

形式

```
[EXEC_COND_STEPSEARCH { BJEX | COMPATIV3 }]
```

機能

COND 属性および逆方向参照の検索範囲を指定する。詳細については、「[3.6.3 ジョブステップを条件付きで実行するときのステップ名の検索範囲](#)」を参照のこと。

オペランド

EXEC_COND_STEPSEARCH ~{BJEX | COMPATIV3}

COND 属性および逆方向参照でジョブステップ名だけを指定した場合の検索範囲を指定する。このパラメータの指定が省略されている場合、オペランドに BJEX が指定されたものとする。

BJEX：次のジョブステップ名の検索範囲をバッチジョブ実行システムで定義した範囲とする。

- ジョブステップを条件付きで実行する場合
- 逆方向参照を行う場合

COMPATIV3：次のジョブステップ名の検索範囲を VOS3 と同等にする。

- ジョブステップを条件付きで実行する場合
- 逆方向参照を行う場合

EXEC_PARMSPLIT (EXEC 要素の PARMSPLIT 属性での省略値定義)

形式

```
[EXEC_PARMSPLIT {YES|NO}]
```

機能

ジョブ定義 XML ファイルの EXEC 要素の PARMSPLIT 属性での省略値を定義する。

オペランド

EXEC_PARMSPLIT ~{YES | NO}

EXEC 要素の PARM 属性値が空白を含む場合、引数がどのようにプログラムに渡されるかを指定する。

YES : UNIX の一般的な引数インタフェースである C 言語インタフェースに従った形式を使用する場合に、YES を指定する。EXEC 要素の PARM 属性値が空白を含む場合、空白で区切られた複数の引数としてプログラムに渡される。

NO : VOS3 から移行した COBOL プログラムなどの VOS3 インタフェースに従った形式を使用する場合に、NO を指定する。EXEC 要素の PARM 属性値が空白を含む場合、空白を含む 1 つの引数としてプログラムに渡される。

GDG_REGISTER_JOBRC_THRESHOLD (世代ファイルの世代管理ファイルへの登録タイミングの変更と登録させられる条件の定義)

形式

[GDG_REGISTER_JOBRC_THRESHOLD しきい値]

機能

ジョブステップで生成した世代ファイルの世代管理ファイルへの登録タイミングの変更と登録させられるジョブ終了コードをしきい値によって判定する。

TYPE="GDG"を指定した DD 要素で DISP 属性の第 2 オペランドに"KEEP"を指定している場合、"PASS"を指定した場合の動作を行う。

ジョブ終了時、ジョブ終了コードがしきい値未満の場合、TYPE="GDG"を指定した DD 要素の DISP 属性の第 2 オペランド値を"PASS"から"KEEP"に変更し、ファイルの保存と世代管理ファイルへの登録を行う。終了コードがしきい値以上の場合、"PASS"を"DELETE"に変更しファイルの削除を行う。このとき、世代管理ファイルへの登録も行われない。

このしきい値によるファイルの登録・削除の対象となるのは、ジョブ中で新規に世代ファイルを生成するものだけである。ジョブ実行開始以前から存在する世代ファイルについては、"PASS"は"KEEP"に変更され、しきい値では削除しない。

オペランド

しきい値 ~<符号なし整数>((1~255))

新規作成した世代ファイルの保存と世代管理ファイルへの登録可否をしきい値によって制御する。指定された「しきい値-1」の値までのジョブ終了コードでジョブが終了した場合は、世代ファイルの保存と世代管理ファイルへの登録を行い、「しきい値」以上の値のジョブ終了コードでジョブが終了した場合は、世代ファイルの保存と世代管理ファイルへの登録は行わない。

ISAMLIB_MODE (ISAM ライブラリ (32 ビット版または 64 ビット版) の定義) 【AIX 限定】

形式

```
[ISAMLIB_MODE {M32|M64}]
```

機能

一時 ISAM ファイルを割り当てる場合に使用する ISAM ライブラリ (32 ビット版または 64 ビット版) を定義する。

オペランド

ISAMLIB_MODE ~{M32 | M64}

ISAM ライブラリの 32 ビット版または 64 ビット版のどちらを使用するかを指定する。この指定を省略した場合、オペランドに M32 が指定されたものとする。

M32：一時 ISAM ファイルを割り当てる場合に ISAM ライブラリの 32 ビット版を使用する。bjexec から ISAM ライブラリの 32 ビット版を使用するコマンド処理がサポートプログラムで実行される。

M64：一時 ISAM ファイルを割り当てる場合に ISAM ライブラリの 64 ビット版を使用する。

注意事項

- 一時 ISAM ファイル使用時の、使用する ISAM を次の表に示す。

使用 ISAM	ISAM(32)	ISAM(64)	ISAM(32), ISAM(64)を併用
ISAMLIB_MODE M32 の指定あり	○	×	○ ISAM(32)
ISAMLIB_MODE M64 の指定あり	×	○	○ ISAM(64)

(凡例)

○：使用できる。

×：使用できない。

ISAM(32)：32 ビットの ISAM を使用する。

ISAM(64)：64 ビットの ISAM を使用する。

- Linux では、このパラメータを指定してもエラーにならないが、M32 を指定した場合も、ISAM ライブラリの 64 ビット版を使用する。

JOB_RCDEFINE (ジョブコントローラの終了コードの決定方法定義)

形式

```
[JOB_RCDEFINE {STEPMAX|STEPLAST}]
```

機能

ジョブコントローラの終了コードの決定方法を定義する。

オペランド

JOB_RCDEFINE ~{STEPMAX | STEPLAST}

ジョブコントローラの終了コードの決定方法を定義する。

この指定が省略されている場合、オペランドに STEPMAX が指定されたものとする。

STEPMAX：ジョブコントローラの終了コードは、ジョブステップの最大の終了コードとなる。

STEPLAST：ジョブコントローラの終了コードは、最後に実行完了したメインステップの終了コードとなる。

注意事項

- COND 属性によってスキップしたジョブステップは、ジョブコントローラの終了コードに影響を与えない。
- JP1/AJS によるジョブの強制終了を行った場合、JOB_RCDEFINE パラメータの指定に関係なく、ジョブの終了コードは 255 となる。
- JOB_RCDEFINE パラメータに STEPMAX を指定した場合、ジョブステップの終了コードの最大値が 240 より大きい場合、ジョブの終了コードは 240 とする。
- JOB_RCDEFINE パラメータに STEPLAST を指定した場合、最後に実行完了したメインステップの終了コードが 240 より大きい場合、ジョブの終了コードは 240 とする。
- JOB_RCDEFINE パラメータに STEPLAST を指定した場合、最後に実行完了したジョブステップの直後に指定した SETMAXRC 要素だけが有効となる。その他の SETMAXRC 要素は、ジョブステップの終了コードの最大値を書き換えたことを示す KAKC1909-I は出力されるが、ジョブの終了コードには影響を与えない。例を次に示す。

ジョブ定義 XML ファイル

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="TESTJOB">
    <STEP NAME="STEP1_RC4"> <!-- 終了コード4のステップ -->
      <EXEC PGM="CBLRTN" PARM="004" LANG="COBOL" />
    </STEP>
    <STEP NAME="STEP2_RC0"> <!-- 終了コード0のステップ -->
      <EXEC PGM="CBLRTN" PARM="000" LANG="COBOL" JSPEC="YES" />
    </STEP>
    <SETMAXRC VALUE="16" />
    <STEP NAME="STEP3_RC8"> <!-- 終了コード8のステップ -->
      <EXEC PGM="CBLRTN" PARM="008" LANG="COBOL" />
    </STEP>
    <STEP NAME="STEP4_RC0"> <!-- 終了コード0のステップ -->
      <EXEC PGM="CBLRTN" PARM="000" LANG="COBOL" />
    </STEP>
    <SETMAXRC VALUE="4" />
  </JOB>
</HitachiBatchJobExec>
```

JOBLOG

```
***** BJEX MESSAGE *****
21:08:17 001212 KAKC0091-I TESTJOB Job started.
21:08:17 001212 KAKC0092-I TESTJOB.STEP1_RC4 Step started.
21:08:17 001212 KAKC0097-I TESTJOB.STEP1_RC4 Step ended. rc=4 E-Time=0.488s C-Time=0.000s
21:08:17 001212 KAKC0092-I TESTJOB.STEP2_RC0 Step started.
21:08:17 001212 KAKC0092-I TESTJOB.STEP3_RC8 Step started.
21:08:18 001212 KAKC0097-I TESTJOB.STEP3_RC8 Step ended. rc=8 E-Time=0.221s C-Time=0.010s
21:08:18 001212 KAKC0097-I TESTJOB.STEP2_RC0 Step ended. rc=0 E-Time=0.620s C-Time=0.020s
21:08:18 001212 KAKC1909-I TESTJOB The maximum value of return code was changed by SETMAXRC element.
rc=16
21:08:18 001212 KAKC0092-I TESTJOB.STEP4_RC0 Step started.
21:08:18 001212 KAKC0097-I TESTJOB.STEP4_RC0 Step ended. rc=0 E-Time=0.253s C-Time=0.020s
21:08:18 001212 KAKC1909-I TESTJOB The maximum value of return code was changed by SETMAXRC element.
rc=4
21:08:18 001212 KAKC0098-I TESTJOB Job ended. rc=4 E-Time=1.361s C-Time=0.050s MaxRSS=1972KB
```

この例では、ジョブステップ STEP2_RC0 の直後に指定した SETMAXRC 要素でジョブの終了コードを 16 に設定しているが、最終的なジョブの終了コードに影響を与えない。ジョブステップ STEP4_RC0 の直後に指定した SETMAXRC 要素はジョブの終了コードを 4 に設定しており、これが最後に実行完了したジョブステップの直後に指定した SETMAXRC 要素であるため、ジョブの終了コードとなる。

- JOB_RCDEFINE パラメータに STEPLAST を指定した場合、サブステップの終了コードはジョブコントローラの終了コードに影響を与えない。ただし、最後に実行完了したサブステップ直後の SETMAXRC 要素は有効となる。
- JOB_RCDEFINE パラメータに STEPLAST を指定した場合、メインステップが 1 つも実行されずにジョブが正常終了したとき、ジョブの終了コードは 0 となる。

JOB_STATUS_DISPLAY (実行中のジョブ情報表示機能を使用するかどうかの定義)

形式

```
[JOB_STATUS_DISPLAY { ENABLE | DISABLE }]
```

機能

実行中のジョブの情報を表示する機能を使用するかどうかを指定する。

オペランド

JOB_STATUS_DISPLAY ~ {ENABLE | DISABLE}

bjexlsexjob コマンドを使用して、実行中のジョブの情報を表示する機能を使用するかどうかを指定する。

ENABLE：使用する。

DISABLE：使用しない。

注意事項

- この指定を行った場合、実行中のジョブ情報を格納するための共用メモリが必要となる。必要となる共用メモリの計算式は、リリースノートを参照のこと。
- 表示可能なジョブ数として、LOCK_JOBMPLEX パラメータで指定したジョブ多重度を使用する。LOCK_JOBMPLEX で指定された多重度を超える場合、ジョブの実行は待たされる。

JOBLOG_CONVERT (ジョブログ出力時の文字コード変換定義)

形式

```
[JOBLOG_CONVERT {YES|NO}]
```

機能

ジョブログを stderr に出力する際に文字コードに変換するかを定義する。スプールジョブディレクトリ内のファイルの文字コードは変換しない。

オペランド

JOBLOG_CONVERT ~{YES | NO}

YES : ジョブログに出力する際にジョブ定義 XML ファイルの encoding で指定した文字コードから環境変数 LANG で指定した文字コードに変換して出力する。変換する文字コードの組み合わせを次の表に示す。

- AIX の場合

encoding 属性	環境変数 LANG	
	Shift-JIS	EUC-JP
Shift-JIS	×	○
UTF-8	○	○
EUC-JP	○	×
ASCII	×	×

(凡例)

- : 文字コードを変換する。
- × : 文字コードを変換しない。

- Linux の場合

encoding 属性	環境変数 LANG	
	UTF-8	C
Shift-JIS	○	×
UTF-8	×	×

(凡例)

- : 文字コードを変換する。
- × : 文字コードを変換しない。

NO : ジョブログに出力する際に文字コードを変換しないで出力する。

注意事項

- この機能は、Hitachi Code Converter が前提となる。
- 環境変数 LANG に JP1/AJS-Agent の言語環境と同じ文字コードを設定すること。
- ジョブ定義 XML ファイルの encoding を省略した場合、文字コードを変換しない。
- カタログドプロシジャの encoding は無視する。

JOBPARM_JSPECAN (ジョブステップ異常時の打ち切り定義)

形式

[JOBPARM_JSPECAN {YES|NO}]

機能

並列実行しているジョブステップの1つが異常終了した場合、実行が終了していないジョブステップを打ち切るかどうかを指定する。

オペランド

JOBPARM_JSPECAN ~{YES | NO}

JSPEC 機能で並列実行したジョブステップが異常終了した場合、実行が終了していないジョブステップを打ち切るかどうかを指定する。

YES : 実行が終了していないジョブステップを打ち切る場合に YES を指定する。ジョブコントローラは実行が終了していないジョブステップに SIGKILL を送って打ち切る。

NO : 実行が終了していないジョブステップを打ち切らない場合は NO を指定する。

JP1_HOSTNAME (JP1 が稼働する論理ホスト名定義)

形式

[JP1_HOSTNAME 論理ホスト名]

機能

物理ホストでバッチジョブ実行システムを稼働させ、論理ホストで稼働している JP1 と連携する場合、JP1 の論理ホスト名を定義する。

オペランド

論理ホスト名 ～<記号名称で指定できる文字と- (ハイフン) >((1～255 文字)) 《バッチジョブ実行システムが稼働するサーバの物理ホスト名》

物理ホストでバッチジョブ実行システムを稼働させ、論理ホストで稼働している JP1 と連携する場合、JP1 の論理ホスト名を指定する。このパラメータを省略した場合は、バッチジョブ実行システムが稼働するサーバで hostname コマンドを実行したときに表示されるホスト名が指定されたものと見なす。バッチジョブ実行システムを論理ホストで稼働させる場合は、このパラメータは無視される。

バッチジョブ実行システムを物理ホストで稼働させるには、バッチジョブ実行システムのコマンドに -h オプションを指定しないで実行する。バッチジョブ実行システムを論理ホストで稼働させるには、バッチジョブ実行システムのコマンドに -h オプションを指定して実行する。詳細については、「[4.9 バッチジョブ実行システムの環境を複数構築する場合の設定](#)」および「[7. コマンド](#)」を参照すること。

このパラメータで指定したホスト上の JP1/Base と通信できない場合、バッチジョブ実行システムの起動に時間を必要とする場合がある。

注意事項

'/'や'¥'など、Unix や Windows のパス区切り文字をホスト名として指定しないこと。指定した場合はスプールジョブの転送時にエラーとなる。

JP1EVENT_JOBINFO (JP1 イベントのジョブ情報の指定)

形式

```
[JP1EVENT_JOBINFO {BJEX|JP1AJS}]
```

機能

バッチジョブ実行システムが発行する JP1 イベントに設定されるジョブの情報として、JP1/AJS の情報を使用するか、バッチジョブ実行システムの情報を使用するか指定する。バッチジョブ実行システムが発行する JP1 イベントは次の文字コードで発行しており、JP1/AJS のジョブの情報と文字コードが異なる場合、文字化けすることがある。その場合、バッチジョブ実行システムの情報を使用することで文字化けしないで表示できる。

- AIX の場合：シフト JIS
- Linux の場合：UTF-8

オペランド

JP1EVENT_JOBINFO ~{BJEX | JP1AJS}

BJEX：バッチジョブ実行システムが発行する JP1 イベントに設定されるジョブの情報を"bjexec"とする。

JP1AJS：バッチジョブ実行システムが発行する JP1 イベントに設定されるジョブの情報として JP1/AJS の情報を使用する。

注意事項

このパラメータは設定ファイル解析に失敗した際に発行する JP1 イベントには有効にならないことがある。

JP1EVENT_SUPPRESS_ID (出力抑止する JP1 イベントのイベント ID 定義)

形式

[JP1EVENT_SUPPRESS_ID イベントID]

機能

JP1 イベントの出力を抑止する。抑止したい JP1 イベントのイベント ID を定義する。

オペランド

イベント ID ～< 16 進数>((00005C00～00005CFF))

出力抑止したい JP1 イベントのイベント ID を指定する。

複数の JP1 イベントを抑止したい場合は、パラメータを複数指定する。

イベント ID を重複して指定した場合は、1 つ指定した場合と同じ扱いとする。

指定可能範囲外のイベント ID を指定した場合はエラーとする。

指定可能範囲内で「[10.6 JP1 イベント一覧](#)」および「[10.7 JP1 イベントの詳細](#)」にないイベント ID を指定してもエラーとしない。無視する。

注意事項

- 障害監視ができなくなるため、メッセージのタイプコードが E の JP1 イベントは抑止しないことを推奨する。
- メッセージ制御機能を使用して出力する JP1 イベント（イベント ID：00005C20 および 00005C21）は抑止されない。

JP1FTP_RETRY_INTERVAL (印刷ジョブ転送のリトライ間隔の指定)

形式

[JP1FTP_RETRY_INTERVAL 印刷ジョブ転送のリトライ間隔]

機能

印刷ジョブ転送時に転送エラーが発生した場合に、再転送するまでのリトライ間隔を定義する。

オペランド

印刷ジョブ転送のリトライ間隔 ～<符号なし整数>((0~1000000)) 《3600》

印刷ジョブ転送のリトライ間隔を秒で指定する。

このオペランドの値に0を指定した場合、スプールジョブ転送デーモンは直ちに転送処理を繰り返す。

注意事項

- FTP ログイン時のパスワード設定エラーなど、自動的に回復する可能性のない恒久的な設定誤りのときにはリトライを繰り返すことになるため、JP1/FTP の設定に注意すること。
- このパラメータは、JP1FTP_RETRY_TIMES パラメータと組み合わせて使用し、運用に合わせて印刷ジョブ転送のリトライ間隔とリトライ回数を調整すること。

JP1FTP_RETRY_TIMES (印刷ジョブ転送のリトライ回数の指定)

形式

[JP1FTP_RETRY_TIMES 印刷ジョブ転送のリトライ回数]

機能

印刷ジョブ転送時に転送エラーが発生した場合に、再転送のリトライ回数を定義する。

オペランド

印刷ジョブ転送のリトライ回数 ~<符号なし整数>((0~255)) 《0》

印刷ジョブ転送のリトライ回数を指定する。

このオペランドの値に 0 を指定した場合、無条件にリトライを繰り返す。

このオペランドの値に 1 以上を指定した場合、指定回数分で転送リトライを中断する。

リトライの中断後は、その他の印刷ジョブの転送を続行する。

転送を中断した印刷ジョブを再度転送したいときは、転送ディレクトリから bjextrjob コマンドで再転送する。

注意事項

このパラメータは、JP1FTP_RETRY_INTERVAL パラメータと組み合わせて使用し、運用に合わせて印刷ジョブ転送のリトライ間隔とリトライ回数を調整すること。

JP1IM_MANAGER_CHKDELAY (イベント到達確認までの待ち時間の指定)

形式

[JP1IM_MANAGER_CHKDELAY イベント到達確認までの待ち時間]

機能

メッセージ制御機能の JP1 イベント送信処理で、JP1 イベント発行から最初のイベント到達確認を実施するまでの時間を指定できる。

なお、イベント到達確認については、「[3.15.8 メッセージ制御機能でのイベント到達確認](#)」を参照のこと。

オペランド

イベント到達確認までの待ち時間 ～<符号なし整数>((0~3000)) 《200》

JP1/IM - Manager へのイベント到達確認を行うまでの待ち時間をミリ秒で指定する。

注意事項

- イベントが到達するまでに時間が掛かる理由の 1 つに、ネットワーク・チューニング・パラメータの設定を行う no コマンドのネットワーク・チューナブル・パラメータである fasttimo に指定した値が影響することがある。このパラメータでは、fasttimo に指定した値以上を指定すること。
- このパラメータに指定する値は、最初のイベント到達確認実施までの時間である。最初のイベント到達確認が失敗した場合は、3 秒ごとに繰り返してイベント到達確認を実施する。通常メッセージ出力については、3 回まで実施する。その後のリトライは行わないで、かつ異常終了させないでジョブを続行させる。応答要求メッセージについては、MESSAGE_RETRY_TIMEOUT パラメータの指定によって、リトライ時間を指定できる。

通信処理などで異常が発生した場合、バッチジョブ実行システムの実行ログに記録を残して、あとから障害解析ができるようにする。

JP1IM_MANAGER_HOST (JP1/IM - Manager が稼働するホスト名定義)

形式

[JP1IM_MANAGER_HOST JP1/IM - Managerが稼働するホスト名]

機能

JP1 イベントを固定のマネージャホストに出力する。

オペランド

JP1/IM - Manager が稼働するホスト名 ～<任意文字列>((1～255 文字)) 《バッチジョブ実行システムが稼働するサーバの物理ホスト名》

JP1/IM - Manager が稼働する物理ホストまたは論理ホストのホスト名を指定する。指定したホストに JP1/IM - View を接続すると、JP1 イベントを確認できる。

JP1/IM - Manager のホスト名については、JP1 のマニュアルを参照のこと。このパラメータを省略した場合は、バッチジョブ実行システムが稼働するサーバで hostname コマンドを実行したときに表示されるホスト名が指定されたものと見なす。

応答待ちイベントに対する応答は、このパラメータで指定したホストからだけ応答できる。

注意事項

JP1/IM - Manager が稼働していないホストの名称をこのパラメータに指定した場合、バッチジョブ実行システムが出力する JP1 イベントが JP1/IM - View に表示されなくなる。

LOCK_DIR_RECURSIVE (排他制御機能適用時に再帰的にロックを確保するかの定義)

形式

```
[LOCK_DIR_RECURSIVE {YES|NO}]
```

機能

排他/共用制御適用時、ディレクトリに対する排他/共用要求時に再帰的にロックを確保するかどうかを定義する。

オペランド

LOCK_DIR_RECURSIVE ~{YES | NO}

排他/共用制御適用時、ディレクトリに対する排他/共用要求時に再帰的にロックを確保するかどうかを定義する。

このパラメータが省略されている場合、オペランドに YES が指定されたものとする。

YES：ディレクトリに対する排他/共用要求時、ディレクトリ配下のファイルおよびディレクトリに対し再帰的にロックを確保する。

NO：ディレクトリに対する排他/共用要求時、ディレクトリ直下のファイルに対してだけロックを確保する。

注意事項

LOCK_OPTION で ENABLE を指定していない場合、このパラメータを指定しても無視される。

LOCK_JOBMPLEX (同時に実行できるジョブ数の定義)

形式

[LOCK_JOBMPLEX ジョブ多重度数]

機能

次の機能で使用するジョブ多重度を定義する。

- バッチジョブ実行システムから起動されたジョブで割り当てる DD 要素に指定したファイルまたはディレクトリの排他/共用制御
- 実行中のジョブ情報表示機能

ジョブ多重度の対象になるジョブを次に示す。

- JOB_STATUS_DISPLAY パラメータに ENABLE オペランドを指定した場合はすべてのジョブ
- JOB_STATUS_DISPLAY パラメータに DISABLE オペランドを指定し、かつ LOCK_OPTION に ENABLE オペランドを指定した場合は、DD 要素に排他/共用制御の対象となるファイルを指定したジョブ

なお、JOB_STATUS_DISPLAY パラメータに DISABLE を指定し、かつ LOCK_OPTION に DISABLE を指定した場合は、このパラメータを指定しても無視される。

オペランド

ジョブ多重度数 ～<数字>((1~255)) 《5》

ジョブ多重度（同時に実行できるジョブ数）を指定する。

このパラメータで指定したジョブ多重度を超えたジョブ数を同時期に実行した場合、このパラメータで指定したセマフォの確保ができないジョブは、KAKC1652-W メッセージを出力し、セマフォ確保待ちとなる。セマフォ待ちが解除された場合、KAKC1653-I メッセージを出力する。

LOCK_MAXFILE (排他/共用制御で使用する排他資源管理テーブル数の定義)

形式

[LOCK_MAXFILE ファイル数]

機能

排他/共用制御で使用する排他資源管理テーブル数を定義する。なお、LOCK_OPTION に DISABLE を指定した場合は、このパラメータを指定しても無視される。

オペランド

ファイル数 ~<数字>((1~65535)) 《1275》

排他資源管理テーブルのエントリ数を指定する。

排他資源管理テーブルは、排他が必要なファイルおよびディレクトリを管理するためのテーブルであり、ジョブ実行時に排他が必要なファイルおよびディレクトリ 1 つに対して 1 エントリを使用する。このオペランドでは、システム運用中に同時に排他が必要となるファイルおよびディレクトリの最大数を指定する必要がある。

同一ジョブ内で指定した排他/共用制御の対象ファイル数が、このパラメータで指定した値を超えている場合、KAKC1728-E メッセージを出力し、ジョブは実行中止となる。

ジョブ実行時、排他資源管理テーブルが不足すると KAKC1654-W メッセージを出力し、ジョブ実行待ちとなる。ジョブ実行待ちが解除された場合、KAKC1655-I メッセージを出力する。

LOCK_OPTION (排他/共用制御を使用するかの定義)

形式

```
[LOCK_OPTION {ENABLE|DISABLE}]
```

機能

排他制御機能またはジョブ間 PREST 機能を使用するかどうかを指定する。

オペランド

LOCK_OPTION ~{ENABLE | DISABLE}

排他制御機能またはジョブ間 PREST 機能を使用するかどうかを指定する。

ENABLE : 排他制御機能を使用する。また、この指定をした場合は、ジョブ間 PREST 機能も使用できる。

DISABLE : 排他制御機能を使用しない。また、この指定をした場合は、ジョブ間 PREST 機能は使用できない。

ENABLE が指定されている場合に使用する共用メモリの見積もり計算式については、リリースノートを参照のこと。

LOCK_KEY_ID (排他制御機能および実行中のジョブ情報表示機能で使用するセマフォ, 共有メモリのキー識別情報の定義)

形式

[LOCK_KEY_ID 識別情報]

機能

排他制御機能および実行中のジョブ情報表示機能で使用するセマフォ, 共有メモリの IPC キーを生成するための識別情報を指定する。

オペランド

識別情報 ~<符号なし整数>((1-255))《1》

排他制御機能および実行中のジョブ情報表示機能で使用するセマフォ, 共有メモリの IPC キーを生成するための識別情報を符号なし整数で指定する。セマフォ, 共有メモリの IPC キーについては, 「[4.14 バッチジョブ実行システムにおける SystemV セマフォ, 共有メモリの管理](#)」を参照のこと。論理ホスト構成で複数のバッチジョブ実行システムを稼働するとき, それぞれのシステムで異なる識別情報を指定することが望ましい。

LOCK_RELEASE (排他制御機能適用時の排他解除契機定義)

形式

```
[LOCK_RELEASE {JOB|STEP}]
```

機能

ファイルおよびディレクトリの排他/共用制御で、排他/共用制御の解除契機を定義する。

オペランド

LOCK_RELEASE ~{JOB | STEP}

ファイルおよびディレクトリの排他/共用制御で、排他/共用制御の解除をジョブ終了時に行うかジョブステップ終了時に行うかを定義する。

このパラメータが省略されている場合、オペランドにJOBが指定されたものとする。

JOB : ジョブ終了時にジョブで使用したすべてのファイルおよびディレクトリの排他/共用制御を一括で解除する。

STEP : ジョブステップ終了時にジョブステップで使用したファイルおよびディレクトリのうち、以降のジョブステップで使用されないファイルおよびディレクトリの排他/共用制御を解除する。

ジョブ定義XMLファイルのEXEC要素のCOND属性や、IF要素でスキップされたジョブステップの場合も同様である。

なお、JSPEC機能で並列に実行されるジョブステップの場合、並列に実行されるジョブステップがすべて終了したときに、並列実行されたジョブステップで使用したファイルおよびディレクトリがまとめて解除される。

注意事項

- LOCK_OPTIONでENABLEを指定していない場合、本パラメータを指定しても無視される。
- STEPオペランドを指定した場合でも、JOB LIB DD要素で指定されたディレクトリはジョブ終了時まで解除されない。

LOG_DIR (実行ログ出力ディレクトリのパス名定義)

形式

[LOG_DIR パス名]

機能

多重実行するジョブのメッセージを実行ログとして1つのファイルに集める。この実行ログを出力するディレクトリのパス名を定義する。

オペランド

パス名 ~<パス名>((1~512文字) 《/opt/hitachi/bjex/log》

実行ログを出力するディレクトリのパス名を指定する。

注意事項

LOG_DIR パラメータは SPOOL_DIR パラメータと異なるファイルシステムに設定することを推奨する。

LOG_FILE_CNT (実行ログをバックアップする面数定義)

形式

[LOG_FILE_CNT 面数]

機能

実行ログをバックアップする面数を定義する。

オペランド

面数 ～<符号なし整数>((1～64))《4》

実行ログをバックアップする面数を指定する。

LOG_FILE_SIZE (実行ログを出力するファイルサイズ定義)

形式

[LOG_FILE_SIZE ファイルサイズ]

機能

実行ログを出力するファイルサイズを定義する。

オペランド

ファイルサイズ ~<符号なし整数>((1~16)) 《2》

実行ログを出力するファイルサイズを MB で指定する。

MAX_RESPONSE_REQUEST (JP1/IM - Manager に同時に受け付けられる 応答待ちイベント数定義)

形式

[MAX_RESPONSE_REQUEST 最大応答要求メッセージ数]

機能

JP1/IM - Manager に同時に受け付けられる応答要求メッセージ数を定義する。

指定値を超えた応答要求メッセージについては、すでに受け付けられた応答要求メッセージの応答後に受け付けられる。

オペランド

最大応答要求メッセージ数 ~<符号なし整数>((1~100))《5》

JP1/IM - Manager に同時に受け付けられる応答要求メッセージ数を指定する。

注意事項

JP1/IM - Manager に滞留できる応答待ちイベントは 2,000 件である。そのため、JP1IM_MANAGER_HOST パラメータで指定した JP1/IM - Manager に、応答待ちイベントを出力するホストでは、次に示す内容となるようにこのパラメータを指定する。

全ホスト※の MAX_RESPONSE_REQUEST パラメータ値の総和 ≤ 2,000

注※ JP1IM_MANAGER_HOST パラメータが同一のホストを示す。

このパラメータの指定値を超えたことにより、応答要求メッセージが待たされている場合も MESSAGE_RETRY_TIMEOUT パラメータによる中断の対象となる。このため、このパラメータの指定値は、システムで同時に受け付ける応答要求メッセージの最大数より大きい値を指定することを推奨する。

MESSAGE_REPLACE_SPACE (半角空白に置き換える 1 バイトコードの定義)

形式

```
[MESSAGE_REPLACE_SPACE コード]
```

機能

メッセージ制御機能で半角空白に置き換える 1 バイトコードを定義する。

オペランド

コード ~< 16 進数>((00~FF))

半角空白に置き換える 1 バイトコードを指定する。

複数の 1 バイトコードを置き換える場合は、パラメータを複数指定する。

同じ 1 バイトコードを重複して指定した場合は、1 つ指定した場合と同じ扱いとする。

COBOL の LOW-VALUE など出力メッセージ中に NULL 文字(X'00')文字が含まれているとメッセージ制御機能がエラーとなるが、「MESSAGE_REPLACE_SPACE 00」を定義することでエラーを防ぐことができる。

指定例

16 進数の 00 と FF を半角空白に置き換える例を示す。

```
MESSAGE_REPLACE_SPACE 00  
MESSAGE_REPLACE_SPACE FF
```

MESSAGE_RETRY_TIMEOUT (メッセージ制御機能でリトライを中断する定義)

形式

[MESSAGE_RETRY_TIMEOUT リトライ打ち切り時間]

機能

相手ホストの JP1/Base へのイベント送信が完了しない場合、応答要求メッセージのイベント送信および到達確認のリトライを中断する。応答要求メッセージを発行してから指定した打ち切り時間までにイベントの送信および到達確認ができなかった場合に、処理を終了する。

ただし、JP1 イベント API でリトライ不可能なエラーが発生した場合は無条件に処理を中断する。

なお、イベントの送信および到達確認の実施間隔は、それぞれ 3 秒である。

オペランド

リトライ打ち切り時間 ~<符号なし整数>((0-1440)) 《0》

メッセージ制御機能の応答要求メッセージのイベントの送信および到達確認について、リトライを打ち切るまでの時間を分単位で指定する。0 を指定した場合は、リトライの打ち切りを行わない。

指定したリトライ打ち切り時間までイベントの送信および到達確認を行う。この間にイベント出力が完了しなかった場合、メッセージ制御機能は RC=6 で終了する (COBOL プログラムは異常終了する)。

注意事項

- このパラメータは、メッセージ制御機能のうち、応答要求メッセージについてだけ有効となる。
- 通常メッセージについては、パラメータの指定に関係なく、常にイベント送信および到達確認を 3 回まで実施する。この間に動作が完了しなかった場合、メッセージ制御機能は RC=0 で終了する (COBOL プログラムは正常終了する)。
- 応答要求メッセージ数が、MAX_RESPONSE_REQUEST パラメータの指定値を超えて待たされている場合も中断の対象となる。したがって、このパラメータには応答要求メッセージに回答できる十分な時間を指定すること。

MSG_KEY_ID (メッセージ制御機能で使用するセマフォ, 共有メモリのキー識別情報の定義)

形式

[MSG_KEY_ID 識別情報]

機能

メッセージ制御機能で使用するセマフォ, 共有メモリの IPC キーを生成するための識別情報を指定する。

オペランド

識別情報 ~<符号なし整数>((1-255))《77》

メッセージ制御機能で使用するセマフォ, 共有メモリの IPC キーを生成するための識別情報を符号なし整数で指定する。セマフォ, 共有メモリの IPC キーについては, 「[4.14 バッチジョブ実行システムにおける SystemV セマフォ, 共有メモリの管理](#)」を参照のこと。論理ホスト構成で複数のバッチジョブ実行システムを稼働するとき, それぞれのシステムで異なる識別情報を指定することが望ましい。

MSGCLASS_DEFAULT (JOB 要素の MSGCLASS 属性での省略値定義)

形式

MSGCLASS_DEFAULT 出力クラス名

機能

ジョブ定義 XML ファイルの JOB 要素の MSGCLASS 属性での省略値を定義する。

オペランド

出力クラス名 ~<クラス名>((1~31 文字))

MSGCLASS 属性の出力クラス名を指定する。

このパラメータの指定は必須とし、指定がない場合はエラーとする。

出力クラス名を SysoutClass パラメータで定義していない場合、設定ファイルエラーとなる。

PERMANENT_FILE_DIR (恒久ファイル用ディレクトリのパス名定義)

形式

[PERMANENT_FILE_DIR パス名]

機能

恒久ファイルを定義する DD 要素の TYPE 属性値が FILE, DIR, GDG, または LIB の場合に, DSN 属性の値が相対パス指定であると, ファイルが見つからない可能性があるため, 恒久ファイルを検索するためのカレントとなるディレクトリのパス名を定義する。

ただし, DSN 属性が次に示す場合は, このディレクトリから検索はしない。

- 先頭が"/"で始まる絶対パス
- 先頭が"./"で始まる相対パス
- 先頭が\${BJEX_DIR_任意名}で始まる特定ディレクトリパス

オペランド

パス名 ~<パス名>((1~512 文字)) 《/var/opt/bjex/perm》

恒久ファイルを検索するディレクトリのパス名を指定する。

注意事項

ジョブ定義 XML ファイルでの DD 要素の DSN 属性に指定された相対パスは, このパラメータに指定されたパスからの相対パスとなる。そのため, 最終的なパスの長さが, OS の上限値を超えないように注意すること。

PERMISSION_DIR (PERMISSION 要素の DIR 属性での省略値定義)

形式

[PERMISSION_DIR ディレクトリのパーミッション]

機能

ジョブ定義 XML ファイルの PERMISSION 要素の DIR 属性での省略値を定義する。

ジョブ定義 XML ファイルで PERMISSION 要素の DIR 属性または DD 要素の FPERM 属性を指定した場合は、それらの指定が有効になる。

オペランド

ディレクトリのパーミッション ~< 8 進数>((0~1777)) 《0700》

新規作成するディレクトリのパーミッションを指定する。指定方法は chmod コマンドに従う。

PERMISSION_FILE (PERMISSION 要素の FILE 属性での省略値定義)

形式

[PERMISSION_FILE ファイルのパーミッション]

機能

ジョブ定義 XML ファイルの PERMISSION 要素の FILE 属性での省略値を定義する。

ジョブ定義 XML ファイルで PERMISSION 要素の FILE 属性または DD 要素の FPERM 属性を指定した場合は、それらの指定が有効になる。

オペランド

ファイルのパーミッション ~< 8 進数>((0~777)) 《600》

新規作成するファイルのパーミッションを指定する。指定方法は chmod コマンドに従う。

PERMISSION_SPOOL_STORE_JOB (保管するスプールジョブの権限の定義)

形式

[PERMISSION_SPOOL_STORE_JOB パーミッション]

機能

スプールジョブ保管機能を使用するとき、保管先のスプールジョブのディレクトリとファイルのパーミッションを指定する。省略した場合、保管するスプールジョブのディレクトリのパーミッションは 0700 で、SYSOUT データファイル、スプールジョブ管理ファイルは元の属性を引き継ぐ。

オペランド

パーミッション ~< 8 進数>((0~1777))

保管するスプールジョブに設定する権限を指定する。指定方法は chmod コマンドに従う。

Sticky ビットはスプールジョブのディレクトリにだけ有効であり、ファイルに対しては有効にならない。

PGM_RCABEND (ジョブステップを異常終了とするユーザプログラムの終了コードの定義)

形式

```
[PGM_RCABEND 終了コード[_終了コード]]
```

機能

ジョブステップを異常終了とするユーザプログラムの終了コードを定義する。

オペランド

終了コード ~<符号なし整数>((0~255))

ジョブステップを異常終了とする、ユーザプログラムの終了コードの値を指定する。指定する終了コードが複数ある場合、終了コードと終了コードの間を"_"で結んで昇順で指定する。

指定例

ユーザプログラムの終了コードが 1 および 12 以上の場合、ジョブステップを異常終了とする場合の設定ファイルの記述例を示す。

```
PGM_RCABEND 1  
PGM_RCABEND 12_255
```

注意事項

- ジョブステップがシグナルで異常終了したか、このパラメータによって異常終了したかを区別するには、JOBLOG に出力される KAKC1932-I メッセージ、または JP1 イベント (イベント ID: 00005C06) のエラー詳細で確認する。
- このパラメータの指定によって、SETMAXRC 要素の動作に影響を与えることはない。また、このパラメータによってジョブステップが異常終了した場合でも、該当するジョブステップより後ろに SETMAXRC 要素が指定されていれば、SETMAXRC 要素の VALUE 属性値で指定した値がジョブステップの終了コードの最大値となる。

PROCEDURE_DIR (プロシジャライブラリのパス名定義)

形式

[PROCEDURE_DIR ディレクトリパス名 [プロシジャグループ名]]

機能

プロシジャライブラリ (カタログドプロシジャを格納するディレクトリ) のパス名とグループ名を定義する。最大 255 個指定でき、同一グループの定義は連結されたライブラリとして扱う。連結定義した場合、定義順にライブラリを検索する。

オペランド

ディレクトリパス名 ~<パス名>((1~512 文字)) 《/var/opt/bjex/proc》

カタログドプロシジャのディレクトリのパス名を指定する。

プロシジャグループ名 ~<記号名称>((1~31 文字))

登録するプロシジャグループ名を指定する。

同一グループ名のパラメータを複数指定した場合、連結されたディレクトリとして処理する。

また、省略した場合はグループ名なしのグループとして扱われ、明示的にプロシジャグループの指定がないときのデフォルトとして扱われる。

注意事項

ファイル連結している場合、ジョブ定義 XML ファイルで指定したカタログドプロシジャファイルが、途中のディレクトリに存在しないとき、またはアクセス権限などのエラーがあったときでも検索を中止しないで、最後に指定されたディレクトリまで検索を続ける。

PROCGROUP_QUEUEENAME (JP1/AJS のキュー名をプロシジャグループ名として使用する定義)

形式

```
[PROCGROUP_QUEUEENAME {YES|NO}]
```

機能

JP1/AJS のキュー名をプロシジャグループ名として使用することを定義する。

この定義を有効にした場合、JP1/AJS のキュー名と一致するグループ名のプロシジャグループが定義しているとき、そのプロシジャグループのプロシジャライブラリを使用する。

オペランド

PROCGROUP_QUEUEENAME ~{YES | NO}

JP1/AJS のキュー名をプロシジャグループ名として使用するかどうか指定する。

YES : JP1/AJS のキュー名をプロシジャグループ名として使用する。

NO : JP1/AJS のキュー名をプロシジャグループ名として使用しない。

注意事項

- JP1/AJS のキュー名のプロシジャグループ名と一致しない場合、プロシジャグループは使用しないで、バッチジョブ実行システムの設定ファイルで定義したグループ名なしのカタログドプロシジャのディレクトリ下のカタログドプロシジャを使用する。
- PROCGROUP_QUEUEENAME パラメータと JOBPARM 要素でプロシジャグループが同時に指定されている場合、バッチジョブ実行システムは JOBPARM 要素の指定内容を優先的に参照し、JP1/AJS のキュー名は参照しない。
- JP1/AJS のキュー名が 32 文字以上の場合、先頭から 31 文字までをプロシジャグループ名として利用する。

RNW_FILE_METHOD (ファイル再作成の定義)

形式

```
[RNW_FILE_METHOD {REMAKE|TRUNC}]
```

機能

RNW を指定してファイルを再割り当てする場合のファイル再作成の方法を定義する。

オペランド

RNW_FILE_METHOD ~{REMAKE | TRUNC}

REMAKE : RNW を指定してファイルを再割り当てする場合、ファイルをいったん削除して再作成する。

TRUNC : RNW を指定してファイルを再割り当てする場合、ファイルを削除しないでファイルの内容を消すことでファイル容量を 0 バイトにする。

注意事項

- RNW_FILE_METHOD パラメータで REMAKE を指定した場合は、ファイルを再作成するためファイルの情報（所有者、権限、作成日付）が変更になる。RNW_FILE_METHOD パラメータで TRUNC を指定した場合は、ファイルの内容は削除されるがファイルの情報は変更されない。
- RNW_FILE_METHOD パラメータで REMAKE を指定した場合は、ファイルを再作成するためファイルが削除できることが必要である。RNW_FILE_METHOD パラメータで TRUNC を指定した場合は、ファイルの内容を変更するためファイルが書き込みできることが必要である。

SPOOL_DIR (スプールディレクトリのパス名定義)

形式

[SPOOL_DIR パス名]

機能

ジョブの実行結果（ジョブログおよびジョブステップのプログラムが出力した SYSOUT データファイル）をジョブごとにディレクトリを作成して出力するスプールディレクトリのパス名を定義する。

オペランド

パス名 ~<パス名>((1~128 文字) 《/var/opt/bjex/spool》

スプールディレクトリのパス名を指定する。

注意事項

SPOOL_DIR パラメータは LOG_DIR パラメータと異なるファイルシステムに設定することを推奨する。

SPOOL_JOB_STORE (ジョブ終了時にスプールジョブを保管する機能の定義)

形式

```
[SPOOL_JOB_STORE {YES|NO}]
```

機能

ジョブ終了時にスプールジョブを保管する機能の使用を定義する。

オペランド

SPOOL_JOB_STORE ~{YES | NO}

YES：ジョブ終了時にスプールジョブを保管する。

NO：ジョブ終了時にスプールジョブを保管しない。

SPOOL_STORE_DIR パラメータの指定がない場合、このパラメータの指定は無効になる。

SPOOL_JOB_TRANSFER (スプールジョブの転送定義)

形式

```
[SPOOL_JOB_TRANSFER {YES|NO}]
```

機能

スプールジョブを転送するかどうかを指定する。

NO を指定した場合、JP1/FTP がインストールされていないときでも bjexpd コマンドが動作する。また、ジョブ定義 XML ファイルで HOLD=NO または bjexchjob コマンドで -R オプションを指定した場合でも、印刷ジョブを作成しない。

オペランド

SPOOL_JOB_TRANSFER ~{YES | NO}

スプールジョブを転送するかどうかを指定する。

YES：スプールジョブを転送する。

NO：すべての出力クラスの SYSOUT データを転送しない。

SPOOL_STORE_DIR (スプールジョブの保管先の定義)

形式

[SPOOL_STORE_DIR パス名]

機能

スプールジョブを保管するときの保管先のディレクトリのパス名を指定する。

オペランド

パス名 ~<パス名>((1~128 文字))

スプールジョブを保管する保管先ディレクトリのパス名を指定する。保管先のパスには 1777 のパーミッションを設定する必要がある。ただし、Sticky ビットはシステムの運用方針に合わせて設定する。

SPOOL_STORE_DIRNAME (スプールジョブの保管先ディレクトリ名称の定義)

形式

[SPOOL_STORE_DIRNAME書式]

機能

スプールジョブを保管するときの保管先ディレクトリ名称を定義する。

オペランド

SPOOL_STORE_DIRNAME 書式 ~ 《%j_%h_%i_%u》

スプールジョブの保管先ディレクトリ名称を決定するための書式を指定する。

書式 ~変換指定子 [_ (アンダースコア) 変換指定子] *

書式には、次の表に示す変換指定子を使用し、各変換指定子を"_" (アンダースコア) で結んだ形で指定する。各変換指定子を重複して指定できない。

表 8-6 変換指定子と意味

変換指定子	意味*	必須/任意
%j	ジョブ名に変換する。	○
%h	ホスト名に変換する。	○
%i	ジョブ識別子に変換する。	◎
%u	ユーザ名に変換する。	○
%a	AJSJOBNAME に変換する。	○

(凡例)

◎：指定は必須。

○：指定は任意。

注※ 各値の内容については、「[3.16.4 保管先ディレクトリの名称](#)」を参照のこと。

指定例

- 保管先ディレクトリ名称を、ジョブ名_ホスト名_ジョブ識別子_ユーザ名_AJSJOBNAME とする場合の記述例を示す。

```
SPOOL_STORE_DIRNAME %j_%h_%i_%u_%a
```

- 保管先ディレクトリ名称を、AJSJOBNAME_ジョブ識別子_ホスト名とする場合の記述例を示す。

```
SPOOL_STORE_DIRNAME %a_%i_%h
```

SPOOL_STORE_SUBDIR (スプールジョブの保管先のサブディレクトリの定義)

形式

```
[SPOOL_STORE_SUBDIR {DATE|DATEHOUR|NO}]
```

機能

スプールジョブを保管するときの保管先ディレクトリのサブディレクトリの作成方法を指定する。

オペランド

SPOOL_STORE_SUBDIR ~{DATE | DATEHOUR | NO}

設定ファイルの SPOOL_STORE_DIR パラメータで指定したディレクトリ下に作成するサブディレクトリの作成方法を指定する。サブディレクトリのパーミッションは 1777 で作成する。このパラメータが省略されている場合、オペランドに NO が指定されたものとする。

DATE：スプールジョブ保管先のディレクトリの下にジョブ実行開始日を名前としたサブディレクトリ ("yyyymmdd"の形式) の下にスプールジョブを保管する。

DATEHOUR：スプールジョブ保管先のディレクトリの下にジョブ実行開始日+ジョブ実行開始時間を名前としたサブディレクトリ ("yyyymmdd-hh00"の形式 (hh の値は 00~23)) の下にスプールジョブを保管する。

NO：サブディレクトリの作成をしないで、設定ファイルの SPOOL_STORE_DIR パラメータで指定した保管先のディレクトリ下に保管する。

注意事項

ジョブを投入した際、ジョブ間ファイル排他待ちが発生した場合は、排他待ちが解除された後の日付と時刻を使用してサブディレクトリを作成する。

STDERRCLASS_DEFAULT (標準エラー出力を格納するファイルの出力クラスの省略値定義)

形式

```
[STDERRCLASS_DEFAULT 出力クラス名]
```

機能

JOB 要素の STDERRCLASS 属性と、EXEC 要素の STDERRCLASS 属性を両方とも省略した場合の省略値を定義する。

オペランド

出力クラス名 ~<クラス名>((1~31 文字))

ユーザプログラムの標準エラー出力を格納するファイルの出力クラス名を指定する。この指定を省略した場合、出力クラスはジョブのメッセージクラスを仮定する。出力クラス名を SysoutClass パラメータで定義していない場合、KAKC0405-E メッセージを出力して設定ファイルエラーとなる。

指定例

```
STDERRCLASS_DEFAULT A
```

STDOUTCLASS_DEFAULT (標準出力を格納するファイルの出力クラスの省略値定義)

形式

```
[STDOUTCLASS_DEFAULT 出力クラス名]
```

機能

JOB 要素の STDOUTCLASS 属性と、EXEC 要素の STDOUTCLASS 属性を両方とも省略した場合の省略値を定義する。

オペランド

出力クラス名 ~<クラス名>((1~31 文字))

ユーザプログラムの標準出力を格納するファイルの出力クラス名を指定する。

STDOUTCLASS_DEFAULT パラメータの指定を省略した場合、出力クラスはジョブのメッセージクラスを仮定する。出力クラス名を SysoutClass パラメータで定義していない場合、KAKC0405-E メッセージを出力して設定ファイルエラーとなる。

指定例

```
STDOUTCLASS_DEFAULT A
```

SYSOUT_APPEND_WRITE (SYSOUT データファイルへの書き込みモード定義)

形式

```
[SYSOUT_APPEND_WRITE {YES|NO}]
```

機能

SYSOUT データファイルへの書き込みモードを定義する。

SYSOUT_APPEND_WRITE パラメータは、EXEC 要素の LANG 属性に"COBOL"を指定したジョブステップに対して有効となる。

オペランド

SYSOUT_APPEND_WRITE ~{YES | NO}

COBOL プログラムが SYSOUT データファイルへの書き込みを行うときの書き込みモードを定義する。SYSOUT_APPEND_WRITE パラメータの指定が省略されている場合、オペランドに NO が指定されたものとする。

YES : COBOL プログラムは SYSOUT データファイルに対して追加モードで書き込みを行う。

NO : COBOL プログラムは SYSOUT データファイルに対してオープンで指定されたモードで書き込みを行う。

SysoutClass (出力クラスの属性値定義)

形式

```
<SysoutClass クラス名>
[CHARS {文字配列テーブル名|文字セットイメージ名}]
[COPIES 複写回数]
[COPYG グループ複写回数]
[DEST 出力先識別子]
[DUMMY {YES|NO}]
[ENCODE {SJIS|UTF8}]
[FCB FCB名称]
[FCDFLT デフォルト機能キャラクタ名]
[FORMS 書式番号]
[FTPCNTL スプールジョブ管理ファイル転送先ディレクトリ名]
[FTPDATA SYSOUTデータファイル転送先ディレクトリ名]
[FTYPE {1|2}]
[GAIJI1 レベル1外字名]
[HOLD {YES|NO}]
[JP1FTPCARD 伝送カード名]
[KEEP {YES|NO}]
[MODE {AN|K}]
[OVERC 適用枚数]
[OVERLAY 書式オーバーレイ名]
[RECFM {A|M|N|X}]
[TRANSFER_EMPTY_FILE {YES|NO}]
[TRUNC {YES|NO}]
</SysoutClass>
```

機能

出力クラスの属性値を定義する。

オペランド

クラス名 ～<記号名称>((1～31 文字))

出力クラス名を指定する。JOB 要素の MSGCLASS 属性に指定するクラス名または SYSOUT データを定義する DD 要素の TYPE 属性値が SYSOUT の場合、SYSOUT 属性に指定するクラス名となる。

CHARS {文字配列テーブル名 | 文字セットイメージ名} ～<記号名称>((1～4 文字))

文字セットイメージ名または文字配列テーブル名を指定する。このオペランドの指定は、SYSOUT データを印刷する場合に、PDE で有効となる。

COPIES 複写回数 ～<符号なし整数>((1～255))

複写回数を指定する。このオペランドの指定は、SYSOUT データを印刷する場合に、PDE で有効となる。

COPYG グループ複写回数 ～<符号なし整数>((1～255))

グループ複写回数を指定する。このオペランドの指定は、SYSOUT データを印刷する場合に、PDE で有効となる。

DEST 出力先識別子 ～<記号名称>((1～31 文字))

出力先識別子を指定する。このオペランドの指定は、SYSOUT データを印刷する場合に、PDE で有効となる。

DUMMY ～{YES | NO}

ジョブ実行後、スプールジョブを転送しないで削除するかどうかを指定する。

YES：HOLD および KEEP オペランド、または DD 要素の HOLD および KEEP 属性の指定に関係なく、スプールジョブを転送しないで削除する。

NO：スプールジョブの転送と削除は、HOLD および KEEP オペランド、または DD 要素の HOLD および KEEP 属性の指定に従う。

ENCODE ～{SJIS | UTF8}

ジョブ定義 XML ファイルで指定がないときに仮定する SYSOUT データファイルのエンコード種別を指定する。この指定は PDE に対しても意味がある指定である。

SJIS：シフト JIS コード

UTF8：UTF-8 コード

FCB FCB 名称 ～<記号名称>((1～4 文字))

FCB 名を指定する。このオペランドの指定は、SYSOUT データを印刷する場合に、PDE で有効となる。

FCDFLT デフォルト機能キャラクタ名 ～<記号名称>((1～4 文字))

デフォルト機能キャラクタイメージ名を指定する。このオペランドの指定は、SYSOUT データを印刷する場合に、PDE で有効となる。

FORMS 書式番号 ～<記号名称>((1～4 文字))

書式番号を指定する。

FTPCNTL スプールジョブ管理ファイル転送先ディレクトリ名 ～<任意文字列>((1～128 文字))

スプールジョブ管理ファイル転送先のディレクトリ名を指定する。

JP1FTPCARD オペランドの指定がある場合、このパラメータの指定は必須とし、指定がない場合はエラーとする。ただし、DUMMY オペランドに YES を指定している場合、このパラメータは JP1FTPCARD オペランドを指定していても省略できる。

FTPDATA SYSOUT データファイル転送先ディレクトリ名 ～<任意文字列>((1～128 文字))

SYSOUT データファイル転送先のディレクトリ名を指定する。

JP1FTPCARD オペランドの指定がある場合、このパラメータの指定は必須とし、指定がない場合はエラーとする。ただし、DUMMY オペランドに YES を指定している場合、このパラメータは JP1FTPCARD オペランドを指定していても省略できる。

FTYPE ～{1 | 2}

SYSOUT データファイルの形式を指定する。

この指定は JOBLOG, SYSMMSG, XML, STDOUT, および STDERR には無効であり、これらのファイルはテキストファイルとして処理される。

1：PDE ファイル形式 1

テキストファイルなどの任意のファイル形式

2：PDE ファイル形式 2

COBOL2002 で PDE による書式、行制御印刷を行うファイル形式

GAJ11 レベル 1 外字名 ～<記号名称>((1～4 文字))

PDE と連携して外字フォント名を指定した印刷を行う場合に、レベル 1 外字名を指定する。レベル 1 外字の名称は、VOS3 のレベル 1 外字イメージ名の指定に対応した機能であることに由来する。指定を省略した場合は、PDE の設定に従って印刷処理が行われる。

HOLD ～{YES | NO}

ジョブ実行後、SYSOUT データを即時印刷するか、または出力保留を行い、あとからコマンドの指示によって印刷するかどうかを指定する。なお、EUR 帳票印刷はこの指定に関係なく、ジョブ終了時に印刷する。

YES：出力保留にして即時印刷しない。

NO：出力保留にしないで、即時印刷する。

JP1FTPCARD 伝送カード名 ～<任意文字列>((1～20 文字))

スプールジョブを転送する場合に使用する伝送カード名を指定する。

このオペランドを省略した場合、この出力クラスの SYSOUT データは転送しない。また、SPOOL_JOB_TRANSFER パラメータで NO を指定した場合は、すべての出力クラスの SYSOUT データファイルを転送しない。この場合、エラーメッセージは出力しない。

KEEP ～{YES | NO}

SYSOUT データファイルを出力後保存するかどうかを指定する。

YES：SYSOUT データファイルをスプールジョブ転送サービスに転送指示後、スプールから削除しない (SYSOUT データファイルを出力後保存属性にする)。

NO：SYSOUT データファイルをスプールジョブ転送サービスに転送指示後、スプールから削除する (SYSOUT データファイルを出力後保存属性にしない)。

MODE ～{AN | K}

プリンタモードを指定する。このオペランドの指定は、SYSOUT データを印刷する場合に、PDE で有効となる。

AN：ノンインパクトプリンタモード

K：漢字プリンタモード

OVERC 適用枚数 ～<符号なし整数>((1～255))

書式オーバーレイ情報のイメージ名の適用枚数を指定する。このオペランドの指定は、SYSOUT データを印刷する場合に、PDE で有効となる。

OVERLAY 書式オーバーレイ名 ～<記号名称>((1～4 文字))

書式オーバーレイ情報のイメージ名を指定する。このオペランドの指定は、SYSOUT データを印刷する場合に、PDE で有効となる。

RECFM ～{A | M | N | X}

SYSOUT データの制御文字形式を指定する。

この指定はJOBLOG, SYSMMSG, XML, STDOUT, およびSTDERR の設定には無効であり、これらのファイルは制御文字なしとして処理される (RECFM=X と同等)。

A: ANSI 制御文字。

M: 機械制御文字。

N: 制御文字なし。

X: RECFM 属性を設定しない。RECFM の属性は PDE や Prinfinamanager の設定に従う。

TRANSFER_EMPTY_FILE ~{YES | NO}

SYSOUT データファイルを転送する場合、ファイルサイズが 0 バイトのファイルを転送するかどうかを指定する。SYSOUT データを PDE と連携して印刷する場合、YES を指定してはならない。

YES: 転送する。転送後は KEEP 属性に従って削除または保存処理をする。

NO: 転送しない。

TRUNC ~{YES | NO}

PDE ファイル形式 2 の SYSOUT データの行末の空白を削除するかどうかを指定する。この指定は、ほかの指定との整合性はチェックしない。また、PDE ファイル形式 2 以外に指定しても無効であり、行末の空白は削除しない。

YES: 行末の空白を削除する。

NO: 行末の空白を削除しない。

注意事項

- DUMMY オペランドに YES を指定している場合は、HOLD および KEEP オペランド、または DD 要素の HOLD および KEEP 属性の指定は無効になる。スプールジョブはジョブ終了後に削除される。
- TRANSFER_EMPTY_FILE オペランドは、DUMMY オペランド、HOLD オペランド、および DD 要素の HOLD 属性と併用できる。バッチジョブ実行システムは、それぞれのオペランドや属性に従って動作する。

TEMP_FILE_DIR (一時ファイルディレクトリのパス名定義)

形式

[TEMP_FILE_DIR パス名]

機能

一時ファイルまたは PREST ファイルを格納するディレクトリのパス名を定義する。

一時ファイルとはジョブ中で作成され、ジョブ終了時にはすべて消去する暫定的に利用するファイルである。DD 要素の TYPE 属性値が TEMP または TEMPISAM である場合に、このパス名配下に一時ファイルを作成する。

DD 要素の TYPE 属性値が PREST または TEMPPREST である場合に、このパス名配下に PREST ファイルを作成する。

TYPE 属性値に TEMP, TEMPISAM, または TEMPPREST を指定した場合に作成されるファイルの名称については次に示す個所を参照のこと。

- TYPE 属性値に TEMP を指定した場合に作成されるファイルの名称
[3.4.4 一時ファイル]
- TYPE 属性値に TEMPISAM を指定した場合に作成されるファイルの名称
[3.4.5 一時 ISAM ファイル]
- TYPE 属性値に TEMPPREST を指定した場合に作成されるファイルの名称
[3.12.2(2) ジョブステップ間 PREST ファイル]

オペランド

パス名 ~<パス名>((1~512 文字)) 《/tmp》

一時ファイルまたは PREST ファイルを格納するディレクトリのパス名を指定する。

TEMP_FILE_DIR_XML_USE (一時ファイルディレクトリのパス名を DD 要素の DSN 属性で指定可能とする定義)

形式

```
[TEMP_FILE_DIR_XML_USE {YES|NO} ]
```

機能

ジョブ定義 XML ファイルの DD 要素の DSN 属性で指定したディレクトリ名称を、一時ファイルを格納するディレクトリとするかどうかを指定する。

オペラント

TEMP_FILE_DIR_XML_USE ~ {YES | NO}

ジョブ定義 XML ファイルの DD 要素の DSN 属性で指定したディレクトリ名称を、一時ファイルを格納するディレクトリとするかどうかを指定する。

このパラメータが省略されている場合、オペラントに NO を指定されたものとする。

YES : ジョブ定義 XML ファイルの DD 要素の DSN 属性で指定したディレクトリ名称を、一時ファイル格納先ディレクトリパス名とする。ディレクトリパス名は、絶対パスで指定する必要がある。相対パスあるいは DSN 属性の指定がない場合、設定ファイルの TEMP_FILE_DIR パラメータで指定したディレクトリパスを一時ファイル格納先ディレクトリパス名とする。

NO : 一時ファイル格納先ディレクトリパス名は、設定ファイルの TEMP_FILE_DIR パラメータで指定したディレクトリパス名とする。

TRACE_DIR (トレースを出力するディレクトリのパス名定義)

形式

[TRACE_DIR パス名]

機能

トレースを出力するディレクトリのパス名を定義する。

オペランド

パス名 ~<パス名>((1~512文字)《/opt/hitachi/bjex/trace》)

トレースを出力するディレクトリのパス名を指定する。

TRACE_FILE_CNT (トレース面数の定義)

形式

[TRACE_FILE_CNT 面数]

機能

トレースを出力する面数を定義する。

オペランド

面数 ～<符号なし整数>((1~64)) 《4》

トレースを出力する面数を指定する。通常は 4 を指定すること。

TRACE_FILE_SIZE (トレースファイルサイズの定義)

形式

[TRACE_FILE_SIZE ファイルサイズ]

機能

トレースを出力するファイルサイズを定義する。

オペランド

ファイルサイズ ~<符号なし整数>((1~16)) 《2》

トレースを出力するファイルサイズを MB で指定する。通常は 2 を指定すること。

TRACE_LEVEL (トレース出力レベルの定義)

形式

[TRACE_LEVEL トレースレベル]

機能

トレースを出力するレベルを定義する。

オペランド

トレースレベル ~<符号なし整数>((0, 10, 20, 30)) 《0》

トレースを出力するレベルを指定する。数値が大きいほど詳細なトレースを出力する。通常は 0 を指定すること。

設定ファイルの使用例

設定ファイルの使用例を次に示す。

```
## filesystem configure
SPOOL_DIR      /var/opt/bjex/spool
LOG_DIR        /opt/hitachi/bjex/log
PROCEDURE_DIR  /var/opt/bjex/proc
PERMANENT_FILE_DIR /var/opt/bjex/perm
TEMP_FILE_DIR  /tmp
TRACE_DIR      /opt/hitachi/bjex/trace
BJEX_DIR_GYOMU /home/gyomu
##Select the SysoutClass name.###
MSGCLASS_DEFAULT A
##sysout data
<SysoutClass A>
HOLD           NO
JP1FTPCARD    CARD-A
FTPCNTL       BJEX/CNTL
FTPDATA       BJEX/DATA
FCB           A4L6
MODE          K
</SysoutClass>
```

9

ジョブ定義 XML ファイル

バッチジョブ実行システムのジョブ定義 XML ファイルの次に示す内容について説明する。

- ・ ジョブ定義 XML ファイルの文法
- ・ ジョブ定義 XML ファイル内に指定できる要素および要素内に指定できる属性

ジョブ定義 XML ファイルの文法

ここでは、ジョブ定義 XML ファイルの文法について説明する。

記述方法

ジョブ定義 XML ファイルの記述方法を示す。

ジョブ定義 XML ファイルは、W3C 勧告「拡張可能なマーク付け言語 (XML) 1.0」REC-xml-980210 のサブセットである。このマニュアルに特に記述がなければ、W3C 勧告「拡張可能なマーク付け言語 (XML) 1.0」REC-xml-980210 に従って記述する。ただし、日本語はコメント、CDATA、および日本語が使用できる属性値内だけで使用できる。

実体参照

ジョブ定義 XML ファイルでは、次に示す文字を属性値に直接記述するのではなく、実体参照で記述しなければならない。また、要素の内容も、CDATA セクション (文字列「<![CDATA[と]]>」の囲み) を使用しない場合は、実体参照で記述しなければならない。

表 9-1 直接記述できない文字と実体参照

文字	実体参照
&	&
<	<
>	>
' (シングルクォート)	'
" (ダブルクォート)	"

注 ジョブ定義 XML ファイルでは、ユーザが任意の実体参照を定義して記述することはできない。

例

文字列 AB&CD<EF>GH'IJ" と記述した場合、ジョブ定義 XML ファイルでは次のように記述する。
記述したい文字列

```
AB&CD<EF>GH'IJ"
```

ジョブ定義 XML ファイルでの記述

```
AB&amp;CD&lt;EF&gt;GH&apos;IJ&quot;
```

表記法

ジョブ定義 XML ファイルの文法および使用例について次に示す。

文法

ジョブ定義 XML ファイルの文法を次に示す。

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec [version="DTDバージョン番号" [os="OS種別"] >
<JOB要素>
  [<JOBPARM要素/>]
  [<PERMISSION要素/>]
  [<DD要素/>|<SETENV要素/>][<DD要素/>|<SETENV要素/>][...]
  [<PROC要素>
    [<SYMBOL要素/>][...]
    <STEP要素>
      <EXEC要素/>
      [<DD要素/>|<SETENV要素/>|<SETENVFILE要素/>][...]
    </STEP>
    [<SETMAXRC要素/>]
    [<IF要素/>|<ELSE要素/>|<ENDIF要素/>][...]
    [<STEP要素/>|<SETMAXRC要素/>][<IF要素/>|<ELSE要素/>|<ENDIF要素/>][...][...]
  </PROC>][...]
  <STEP要素>
    <EXEC要素/>|
    <CALL要素>
      [<SETSYMBOL要素/>][...]
      [<REPLACE要素>
        [<EXEC要素/>]
        [<DD要素/>][...]
        [<REPLACESTEP要素>
          [<EXEC要素/>]
          [<DD要素/>][...]
        </REPLACESTEP>][<REPLACESTEP要素/>][...]
      </REPLACE>
    </CALL>
    [<DD要素/>|<SETENV要素/>|<SETENVFILE要素/>][...]
  </STEP>
  [<SETMAXRC要素/>]
  [<IF要素/>|<ELSE要素/>|<ENDIF要素/>][...]
  [<STEP要素/>|<SETMAXRC要素/>][<IF要素/>|<ELSE要素/>|<ENDIF要素/>][...][...]
</JOB>|
[<PROC要素>
  [<SYMBOL要素/>][...]
  <STEP要素>
    <EXEC要素/>
    [<DD要素/>|<SETENV要素/>|<SETENVFILE要素/>][...]
  </STEP>
  [<SETMAXRC要素/>]
  [<IF要素/>|<ELSE要素/>|<ENDIF要素/>][...]
  [<STEP要素/>|<SETMAXRC要素/>][<IF要素/>|<ELSE要素/>|<ENDIF要素/>][...][...]
</PROC>]
</HitachiBatchJobExec>

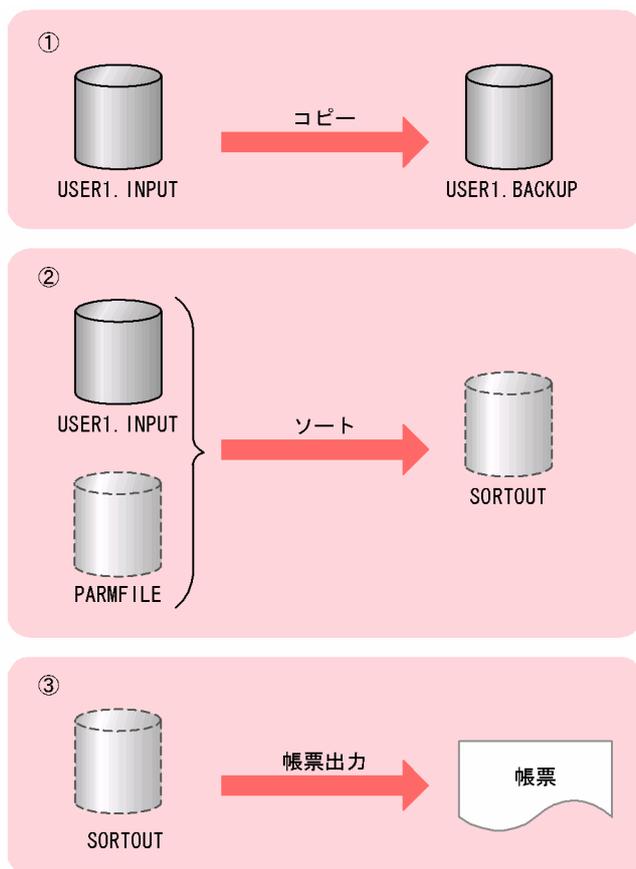
```

使用例

ジョブ定義 XML ファイルの例

入力ファイル USER1.INPUT をバックアップファイル USER1.BACKUP にコピーしたあと、入力ファイル USER1.INPUT を入力にソートし、ソート結果を COBOL プログラムの入力として帳票出力するプログラムを実行する場合の流れとジョブ定義 XML ファイルの例（項番は項目の説明の番号と対応している）を次に示す。

図 9-1 帳票出力するプログラムを実行する場合の流れ



```

<?xml version="1.0" encoding="Shift-JIS" ?> . . . 1
<HitachiBatchJobExec version="1.3" os="unix" > . . . 2
  <JOB NAME="SAMPLE1" COND="0, NE" > . . . 3
    <DD NAME="JOBLIB" TYPE="LIB" DSN="/home/bjex/batch/job1/bin" /> . . . 4
    <DD NAME="JOBLIB" TYPE="LIB" DSN="/home/bjex/batch/job2/bin" /> . . . 4
    <!-- ***** -->
    <!-- ステップ1. 入力ファイルUSER1.INPUTをUSER1.BACKUPにコピー -->
    <!-- ***** -->
    <STEP NAME="BACKUP" > . . . 5
      <EXEC PGM="*" > . . . 6
        <![CDATA[!cp ${DDN_SYSUT1} ${DDN_SYSUT2} !]]> . . . 7
      </EXEC> . . . 8
      <DD NAME="SYSUT1" TYPE="FILE" DSN="USER1.INPUT" DISP="SHR" /> . . . 9
      <DD NAME="SYSUT2" TYPE="FILE" DSN="USER1.BACKUP" DISP="OLD" /> . . . 10
    </STEP> . . . 11
    <!-- ***** -->
    <!-- ステップ2. 入力ファイルUSER1.INPUTをソートし -->
    <!-- 結果を一時ファイルに出力 -->
    <!-- ***** -->
    <STEP NAME="SORT" > . . . 12
      <EXEC PGM="*" > . . . 13
      <![CDATA[!
exsort -p "${DDN_PARMFILE}"
!]]> . . . 14
      </EXEC> . . . 15
      <DD NAME="SORTIN" TYPE="FILE" DSN="USER1.INPUT" DISP="SHR" /> . . . 16
      <DD NAME="SORTOUT" TYPE="TEMP" DSN="SORTOUT" DISP="," PASS" /> . . . 17
      <DD NAME="PARMFILE" TYPE="DATA"> . . . 18
      <![CDATA[!
-function sort . . . 19
-locatajst . . . 19
-cmdopt DUP:OVWRT . . . 19
-key j+1-13 . . . 19
-inpfile fileorg=S . . . 19
      reclen=1040 . . . 19
      file="%DDN_SORTIN%" . . . 19
-outfile file="%DDN_SORTOUT%" . . . 19
-sorttmpdir /tmp . . . 19
!]]> . . . 19
      </DD> . . . 20
    </STEP> . . . 21
    <!-- ***** -->
    <!-- ステップ3. 一時ファイルを入力に帳票作成 -->
    <!-- ***** -->
    <STEP NAME="COBOL" > . . . 22
      <EXEC PGM="UAP" LANG="COBOL" /> . . . 23
      <DD NAME="INPUT" TYPE="TEMP" DSN="SORTOUT" DISP="OLD, DELETE" /> . . . 24
      <DD NAME="SYS001" TYPE="SYSOUT" SYSOUT="A" FCB="M6TN" /> . . . 25
    </STEP> . . . 26
  </JOB> . . . 27
</HitachiBatchJobExec> . . . 28

```

1. XML 宣言を示す。
2. ジョブ定義 XML ファイルのルート要素の開始を示す。
3. JOB 要素の開始を示す。ジョブステップの終了コードが 0 でない場合にジョブは終了し、後続ジョブステップの実行を実施しないように指定する。
4. DD 要素を示す。実行するプログラムの検索パスを指定する。
5. STEP 要素の開始を示す。ジョブステップ BACKUP を開始する。
6. EXEC 要素の開始を示す。ジョブステップ BACKUP で、CDATA セクションで示したプログラム（ファイル USER1.INPUT をファイル USER1.BACKUP にコピーする）を実行するように指定する。

7. CDATA セクションの内容を示す。
8. EXEC 要素の終了を示す。
9. DD 要素を示す。TYPE="FILE"は恒久ファイルを示す。コピー元ファイルとして恒久ファイル USER1.INPUT を指定する。
10. DD 要素を示す。TYPE="FILE"は恒久ファイルを示す。コピー先ファイルとして恒久ファイル USER1.BACKUP を指定する。
11. STEP 要素の終了を示す。
12. STEP 要素の開始を示す。ジョブステップ SORT を開始する。
13. EXEC 要素の開始を示す。ジョブステップ SORT で、CDATA セクションで示したソートプログラム SORT EE (exsort -p パラメータファイル) を実行するように指定する。
14. CDATA セクションの内容を示す。
15. EXEC 要素の終了を示す。
16. DD 要素を示す。TYPE="FILE"は恒久ファイルを示す。ソートプログラムの入力ファイル (パラメータファイルの file オペランド値) を設定する。
17. DD 要素を示す。TYPE="TEMP"は一時ファイルを示す。ソートプログラムの出力ファイルを指定する。DISP=",PASS"で、後続ジョブステップに渡すことを示す。
18. DD 要素の開始を示す。TYPE="DATA"は CDATA セクションで指定したデータを一時ファイルに割り当てプログラムの入力データとすることを示す。ソートパラメータを CDATA セクションで指定する。
19. CDATA セクションの内容を示す。
20. DD 要素の終了を示す。
21. STEP 要素の終了を示す。
22. STEP 要素の開始を示す。ジョブステップ COBOL を開始する。
23. EXEC 要素を示す。ジョブステップ COBOL で、UAP プログラムの実行を指定する。UAP プログラムは COBOL 言語プログラムであることを LANG="COBOL"で示す。
24. DD 要素を示す。TYPE="TEMP"は一時ファイルを示す。ジョブステップ SORT で割り当てられた一時ファイルを UAP プログラムの入力データとして設定する。
25. DD 要素を示す。TYPE="SYSOUT"は SYSOUT データファイルを割り当てることを示す。UAP プログラムの出力データは SYSOUT データファイルに出力する。
26. STEP 要素の終了を示す。
27. JOB 要素の終了を示す。
28. ルート要素の終了を示す。

カタログドプロシジャ実行のジョブ定義 XML ファイルの例

カタログドプロシジャの使用例を次に示す (項番は項目の説明の番号と対応している)。

<?xml version="1.0" encoding="Shift_JIS" ?>	...	1
<HitachiBatchJobExec version = "1.3" os="unix" >	...	2
<PROC NAME="SAMPLE2_PROC" >	...	3
<STEP NAME="SAMPLE2_PROC_STEP1">	...	4
<EXEC PGM="/bin/echo" PARM="%%%%%%%%SAMPLE2_PROC_STEP1%%%%%%%%" />	...	5
</STEP>	...	6
</PROC>	...	7
</HitachiBatchJobExec>	...	8

1. XML 宣言を示す。
2. ジョブ定義 XML ファイルのルート要素の開始を示す。
3. PROC 要素の開始を示す。
4. STEP 要素の開始を示す。
5. EXEC 要素を示す。PARM 属性の指定値を標準出力に出力するプログラムを指定する。
6. STEP 要素の終了を示す。
7. PROC 要素の終了を示す。
8. ルート要素の終了を示す。

カタログドプロシジャの呼び出しの使用例を次に示す（項番は項目の説明の番号と対応している）。

<?xml version="1.0" encoding="Shift_JIS" ?>	...	1
<HitachiBatchJobExec version = "1.3" os="unix" >	...	2
<JOB NAME="SAMPLE2_JCL" >	...	3
<STEP NAME="STEP1" >	...	4
<EXEC PGM="/bin/echo" PARM="%%%%%%%%SAMPLE2_JCL_STEP1%%%%%%%%" />	...	5
</STEP>	...	6
<STEP NAME="STEP2" >	...	7
<CALL NAME="SAMPLE2_PROC" />	...	8
</STEP>	...	9
<STEP NAME="STEP3" >	...	10
<CALL NAME="SAMPLE2_PROC" >	...	11
<REPLACE>	...	12
<REPLACESTEP NAME="SAMPLE2_PROC_STEP1">	...	13
<EXEC PARM="%%%%%%%%replace_SAMPLE2_PROC_STEP1%%%%%%%%" />	...	14
</REPLACESTEP>	...	15
</REPLACE>	...	16
</CALL>	...	17
</STEP>	...	18
</JOB>	...	19
</HitachiBatchJobExec>	...	20

1. XML 宣言を示す。
2. ジョブ定義 XML ファイルのルート要素の開始を示す。
3. JOB 要素の開始を示す。
4. STEP 要素の開始を示す。ジョブステップ STEP1 を開始する。
5. EXEC 要素を示す。PARM 属性の指定値を標準出力に出力するプログラムを指定する。
6. STEP 要素の終了を示す。
7. STEP 要素の開始を示す。ジョブステップ STEP2 を開始する。
8. カタログドプロシジャ SAMPLE2_PROC 呼び出しのための CALL 要素を示す。

9. STEP 要素の終了を示す。
10. STEP 要素の開始を示す。ジョブステップ STEP3 を開始する。
11. カタログドプロシジャ SAMPLE2_PROC 呼び出しのための CALL 要素の開始を示す。
12. 呼び出されるカタログドプロシジャの EXEC 要素の属性を変更する REPLACE 要素の開始を示す。
13. 変更対象のジョブステップを示すための REPLACESTEP 要素の開始を示す。
14. EXEC 要素の PARM 属性値の変更を示す。
15. REPLACESTEP 要素の終了を示す。
16. REPLACE 要素の終了を示す。
17. CALL 要素の終了を示す。
18. STEP 要素の終了を示す。
19. JOB 要素の終了を示す。
20. ルート要素の終了を示す。

カタログドプロシジャ実行のジョブ定義 XML ファイルの実行例を次に示す。

```

%%%%%%%%SAMPLE2_JCL_STEP1%%%%%%%%
%%%%%%%%SAMPLE2_PROC_STEP1%%%%%%%%
%%%%%%%%replace_SAMPLE2_PROC_STEP1%%%%%%%%

```

入カストリームからのプロシジャ呼び出し例

入カストリームからのプロシジャ呼び出しのためのジョブ定義 XML ファイルの例を次に示す（項番は項目の説明の番号と対応している）。

```

<?xml version="1.0" encoding="Shift_JIS" ?> . . . 1
<HitachiBatchJobExec version = "1.3" os="unix" > . . . 2
<JOB NAME="SAMPLE3_JCL" > . . . 3
  <PROC NAME="SAMPLE3_PROC" > . . . 4
    <STEP NAME="SAMPLE3_PROC_STEP1"> . . . 5
      <EXEC PGM="/bin/echo" PARM="%%%%%%%%SAMPLE3_PROC_STEP1%%%%%%%%" /> . . . 6
    </STEP> . . . 7
  </PROC> . . . 8
  <STEP NAME="STEP1" > . . . 9
    <EXEC PGM="/bin/echo" PARM="%%%%%%%%SAMPLE3_JCL_STEP1%%%%%%%%" /> . . . 10
  </STEP> . . . 11
  <STEP NAME="STEP2" > . . . 12
    <CALL NAME="SAMPLE3_PROC" /> . . . 13
  </STEP> . . . 14
  <STEP NAME="STEP3" > . . . 15
    <CALL NAME="SAMPLE3_PROC" > . . . 16
      <REPLACE> . . . 17
        <REPLACESTEP NAME="SAMPLE3_PROC_STEP1"> . . . 18
          <EXEC PARM="%%%%%%%%replace SAMPLE3_PROC_STEP1%%%%%%%%" /> . . . 19
        </REPLACESTEP> . . . 20
      </REPLACE> . . . 21
    </CALL> . . . 22
  </STEP> . . . 23
</JOB> . . . 24
</HitachiBatchJobExec> . . . 25

```

1. XML 宣言を示す。

2. ジョブ定義 XML ファイルのルート要素の開始を示す。
3. JOB 要素の開始を示す。
4. PROC 要素の開始を示す。
5. STEP 要素の開始を示す。
6. EXEC 要素を示す。PARM 属性の指定値を標準出力に出力するプログラムを指定する。
7. STEP 要素の終了を示す。
8. PROC 要素の終了を示す。
9. STEP 要素の開始を示す。ジョブステップ STEP1 を開始する。
10. EXEC 要素を示す。PARM 属性の指定値を標準出力に出力するプログラムを指定する。
11. STEP 要素の終了を示す。
12. STEP 要素の開始を示す。ジョブステップ STEP2 を開始する。
13. カタログドプロシジャ SAMPLE3_PROC 呼び出しのための CALL 要素を示す。
14. STEP 要素の終了を示す。
15. STEP 要素の開始を示す。ジョブステップ STEP3 を開始する。
16. プロシジャ SAMPLE3_PROC 呼び出しのための CALL 要素の開始を示す。
17. 呼び出されるプロシジャの EXEC 要素の属性を変更する REPLACE 要素の開始を示す。
18. 変更対象のジョブステップを示すための REPLACESTEP 要素の開始を示す。
19. EXEC 要素の PARM 属性値の変更を示す。
20. REPLACESTEP 要素の終了を示す。
21. REPLACE 要素の終了を示す。
22. CALL 要素の終了を示す。
23. STEP 要素の終了を示す。
24. JOB 要素の終了を示す。
25. ルート要素の終了を示す。

入力ストリームからのプロシジャ呼び出しのためのジョブ定義 XML ファイルの実行例を次に示す。

```
%%%%%%%%SAMPLE3_JCL_STEP1%%%%%%%%  
%%%%%%%%SAMPLE3_PROC_STEP1%%%%%%%%  
%%%%%%%%replace SAMPLE3_PROC_STEP1%%%%%%%%
```

prolog (XML の宣言)

形式

```
[version="XMLの仕様の版番号"]  
[encoding="文字符号化方式名"]  
[standalone="{yes|no}"]
```

機能

ジョブ定義 XML ファイルでの XML 宣言を文書の先頭に指定する。

属性

version="XML の仕様の版番号" ~<名前トークン>((1.0)) 《1.0》

XML の仕様の版番号"1.0"を指定する。

encoding="文字符号化方式名" ~<名前トークン>((Shift_JIS | ASCII | UTF-8 | EUC_JP))

XML 文書内での文字の符号化方式の名前が EncName であることを示す。

EncName には、次に示す内容が指定できる。

- Shift_JIS
シフト JIS コード (ただし、Linux の場合、日本語を使用できるのはコメントだけである)
- ASCII **【AIX 限定】**
ASCII コード
- UTF-8
UTF-8 コード (ただし、AIX の場合、ASCII の文字範囲だけ使用できる)
- EUC_JP **【AIX 限定】**
EUC コード

文字符号化方式名は上記を標準名とする。ただし、次に示す相違があっても同一と解釈する。

- 大文字と小文字
- アンダースコア(_)とハイフン(-)

standalone="{yes | no}"

スタンドアロン文書宣言であることを示す。

- yes
文書実体の外部に、XML プロセサからアプリケーションへと渡される情報に影響するマーク付け宣言が存在しないことを示す。
- no
外部マーク付け宣言が存在する、または存在する可能性があることを示す。外部にマーク付け宣言は存在しなければ、スタンドアロン文書宣言は意味を持たない。

ジョブ定義 XML ファイルでは、外部にマーク付け宣言は存在しないので、スタンドアロン文書宣言は意味をもたない。このため、この属性は指定してもよいが、無視される。

注意事項

- prolog は文書の先頭に記述すること。
- prolog は省略できる。
- encoding を省略した場合、ジョブ定義 XML ファイルには、ASCII 文字コードの範囲中の制御文字を除いた範囲 (0x20~0x7e) だけ使用できる。
- encoding を省略した場合、ジョブログを出力する際、文字コード変換を実施しない。
- BOM による符号化方式の指定は無視する。

使用例

```
<?xml version="1.0" encoding="Shift_JIS" ?>
```

XML ジョブ制御要素一覧

バッチジョブ実行システムの XML ジョブ制御要素一覧を次の表に示す。

表 9-2 XML ジョブ制御要素一覧

XML ジョブ制御要素	定義内容
CALL	呼び出すプロシジャを定義する。
DD	ジョブまたはジョブステップで使用するファイルを定義する。
ELSE	IF 要素と組み合わせて ELSE 節の開始を定義する。
ENDIF	IF 要素と組み合わせて THEN 節および ELSE 節の終わりを定義する。
EXEC	実行するプログラムを定義する。
HitachiBatchJobExec	ジョブ定義 XML ファイルのルート要素であり、このファイルがジョブ定義 XML ファイルであることを示す。また、ジョブ定義 XML ファイルのバージョンや動作環境を定義する。
IF	条件式の判定結果でジョブステップの実行を制御する。
JOB	ジョブの属性を定義する。
JOBPARM	ジョブ単位の制御情報を定義する。
PERMISSION	新規作成するファイルまたはディレクトリのパーミッションを定義する。
PROC	プロシジャを定義する。
REPLACE	カタログドプロシジャまたは入出力ストリームプロシジャ内の先頭ジョブステップの DD 要素、EXEC 要素内の属性の追加、変更、または削除をする場合、変更内容を定義する。
REPLACESTEP	カタログドプロシジャまたは入出力ストリームプロシジャ内の DD 要素、EXEC 要素内の属性の追加、変更、または削除をする場合、その対象となるジョブステップを定義する。
SETENV	環境変数を定義する。
SETENVFILE	環境変数格納ファイルを定義する。
SETMAXRC	ジョブの途中で、先行ジョブステップの終了コードの最大値を任意の値で定義する。
SETSYMBOL	カタログドプロシジャまたは入力ストリームプロシジャ中で定義された記号パラメータに値を定義する。
STEP	実行するプログラムまたは呼び出すカタログドプロシジャを定義するとともに、それらを実行する場合の各要素の開始を定義する。
SYMBOL	PROC 要素のプロシジャ内の記号パラメータに省略時の値を定義する。

XML ジョブ制御要素を定義する場合の注意事項を次に示す。

- 要素名、属性名、属性値、および要素の内容の大文字小文字は区別する
- DD 要素と EXEC 要素以外に要素の内容が指定されている場合は無視する

- ドキュメントの最上位（HitachiBatchJobExec 要素の外部）に空白、タブ、および改行コード以外の文字を指定してはならない

CALL 要素 (プロシジャの呼び出し)

形式

```
NAME="プロシジャ名"
```

機能

呼び出すプロシジャを定義する。

CALL 要素内には、次に示す順番で要素を含めることができる。

- SETSYMBOL 要素
複数回記述，または省略できる。
- REPLACE 要素
1 回だけ記述，または省略できる。

属性

NAME="プロシジャ名" ～<記号名称>((1～31 文字))

呼び出すカタログドプロシジャまたは入力ストリームプロシジャ名を指定する。

呼び出すカタログドプロシジャまたは入力ストリームプロシジャ名に同一名がある場合，入力ストリームプロシジャを優先する。

注意事項

CALL 要素は，STEP 要素内に 1 つだけ指定できる。ただし，同一 STEP 要素内に EXEC 要素がある場合には指定できない。

使用例

カタログドプロシジャ ASMCG を実行する場合の使用例を次に示す。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="ASM">
    <STEP NAME="STEP">
      <CALL NAME="ASMCG" >
        <SETSYMBOL NAME="PARM1" VALUE="/home/user1/" />
      </CALL>
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

DD 要素 (ファイルの定義)

形式

```
[CHARS="文字セットイメージ名[, 文字セットイメージ名]…"]
[CMT="注釈"]
[COPIES="複写回数"]
[COPIYG="グループ複写回数"]
[DEST="SYSOUT出力先識別子"]
[DISP="[ {NEW|RNW|OLD|SHR|MOD} ][, {DELETE|KEEP|PASS} ][, {DELETE|KEEP} ]"]
[DSN=
  "{一時ファイル識別名
  |PREST識別名
  |一時ISAMファイル識別名
  |世代データ群パス名 (相対世代番号 | 日付世代番号)
  |ファイルパス名
  |ディレクトリパス名
  |* [. ジョブステップ名 [. プロシジャステップ名] ]. DD名
  }"]
[ENCODE="{SJIS|UTF8}"]
[EURENCODE="EURエンコード種別"]
[EURFORM="EUR定義ファイル名"]
[FCB="FCB名"]
[FCDFLT="デフォルト機能キャラクタイメージ名"]
[FORMS="書式番号"]
[FTYPE="{1|2}"]
[FPERM="パーミッション"]
[GAIJI1="レベル1外字名称"]
[HOLD="{YES|NO}"]
[ISKEY="キー長, キー位置[, {C|I|L|F|D}"]
[ISRECFM="{F|V}"]
[ISRECL="レコード長[, 最小レコード長"]
[KEEP="{YES|NO}"]
[MODE="{AN|K}"]
NAME="DD名"
[OPTCDJ="{YES|NO}"]
[OVERC="適用枚数[, 適用枚数]…"]
[OVERLAY="書式オーバーレイ情報のイメージ名[, 書式オーバーレイ情報のイメージ名]…"]
[RECFM="{A|M|N|X}"]
[RENAME="環境変数名"]
[SAMF="{YES|NO}"]
[SYSOUT="{クラス名|*}"]
[TYPE="{DATA|DUMMY|FILE|TEMP|TEMPISAM|PREST|TEMPPREST|GDG|LIB|SYSOUT|DIR}"]
```

機能

ジョブまたはジョブステップで使用するファイルを定義する。

属性

CHARS="文字セットイメージ名[, 文字セットイメージ名]…" ～<記号名称>((1～4文字))

PDE で使用する文字セットイメージ名を指定する。

コンマで区切って 4 個まで指定できる。

この属性を省略した場合、SysoutClass パラメータの指定値が有効になる。

CMT="注釈" ~<任意文字列>((1~255))

任意の文字列を指定する。Prinfina MANAGER SP でセパレータなどに利用されるが、この場合は前後の空白は削除される。

COPIES="複写回数" ~<符号なし整数>((1~255))

PDE で使用する SYSOUT データファイルの複写回数を指定する。

この属性を省略した場合、SysoutClass パラメータの指定値が有効になる。

COPYG="グループ複写回数" ~<符号なし整数>((1~255))

PDE で使用する SYSOUT データファイルをプリンタに出力する場合のグループ複写回数を指定する。

この属性を省略した場合、SysoutClass パラメータの指定値が有効になる。

DEST="SYSOUT 出力先識別子" ~<記号名称>((1~31 文字))

PDE で使用する SYSOUT データファイルの出力先を識別するための出力先識別子を指定する。

この属性を省略した場合、SysoutClass パラメータの指定値が有効になる。

DISP="[{NEW | RNW | OLD | SHR | MOD}][, {DELETE | KEEP | PASS}][, {DELETE | KEEP}]"

ファイルの状態および後処理に関する情報を指定する。DISP 属性の省略値については、「[3.4.2 ファイルの状態と後処理](#)」を参照のこと。なお、排他/共用制御の詳細については、「[3.13 ジョブ単位のファイルおよびディレクトリに対する排他/共用制御機能](#)」を参照のこと。

第 1 パラメータ

ジョブステップ実行直前のファイルの状態を表す。

- **NEW**

ファイルが該当するジョブステップ中で利用する。ジョブステップ開始前に 0 バイトのファイルを作成する。この作成時、DSN 属性値に指定したファイルが存在する場合は、エラーとなりジョブは終了する。

- **RNW**

ジョブステップ実行前に DSN 属性値に指定したファイルがすでに存在する場合、該当するジョブを異常終了としないでそのファイルを新しく作成し直す。DSN 属性値に指定されたファイルが存在する場合は、ファイルをいったん削除し、再度ファイルを生成する。DSN 属性値に指定したファイルがない場合は、ファイルを生成する。

ディレクトリに対して指定した場合、ジョブ実行時にエラーとなって、ジョブが終了する。

なお、DSN 属性値に指定されたファイルがシンボリックリンクのときは、リンク先のファイル実体の有無に関係なくシンボリックリンクをいったん削除し、新たに同じ名称のファイルを生成する。新たなシンボリックリンクが生成されるのではない点に注意する必要がある。

- **OLD**

ファイルが該当するジョブステップ以前にすでに存在する場合、ほかのジョブとファイルを排他するときに指定する。ファイル排他が適用されていない場合や適用できない TYPE 属性値の場合は、SHR パラメータ指定と同じになる。ジョブステップ開始前にファイルパスの存在の有無を確認するかどうかは、DD_EXISTCHECK_COBOL パラメータ、または DD_EXISTCHECK

パラメータの指定に従う。これらのパラメータの詳細については、「8. 設定ファイル」の「DD_EXISTCHECK_COBOL (DD 要素に指定したファイルパスの存在確認要否を実施するための定義)」, または「DD_EXISTCHECK (DD 要素に指定したファイルパスの存在確認要否を実施するための定義)」を参照のこと。

- SHR

ファイルが該当するジョブステップ以前にすでに存在する場合、ほかのジョブとファイルを共有するときに指定する。ジョブステップ開始前にファイルパスの存在の有無を確認するかどうかは、DD_EXISTCHECK_COBOL パラメータ、または DD_EXISTCHECK パラメータの指定に従う。これらのパラメータの詳細については、「8. 設定ファイル」の「DD_EXISTCHECK_COBOL (DD 要素に指定したファイルパスの存在確認要否を実施するための定義)」, または「DD_EXISTCHECK (DD 要素に指定したファイルパスの存在確認要否を実施するための定義)」を参照のこと。

- MOD

ファイルが該当するジョブステップ以前にすでに存在する場合、ファイル操作ユーティリティまたは COBOL プログラムがファイルに対して追加書きを行うときに指定する。

ファイルが該当するジョブステップ以前に存在しない場合は、DISPMOD_NOFILE パラメータの指定に従う。詳細については、「8. 設定ファイル」の「DISPMOD_NOFILE (DD 要素の DISP 属性に MOD を指定したファイルが存在しない場合の処理の定義)」を参照のこと。

EXEC 要素の LANG 属性が COBOL であり、かつ DD 要素の TYPE 属性の値が FILE, TEMP, または GDG (既存世代指定) である場合だけ使用できる。その他の値の場合は、次のとおり動作する。

- DD 要素の TYPE 属性の値が LIB, PREST, または TEMPPREST の場合は、SHR パラメータが指定されたものとして動作する。
- DD 要素の TYPE 属性の値が TEMPISAM の場合は、MOD として動作するが、書き込み動作はユーザプログラムに依存する。
- DD 要素の TYPE 属性の値が DIR の場合は、ジョブ定義 XML ファイル解析エラーとなる。

第 2 パラメータ

ジョブステップが正常終了した場合のファイルの後処理を指定する。

- DELETE

該当するジョブステップ終了後にファイルを削除する。

DSN 属性値に指定されたファイルがシンボリックリンクのときは、リンク先の実体の有無に関係なくシンボリックリンクを削除する。シンボリックリンクがリンクしているファイルが削除されるのではない点に注意する必要がある。

- KEEP

該当するジョブステップ終了後にファイルを保存する。ただし、ファイルを作成したジョブステップが異常終了した場合、第 3 パラメータに DELETE を指定していれば削除する。

- PASS

一時ファイルおよび一時 ISAM ファイルでだけ使用できる。

ジョブステップでファイルを使用したあと、該当するファイルを後続ジョブステップで利用できるようにすること（パスすること）を指示する。

このパラメータで指定したファイルを受け取れるのは、後続ジョブステップ中のどれか 1 回だけである。したがって、受け取ったあと、さらにこれを後続ジョブステップにパスする場合は、そのつど DISP 属性値で PASS パラメータを指定する必要がある。

このパラメータを指定した場合、第 3 パラメータに指定したパラメータに関係なくジョブステップ終了時（異常終了時も含む）、該当するファイルをパスする。

第 3 パラメータ

ジョブステップが異常終了した場合のファイルの後処理を指定する。

第 3 パラメータは、第 2 パラメータ中の PASS パラメータを除いたものである。各パラメータの意味については、「第 2 パラメータ」を参照のこと。

DSN=

```
"{一時ファイル識別名  
| PREST 識別名  
| 一時 ISAM ファイル識別名  
| 世代データ群パス名 (相対世代番号 | 日付世代番号)  
| ファイルパス名  
| ディレクトリパス名  
| *.[ジョブステップ名].[プロシジャステップ名].DD 名  
}"
```

• 一時ファイル識別名 ～<パス名>

TYPE 属性値に TEMP または TEMPPREST を指定した場合、一時ファイルの識別名を指定する。実際のファイル名は、システムでユニークな名称を与える。ディレクトリ区切り文字の"/"（スラント）を含めた文字列を指定した場合、最後のディレクトリ区切り文字"/"（スラント）より後ろの文字列を一時ファイル識別名と判断する。

TYPE=TEMP の場合

[3.4.4(3) 一時ファイルを利用する場合の注意事項] で示した一時ファイルの名称が、プラットフォームで決められた最大長を超えるような一時ファイル識別名を指定してはならない。超えた場合は、ジョブステップ実行開始時のファイル割り当て処理でエラーとなる。生成されるファイル名の長さは、一時ファイル識別名の長さ、JOB 要素の NAME 属性値の長さによって変わる。

TYPE=TEMPPREST の場合

ジョブステップ間 PREST ファイルの名称が、プラットフォームで決められた最大長を超えるような一時ファイル識別名を指定しないこと。超えた場合は、ジョブステップ実行開始時のファイル割り当て処理でエラーとなる。生成されるファイル名の長さは、一時ファイル識別名、ジョブ識別子、および JOB 要素の NAME 属性値の長さによって変わる。

ジョブステップ間 PREST ファイルの名称の詳細については、「3.12.2(2) ジョブステップ間 PREST ファイル」を参照のこと。

• PREST 識別名 ～<パス名>

TYPE 属性値に PREST を指定した場合、ジョブ間 PREST の識別名を指定する。この識別名とバッチジョブ実行システムの一部ファイルディレクトリ内でユニークな名称を組み合わせ、ジョブ間 PREST ファイル名とする。実際のファイル名はシステムでユニークな名称を与える。ディレクトリ区切り文字の"/" (スラント) を含めた文字列を指定した場合、最後のディレクトリ区切り文字"/" (スラント) より後ろの文字列を PREST 識別名と判断する。

ジョブ間 PREST ファイル名が、プラットフォームで決められた最大長を超えるような PREST 識別名は指定しないこと。超えた場合は、ジョブステップ実行開始時のファイル割り当て処理でエラーとなる。生成されるファイル名の長さは、PREST 識別名の長さによって変わる。

ジョブステップ間 PREST ファイルの名称の詳細については、「[3.12.2\(2\) ジョブステップ間 PREST ファイル](#)」を参照のこと。

- **一時 ISAM ファイル識別名 ～<パス名>**

TYPE 属性値に TEMPISAM を指定した場合、一時 ISAM ファイルの識別名を指定する。実際のファイル名は、システムでユニークな名称を与える。ディレクトリ区切り文字の"/" (スラント) を含めた文字列を指定した場合、最後のディレクトリ区切り文字"/" (スラント) より後ろの文字列を一時 ISAM ファイル識別名と判断する。

「[3.4.5 一時 ISAM ファイル](#)」で示した一時 ISAM ファイルの名称が、プラットフォームで決められた最大長を超えるような一時 ISAM ファイル識別名は指定しないこと。超えた場合は、ジョブステップ実行開始時のファイル割り当て処理でエラーとなる。生成されるファイル名の長さは、一時 ISAM ファイル識別名の長さとして JOB 要素の NAME 属性値の長さによって変わる。

- **世代データ群パス名 (相対世代番号 | 日付世代番号) ～<パス名>((1~1023 文字))**

TYPE 属性値に GDG を指定した場合、世代データ群のパス名と世代ファイルの相対世代番号または日付世代番号を指定する。各世代ファイルに対して、次に示すユニークなファイルパス名を与える。

- **相対世代番号**

世代データ群パス名.Gmmmm*

注※ mmmm は世代番号を表す 0001~9999 の整数である。

相対世代番号は、0 または符号を付けた 1~255 の整数で指定する。ただし、0 には符号を付けない。すでにある世代ファイルを参照する場合、最近作った世代ファイルに対して相対世代番号 0 を指定し、世代が古くなるに従い、相対世代番号を「-1, -2, -3, …」と指定する。

また、新しく世代ファイルを作成する場合には、相対世代番号を作成順に「+1, +2, +3, …」と指定する。同一世代ファイルに対しては、同一ジョブ中で同じ相対世代番号を指定する。

- **日付世代番号**

世代データ群パス名.Dyyyyymmdd*¹

注※¹ yyyy が西暦、mm は月、dd は日である。

日付世代番号は、次の 2 種類で指定する。

- EXD または EXD-n*²

- Dyyyyymmdd*³

注※² n は 1~511 までの整数で指定する。

注※3 yyyy はバッチジョブ実行日の西暦を指定する。mm は 01～12 までを指定する。dd は 01～を指定する。ただし、yyyy に指定できる西暦は、DISP 属性の第 1 パラメータが NEW の場合、バッチジョブ実行日の西暦の±2 年以内となる。例えば、バッチジョブ実行日が 2013 年 6 月 15 日の場合、2011 年 1 月 1 日～2015 年 12 月 31 日まで指定可能となる。また、存在しない日を指定することはできない。

すでに存在する世代ファイルを参照する場合、EXD、EXD-n、または Dyyyyymmdd を指定する。EXD と指定すると、JP1/AJS のルートジョブネットの実行開始予定日に作成された世代ファイルを参照し、世代が古くなるにしたがって、「EXD-1、EXD-2、EXD-3、…」と指定する。Dyyyyymmdd と指定すると、世代データ群名.yyyyyymmdd を参照する。また、新しく世代ファイルを作成する場合には、「EXD」または「Dyyyyymmdd」を指定する。同一世代ファイルを EXD-n で指定する場合、同一ジョブ中で n に同一の数値を指定する。

- **ファイルパス名 ～<パス名>((1～1023 文字))**

TYPE 属性値に FILE を指定した場合、ファイルのパス名を指定する。

- **ディレクトリパス名 ～<パス名>((1～1023 文字))**

TYPE 属性値に LIB を指定した場合、ジョブから実行するコマンドとプログラムを格納したディレクトリのパス名を指定する。

TYPE 属性値に DIR を指定した場合、ディレクトリのパス名を指定する。

- ***[.ジョブステップ名[.プロシジャステップ名]].DD 名**

ジョブステップ名 ～<記号名称>((1～31 文字))

プロシジャステップ名 ～<記号名称>((1～31 文字))

*[.ジョブステップ名[.プロシジャステップ名]].DD 名で示す DD 要素の DSN 属性値を複写するように指定する。DD 要素の TYPE 属性値と、複写元 DD 要素の TYPE 属性値は FILE にする必要がある。どちらか一方、または両方が FILE でない場合はエラーとなる。

ENCODE="{SJIS | UTF8}"

SYSOUT データファイルのエンコード種別を指定する。PDE と連携する場合に意味がある。

- **SJIS**

ファイルをシフト JIS のコードで作成する場合に指定する。

- **UTF8**

ファイルを UTF-8 のコードで作成する場合に指定する。

EURENCODE="EUR エンコード種別" ～<任意文字列>((1～31 文字))

作成する EUR のマッピングデータのエンコード種別を指定する。

指定できるエンコード種別は設定ファイルの EUR_LANG パラメータで指定したエンコード種別だけである。

通常は指定する必要はなく、バッチジョブ実行システムの LANG 環境変数と UAP の LANG 環境変数が異なるときに指定する。指定を省略した場合、バッチジョブ実行システムの LANG 環境変数を基に EUR がデータファイルのエンコード種別を判断する。

EURENCODE 属性は、EURFORM 属性の指定がない場合は無効となる。

EURFORM="EUR 定義ファイル名" ～<記号名称>((1～31 文字))

EUR で印刷するとき使用する EUR 定義ファイル名を指定する。

バッチジョブ実行システムの設定ファイルで EUR_DEFINITION_FILE_DIR パラメータの指定をしない場合は無効となる。

FCB="FCB 名" ～<記号名称>((1～4 文字))

PDE で使用する FCB 名を指定する。

この属性を省略した場合、JOBPARM 要素の FCB 属性が有効となる。JOBPARM 要素の FCB 属性を省略した場合は、SysoutClass パラメータの指定値が有効になる。

FCDFLT="デフォルト機能キャラクタイメージ名" ～<記号名称>((1～4 文字))

PDE で使用するデフォルト機能キャラクタイメージ名を指定する。

この属性を省略した場合、SysoutClass パラメータの指定値が有効になる。

FORMS="書式番号" ～<記号名称>((1～4 文字))

PDE で使用する書式番号を指定する。

この属性を省略した場合、JOBPARM 要素の FORMS 属性が有効となり、JOBPARM 要素の FORMS 属性を省略した場合は、SysoutClass パラメータの指定値が有効となる。

FTYPE="{1 | 2}"

SYSOUT データファイルのファイル形式を指定する。

この指定を省略した場合、SysoutClass パラメータの指定値が有効になる。

この指定は JOBLOG, SYSMSG, XML, STDOUT, および STDERR には無効であり、これらのファイルはテキストファイルとして処理される。

- 1 : PDE ファイル形式 1
テキストファイルなどの任意のファイル形式
- 2 : PDE ファイル形式 2
COBOL2002 で PDE による書式、行制御印刷を行うファイル形式

FPERM="パーミッション" ～< 8 進数>((0～1777))

ファイルのパーミッションを指定する。指定方法は、chmod コマンドに従う。この指定をしない場合は、PERMISSION 要素の属性に従う。

1000 : ディレクトリにリンク許可を属性、またはファイルに save-text 属性を設定する。

0400 : バッチジョブ実行システムを起動したユーザへの読み取り属性を設定する。

0200 : バッチジョブ実行システムを起動したユーザへの書き込み属性を設定する。

0100 : バッチジョブ実行システムを起動したユーザへの実行または検索属性を設定する。

0040 : バッチジョブ実行システムを起動したグループへの読み取り属性を設定する。

0020 : バッチジョブ実行システムを起動したグループへの書き込み属性を設定する。

0010 : バッチジョブ実行システムを起動したグループへの実行または検索属性を設定する。

0004 : ほかのユーザへの読み取り属性を設定する。

0002 : ほかのユーザへの書き込み属性を設定する。

0001：ほかのユーザへの実行または検索属性を設定する。

GAIJI1="レベル 1 外字名称" ~<記号名称>((1~4 文字))

PDE で使用する外字名称を指定する。

この属性を省略した場合、SysoutClass パラメータの指定値が有効になる。

HOLD="{YES | NO}"

ジョブ実行後、SYSOUT データファイルを即時印刷するか、または出力保留をし、あとからコマンドの指示によって印刷するかどうかを指定する。なお、EUR 帳票印刷はこの指定に関係なく、ジョブ終了時に印刷する。

この属性を省略した場合、SysoutClass パラメータの指定値が有効になる。

- YES

出力保留にして、即時印刷しない。

- NO

出力保留にしないで、即時印刷する。

ISKEY="キー長,キー位置[,IC | I | L | F | D]"

TYPE 属性に TEMPISAM を指定して一時 ISAM ファイルを作成する場合に、一時 ISAM ファイルのキー長、相対キー位置、およびキー属性を指定する。

なお、キー位置とキー長の合計は ISRECL 属性で指定するレコード長または最小レコード長以下にする必要がある。

- キー長

キー属性に応じて、次に示すように指定する。

C (文字型) の場合：1~255 の範囲

I (2 バイト整数型) の場合：2

L (4 バイト整数型) の場合：4

F (計算機浮動型) の場合：4

D (計算機倍長型) の場合：8

- キー位置

ISRECFM 属性で指定するレコード種別に応じて、次に示すように指定する。

ISRECFM="F"の場合：0~ (ISRECL 属性で指定するレコード長-1) の範囲

ISRECFM="V"の場合：0~ (ISRECL 属性で指定する最小レコード長-1) の範囲

- {C | I | L | F | D}

キー属性を次に示すどれかのパラメータで指定する。

C：文字型

I：2 バイト整数型

L：4 バイト整数型

F：計算機浮動型

D：計算機倍長型

ISRECFM="{F | V}"

TYPE 属性に TEMPISAM を指定して一時 ISAM ファイルを作成する場合に、一時 ISAM ファイルのレコード種別を指定する。

- F
レコード種別が順編成固定長であることを示す。
- V
レコード種別が可変長であることを示す。

ISRECL="レコード長[,最小レコード長]"

TYPE 属性に TEMPISAM を指定して一時 ISAM ファイルを作成する場合に、一時 ISAM ファイルのレコード長を指定する。

また、ISRECFM 属性に V を指定した場合は、最小レコード長も指定する必要がある。

なお、ISRECFM 属性に F を指定した場合は、最小レコード長を指定しないこと。指定した場合は、ジョブ定義 XML ファイルの解析時にエラーとなる。レコード長または最小レコード長は次に示す範囲で指定する。

- レコード長
1~65,503 の範囲で指定する。
- 最小レコード長
1~65,503 の範囲で指定する。
なお、最小レコード長 ≤ レコード長にする必要がある。

KEEP="{YES | NO}"

SYSOUT データファイルを出力後保存するかどうかを指定する。この属性を省略した場合、SysoutClass パラメータの指定値が有効になる。

- YES
SYSOUT データファイルをスプールジョブ転送サービスに転送指示後、スプールから削除しない (SYSOUT データファイルを出力後保存属性にする)。
- NO
SYSOUT データファイルをスプールジョブ転送サービスに転送指示後、スプールから削除する (SYSOUT データファイルを出力後保存属性にしない)。

MODE="{AN | K}"

PDE で使用するプリンタモードを指定する。

この属性を省略した場合、SysoutClass パラメータの指定値が有効になる。

- AN
ノンインパクトモード
- K
漢字プリンタモード

NAME="DD 名" ~<環境変数名>((1~31 文字))

ジョブステップ内で、DD 要素や DD 要素で割り当てたファイルを識別する識別名を指定する。この DD 名の先頭に文字列を付加した名称の環境変数に、割り当てたファイルのパス名を格納する。

付加する文字列は EXEC 要素のプログラムが COBOL の場合は "CBL_" となり、COBOL 以外の場合は "DDN_" となる。

DD 名に JOBLIB を指定した場合は、DSN 属性値に指定した文字列を、PATH 環境変数に追加する。追加した文字列は、ジョブ全体で有効となる。この場合、DD 要素は、JOB 要素内の STEP 要素外に指定する必要がある。また、TYPE 属性値は LIB にする必要がある。

DD 名に STEPLIB を指定した場合は、DSN 属性値に指定した文字列を、PATH 環境変数に追加する。追加した文字列は、ジョブステップ内でだけ有効となる（次のジョブステップでは、PATH 環境変数から除外する）。この場合、DD 要素は STEP 要素内に指定する必要がある。また、TYPE 属性値は LIB にする必要がある。

OPTCDJ="{YES | NO}"

PDE で、行単位にフォントと文字ピッチを切り替える文字配列テーブルを変更するかどうかを指定する。

- YES
行単位に使用する文字配列テーブルを変更する。
- NO
行単位に使用する文字配列テーブルを変更しない。

OVERC="適用枚数[,適用枚数]..." ~<符号なし整数>((1~255))

PDE で使用する書式オーバーレイの適用枚数を指定する。

コンマで区切って 8 個まで指定できる。

この属性を省略した場合、SysoutClass パラメータの指定値が有効になる。

OVERLAY="書式オーバーレイ情報のイメージ名[,書式オーバーレイ情報のイメージ名]..." ~<記号名称>((1~4 文字))

PDE で使用する書式オーバーレイ情報のイメージ名を指定する。

コンマで区切って 8 個まで指定できる。

VOS3 から移行した COBOL プログラムに APPLY FORMS-OVERLAY 指定がある場合、書式オーバーレイ名をここで指定することで書式オーバーレイ印刷ができる。ただし、ページごとの切り替えはできない。

この属性を省略した場合、SysoutClass パラメータの指定値が有効になる。

RECFM="{A | M | N | X}"

PDE で使用する出力データに指定しているプリンタ制御文字の種別を指定する。

RECFM 属性は JOBLG, SYMSG, XML, STDOUT, および STDERR の設定には無効であり、これらのファイルは制御文字なしとして処理される (RECFM=X と同等)。

- A
ANSI 制御文字
- M

機械制御文字

- N

制御文字なし (ASCII 制御文字)

- X

RECFM 属性を設定しない。この場合、PDE や Prinfinamanager の設定に従う。

RENAME="環境変数名" ~<環境変数名>((1~31 文字))

割り当てられたファイルのパス名を格納する環境変数名を変更する。この属性を省略した場合、環境変数名は、NAME 属性で指定した DD 名の先頭に "DDN_" または "CBL_" を付加した名称になる。

この属性は、COBOL2002 の特殊な DD 名称など、環境変数を "DDN_" または "CBL_" で開始しない名称にする必要がある場合に使用する。

RENAME 属性に関する注意事項については、「[3.4.7\(4\) 環境変数名を任意の名称に設定する方法](#)」を参照のこと。

SAMF="{YES | NO}"

ジョブ定義 XML ファイルに記述した SYSIN データを、ジョブコントローラが COBOL プログラムに渡す場合のデータの形式を指定する。

この属性を省略した場合、設定ファイルの DD_SAMF パラメータの指定値が有効になる。

この属性は、TYPE 属性に "DATA" が指定されていて、かつ同一ジョブステップの EXEC 要素の LANG 属性に "COBOL" が指定されている場合にだけ指定できる。それ以外の場合は、ジョブ定義 XML ファイルの解析エラーとなり、ジョブを実行しないで終了する。

- YES

COBOL が順編成固定長ファイルとして SYSIN データを入力する場合に、"YES" を指定する。ジョブコントローラは、SYSIN データをレコード長 80 バイトの順編成固定長ファイルに変換し、COBOL プログラムに渡す。1 行が改行を除いて 81 バイト以上であり、81 バイト目以降に空白以外の文字がある場合、ジョブ定義 XML ファイルの解析エラーとなり、ジョブを実行しないで終了する。

- NO

COBOL がテキスト編成ファイルとして SYSIN データを入力する場合に、"NO" を指定する。ジョブコントローラは、ジョブ定義 XML ファイルに記述した SYSIN データを変更しないで、COBOL プログラムに渡す。

SYSOUT="{クラス名 | *}"

SYSOUT データファイルの出力に関する情報を指定する。

- クラス名 ~<記号名称>((1~31 文字))

出力クラス名を指定する。このクラス名は、あらかじめ設定ファイルに定義しておく必要がある。

- *

ジョブ要素で指定した MSGCLASS 属性の出力クラス名と同じクラスを割り当てる。ジョブ要素で MSGCLASS 属性を指定していない場合は、設定ファイルの MSGCLASS_DEFAULT パラメータで指定したクラスを割り当てる。

TYPE="{DATA | DUMMY | FILE | TEMP | TEMPISAM | PREST | TEMPPREST | GDG | LIB | SYSOUT | DIR}"

割り当てるファイル種別を指定する。

この属性を省略した場合、SYSOUT 属性を指定していれば TYPE="SYSOUT"が仮定され、SYSOUT 属性を指定していなければ TYPE="FILE"が仮定される。

- **DATA**

DD 要素の内容に、ジョブステッププログラムへの入力データがある。

- **DUMMY**

ダミーファイルを割り当てる。

DUMMY_FILE_TYPE パラメータに DEVNULL を指定した場合、/dev/null を割り当てる。

DUMMY_FILE_TYPE パラメータに TEMP を指定した場合、DD 要素で TYPE="DUMMY"の代わりに TYPE="TEMP"が指定されたものとして一時ファイルを割り当てる。

詳細については、「[3.4.6 ダミーファイル](#)」を参照のこと。

- **FILE**

恒久ファイルを割り当てる。

- **TEMP**

一時ファイルを割り当てる。

- **TEMPISAM**

一時 ISAM ファイルを割り当てる。

- **PREST**

ジョブ間にわたって有効なジョブ間 PREST ファイルを割り当てる。ただし、LOCK_OPTION パラメータに ENABLE を指定していない場合、この属性に PREST を指定するとジョブ定義 XML ファイルの解析エラーとなる。

- **TEMPPREST**

ジョブ内でだけ有効なジョブステップ間 PREST ファイルを割り当てる。

- **GDG**

世代ファイルを割り当てる。

- **LIB**

JOBLIB/STEPLIB を割り当てる。

- **SYSOUT**

SYSOUT データファイルを割り当てる。

- **DIR**

ディレクトリを割り当てる。

内容

TYPE 属性値に"DATA"を指定した場合、SYSIN データ（シェルスクリプトのヒアドキュメント相当、ジョブステッププログラムへの入力データをインライン記述したもの）をこの要素の内容に指定する。複数行からの構成または<や"&の文字が含まれている場合は、記述するデータは<![CDATA[!と!]]>で囲んで指定する。記述するデータの詳細については、「3.10 SYSIN データの読み込み」を参照のこと。

注意事項

- ファイルを連結する場合を除いて、同一の JOB または STEP 要素内には、同じ NAME 属性値の DD 要素を複数指定しないこと。
- 直前の DD 要素の NAME 属性値と同じ NAME 属性値の DD 要素を指定した場合、ファイルの連結と判断する。
- NAME 属性値が同一の DD 要素が、JOB 要素または STEP 要素内で連続していない場合、ジョブはエラーとなる。
- JOB 要素内の STEP 要素外に指定できる DD 要素は、NAME 属性値に JOBLIB を指定した場合だけである。
- NAME 属性値に JOBLIB または STEPLIB を指定した場合、TYPE 属性値に LIB を指定しなければならない。
- DD 要素内の属性の追加、変更、または削除のために、DD 要素を REPLACE 要素内および REPLACESTEP 要素内に指定した場合は、NAME 属性と追加、変更、または削除する属性の属性名と属性値を指定する。削除の場合は属性値に""を指定する。
- DD 要素を追加するために、DD 要素を REPLACE 要素内および REPLACESTEP 要素内に指定する場合は、追加する DD 要素全体を指定する。
- DD 要素を削除するために、DD 要素を REPLACE 要素内または REPLACESTEP 要素内に指定する場合は、NAME 属性値と TYPE 属性値で DUMMY を指定する。TYPE 属性値で DUMMY を指定した場合、NAME 以外の属性値は無効となる。
- TYPE 属性値が"LIB"の場合に DISP 属性を指定した場合でも、指定値に関係なく DISP="SHR"と見なして処理する。
- TYPE 属性値が"TEMP"および"TEMPISAM"以外の場合、DISP 属性の第 2 パラメータに PASS を指定することはできない。
- RECFM 属性は VOS3 の行制御の種別を指定する項目であり、印刷データの形式に合わせて指定する必要がある。VOS3 から移行した COBOL プログラムでは次に示す内容を考慮して指定をする必要がある。

PDE ファイル形式 1 を使用する場合

- WRITE 指定の場合

印刷データに付加している制御コードに合わせて指定する。通常は VOS3 の JCL (Job Control Language) で指定していた値と同じ値を指定する。また、COBOL プログラムの移行時に実行時

環境変数 CBLD_ファイル名 (SELECT 句に指定するファイル名) に SAMAADV を設定する必要がある。

RECFM="A"の場合、印刷制御文字を 16 進で指定しているときは、ASCII コードに変換する必要がある。

- WRITE ADVANCING 指定の場合
COBOL プログラムの移行時に印刷データの先頭に制御文字の領域を定義している場合は、コンパイラオプションで制御文字の領域を無効にする。また、RECFM="N"を指定する。
- WRITE POSITIONING 指定の場合
COBOL プログラム移行時に POSITIONING は ADVANCING 1 LINE と扱われるため、必要に応じて ADVANCING 指定に変更する必要がある。また、RECFM="N"を指定する。
- DISPLAY UPON SYSOUT, DISPLAY UPON SYSLST 指定の場合
UPON SYSLST は指定しないこと。また、COBOL プログラムに WITH NO ADVANCING の指定がある場合、削除する必要がある。
RECFM="N"を指定する。

PDE ファイル形式 2 を使用する場合

- WRITE 指定の場合
印刷データに付加している制御コードに合わせて指定する。通常は VOS3 の JCL で指定していた値と同じ値を指定する。
RECFM="A"の場合、印刷制御文字を 16 進で指定しているときは、ASCII コードに変換する必要がある。
- WRITE ADVANCING 指定の場合
COBOL プログラムの移行時に印刷データの先頭に制御文字の領域を定義している場合は、コンパイラオプションで制御文字の領域を無効にする。また、RECFM="N"を指定する。
- WRITE POSITIONING 指定の場合
コンパイラオプションで制御文字の領域を無効にする。また、RECFM="M"を指定する。
- DISPLAY UPON SYSOUT, DISPLAY UPON SYSLST 指定の場合
PDE ファイル形式 2 には対応していないので、PDE ファイル形式 1 の場合と同様の指定をする。
- DD 要素を定義するとき、TYPE 属性の値によって指定必須の属性、指定不可の属性、省略できる属性が、それぞれ排他的関係として決められている。指定必須の属性を指定しない、あるいは指定不可の属性を指定するとエラーとなる。DD 要素の属性の排他的関係一覧を次の表に示す。表の列は TYPE 属性の値を、行は各属性を表しており、TYPE 属性の値に対して各属性の指定の可否を表している。

表 9-3 DD 要素の属性の排他的関係一覧

DD 要素の属性	DD 要素の TYPE 属性										
	FILE	DIR	GDG	TEMP	DATA	DUM MY	SYSOU T	LIB	TEMPISA M	PRES T	TEMPPR EST
NAME	○	○	○	○	○	○	○	○	○	○	○

DD 要素の属性	DD 要素の TYPE 属性										
	FILE	DIR	GDG	TEMP	DATA	DUM MY	SYSOU T	LIB	TEMPISA M	PRES T	TEMPPR EST
DSN	○	○	○	△	×	×	×	○	△	○	○
DISP	△	△	△	△	×	×	×	△	△	△	△
FPERM	△	△	△	×	×	×	×	×	×	△	×
SYSOUT	×	×	×	×	×	×	○	×	×	×	×
CHARS	×	×	×	×	×	×	△	×	×	×	×
CMT	×	×	×	×	×	×	△	×	×	×	×
COPIES	×	×	×	×	×	×	△	×	×	×	×
COPYG	×	×	×	×	×	×	△	×	×	×	×
DEST	×	×	×	×	×	×	△	×	×	×	×
ENCODE	×	×	×	×	×	×	△	×	×	×	×
EURENC ODE	×	×	×	×	×	×	△	×	×	×	×
EURFOR M	×	×	×	×	×	×	△	×	×	×	×
FCB	×	×	×	×	×	×	△	×	×	×	×
FCDFLT	×	×	×	×	×	×	△	×	×	×	×
FORMS	×	×	×	×	×	×	△	×	×	×	×
FTYPE	×	×	×	×	×	×	△	×	×	×	×
GAII1	×	×	×	×	×	×	△※1	×	×	×	×
HOLD	×	×	×	×	×	×	△	×	×	×	×
KEEP	×	×	×	×	×	×	△	×	×	×	×
MODE	×	×	×	×	×	×	△	×	×	×	×
OPTCDJ	×	×	×	×	×	×	△	×	×	×	×
OVERC	×	×	×	×	×	×	△	×	×	×	×
OVERLA Y	×	×	×	×	×	×	△	×	×	×	×
RECFM	×	×	×	×	×	×	△	×	×	×	×
ISKEY	×	×	×	×	×	×	×	×	○※2	×	×
ISRECFM	×	×	×	×	×	×	×	×	○※2	×	×
ISRECL	×	×	×	×	×	×	×	×	○※2	×	×

DD 要素の属性	DD 要素の TYPE 属性										
	FILE	DIR	GDG	TEMP	DATA	DUM MY	SYSOU T	LIB	TEMPISA M	PRES T	TEMPPR EST
RENAME	△	△	△	△	△	△	△	×	△	△	△
SAMF	×	×	×	×	△	×	×	×	×	×	×

(凡例)

- ：必ず指定する。
- △：指定を省略できる。
- ×

注※1

省略した場合は SysoutClass パラメータの指定値が有効になる。

注※2

一時 ISAM ファイルを作成する場合は必ず指定する。DISP="SHR"または DISP="OLD"による一時 ISAM ファイルの割り当て時に指定した場合は、無効となる。ただし、属性値のチェックは行われる。DISP="MOD"の場合は、先行ジョブステップで PASS されていない一時 ISAM ファイルには必ず指定する。

- プロシジャ内の DD 要素は、属性の追加および削除をしたあとに、排他関係をチェックする。ただし、TYPE 属性を置き換えた場合は、すでに存在する指定不可の属性は、指定されていないものと判断する。そのため、排他関係にある属性を明示的に削除する必要はない。
- 次に示す文字列は、特別な意味を持つ DD 名として将来バッチジョブ実行システムが使用する可能性がある。そのため、これらの値を NAME 属性に指定してはならない。指定した場合はエラーとなる。AJOBLIB, ASTEPLIB, SYSUDUMP, SYSABEND, SYSNDUMP, SYSCHK, SYSCKEOV, JOBCAT, STEPCAT, KANJLIB, IMGLIB, SYSMAPLB
- CALL 要素を定義したジョブステップ内に DD 要素を定義しても、その DD 要素は無視する。
- ファイル連結できない TYPE 属性を指定した DD 要素を連結した場合は、警告メッセージを出力してジョブを続行する。詳細については、「3.4.3 ファイルの連結処理」を参照のこと。
- 同一の DSN 属性で複数の一時ファイルを作成し、PASS 指定することは、誤りの原因となるため、使用しないこと。
- DSN 属性にファイル名を指定する場合は、ファイル名の末尾にパス区切り文字 ('/') を指定しないこと。指定した場合はエラーとなる。
- DISP 属性の第 1 パラメータに MOD を指定し、COBOL プログラムが入力としてオープンした場合、オープンの指定どおり入力ファイルとして使用する。この場合、ファイル連結指定をすることもできる。

使用例

1 枚目にだけ書式オーバーレイモジュール「FOPA」を使用し出力する場合の使用例を次に示す。

```
<DD NAME="EXA2" TYPE="SYSOUT" SYSOUT="*" OVERLAY="FOPA" OVERC="1"/>
```

ELSE 要素 (IF 要素に対する分岐)

形式

指定できる属性はない。

機能

IF 要素と組み合わせて使用し、THEN 節の終わりと ELSE 節の開始を指定する。

ELSE 要素は省略できる。ELSE 要素を指定しない場合、ELSE 節はなく、IF 要素から対応する ENDIF 要素までが THEN 節となる。

属性

なし。

注意事項

- SETSYMBOL, REPLACE または REPLACESTEP の置き換えの対象外である。

使用例

ELSE 要素を使用した例については、「[3.7.2 ジョブステップの実行制御の使用例](#)」を参照のこと。

ENDIF 要素 (ジョブステップの実行制御の終了)

形式

指定できる属性はない。

機能

IF 要素と組み合わせて使用し、THEN 節または ELSE 節の終わりを示す。

IF 要素を指定した場合、ENDIF 要素を指定する必要がある。

属性

なし。

注意事項

- SETSYMBOL, REPLACE または REPLACESTEP の置き換えの対象外である。

使用例

ENDIF 要素を使用した例については、「[3.7.2 ジョブステップの実行制御の使用例](#)」を参照のこと。

EXEC 要素 (プログラムの実行)

形式

```
[COND="
  [(しきい値, 演算符号 [, ジョブステップ名 [, プロシジャステップ名]] [ ])]
  [, (しきい値, 演算符号 [, ジョブステップ名 [, プロシジャステップ名]])] +
  [, {EVEN|ONLY}] " ]
[Group="Javaバッチアプリケーション実行時のスケジュールグループ名"]
[JSPEC="{YES|NO}"]
[LANG="{COBOL|JP1Script|Java}"]
[Option="{Javaのクラスパス | Javaのシステムプロパティ}"]
[PGM="{*|コマンド名|スクリプトファイル名|
  Javaアプリケーションクラス名}"]
[PARM="{引数|mainメソッドに渡す引数}"]
[PARMSPLIT="{YES|NO}"]
[STDOUTCLASS="クラス名"]
[STDERRCLASS="クラス名"]
```

機能

実行するプログラムとプログラムを実行する条件を定義する。

属性

COND="

```
[(しきい値, 演算符号 [, ジョブステップ名 [, プロシジャステップ名]] [ ])]
[, (しきい値, 演算符号 [, ジョブステップ名 [, プロシジャステップ名]])] +
[, {EVEN | ONLY}] "
```

先行する全ジョブステップまたは特定のジョブステップの終了コードや、異常終了した先行ジョブステップの有無によって、このジョブステップを実行するかどうかを決定するための条件を指定する。

- **しきい値** ~<符号なし整数>((0~255))
先行する全ジョブステップまたは特定のジョブステップの終了コードと比較するしきい値を指定する。1つでも条件を満たした場合、該当するジョブステップをスキップする。
- **演算符号** ~((**EQ | NE | GT | GE | LT | LE**))
演算符号の意味を次に示す。
EQ: しきい値は終了コードと等しい。
NE: しきい値は終了コードと等しくない。
GT: しきい値は終了コードより大きい。
GE: しきい値は終了コードより大きいか等しい。
LT: しきい値は終了コードより小さい。
LE: しきい値は終了コードより小さいか等しい。
- **ジョブステップ名** ~<記号名称>((1~31文字))

判定対象となる終了コードを返すジョブステップ名を指定する。ただし、該当するジョブステップがプロシジャ中のジョブステップの場合には、ジョブステップ名としてプロシジャを呼び出した CALL 要素のジョブステップ名を指定したあと、続いてプロシジャステップ名を指定する。

プロシジャステップ名を指定しないでジョブステップ名だけ記述した場合、ジョブコントローラは次の順序でジョブステップ名を検索する。

AIX の場合

ジョブ定義 XML ファイルの先頭から、プロシジャ外のジョブステップ名を検索する。

Linux の場合

該当するプロシジャ内のジョブステップ名を検索し、見つからない場合はジョブ定義 XML ファイルの先頭から、プロシジャ外のジョブステップ名を検索する。

- **プロシジャステップ名** ~<記号名称>((1~31 文字))

該当するジョブステップがプロシジャ中のジョブステップの場合に、判定対象となる終了コードを返すプロシジャ中の STEP 要素に指定したジョブステップ名を指定する。

- **EVEN**

先行したジョブステップのどれかが異常終了したかどうかに関係なく、無条件にこのジョブステップを実行する。

- **ONLY**

先行したジョブステップのどれかが異常終了した場合に、このジョブステップを実行する。

Group="Java バッチアプリケーション実行時のスケジュールグループ名" ~<グループ名>((1~63))

Java バッチアプリケーション実行時、ジョブスケジューリング機能を使用する場合に、CTM がバッチアプリケーションの実行をスケジューリングするときに割り当てるバッチサーバのスケジュールグループを指定する。スケジュールグループ名は、usrconf.cfg (バッチアプリケーション用オプション定義ファイル) でも指定できる。

デフォルト値は JOBGROUP となる。スケジュールグループ名の指定の優先順位は次のとおりになる。

1. コマンドライン
2. usrconf.cfg (バッチアプリケーション用オプション定義ファイル)
3. デフォルト値

詳細については、マニュアル「Cosminexus アプリケーションサーバ V8 リファレンス コマンド編」を参照のこと。

JSPEC="{YES | NO}"

ジョブステップを並列実行する場合に、そのジョブステップの種類を指定する。

- **YES**

並列実行するサブステップに指定する。ジョブの最終ジョブステップにこの属性を指定しても無視される (ジョブの最終ジョブステップは必ずメインステップとなる)。

- **NO**

メインステップに指定する。先行するジョブステップに JSPEC = "YES" の指定がなければ、このジョブステップは並列実行しない。

LANG="{COBOL | JP1Script | Java}"

- COBOL

COBOL2002 でコンパイルしたプログラムを実行する場合、実行するプログラム言語"COBOL"を指定する。プログラム言語が COBOL 以外の場合は指定しないこと。

この属性の指定によって、ジョブステップ内の DD 要素で指定した DD 要素名から生成する環境変数に"CBL_"を付加し、"CBL_+"DD 要素名"という名称に変換する。

この属性を省略した場合、ジョブステップ内の DD 要素で指定した DD 要素名から生成する環境変数に"DDN_"を付ける。

- JP1Script

JP1/Script を実行する場合、実行するプログラム言語として"JP1Script"を指定する。JP1/Script 以外を実行する場合は指定しないこと。この属性の指定によって、PGM 属性にスクリプトファイル名を指定し、PARM 属性に位置変数とオプションを指定する。この指定によってスクリプトファイル名、位置変数、オプションを引数とする JP1/Script 実行コマンド sptxe を実行する。

- Java

Cosminexus 08-00 以降でバッチアプリケーションを実行する場合、実行するプログラム言語として"Java"を指定する。プログラム言語が Java 以外の場合は指定しないこと。この属性の指定によって、PGM 属性に Java アプリケーションクラス名、PARM 属性に main メソッドに渡す引数、Group 属性にスケジュールグループ名、Option 属性に Java オプションを引数とするバッチアプリケーション実行コマンド cjexecjob を実行する。

Option="{Java のクラスパス | Java のシステムプロパティ}" ~<任意文字列>((1~1022))

- Java のクラスパス

-cp または -classpath オプションを指定する。ジョブコントローラは、この属性に指定されたオプションを、そのまま Java オプションとして cjexecjob コマンドに渡す。

複数のクラスパスを異なる手段で指定した場合の検索順は次のとおりになる。

1. usrconf.cfg (バッチサーバ用オプション定義ファイル)
2. コマンドライン引数
3. usrconf.cfg (バッチアプリケーション用オプション定義ファイル)

- Java のシステムプロパティ

cjexecjob コマンドに渡す、Java 実行環境のシステムプロパティを指定する。同じキーのシステムプロパティを指定した場合、あとに指定したプロパティが優先される。同じキーのシステムプロパティを異なる手段で指定した場合の優先順位は次のとおりになる。

1. コマンドライン引数
2. usrconf.properties (バッチアプリケーション用ユーザプロパティファイル)
3. usrconf.cfg (バッチアプリケーション用オプション定義ファイル)
4. usrconf.properties (バッチサーバ用ユーザプロパティファイル)
5. usrconf.cfg (バッチサーバ用オプション定義ファイル)

詳細については、マニュアル「Cosminexus アプリケーションサーバ V8 リファレンス コマンド編」を参照のこと。

PGM="{* | コマンド名 | スクリプトファイル名 | Java アプリケーションクラス名}"

実行するプログラムを指定する。この属性は、REPLACE 要素内で EXEC 要素を置換する場合にだけ省略できる。

- *
実行するプログラム、コマンド、シェルスクリプトの名称と、その引数を要素の内容に指定していることを示す。
- コマンド名～<任意文字列>((1～64 文字))
実行するプログラム、コマンド、シェルスクリプトの名称を指定する。
- スクリプトファイル名～<任意文字列>((1～1023 文字))
JP1/Script のスクリプトファイル名を指定する。
- Java アプリケーションクラス名～<任意文字列>((1～1023 文字))
パッケージ名を含めた Java アプリケーションクラス名を指定する。

PARM="{引数 | main メソッドに渡す引数}" ～<任意文字列>((1～958 文字))

- 引数
実行するプログラムに与える引数を指定する。ただし、PGM 属性が"*"の場合は指定しても無視する。
文字列中に空白を含む場合は、この属性が指定された EXEC 要素の PARMSPLIT 属性の指定に従い、プログラムにどのように引数を与えるかを決定する。
LANG="JP1Script"の場合は JP1/Script に渡す引数として sptxe の位置変数とオプションを指定する。位置変数とオプションは、空白で区切って指定する。この場合は、PARMSPLIT 属性の指定は無視される。
- メソッドに渡す引数
LANG="Java"の場合は Java アプリケーションの main メソッドに渡す引数を指定する。

PARMSPLIT="{YES | NO}"

PARM 属性値が空白を含む場合、プログラムにどのように引数を与えるかを指定する。この属性を省略した場合、設定ファイルの EXEC_PARMSPLIT パラメータの指定値が有効になる。

- YES
UNIX の一般的な引数インタフェースである C 言語インタフェースに従った形式を使用する場合に、YES を指定する。PARM 属性値が空白を含む場合、空白で区切られた複数の引数としてプログラムに渡される。
- NO
VOS3 から移行した COBOL プログラムなどの VOS3 インタフェースに従った形式を使用する場合に、NO を指定する。PARM 属性値が空白を含む場合、空白を含む 1 つの引数としてプログラムに渡される。

EXEC 要素の PARMSPLIT 属性と設定ファイルの EXEC_PARMSPLIT パラメータの指定による PARM 属性の扱いを次の表に示す。

表 9-4 EXEC 要素の PARMSPLIT 属性と設定ファイルの EXEC_PARMSPLIT パラメータの指定による PARM 属性の扱い

EXEC_PARMSPLIT パラメータ	PARMSPLIT 属性		
	YES	NO	省略
YES または省略	分割する※1。	分割しない※2。	分割する※1。
NO	分割する※1。	分割しない※2。	分割しない※2。

注※1 空白で分割した複数の引数とする。

注※2 空白を含む 1 つの引数とする。

STDOUTCLASS="クラス名" ～<記号名称>((1～31 文字))

ユーザプログラムの標準出力を格納するファイルの出力クラスを指定する。この属性を省略した場合は、JOB 要素の STDOUTCLASS 属性で指定された出力クラスとなる。

出力クラス名を SysoutClass パラメータで定義していない場合は、KAKC1124-E メッセージを出力してジョブ定義 XML ファイルエラーとなる。

STDERRCLASS="クラス名" ～<記号名称>((1～31 文字))

ユーザプログラムの標準エラー出力を格納するファイルの出力クラスを指定する。この属性を省略した場合は、JOB 要素の STDERRCLASS 属性で指定された出力クラスとなる。

出力クラス名を SysoutClass パラメータで定義していない場合は、KAKC1124-E メッセージを出力してジョブ定義 XML ファイルエラーとなる。

内容

PGM 属性値に "*" を指定した場合、この要素の内容に、<![CDATA[!と!]]>で囲んで、プログラム名、引数など、シェルスクリプトを指定する。プログラム名や引数に、属性値の引用符内の文字列に指定できない、", または & の文字が含まれている場合に、この要素の内容に記述する。

JP1/AJS で起動した Advanced Shell のスクリプトから実行するときは、JP1/AJS が環境変数 SHELL に Advanced Shell を設定することがあるので、PGM 属性値に "*" を指定した場合、注意が必要である。詳細は、「表 5-1 JP1/AJS からバッチジョブを起動した場合の強制終了の方法と制限事項」を参照のこと。

注意事項

- EXEC 要素は、STEP 要素内に 1 つだけ指定できる。ただし、同一 STEP 要素内に CALL 要素がある場合には指定できない。
- COND 属性の注意事項を次に示す。
 - 条件パラメータにジョブステップ名およびプロシジャステップ名を記入していない場合、先行するすべてのジョブステップが判定の対象となる。
 - 条件パラメータで指定したジョブステップ名およびプロシジャステップ名が先行するジョブステップ中に 2 つ以上ある場合、該当するジョブステップ中の先頭が判定の対象となる。

- 条件パラメータは、EVEN または ONLY を含めて 8 つまで指定できる。
- 1 つの条件パラメータだけ記入する場合、両端の括弧 " () " を省略できる。
- EVEN または ONLY は、必ずしも条件パラメータの最後に記入しなくてもよい。
- EXEC 要素と JOB 要素の両方で COND 属性を指定した場合、JOB 要素で指定した条件を満たすと、EXEC 要素で指定した条件を満たさないときも、後続のジョブステップを実行しない。
- 指定した先行するジョブステップが条件付き実行によってすでに実行がスキップされている場合、COND 属性は無視される。
- 自ジョブステップと並列実行するジョブステップを参照しても無効となる。
- 空白で分割した複数の引数としてプログラムに渡される場合、次に示すような引数に空白を含む指定はできない。

```
引数 1 : AAA
引数 2 : BBB△CCC
```

このような指定をしたい場合は、PGM 属性に "*" を指定してシェルスクリプトを記述し、シェルの仕様に従って空白を含む引数を指定する。

```
<EXEC PGM="*">
<![CDATA[
!
program△AAA△"BBB△CCC"
!
]]>
</EXEC>
```

使用例

プログラム PROG1 を実行する場合の使用例を次に示す。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="JOB1">
    <STEP NAME="STEP">
      <EXEC PGM="PROG1" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

また、シェルとして実行する場合の使用例を次に示す。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="JOB1">
    <STEP NAME="STEP">
      <EXEC PGM='*'>
        <![CDATA[
!
PROG1
!
]]>
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

```
]]>  
</EXEC>  
</STEP>  
</JOB>  
</HitachiBatchJobExec>
```

HitachiBatchJobExec 要素 (ジョブ定義 XML ファイルの宣言)

形式

```
[os="os種別"]  
[version="メジャーバージョン番号.マイナーバージョン番号"]
```

機能

ジョブ定義 XML ファイルのルート要素であり、このファイルがジョブ定義 XML ファイルであることを示す。また、ジョブ定義 XML ファイルのバージョンや動作環境を定義する。

属性

os="os 種別" ~<任意文字列>((1~31 文字))

ジョブ定義 XML ファイルに指定したジョブを実行する OS 種別を指定する。

指定は任意である。また、指定した内容のチェックを行わない。

例

- unix
ジョブ定義 XML ファイルを UNIX システムで実行することを指定する。
- windows
ジョブ定義 XML ファイルを Windows システムで実行することを指定する。

version="メジャーバージョン番号.マイナーバージョン番号" ~<任意文字列>((3~4 文字))

ジョブ定義 XML ファイルの DTD のメジャーバージョン番号とマイナーバージョン番号を指定する。指定は任意である。また、指定した内容のチェックを行わない。

例

- 1.4
ジョブ定義 XML ファイルの DTD のバージョン番号が 1.4 であることを指定する。
バッチジョブ実行システムのバージョンと version 属性に指定する値の対応を次の表に示す。

表 9-5 バッチジョブ実行システムのバージョンと version 属性に指定する値の対応

バッチジョブ実行システムのバージョン	version 属性に指定する値	
	AIX	Linux
08-05 以降	1.4	1.3
08-00	1.3	1.2
07-61	1.2	—
07-60	1.1	1.1

バッチジョブ実行システムのバージョン	version 属性に指定する値	
	AIX	Linux
01-00~07-50	1.0	—

(凡例)

—：該当する OS ではこのバージョンのバッチジョブ実行システムをサポートしていない。

注意事項

- HitachiBatchJobExec 要素はジョブ定義 XML ファイルのルート要素であり、ファイル中（XML 宣言よりあと）に指定している必要がある。また、同一ファイル中には複数指定しないこと。
- HitachiBatchJobExec 要素内には、JOB 要素または PROC 要素のどちらかを必ず指定すること。

使用例

```
<HitachiBatchJobExec version="1.3" os="unix">
```

IF 要素 (ジョブステップの実行制御の開始)

形式

```
OP="条件式"
```

機能

ジョブステップのリターンコードを判定し、条件が成立した場合、THEN 節のジョブステップを実行する。条件が成立しない場合は ELSE 節のジョブステップを実行する。IF 要素の使用の規則を次に示す。

- IF 要素は ELSE 要素と ENDIF 要素と組み合わせて使用する。
- IF 要素から ELSE 要素までを THEN 節と呼ぶ。また、ELSE 要素から ENDIF 要素までを ELSE 節と呼ぶ。
- ELSE 要素は省略できる。ELSE 要素を省略した場合、ELSE 節は存在しないため IF 要素から ENDIF 要素までが THEN 節となる。
- THEN 節または ELSE 節内のジョブステップからステップリスタートを実行できない。

属性

OP="条件式" ～((1～2048 文字))

判定条件を記述する。

条件式 ～{単一条件式 | NOT 条件式 | 条件式{AND | OR}条件式 | (条件式)}

複数の単一条件式を、論理積"AND"、論理和"OR"、否定"NOT"、括弧"()"で組み合わせたものである。

条件式は入れ子（ネスト）にすることができる。論理符号の優先順位は次のとおりになる（優先順位は 1 が最も高い）。

1. 括弧"()"：優先的に条件式を判定する。
2. 否定"NOT"：条件式が不成立の場合に真となる。
3. 論理積"AND"：2 つの条件式の両方が成立した場合に真となる。
4. 論理和"OR"：2 つの条件式のどちらかが成立した場合に真となる。

単一条件式 ～(*|?|[ジョブステップ名[.プロシジャステップ名]],演算符号,[リターンコード])

対象のジョブステップを実行するかどうかを判定するための条件の最小単位である。

*|?|[ジョブステップ名[.プロシジャステップ名]

条件の判定対象のジョブステップを指定する。

- *

演算符号に、EQ, NE, GT, GE, LT, LE のどれかを指定した場合、

リターンコードが最大値のジョブステップが条件に一致するか判定する。SETMAXRC 要素で変更した値は判定されない。

演算符号に、AB、NA のどちらかを指定した場合、

実行済みのジョブステップのすべてが条件に一致するか判定する。

- ?

実行済みのジョブステップのどれかが条件に一致するか判定する。

- ジョブステップ名[.プロシジャステップ名]

特定のジョブステップを判定する。

ジョブステップ名を検索する場合、ジョブの先頭から検索して最初に見つかったジョブステップを判定の対象にする。該当するジョブステップがカタログドプロシジャの内か外であるかや実行済みであるかどうかには関係はない。また、後続のジョブステップ名は指定できない。

ジョブステップ名： ~<記号名称>((1~31 文字))

ジョブステップ名を指定する。

プロシジャステップ名： ~<記号名称>((1~31 文字))

プロシジャ内のジョブステップのリターンコードを参照するときはプロシジャのジョブステップ名を指定する。

カタログドプロシジャ内でジョブステップ名だけを指定した場合、CALL 要素で呼び出したときのジョブステップ名を使用して「ジョブステップ名.プロシジャステップ名」と解釈する。このため、同一カタログドプロシジャを複数回呼び出す場合は、それぞれの CALL 要素で呼び出したときのジョブステップ名はジョブ内で固有にする必要がある。固有でない場合は同一の「ジョブステップ名.プロシジャステップ名」が複数存在するため、先頭のジョブステップの情報を参照することになる。

- 省略

判定対象を省略した場合、直前のジョブステップを判定対象にする。

判定対象のジョブステップが実行されていない場合、そのジョブステップのリターンコードは 0、かつ異常終了しなかったと仮定して処理する。

判定方法が EX および NX のときは*または?は指定できない。

演算符号 ~{EQ | NE | GT | GE | LT | LE | EX | NX | AB | NA}

判定方法を指定する。

- EQ：ジョブステップのリターンコード = OP 属性で指定したリターンコード
- NE：ジョブステップのリターンコード ≠ OP 属性で指定したリターンコード
- GT：ジョブステップのリターンコード > OP 属性で指定したリターンコード
- GE：ジョブステップのリターンコード ≥ OP 属性で指定したリターンコード
- LT：ジョブステップのリターンコード < OP 属性で指定したリターンコード
- LE：ジョブステップのリターンコード ≤ OP 属性で指定したリターンコード
- EX：指定したジョブステップを実行した。

- NX：指定したジョブステップを実行していない。
- AB：ジョブステップが異常終了した。
- NA：ジョブステップが異常終了しなかった。

リターンコード ～<符号なし整数>((0～255))

比較するリターンコード値を指定する。

演算符号が EX, NX, AB, NA のどれかのとき、この指定は無視される。

注意事項

- SETSYMBOL, REPLACE または REPLACESTEP の置き換えの対象外である。
- 論理積"AND"または論理和"OR"のあとに NOT を指定した条件を記述する場合、NOT を指定した条件は次に示すように括弧"()"で囲わなければならない。

(例 1) 正しい記述方法の例

```
<IF OP="(STEP01,LE,1)AND(NOT(STEP02,EQ,2) )"/>
```

(例 2) 誤った記述方法の例

```
<IF OP="( STEP01,LE,1)AND NOT(STEP02,EQ,2) "/>
```

- 次の例に示すような矛盾する条件式を記述しても、シンタクス上のエラーがない場合は判定を実施し、処理を続行する。

(例) 矛盾した条件指定の例

```
<IF OP="(STEP01,EX,)AND(STEP01,NX,)" />・・・STEP01 を実行した、かつ STEP01 を実行していない。
```

- 通常、先行するジョブステップが異常終了すると、判定条件が成立していても後続のジョブステップはスキップされる。ただし、演算符号が AB および NA の IF 要素内およびそれに対応する ELSE 要素内のすべてのジョブステップは、判定条件が成立していれば実行される。判定条件によるジョブステップの動作例を次に示す。

判定条件によるジョブステップの動作例

ジョブ開始	
STEP01 (正常終了, RC=0)	… (1) リターンコードが0で正常終了したジョブステップ。
STEP02 (異常終了)	… (2) 異常終了したジョブステップ。
STEP03	… (3) STEP02が異常終了しているため実行しない。
IF要素A (STEP01, EQ, 0)	… (4) 演算符号がEQのIF要素で、条件は真。
↳STEP04	… (5) IF要素Aの条件は真であるが、STEP02が異常終了しているため実行しない。
IF要素B (STEP01, NA,)	… (6) 演算符号がNAのIF要素で、条件は真。
↳STEP05	… (7) IF要素Bの条件が真であり、演算符号がABおよびNAのIF要素内のため実行する。
IF要素C (STEP01, EQ, 4)	… (8) 演算符号がEQのIF要素で、条件は偽。
↳STEP06	… (9) 演算符号がABおよびNAのIF要素内であるが、IF要素Cの条件が偽であるため実行しない。
ELSE要素C	… (10) 演算符号がEQのIF要素に対応するELSE要素で、条件は真。
↳STEP07	… (11) 演算符号がABおよびNAのIF要素内であり、IF要素Cの条件が偽であるため実行する。
STEP08	… (12) STEP02が異常終了しているため実行しない。
ジョブ終了	

使用例

IF 要素を使用した例については、「[3.7.2 ジョブステップの実行制御の使用例](#)」を参照のこと。

JOB 要素 (ジョブの属性と終了条件の定義)

形式

```
NAME="ジョブ名"  
[COND="([条件コード,演算符号]) [(条件コード,演算符号)]+" ]  
[MSGCLASS="クラス名"]  
[STDOUTCLASS="クラス名"]  
[STDERRCLASS="クラス名"]
```

機能

ジョブの初期値となる属性と終了条件を定義する。

ジョブ要素内には、次に示す順番で要素を含めることができる。要素の順番が適切でない場合、ジョブは実行されずにエラーとなる。

- JOBPARM 要素
1 回だけ記述、または省略できる。
- PERMISSION 要素
1 回だけ記述、または省略できる。
- DD 要素または、SETENV 要素
複数回記述、または省略できる。
DD 要素は、NAME 属性が JOBLIB、および TYPE 属性が LIB である DD 要素を記述できる。
- PROC 要素
複数回記述、または省略できる。
- STEP 要素と SETMAXRC 要素の繰り返し
複数回記述する。ただし、SETMAXRC 要素は省略できる。
- IF, ELSE および ENDIF 要素
SETMAXRC 要素の次に記述、または省略できる。

属性

NAME="ジョブ名" ~<記号名称>((1~31 文字))

ジョブを識別するためのジョブ名を指定する。

COND="([条件コード,演算符号D]) [(条件コード,演算符号)]+"

ジョブステップの終了コードによって、ジョブの実行を打ち切る条件を指定する。

- 条件コード ~<符号なし整数>((0~255))
ジョブステップの終了コードと比較するしきい値を指定する。1 つでも条件を満たすとジョブは終了し、後続ジョブステップを実行しない。

- 演算符号 ~((EQ | NE | GT | GE | LT | LE))

演算符号の意味を次に示す。

EQ：しきい値は終了コードと等しい。

NE：しきい値は終了コードと等しくない。

GT：しきい値は終了コードより大きい。

GE：しきい値は終了コードより大きいか等しい。

LT：しきい値は終了コードより小さい。

LE：しきい値は終了コードより小さいか等しい。

MSGCLASS="クラス名" ~<記号名称>((1~31 文字))

ジョブログやシステムから出力されるメッセージの出力クラスを指定する。また、DD 要素で SYSOUT="*"を指定した場合の出力クラスになる。省略した場合、またはジョブ定義 XML ファイルにエラーがあり、この属性を正しく解析できなかった場合、MSGCLASS_DEFAULT パラメータで指定されたクラスになる。

STDOUTCLASS="クラス名" ~<記号名称>((1~31 文字))

EXEC 要素の STDOUTCLASS 属性を省略した場合のユーザプログラムの標準出力を格納するファイルの出力クラスを指定する。この属性を省略した場合は、設定ファイルの STDOUTCLASS_DEFAULT パラメータを仮定する。

出力クラス名を SysoutClass パラメータで定義していない場合は、KAKC1124-E メッセージを出力してジョブ定義 XML ファイルエラーとなる。

STDERRCLASS="クラス名" ~<記号名称>((1~31 文字))

EXEC 要素の STDERRCLASS 属性を省略した場合のユーザプログラムの標準エラー出力を格納するファイルの出力クラスを指定する。この属性を省略した場合は、設定ファイルの STDOUTCLASS_DEFAULT パラメータを仮定する。

出力クラス名を SysoutClass パラメータで定義していない場合は、KAKC1124-E メッセージを出力してジョブ定義 XML ファイルエラーとなる。

注意事項

- COND 属性での条件パラメータは 8 つまで指定できる。条件パラメータを 1 つだけ指定する場合は、外側の括弧"()"を省略できる。
- JOB 要素と EXEC 要素の両方で COND 属性を指定した場合、JOB 要素で指定した条件を満たすと、EXEC 要素で指定した条件を満たさないときも、後続のジョブステップを実行しない。

使用例

JOB 要素の直下に指定できるすべての要素を指定したジョブの例を次に示す。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version = "1.3" os="unix" >
  <JOB NAME="JOB1" COND="0,LT" MSGCLASS="A">
    <JOBPARM FCB="FCB1"/>
    <PERMISSION FILE="755"/>
  </JOB>
</HitachiBatchJobExec>
```

```
<DD NAME="JOBLIB" TYPE="LIB" DSN="DIR1" DISP="SHR"/>
<DD NAME="JOBLIB" TYPE="LIB" DSN="DIR2" DISP="SHR"/>
<SETENV NAME="ENV1" VALUE="VALUE1"/>
<SETENV NAME="ENV2" VALUE="VALUE2"/>
<PROC NAME="PROC1">
  <STEP>
    <EXEC PGM="PROG1"/>
  </STEP>
</PROC>
<PROC NAME="PROC2">
  <STEP>
    <EXEC PGM="PROG2"/>
  </STEP>
</PROC>
<STEP NAME="STEP1">
  <EXEC PGM="PROG2"/>
</STEP>
<STEP NAME="STEP2">
  <CALL NAME="PROC1"/>
  <DD NAME="DD2" TYPE="FILE" DSN="DSN1" DISP="NEW,DELETE"/>
</STEP>
<SETMAXRC VALUE="0"/>
<STEP NAME="STEP3">
  <EXEC PGM="PROG3"/>
  <DD NAME="DD3" TYPE="SYSOUT" SYSOUT="A"/>
</STEP>
</JOB>
</HitachiBatchJobExec>
```

JOBPARM 要素 (ジョブ単位の制御情報の定義)

形式

```
[FCB="FCB識別子"]  
[FORMS="書式番号"]  
[PROCLIB="プロシジャグループ名"]  
[ROOM="仕分け番号"]
```

機能

ジョブ単位の制御情報を定義する。

属性

FCB="FCB 識別子" ～<記号名称>((1～4 文字))

JOB 要素内の各 DD 要素の FCB 属性値に、FCB 名の指定がない場合の FCB 名を指定する。

FORMS="書式番号" ～<記号名称>((1～4 文字))

JOB 要素内の各 DD 要素の SYSOUT 属性値に書式番号指定がない場合に与える初期値の書式番号を指定する。

PROCLIB="プロシジャグループ名" ～<記号名称>((1～31 文字))

使用するプロシジャライブラリのグループ名を指定する。

使用されるカタログドプロシジャは、設定ファイルで指定したカタログドプロシジャのプロシジャグループの定義に従う。この指定がない場合、デフォルトのグループ名なしのプロシジャグループを使用するか、PROCGROUP_QUEUENAME パラメータで YES を指定した場合には、JP1/AJS のキュー名からプロシジャグループ名を決定する。

定義したグループが設定ファイルに定義されていない場合は、デフォルトのグループを仮定する。

ROOM="仕分け番号" ～<記号名称>((1～4 文字))

PDE で使用する仕分け番号を指定する。

注意事項

JOBPARM 要素は、JOB 要素内に指定すること。

使用例

ジョブの仮定値として FCB 名を FCB1、仕分け番号を 1 とする場合の使用例を次に示す。

```
:  
<JOB NAME="DBJNL">  
<JOBPARM FCB="FCB1" ROOM="1"/>  
:
```

PERMISSION 要素（ファイルまたはディレクトリのパーミッションの定義）

形式

```
[DIR="ディレクトリのパーミッション"]  
[FILE="ファイルのパーミッション"]
```

機能

新規作成するファイル、ジョブ間 PREST ファイル、またはディレクトリのパーミッションを定義する。この要素を指定しない場合は、設定ファイルの PERMISSION_DIR パラメータおよび PERMISSION_FILE パラメータの指定に従う。DD 要素の FPERM 属性を指定した場合は、その指定が有効になる。

属性

DIR="ディレクトリのパーミッション" ～< 8 進数>((0~1777))

新規作成するディレクトリのパーミッションを指定する。指定方法は chmod コマンドに従う。

FILE="ファイルのパーミッション" ～< 8 進数>((0~777))

新規作成するファイルまたはジョブ間 PREST ファイルのパーミッションを指定する。指定方法は chmod コマンドに従う。

注意事項

PERMISSION 要素は、次に示す場合は有効とならない。

- 新規作成以外のファイル
- 一時ファイル
- 一時 ISAM ファイル
- ジョブログ
- SYSOUT
- ジョブステップ間 PREST ファイル

使用例

ジョブが作成する新規ファイルの権限の仮定値として、ファイルには 644、ディレクトリには 750 の属性を設定する場合の使用例を次に示す。

```
:  
<JOB NAME="SAMPLE" >  
<PERMISSION FILE="644" DIR="750" />  
:
```

PROC 要素（プロシジャの定義）

形式

```
NAME="プロシジャ名"
```

機能

プロシジャを定義する。プロシジャは、頻繁に使用する一連の要素を毎回指定することの手間を省くことと、要素記述量を減少させて誤りを少なくするために用いる。CALL 要素の NAME 属性値にプロシジャ名を指定することによって内容が展開され、展開された内容が指定したものと同等に扱われる。

PROC 要素内には、次に示す順番で要素を含めることができる。

- SYMBOL 要素
複数回記述，または省略できる。
- STEP 要素
1 回以上記述する。
- SETMAXRC 要素
STEP 要素の次に記述，または省略できる。
- IF, ELSE および ENDIF 要素
SETMAXRC 要素の次に記述，または省略できる。

属性

NAME="プロシジャ名" ～<記号名称>((1～31 文字))

定義するカタログドプロシジャまたは入力ストリームプロシジャのプロシジャ名を指定する。

注意事項

- カタログドプロシジャの場合は，同一ファイル内に PROC 要素を 1 つだけ指定し，JOB 要素は指定しない。PROC 要素は，HitachiBatchJobExec 要素内に指定する。
- 入力ストリームプロシジャの場合は，PROC 要素を複数指定できる。この場合，PROC 要素を JOB 要素中の最初の STEP 要素より前に指定する。ただし，同じ NAME 属性で PROC 要素を複数指定しないこと。
- JOB 要素がない場合，PROC 要素を複数指定するとエラーとなる。
- バッチジョブ実行システムは，REPLACE 要素を適用したあとでプロシジャを解析する。例えば，DSN 属性として*.OLDSTEP.DDNAME を指定した DD 要素に対して，REPLACE 要素で DSN 属性を*.NEWSTEP.DDNAME に置き換えている場合，バッチジョブ実行システムは DSN 属性部分を*.NEWSTEP.DDNAME に置き換えてからプロシジャを解析する。

使用例

プログラム PROG1 をカタログドプロシジャとして定義する場合の使用例を次に示す。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <PROC NAME="ASMG">
    <STEP NAME="STEP">
      <EXEC PGM="PROG1" />
    </STEP>
  </PROC>
</HitachiBatchJobExec>
```

REPLACE 要素 (DD 要素または EXEC 要素の属性の変更内容の定義)

機能

カタログドプロシジャまたは入出力ストリームプロシジャ内の先頭ジョブステップの DD 要素, EXEC 要素内の属性の追加, 変更, または削除をする場合, 変更内容を定義する。

REPLACE 要素内には, 次に示す順番で要素を含めることができる。

- EXEC 要素
1 回だけ記述, または省略できる。
- DD 要素
複数回記述できる。または省略できる。
- REPLACESTEP 要素
複数回記述できる。または省略できる。

属性

なし。

注意事項

- EXEC/DD 要素内の属性値の置換
REPLACE 要素内に, 置き換えする EXEC/DD 要素と, 置き換えする属性を記述する。プロシジャ内の先頭ジョブステップ以外のジョブステップを置き換えする場合は, REPLACESTEP 要素を利用する。EXEC 要素の PGM 属性および DD 要素の NAME 属性は置き換えできない。
- EXEC/DD 要素内の属性値の追加
PROC 要素内の EXEC/DD 要素内に, 属性として定義されていない属性を追加する場合は, 置き換えと同様に, 追加する EXEC/DD 要素と属性を記述する。REPLACE 要素内に, EXEC/DD 要素を追加した場合, プロシジャ内の先頭のジョブステップに対して追加する。
- EXEC/DD 要素内の属性値の削除
PROC 要素内の EXEC/DD 要素内に属性として定義されている属性を削除する場合は, 削除する EXEC/DD 要素と属性を記述し, その属性値を NULL とする。REPLACE 要素内に EXEC/DD 要素を記述した場合, プロシジャ内の先頭のジョブステップに対して削除する。DD 要素自体を削除する場合, TYPE 属性に DUMMY を指定する。

使用例

カタログドプロシジャ UTILITY の, DD 名が SYSPRINT の DD 要素の出力先を /opt/var/sysout に変更する場合の使用例を次に示す。

```
:  
<CALL NAME="UTILITY">
```

```
<REPLACE>  
  <DD NAME="SYSPRINT" TYPE="FILE" DSN="/opt/var/sysout"/>  
</REPLACE>  
</CALL>
```

REPLACESTEP 要素 (DD 要素または EXEC 要素の属性を変更対象とするジョブステップの定義)

形式

```
NAME="ジョブステップ名"
```

機能

カタログドプロシジャまたは入出力ストリームプロシジャ内の DD 要素, EXEC 要素内の属性の追加, 変更, または削除をする場合, その対象となるジョブステップを定義する。

REPLACESTEP 要素内には, 次に示す順番で要素を含めることができる。

- EXEC 要素
1 回だけ記述, または省略できる。
- DD 要素
複数回記述できる。または省略できる。

属性

NAME="ジョブステップ名" ~<記号名称>((1~31 文字))

属性の追加, 変更, または削除をするジョブステップ名を指定する。

注意事項

- EXEC 要素の PGM 属性および DD 要素の NAME 属性は置き換えできない。
- バッチジョブ実行システムは, REPLACESTEP 要素を適用したあとでプロシジャを解析する。例えば, ジョブステップ名として OLDSTEP を指定した COND 属性に対して, REPLACESTEP 要素でジョブステップ名部分を NEWSTEP に置き換えている場合, バッチジョブ実行システムはジョブステップ名部分を NEWSTEP に置き換えてから COND 属性を解析する。

使用例

カタログドプロシジャ UTILITY の STEP1 にある EXEC 要素の PARM 属性を削除する場合の使用例を次に示す。

```
:  
<CALL NAME="UTILITY">  
<REPLACE>  
  <REPLACESTEP NAME="STEP1">  
    <EXEC PARM="" />  
  </REPLACESTEP>  
</REPLACE>  
</CALL>
```

SETENV 要素 (環境変数の定義)

形式

```
NAME="環境変数名"  
VALUE="環境変数設定値"
```

機能

環境変数を定義する。

属性

NAME="環境変数名" ～<環境変数名>((1～255 文字))

設定する環境変数名を指定する。

VALUE="環境変数設定値" ～<任意文字列>((0～1023 文字))

環境変数に設定する値を指定する。

注意事項

- 同一の JOB または STEP 要素内には、SETENV 要素を 255 個まで指定できる。
- 同一の JOB または STEP 要素内には、NAME 属性値が同じ SETENV 要素を複数指定しないこと。
- SETENV 要素を JOB 要素内の STEP 要素外に指定した場合、その環境変数はジョブ全体で有効となる。
- SETENV 要素を STEP 要素内に指定した場合、その環境変数はそのジョブステップ内でだけ有効となる。
- CALL 要素を定義したジョブステップ内に、SETENV 要素を定義しても、その SETENV 要素は無視する。

使用例

環境変数 MAIL に /usr/spool/mail/user を設定する場合の使用例を次に示す。

```
:  
<SETENV NAME="MAIL" VALUE="/usr/spool/mail/user"/>  
:
```

SETENVFILE 要素 (環境変数格納ファイルの定義)

形式

DDNAME="環境変数格納ファイルを定義しているDD要素のDD名"

機能

環境変数格納ファイルを定義する。

属性

DDNAME="環境変数格納ファイルを定義している DD 要素の DD 名" ~<環境変数名>((1~31 文字))

環境変数格納ファイルを定義している DD 要素の DD 名を指定する。

同一 STEP 要素内に複数の SETENVFILE 要素を指定した場合、指定した順に環境変数を設定する。

なお、同じ環境変数名を複数指定している場合、最後に指定した環境変数値が設定される。

また、DD 名で示される環境変数格納ファイルが連結されている場合、ファイルの連結順に環境変数が設定される。

環境変数格納ファイルとして世代データ群名を指定した場合、指定した世代群に登録されているすべての世代ファイルが連結されて、環境変数が設定される。

注意事項

- 同一の STEP 要素内には、SETENVFILE 要素を複数指定できる。
- 同一の STEP 要素内に、DDNAME 属性値が同じ SETENVFILE 要素を複数指定した場合、最後に指定した SETENVFILE 要素が有効となる。
- SETENVFILE 要素で指定した DD 名を持つ DD 要素が見つからない場合、XML 解析時にエラーとなる。
- CALL 要素を定義したジョブステップ内に SETENVFILE 要素を定義しても、その SETENVFILE 要素は無視する。
- 同一の STEP 要素内に、SETENV 要素と共に指定した場合、SETENV 要素は指定したジョブステップ内だけ有効となり、後続ジョブステップには有効とならない。

使用例

環境変数格納ファイルを DD 名"ENVFILE"で指定する場合の使用例を次に示す。

```
:  
<DD NAME="ENVFILE" TYPE="FILE" DSN="Envvarfile" DISP="SHR" />  
<SETENVFILE DDNAME="ENVFILE" />  
:
```

SETMAXRC 要素 (ジョブステップの終了コードの最大値, または最終ジョブステップの終了コードの変更)

形式

```
VALUE="最大終了コード"
```

機能

- 設定ファイルの JOB_RCDEFINE パラメータが STEPMAX の場合
ジョブの途中で、ジョブコントローラの終了コードであるジョブステップの終了コードの最大値を変更する。
- 設定ファイルの JOB_RCDEFINE パラメータが STEPLAST の場合
ジョブコントローラの終了コードである最後に実行完了したメインステップの終了コードを変更する。

SETMAXRC 要素は、先行ジョブステップの終了コードを無視したい場合や、JP1/AJS のジョブ終了判定で、しきい値による判定をしたい場合などに指定する。

属性

VALUE="最大終了コード" ~<符号なし整数>((0~255))

- 設定ファイルの JOB_RCDEFINE パラメータが STEPMAX の場合
先行ジョブステップの終了コードの最大値を指定する。
- 設定ファイルの JOB_RCDEFINE パラメータが STEPLAST の場合
最終ジョブステップの終了コードを変更する。

注意事項

- SETMAXRC 要素は、STEP 要素のあとに記述する。SETMAXRC 要素の直前のジョブステップ終了後、終了コードの最大値を変更する。
- SETMAXRC 要素の直前のジョブステップを COND 属性の条件でスキップした場合であっても、終了コードの最大値を変更する。IF 要素でスキップした場合は終了コードの最大値は変更しない。
- SETMAXRC 要素で指定した終了コードは、COND 属性および IF 要素の判定では使用しない。
- 設定ファイルの JOB_RCDEFINE パラメータに STEPLAST を指定した場合、最後に実行完了したジョブステップの直後に指定した SETMAXRC 要素だけが有効となる。その他の SETMAXRC 要素は、ジョブステップの終了コードの最大値を書き換えたことを示す KAKC1909-I メッセージは出力されるが、ジョブの終了コードには影響を与えない。
- JSPEC 機能を用いてジョブステップを並列実行する場合、SETMAXRC 要素はメインステップの直後に指定すること。複数のサブステップの直後に SETMAXRC 要素を指定した場合、最終的にどの SETMAXRC 要素の指定が有効になるかはサブステップの終了順序に依存し、最後に完了したサブステップの直後に指定した SETMAXRC 要素が有効になる。

使用例

SETMAXRC 要素で先行ジョブステップの終了コードを無視する場合の使用例を次に示す。

SETMAXRC 要素より先に記述したジョブステップ STEP01, および STEP02 の終了コードがどのような値であっても, ジョブステップの終了コードの最大値を 0 に変更する。STEP01 または STEP02 の終了コードが, STEP03 の終了コードより大きい場合であっても, STEP03 の終了コードがジョブの終了コードとなる。

```
<STEP NAME="STEP01" >  
:  
</STEP>  
<STEP NAME="STEP02" >  
:  
</STEP>  
<SETMAXRC VALUE="0" />  
<STEP NAME="STEP03" >  
:  
</STEP>
```

SETSYMBOL 要素（記号パラメータの値の定義）

形式

```
NAME="記号パラメータ名"  
VALUE="記号パラメータ設定値"
```

機能

カタログドプロシジャまたは入力ストリームプロシジャ中で定義した記号パラメータに値を定義する。PROC 要素内の記号パラメータ ("# {記号パラメータ名} ") がある場合、記号パラメータ名が一致する SYMBOL 要素内の VALUE 属性値で置き換える。置き換えができるのは、EXEC 要素と DD 要素および SETENV 要素の中の属性値 (" で囲まれている中) である。ただし、EXEC 要素の PGM 属性と DD 要素の NAME 属性および SETENV 要素の NAME 属性は置き換えできない。

属性

NAME="記号パラメータ名" ～<記号名称>((1～31 文字))

値を設定する記号パラメータ名を指定する。

VALUE="記号パラメータ設定値" ～<任意文字列>((0～255 文字))

記号パラメータに設定する値を指定する。

注意事項

- プロシジャ中の要素の属性値に、記号パラメータの記述目的以外で「#{…}」, または「#{ }」は記述できない。
- 同一 CALL 要素内に、NAME 属性値が同じ SETSYMBOL 要素を複数指定した場合、最後に指定した VALUE 属性値が有効となる。

使用例

カタログドプロシジャの呼び出し時に PARM1 を置き換える場合の使用例を次に示す。

```
<CALL NAME="ASMCG" >  
<SETSYMBOL NAME="PARM1" VALUE="/home/user1/" />  
</CALL>
```

STEP 要素 (ジョブステップの定義)

形式

```
[NAME="ジョブステップ名"]
```

機能

実行するプログラムまたは呼び出すプロシジャを定義するとともに、これらを実行する場合の各要素の開始を定義する。

STEP 要素内には、次に示す順番で要素を含めることができる。

- EXEC 要素, または CALL 要素
必ずどちらか一方を指定する。
- DD 要素, SETENV 要素, または SETENVFILE 要素
複数回記述, または省略できる。ただし, このジョブステップ内に CALL 要素を指定した場合は, DD 要素, SETENV 要素, または SETENVFILE 要素を指定しても無視する。

属性

NAME="ジョブステップ名" ~<記号名称>((1~31 文字))

該当するジョブステップ名を指定する。

注意事項

- STEP 要素は, JOB 要素内または PROC 要素内に指定すること。
- 同一 JOB 要素内には, STEP 要素を 255 個まで指定できる。ただし, STEP 要素内に CALL 要素がある場合は, その STEP 要素は含めずに代わりに CALL 要素の NAME 属性が示すプロシジャ内の STEP 要素数を含める必要がある。

使用例

ジョブステップ名に STEP01 というジョブステップを定義する場合の使用例を次に示す。

```
<STEP NAME="STEP01" >  
:  
</STEP>
```

SYMBOL 要素 (記号パラメータの定義)

形式

```
NAME="記号パラメータ名"  
[VALUE="記号パラメータ値"]
```

機能

PROC 要素のプロシジャ内の記号パラメータに省略時の値を定義する。

置き換えができるのは、EXEC 要素、DD 要素および SETENV 要素の中の属性値 (""で囲まれている中) である。ただし、EXEC 要素の PGM 属性と DD 要素および SETENV 要素の NAME 属性は置き換えできない。

属性

NAME="記号パラメータ名" ～<記号名称>((1～31 文字))

プロシジャ内の SYMBOL 要素以外の要素・要素の内容に現れる記号パラメータの先頭の「#{」と末尾の「}」とを取ったものである。記号パラメータをプロシジャ中の EXEC 要素や DD 要素中で定義するとき、記号パラメータの前に「#{」を、あとに「}」を付けて指定する。

例 # {ABC}

なお、XML 中に記号パラメータの記述目的以外で「#{…}」または「#{}」は記述できない。

VALUE="記号パラメータ値" ～<任意文字列>((0～255 文字)) 《NULL》

PROC 要素内に SYMBOL 要素を指定した場合、プロシジャ中に現れる"# {記号パラメータ}"と置き換える文字列の省略値を指定する。

注意事項

- SYMBOL 要素は、PROC 要素内に 255 個まで指定できる。
- 同一 PROC 要素内に NAME 属性値が同じ SYMBOL 要素を複数指定した場合、最初に指定した VALUE 属性値が有効となる。

使用例

カタログドプロシジャ PARM の記号パラメータ PARM1 の仮定値を、/USER/SYS1/と定義する場合の使用例を次に示す。

```
<PROC NAME="PARM">  
  <SYMBOL NAME="PARM1" VALUE="/USER/SYS1/" />  
  <STEP NAME="CHANGE">  
    <EXEC PGM="COPY" PARM="# {PARM1}DATA" />  
  </STEP>  
</PROC>
```

10

メッセージと終了コード

この章では、バッチジョブ実行システムが出力するメッセージおよび終了コードについて説明する。

10.1 メッセージの形式

10.1.1 メッセージの出力形式

出力するメッセージの形式を次に示す。

```
KAKCnnnn-t メッセージテキスト  
KAKDnnnn-t メッセージテキスト
```

- KAKC および KAKD
バッチジョブ実行システムのメッセージであることを示す。
- nnnn
メッセージ番号を示す。
- t
タイプコードを次の表に示す。メッセージに対する処置の指標を示すものであり、次に示す種類がある。

表 10-1 タイプコード

タイプコード	種類	意味
E	エラー (Error)	<ul style="list-style-type: none">• 各ライブラリ、コマンド、サーバの機能が働かない障害が起きたことを示す。• 定義誤り、コマンドのオペランド指定誤りによって、動作できないことを示す。
W	警告 (Warning)	メッセージ出力後、処理は続けられる。
I	情報 (Information)	ユーザに情報を知らせる。

なお、ジョブログに出力するメッセージには時刻・ジョブ識別子を与える。詳細については、「[3.1.3\(2\) ジョブログ出力機能](#)」を参照のこと。

10.1.2 メッセージの記載形式

出力するメッセージの記載形式を次に示す。

英語メッセージテキストは、標準出力、標準エラー出力、JOBLOG、SYSMSG、または実行ログに出力するメッセージテキストを示す。JP1 イベント日本語メッセージテキストは、JP1 イベントで送るメッセージテキストを示す。メッセージテキストは、メッセージの出力先によって英語メッセージテキストと JP1 イベント日本語メッセージテキストの両方を記載している場合と、どちらか一方だけを記載している場合がある。

メッセージテキスト中のゴシックで書かれている部分は、表示内容が変わる個所を示す。

メッセージはメッセージ ID 順に記載している。記載形式の例を次に示す。

メッセージ ID

英語メッセージテキスト

JPI イベント日本語メッセージテキスト

メッセージの説明文

(S)システムの処置を示す。

(O)メッセージが出力された場合に、オペレーターの取る処置を示す。

10.1.3 メッセージ番号の割り当て

(1) KAKC で始まるメッセージ番号の割り当て

KAKC で始まるメッセージ番号の割り当てを次の表に示す。

表 10-2 KAKC で始まるメッセージ番号の割り当て

メッセージ番号	用途および内容
0001~0299	基本稼働情報
0300~0399	コマンド引数関連メッセージ
0400~0699	初期設定ファイル関連メッセージ
0700~0899	ジョブログファイル関連メッセージ
0900~1099	XML パーサ関連メッセージ
1100~1599	ジョブ定義 XML ファイル文法関連メッセージ
1600~1899	アロケータ関連メッセージ
1900~2199	ジョブステッププロセス実行関連メッセージ
2200~2499	メッセージ処理関連メッセージ
3000~3999	スプールジョブ転送デーモン関連メッセージ
4000~4199	世代コマンド関連メッセージ
4200~4599	スプールジョブ管理コマンドメッセージ
5000~5199	メッセージ制御ライブラリメッセージ
8000~8999	トレース用

(2) KAKD で始まるメッセージ番号の割り当て

KAKD で始まるメッセージ番号の割り当てを次の表に示す。

表 10-3 KAKD で始まるメッセージ番号の割り当て

メッセージ番号	用途および内容
1000~1499	bjexu_pcopy 関連メッセージ
2000~2499	bjexu_scopy 関連メッセージ
3000~3499	bjexu_ismut および bjexu_ismut64 関連メッセージ

10.1.4 システム管理者への指示

- 運用中にトラブルが発生した場合には、「6. 障害対策」を参照してログ情報を採取し、初期調査をすること。
- トラブル要因の初期調査をする場合は、OS のログ情報（UNIX の場合は syslog など）や、バッチジョブ実行システムおよび JP1 が出力する各種ログ情報を参照する。これらのログ情報でトラブル発生時間帯の内容を参照して、トラブルの回避および対処をすること。
- トラブルが発生するまでの操作方法などは記録しておき、同時にできるだけ再現性の有無を確認すること。
- **【Linux 限定】**
トラブル要因の初期調査が迅速に行えるように適切な core 出力設定をすること。

10.1.5 メッセージの出力

設定ファイルの同一行、およびジョブ定義 XML ファイルの同一要素に、複数の誤りがある場合、KAKC0432-E メッセージなど、文法エラーを示す同一メッセージ ID のメッセージを複数出力することがある。

10.2 メッセージの出力先一覧

バッチジョブ実行システムが出力する各メッセージの出力先一覧を次に示す。

10.2.1 KAKC で始まるメッセージの出力先一覧

KAKC で始まるメッセージの出力先一覧を次の表に示す。

表 10-4 KAKC で始まるメッセージの出力先一覧

メッセージ ID の範囲	stdout	stderr	JOBLOG	SYSMSG	実行ログ	JP1
KAKC0001-E	—	○	○	○	○	—
KAKC0002-W, KAKC0003-W	—	○	—	—	—	—
KAKC0004-I	—	—	—	—	○	—
KAKC0005-E	—	○※5	○※5	○※5	○※5	—
KAKC0006-E~KAKC0012-E	—	○	—	—	○	—
KAKC0013-E	—	—	—	—	—	○
KAKC0050-E	—	○	—	—	—	—
KAKC0060-E	—	○	—	—	—	—
KAKC0080-I	—	○	—	—	○	—
KAKC0090-I	—	○	—	—	—	—
KAKC0091-I, KAKC0092-I	—	—	○	—	○	○
KAKC0093-E~KAKC0095-E	—	○	—	—	—	—
KAKC0096-E~KAKC0098-I	—	○	○	—	○	○
KAKC0099-I	—	○	—	—	—	—
KAKC0100-E	—	—	—	—	—	○
KAKC0101-E	—	○	○	—	○	○
KAKC0103-E, KAKC0105-E	—	○	—	—	—	—
KAKC0110-E, KAKC0112-I	—	○	○	—	○	—
KAKC0199-E~KAKC0202-E	—	○	—	—	○※1	—
KAKC0210-E~KAKC0238-E	—	○	—	—	○	—
KAKC0299-E	—	○	○	○	○	—
KAKC0300-I~KAKC0702-E	—	○	—	—	—	—
KAKC0703-E	—	○	—	—	○	—

メッセージ ID の範囲	stdout	stderr	JOBLOG	SYSMSG	実行ログ	JP1
KAKC0704-E~KAKC0706-E	-	○	-	-	-	-
KAKC0707-I	-	-	-	○	-	-
KAKC0709-E~KAKC0712-E	-	-	-	○	○	-
KAKC0719-I~KAKC0724-I	-	○	-	-	-	-
KAKC0730-E	-	○	-	-	○	-
KAKC0800-E~KAKC0802-E	-	○	-	-	-	-
KAKC0803-E	-	○	-	○	○	-
KAKC0804-E	-	○	-	-	-	-
KAKC0805-E	-	○	-	○	○	-
KAKC0900-I~KAKC1625-E	-	-	-	○	-	-
KAKC1626-E, KAKC1627-I	-	○※5	-	○※5	-	-
KAKC1628-E~KAKC1648-E	-	-	-	○	-	-
KAKC1650-W~KAKC1655-I	-	○	○	-	○	○
KAKC1660-E~KAKC1691-E	-	-	-	○	-	-
KAKC1717-E~KAKC1719-E	-	○	-	-	-	-
KAKC1720-E~KAKC1723-E	-	○	-	○※1	○※1	-
KAKC1724-E, KAKC1725-E	-	○	-	○	○	-
KAKC1726-E	-	○	○	○	○	-
KAKC1727-E	-	○	○	-	○	-
KAKC1728-E	-	○	-	○	○	-
KAKC1729-E	-	○	○	-	○	-
KAKC1731-E~KAKC1734-E	-	○	-	○	○	-
KAKC1800-I, KAKC1801-E	-	-	-	○	-	-
KAKC1900-E, KAKC1901-E	-	○※2	-	-	○	-
KAKC1902-W~KAKC1905-E	-	○	○	-	○	-
KAKC1906-I~KAKC1911-E	-	-	○	-	○	-
KAKC1912-W	-	-	-	-	○	-
KAKC1915-I~KAKC1922-E	-	-	○	-	○	-
KAKC1923-W	-	○	○	-	○	-
KAKC1932-I	-	-	○	-	-	-

メッセージ ID の範囲	stdout	stderr	JOBLOG	SYSMSG	実行ログ	JP1
KAKC1942-W, KAKC1943-W	—	○	○	—	○	—
KAKC1950-E, KAKC1951-E	—	○	—	—	○	—
KAKC2201-E~KAKC2205-E	—	○	—	—	○	—
KAKC2206-E	—	○	—	—	—	—
KAKC2207-E	—	○	—	—	○	—
KAKC2208-E, KAKC2209-E	—	○*7	_*7	—	—	—
KAKC2211-E	—	○*7	_*7	—	—	—
KAKC2213-E	—	○*7	_*7	—	—	—
KAKC2214-E	—	○	—	—	○	—
KAKC2215-E	—	○*7	_*7	○*7	—	—
KAKC2300-E	—	○	—	—	—	—
KAKC2301-E~KAKC2302-E	—	○	—	—	○	—
KAKC2303-E	—	○	—	—	—	—
KAKC2310-E~KAKC2499-E	—	○	—	—	○	—
KAKC3000-I	○	○	—	—	○	○
KAKC3001-I	—	○	—	—	○	○
KAKC3002-E	○	○	—	—	○	○
KAKC3003-E	—	○	—	—	○	○
KAKC3004-E	—	—	—	—	○	○
KAKC3006-I	—	○	—	—	○	—
KAKC3007-W~KAKC3009-E	○	○	—	—	○	—
KAKC3010-I~KAKC3019-E	—	—	—	—	○	—
KAKC3020-E~KAKC3027-E	—	○	—	—	○	—
KAKC3028-E	—	—	—	—	○	—
KAKC3029-E, KAKC3099-E	—	○	—	—	○	—
KAKC3031-E	—	○	—	—	○	○
KAKC3120-E~KAKC3129-E	—	—	—	—	○	—
KAKC3130-I	—	—	—	—	○	○
KAKC3199-E	—	—	—	—	○	—
KAKC3204-I	—	—	—	—	○	○

メッセージ ID の範囲	stdout	stderr	JOBLOG	SYSMMSG	実行ログ	JP1
KAKC3220-E~KAKC3261-I	-	-	-	-	○	-
KAKC3400-I, KAKC3402-E	○	○	-	-	-	-
KAKC3412-E	-	-	-	-	-	○
KAKC3415-E, KAKC3416-E	-	○	-	-	-	-
KAKC3508-I	-	-	-	-	○	○
KAKC3522-E, KAKC3542-W	-	○	-	-	○	-
KAKC3543-E, KAKC3544-E	-	○	-	-	○	-
KAKC3602-E	○	○	-	-	○	-
KAKC3700-I~KAKC3799-I	○	-	-	-	-	-
KAKC3820-E~KAKC3999-E	-	○	-	-	-	-
KAKC4000-E~KAKC4005-E	-	-	-	○	○	-
KAKC4010-I, KAKC4011-I	-	○	-	-	-	-
KAKC4020-E, KAKC4021-E	-	○	-	○	○	-
KAKC4022-E, KAKC4023-E	-	○	-	-	-	-
KAKC4024-E~KAKC4026-E	-	-	-	○	○	-
KAKC4050-E~KAKC4075-E	-	○	-	○	○	-
KAKC4076-E~KAKC4599-E	-	○	-	-	-	-
KAKC4600-I	○	-	-	-	-	-
KAKC4624-E	-	○	-	-	-	-
KAKC5000-E	-	-	○	-	○	-
KAKC5001-I	-	-	○	-	○	○*6
KAKC5002-I, KAKC5003-I	-	-	-	-	○	-
KAKC5005-E	-	-	○	-	○	-
KAKC5006-E	-	-	○	○	-	-
KAKC5008-I	-	-	-	-	○	○
KAKC5009-E, KAKC5010-E	-	-	-	-	○	-
KAKC5011-E	-	-	○	-	○	-
KAKC5012-I, KAKC5013-I	-	-	○	○	○	-
KAKC5014-I	-	-	-	-	○	-
KAKC5015-E	-	-	○	-	○	-
KAKC5016-I	-	-	-	-	○	-

メッセージIDの範囲	stdout	stderr	JOBLOG	SYMSMSG	実行ログ	JP1
KAKC5017-E	-	-	○	-	○	-
KAKC5018-I	-	-	-	-	○	-
KAKC5019-E, KAKC5020-E	-	-	○	-	○	-
KAKC5021-E	-	-	○	-	-	-
KAKC5100-I~KAKC5102-E	○	-	-	-	-	-
KAKC5205-E	-	○	-	-	-	-
KAKC5208-E	-	-	-	-	-	○
KAKC5309-E, KAKC5323-E	○	-	-	-	○	○※3
KAKC5362-E, KAKC5370-I	○	-	-	-	-	○※4
KAKC5371-E	○	-	-	-	○	○※4
KAKC5372-E	○	-	-	-	○	○※3
KAKC5380-I, KAKC5381-I	-	-	-	-	○	-
KAKC5390-I~KAKC5399-E	○	-	-	-	-	○※4
KAKC5407-E	-	○	-	-	-	-
KAKC5408-I	-	-	-	-	-	○
KAKC5409-E~KAKC5499-E	-	○	-	-	-	-

(凡例)

stdout : 標準出力を示す。

stderr : 標準エラー出力を示す。

JOBLOG : ジョブログを示す。

SYMSMSG : システムメッセージを示す。

実行ログ : ログを示す。

JP1 : JP1 イベントを示す。

○ : 出力する。

- : 出力しない。

注※1 コマンド実行時は出力しない。

注※2 ジョブステップごとの標準エラーに出力する。

注※3 アダプタコマンドが出力するメッセージ。JP1/IM に応答される。

注※4 アダプタコマンドが出力するメッセージ。JP1/IM の画面に表示される。

注※5 ISAMLIB_MODE 設定パラメータの指定によって出力先が異なる。

KAKC0005-E メッセージ

- ・ M64 を指定した場合は、stderr にだけ出力される。
- ・ M32 を指定した場合は、出力指示の通りの出力となる。

KAKC1626-E と KAKC1627-I メッセージ

- ・ M64 を指定した場合は、stderr にだけ出力される。
- ・ M32 を指定した場合は、SYSMSG にだけ出力される。

注※6 環境変数 BJEX_DISPLAY_REPLY の指定によって出力するかどうか異なる。

詳細については、「[3.15.7 DISPLAY 文による JP1 イベント出力の抑止機能](#)」を参照のこと。

注※7 出力するプロセスによって出力先が異なる。

- ・ ジョブコントローラプロセスが出力した場合は、出力指示どおりの出力となる。
- ・ ジョブステッププロセスが出力した場合は、JOBLOG にだけ出力される。

10.2.2 KAKD で始まるメッセージの出力先一覧

KAKD で始まるメッセージの出力先一覧を次の表に示す。

表 10-5 KAKD で始まるメッセージの出力先一覧

メッセージ ID の範囲	stdout	stderr	SYSPRINT	実行ログ	JP1
KAKD1001-E~KAKD1006-E	—	○	○	—	—
KAKD1009-I	—	—	○	—	—
KAKD1010-E~KAKD1091-E	—	○	○	—	—
KAKD1100-I~KAKD1105-I	—	—	○	—	—
KAKD1110-E~KAKD1130-E	—	○	○	—	—
KAKD1131-I	—	—	○	—	—
KAKD1137-E	—	○	○	—	—
KAKD1147-I~KAKD1153-I	—	—	○	—	—
KAKD2001-E~KAKD2005-E	—	○	○	—	—
KAKD2009-I	—	—	○	—	—
KAKD2010-E~KAKD2091-E	—	○	○	—	—
KAKD2100-I	—	—	○	—	—
KAKD2101-E	—	○	○	—	—
KAKD2102-I	—	—	○	—	—

メッセージ ID の範囲	stdout	stderr	SYSPRINT	実行ログ	JP1
KAKD2103-E~KAKD2121-E	-	○	○	-	-
KAKD2130-I	-	-	○	-	-
KAKD2131-E~KAKD2137-E	-	○	○	-	-
KAKD2138-I~KAKD2140-I	-	-	○	-	-
KAKD2150-E	-	○	○	-	-
KAKD3001-E~KAKD3007-E	-	○	○	-	-
KAKD3009-I	-	-	○	-	-
KAKD3010-E~KAKD3093-E	-	○	○	-	-
KAKD3100-I~KAKD3102-I	-	-	○	-	-
KAKD3105-E~KAKD3136-E	-	○	○	-	-
KAKD3137-W	-	○	○	-	-
KAKD3138-I	-	-	○	-	-
KAKD3139-E~KAKD3145-E	-	○	○	-	-

(凡例)

stdout : 標準出力を示す。

stderr : 標準エラー出力を示す。

SYSPRINT : DD 名が"SYSPRINT"の DD 要素で指定した出力先を示す。

実行ログ : ログを示す。

JP1 : JP1 イベントを示す。

○ : 出力する。

- : 出力しない。

10.3 syslog に出力するメッセージの一覧

バッチジョブ実行システムが syslog に出力するメッセージの一覧を次の表に示す。

表 10-6 バッチジョブ実行システムが syslog に出力するメッセージの一覧

メッセージID	ファシリティ	レベル
KAKC3000-I	LOG_USER	LOG_NOTICE
KAKC3001-I		
KAKC3002-E		
KAKC3003-E		
KAKC3004-E		
KAKC3031-E		
KAKC3412-E		

10.4 システムコールのエラー情報

バッチジョブ実行システム的环境で発生しやすいシステムコールのエラー情報に対する原因と対策を次の表に示す。表にないエラーについては、使用している OS のマニュアルを参照すること。

なお、次の表では AIX でのエラー番号を記載している。

バッチジョブ実行システム的环境で発生しやすいシステムコールのエラー情報の内容だけを記載しているため、詳細についてはメッセージで表示されたエラー番号に該当するニモニックを使用している OS の `errno` 定義ファイル (`errno.h`) で調べること。

さらに、エラーとなったシステムコールについては、OS のマニュアルを参照し、該当するニモニックのエラーが返った原因を特定する。

表 10-7 代表的なシステムコールのエラー情報に対する原因と対策

エラー番号	エラー詳細	ニモニック	考えられる主な原因	対策
2	No such file or directory	ENOENT	ファイル、またはディレクトリが見つからない。	ファイルの存在を確認すること。
5	I/O error	EIO	入出力エラーが発生した。	OS またはハードウェアの情報に従うこと。
6	No such device or address	ENXIO	ファイルに対するアクセス権がない。	デバイスが存在するか、またはデバイスが有効化しているか確認すること。デバイスが有効化されていない場合は、有効化すること。それ以外の原因の場合は、使用している OS のマニュアルを参照すること。
7	Arg list too long	E2BIG	処理プログラムの引数または環境変数用の領域が不足している。	処理プログラムの引数を確認する。 DD 要素, SETENV 要素, SETENVFILE 要素を見直し、不要な要素を削除する。
11	Resource temporarily unavailable	EAGAIN	次のどちらかの原因が考えられる。 <ul style="list-style-type: none"> プロセスの数が多過ぎる、または一時的なメモリ不足が発生している。 メッセージキューが満杯である。 	<ul style="list-style-type: none"> 再度実行しても繰り返し発生する場合は、不要なプロセスを停止させること。 スプールジョブ転送デーモンを起動する。
12	Not enough space	ENOMEM	次のどちらかの原因が考えられる。 <ul style="list-style-type: none"> スワップ領域、または仮想メモリ不足のため、プロセスを新しく生成できない。 	<ul style="list-style-type: none"> スワップ領域または仮想メモリが足りない場合は、拡張すること。拡張できない場合は、不要なプロセスを停止させること。

エラー番号	エラー詳細	エラメッセージ	考えられる主な原因	対策
12	Not enough space	ENOMEM	<ul style="list-style-type: none"> プロセスの数が多過ぎる、または一部のプロセスが大量のメモリを消費している。 	<ul style="list-style-type: none"> 一部のプロセスが大量のメモリを消費している場合は、該当するプロセスをいったん停止できないか検討すること。
13	Permission denied	EACCES	アクセス権限が不正である。	ファイルに対するアクセス権限が正しいか確認すること。
14	Bad address	EFAULT	アクセスできない領域に書き込みしようとした。書き込みしようとしたディスクが切り離されたおそれがある。	系切り替えに伴うディスクの切り替え中の場合は、問題ないので無視すること。誤ってディスクを切り離してしまった場合は、該当するファイルをバックアップから回復する、または初期化してから使用すること。それ以外の場合は、システム管理者に連絡すること。
17	File exists	EEXIST	作成しようとしたファイルはすでに存在する。	ファイル名を変更して再度実行する、または既存のファイルが不要であれば、削除してから再度実行すること。
22	Invalid argument	EINVAL	メモリ管理情報の不正を検知した。	システム管理者に連絡すること。
23	File table overflow	ENFILE	ファイルのオープン数がシステムの上限を超えた。	OSのオペレーティングシステムパラメータ（カーネルパラメータ）の、システムでオープンできるファイル最大数（maxuproc*nofiles*）の指定値を大きくすること。
24	Too many open files	EMFILE	該当するプロセスでオープンしているファイル数が多い。	OSのオペレーティングシステムパラメータ（カーネルパラメータ）の、プロセスでオープンできるファイル数の最大nofiles)の値を大きくすること。
27	File too large	EFBIG	ファイルの大きさがシステム制限値を超えた。	使用するファイルサイズを見直すこと。
28	No space left on device	ENOSPC	ファイルシステムに十分な空き領域がない。	空き領域を確保すること。
86	File name too long	ENAMETOOLONG	ファイルの名前が長過ぎる。	ファイルの名前を見直すこと。

10.5 メッセージ一覧

バッチジョブ実行システムが出力するメッセージと対処方法について説明する。

KAKC0001-E

Memory shortage. DETAIL=保守情報

メモリ不足が発生した。保守情報は、8けたの16進数で表示する。保守情報は、システムの内部状態を示す情報である。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生のタイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を終了する。

(O)システム管理者に連絡する。システム管理者は、メモリ見積もりを見直す。

KAKC0002-W

Environment variable "環境変数名" is not specified.

環境変数名で示す環境変数が取得できない。

環境変数名が AJS_BJEX_STOP である場合、JP1/AJS からバッチジョブ実行システムを起動するためのジョブの定義、あるいはサブミットジョブの場合、jppqjobsub コマンドの指定が適切でない。JP1/AJS から起動していない場合や JP1/AJS から Advanced Shell を経由して起動した場合、本メッセージが出力されても問題ないため、(O)に示す対処をとる必要は無い。

(S)処理を続行する。

環境変数名が AJS_BJEX_STOP である場合、JP1/AJS からバッチジョブ実行システムのジョブを強制終了すると、バッチジョブ実行システムの終了処理が行われない。

(O)バッチジョブ実行システムを起動する JP1/AJS のジョブ定義で、環境変数 AJS_BJEX_STOP=TERM を指定する。環境変数 AJS_BJEX_STOP=TERM は、JP1/AJS のジョブ定義で指定する必要がある。

サブミットジョブの場合、jppqjobsub コマンドで -env AJS_BJEX_STOP=TERM を指定する。

問題が解決しない場合、システム管理者に連絡する。

KAKC0003-W

Value of environment variable "環境変数名" must be "環境変数値".

環境変数名で示す環境変数の値が環境変数値でない。

環境変数名が AJS_BJEX_STOP である場合、値が TERM でない。JP1/AJS でのジョブの定義、またはサブミットジョブの場合、jppqjobsub コマンドの指定が適切でない。

(S)処理を続行する。

環境変数名が AJS_BJEX_STOP で、環境変数値が TERM である場合、JP1/AJS からバッチジョブ実行システムのジョブを強制終了した場合、バッチジョブ実行システムの終了処理が行われない。

(O)バッチジョブを起動する JP1/AJS のジョブ定義で、環境変数 AJS_BJEX_STOP=TERM を指定する。環境変数 AJS_BJEX_STOP=TERM は、JP1/AJS のジョブ定義で指定する必要がある。

サブミットジョブの場合、jpqjobsub コマンドで-env AJS_BJEX_STOP=TERM を指定する。

問題が解決しない場合、システム管理者に連絡する。

KAKC0004-I

Jobid=バッチジョブ実行システムのジョブ識別子, JP1NBQSQueueName=環境変数値,
JP1JobID=JP1 ジョブ番号

起動したジョブの JP1/AJS のジョブの情報とバッチジョブ実行システムが与えたジョブ識別子を表示する。

バッチジョブ実行システムのジョブ識別子：このジョブのバッチジョブ実行システムが与えるジョブ識別子

環境変数値：このジョブの JP1NBQSQueueName 環境変数の値

JP1 ジョブ番号：このジョブの JP1/AJS が与えた JP1 ジョブ番号

(S)処理を続行する。

(O)なし。

KAKC0005-E

Failed to load ISAM library. reason=エラー詳細

ジョブ定義 XML ファイルに一時 ISAM ファイルの割り当て要求の DD 要素が指定されたが、ISAM ライブラリがロードできない。

(S)処理を終了する。

(O)エラー詳細を調べ、ISAM 製品が正しくインストールされているか確認する。

KAKC0006-E

Failed to load CodeConv library. reason=エラー詳細

設定ファイルにコード変換を必要とする機能が指定されたが、Hitachi Code Converter ライブラリがロードできない。

(S)処理を続行する。ただし、ジョブログを stderr に出力時、文字コードの変換を実施しない。

(O)エラー詳細を調べ、Hitachi Code Converter 製品が正しくインストールされているか確認する。

KAKC0010-E

Message queue open failed. reason=**エラー詳細**

スプールジョブ転送デーモンへの通知で使用するメッセージキューに対する mq_open 関数で**エラー詳細**を示すエラーが発生した。

このメッセージは、ジョブ開始時に出力する場合は stderr に出力し、ジョブ終了時に出力する場合は stderr と実行ログに出力する。

(S)処理を続行する。

(O)スプールジョブ転送デーモンが起動していることを確認する。問題が解決しない場合はシステム管理者に連絡する。

このメッセージを出力したジョブのスプールジョブは、スプールジョブ転送デーモンによって処理されない。bjexlsjob コマンドの-f オプションで状態を表示した場合は、管理から漏れている (–) または実行中 (X) となる。このスプールジョブを印刷する場合は、bjexchjob コマンドの-S オプションで回復処理をしたあと、印刷を指示する必要がある。

KAKC0011-E

Failed to send a message to message queue. reason=**エラー詳細**

スプールジョブ転送デーモンへの通知で使用するメッセージキューへのデータ追加処理で利用する mq_send 関数で**エラー詳細**を示すエラーが発生した。

このメッセージは、ジョブ開始時に出力する場合は stderr に出力し、ジョブ終了時に出力する場合は stderr と実行ログに出力する。

(S)処理を続行する。

(O)スプールジョブ転送デーモンが起動していることを確認する。問題が解決しない場合はシステム管理者に連絡する。

このメッセージを出力したジョブのスプールジョブは、スプールジョブ転送デーモンによって処理されない。bjexlsjob コマンドの-f オプションで状態を表示した場合は、管理から漏れている (–) または実行中 (X) となる。このスプールジョブを印刷する場合は、bjexchjob コマンドの-S オプションで回復処理をしたあと、印刷を指示する必要がある。

KAKC0012-E

Message queue close failed. reason=**エラー詳細**

スプールジョブ転送デーモンへの通知で使用するメッセージキューへのデータ追加処理で利用する mq_close 関数で**エラー詳細**を示すエラーが発生した。

このメッセージは、ジョブ開始時に出力する場合は stderr に出力し、ジョブ終了時に出力する場合は stderr と実行ログに出力する。

(S)処理を続行する。

(O)スプールジョブ転送デーモンが起動していることを確認する。問題が解決しない場合はシステム管理者に連絡する。

KAKC0013-E

スプールジョブ転送サービスへの通知に失敗しました

スプールジョブ転送デーモンへの通知で使用するメッセージキューに対する処理でエラーが発生した。

(S)処理を続行する。

(O)KAKC0010-E または KAKC0011-E のメッセージに示す問題を解決する。このメッセージを出力したジョブのスプールジョブは、スプールジョブ転送デーモンによって処理されない。bjexlsjob コマンドの -f オプションで状態を表示した場合は、管理から漏れている (-) または実行中 (X) となる。このスプールジョブを印刷する場合は、bjexchjob コマンドの -S オプションで回復処理をしたあと、印刷を指示する必要がある。

KAKC0050-E

Failed to get username. reason=エラー詳細

ユーザ名の取得に失敗した。

(S)処理を終了する。

(O)エラー詳細を基に問題がないか確認する。問題が解決しない場合はシステム管理者に連絡する。

KAKC0060-E

BJEX_DIR ID "環境変数名" putenv failed.

設定ファイルの BJEX_DIR の処理で、環境変数名に示す環境変数の設定に失敗した。

環境変数名に示す環境変数がすでに設定されている場合も、このメッセージを出力する。

(S)処理を終了する。

(O)環境変数名で示す環境変数をすでに設定していないか確認する。環境変数名で示す環境変数をすでに設定している場合、環境変数名で示す環境変数の設定を削除する、または設定ファイルの BJEX_DIR_xxx の名称に変更する。設定パラメータの BJEX_DIR_xxx の名称を変更する場合は、ジョブ定義 XML ファイル内の名称も変更する必要がある。

問題が解決しない場合はシステム管理者に連絡する。

KAKC0080-I

Grid job started. UGPSM_DATAID=環境変数値

環境変数 UGPSM_DATAID の環境変数値で示すグリッドジョブを開始した。

(S)処理を続行する。

(O)なし。

KAKC0090-I

HitachiBatchJobExec started.

ジョブコントローラが処理を開始した。

(S)処理を続行する。

(O)なし。

KAKC0091-I

ジョブ名 Job started.

ジョブ (JP1/AJS のジョブ情報 : ジョブ名) を開始します

ジョブを開始した。

(S)処理を続行する。

(O)なし。

KAKC0092-I

ジョブ名.ジョブステップ名[.プロシジャステップ名] Step started.

ジョブステップ (JP1/AJS のジョブ情報 : ジョブ名.ジョブステップ名 [.プロシジャステップ名]) を開始します

ジョブステップを開始した。

(S)処理を続行する。

(O)なし。

KAKC0093-E

bjexec received terminate request.

ジョブコントローラに対する終了要求を受け取った。

(S)ジョブコントローラは、ジョブステップのプロセスグループに SIGKILL を送り、ファイルの後始末などの必要な後処理を実行してから終了する。

(O)ユーザプログラムが作成した資源を必要に応じて後始末する。

KAKC0094-E

```
bjexec is ended because of terminate request of second times.
```

ジョブコントローラで 2 回目の終了要求のシグナルを受け取った。

(S)ジョブコントローラは、即時に終了する。一時ファイルの削除やファイルの後始末などの後処理を行わない。スプールにあるジョブの出力も自動的に印刷しない。

(O)必要があれば、bjexchjob コマンドでスプールにあるジョブの状態を変更して出力する。

KAKC0095-E

```
bjexec ended abnormally.
```

ジョブコントローラが異常終了した。

(S)一時ファイルの削除やファイルの後始末などの後処理を行わない。スプールにあるジョブの出力も自動的に印刷しない。

(O)必要があれば、bjexchjob コマンドでスプールにあるジョブの状態を変更して出力する。

KAKC0096-E

```
ジョブ名.ジョブステップ名[.プロシジャステップ名] Step ended abnormally. rc=終了コード status=状態コード E-Time=実行時間 s C-Time=CPU 時間 s
```

```
ジョブステップ (JP1/AJS のジョブ情報: ジョブ名.ジョブステップ名 [.プロシジャステップ名]) が異常終了しました
```

ジョブステップが異常終了した。

終了コード: ジョブステップが異常終了したときの終了コードである。詳細については、「[10.8 終了コード](#)」を参照のこと。

状態コード: ジョブステップが終了したときのシグナルコードである。PGM_RCABEND パラメータによって異常終了した場合は 0 が出力される。

実行時間: ジョブステッププロセスが実際に実行していた実時間である (秒単位)。

CPU 時間: ジョブステッププロセスが使用した CPU 時間である (秒単位)。

(S)処理を続行する。

(O)異常終了したジョブステップで実行するプログラムに問題がないか確認する。

KAKC0097-I

```
ジョブ名.ジョブステップ名[.プロシジャステップ名] Step ended. rc=終了コード E-Time=実行時間  
s C-Time=CPU 時間 s
```

ジョブステップ (JP1/AJS のジョブ情報 : ジョブ名.ジョブステップ名 [.プロシジャステップ名]) が
終了しました

ジョブステップが終了した。

終了コード : 各ジョブステップの終了コードである。終了コードは、ジョブステップで実行したプログラムの、main 関数の return 文で指定された式、または exit 関数で指定された状態コードである。

実行時間 : ジョブステッププロセスが実際に実行していた実時間である (秒単位)。

CPU 時間 : ジョブステッププロセスが使用した CPU 時間である (秒単位)。

(S)処理を続行する。

(O)なし。

KAKC0098-I

```
ジョブ名 Job ended. rc=終了コード E-Time=実行時間 s C-Time=CPU 時間 s [MaxRSS=最大  
RSSKB]
```

ジョブ (JP1/AJS のジョブ情報 : ジョブ名) が終了しました

ジョブが終了した。

終了コード : 設定ファイルの JOB_RCDEFINE パラメータで STEPMAX を指定した場合は、各ジョブステップが終了した際の最大終了コードである。設定ファイルの JOB_RCDEFINE パラメータで STEPLAST を指定した場合は、最後に実行完了したメインステップの終了コードである。詳細については、「[10.8 終了コード](#)」を参照のこと。

実行時間 : 全ジョブステッププロセスが実際に実行していた実時間の合計である (秒単位)。gettimeofday 関数で求めたジョブステップの開始時刻と終了時刻の差で求めた参考値である。

CPU 時間 : 全ジョブステッププロセスの利用した合計 CPU 時間である (秒単位)。times 関数のリターン情報である構造体 tms 内の tms_cutime と tms_cstime の和で求めた参考値である。

【AIX 限定】

最大 RSS : 各ジョブステップの最大実メモリとジョブコントローラの最大実メモリを合計した値である (KB 単位)。getrusage 関数のリターン情報である構造体 rusage 内の ru_maxrss の値で求めた参考値である。

(S)処理を続行する。

(O)なし。

KAKC0099-I

HitachiBatchJobExec ended. rc=終了コード

ジョブコントローラが処理を終了する。

(S)処理を続行する。

(O)なし。

KAKC0100-E

bjexec の起動処理でエラーが発生しました

ジョブコントローラの起動処理中に、JP1 イベントの拡張属性のエラー詳細に示すエラーが発生した。

(S)処理を終了する。

(O)エラー詳細および一緒に出力されるほかのメッセージを参照してエラーの原因を取り除き、ジョブを再実行する。

KAKC0101-E

ジョブ名 An error occurred during execution of job.

ジョブ (JP1/AJS のジョブ情報: ジョブ名) の実行中にエラーが発生しました

ジョブコントローラがジョブ実行中にエラーを検知した。

(S)処理を続行する。

(O)一緒に出力されるほかのメッセージを参照してエラーの原因を取り除き、ジョブを再実行する。

KAKC0103-E

コマンド名 received terminate request.

コマンド名のコマンドで終了要求のシグナルを受け取った。

(S)処理を終了する。

(O)なし。

KAKC0105-E

コマンド名 ended abnormally.

コマンド名のコマンドが異常終了した。

(S)処理を終了する。

(O)なし。

KAKC0110-E

Restart step name is not found. stepname=ジョブステップ名[.プロシジャステップ名]

リスタートステップ名が見つからない。または、リスタートステップとして指定できないジョブステップを指定した。

(S)処理を中断する。

(O)リスタートステップ名が正しいか確認する。

KAKC0112-I

Job is restarted from specified step. stepname=ジョブステップ名[.プロシジャステップ名]

ジョブはジョブステップ名[.プロシジャステップ名]からリスタートする。

(S)処理を続行する。

(O)なし。

KAKC0199-E

Failed to execute a command. (jobid=ジョブ識別子, error=理由)

コマンド実行処理でエラーが発生した。

ジョブ識別子：コマンド実行が必要になったジョブのジョブ識別子

理由：エラー内容を表す文字列

理由の表示内容と説明を次に示す。

理由	説明
コマンド名:create sub process:エラー理由	サブプロセスの生成に失敗した。エラー理由は errno から求めた文字列である。
コマンド名:wait process:エラー理由	サブプロセスの終了待ちに失敗した。エラー理由は errno から求めた文字列である。
コマンド名:rc=リターンコード	コマンドがリターンコードで終了した。
コマンド名:sub process:abnormal terminated	サブプロセスがシグナルなどで終了した。

理由	説明
コマンド名:sub process(setgid):エラー理由	サブプロセスがグループ id の設定に失敗した。エラー理由は ermo から求めた文字列である。
コマンド名:sub process(setuid):エラー理由	サブプロセスがユーザ id の設定に失敗した。エラー理由は ermo から求めた文字列である。
コマンド名:sub process(setenv(LANG=環境変数値)):エラー理由	サブプロセスで LANG 環境変数の設定に失敗した。エラー理由は ermo から求めた文字列である。
コマンド名:sub process(execv):エラー理由	サブプロセスがコマンドの起動に失敗した。エラー理由は ermo から求めた文字列である。

(S)コマンド実行処理を中断する。

(O)システム管理者に連絡する。

KAKC0200-E

Failed to store spool job. (jobid=ジョブ識別子, error=理由)

スプールジョブの保管処理でエラーが発生した。

ジョブ識別子：スプールジョブのジョブ識別子

理由：エラー内容を表す文字列

理由の表示内容と説明を次に示す。

理由	説明
spool job control file:エラー理由	スプールジョブ管理ファイルでエラーが発生した。エラー理由は ermo から求めた文字列である。
failed to create sub process	サブプロセスの生成に失敗した。
failed to wait sub process	サブプロセスの待ち処理に失敗した。
command terminated by error	コマンドがエラーで終了した。
open spool job dir:エラー理由	スプールジョブディレクトリのオープンでエラーが発生した。エラー理由は ermo から求めた文字列である。

(S)スプールジョブの保管処理を中断する。

(O)システム管理者に連絡する。

KAKC0201-E

Failed to change spool job permission. (jobid=ジョブ識別子, error=理由)

保管したスプールジョブの権限設定に失敗した。

ジョブ識別子：失敗したスプールジョブのジョブ識別子

理由：エラー内容を表す文字列

理由の表示内容と説明を次に示す。

理由	説明
failed to create sub process	サブプロセスの生成に失敗した。
failed to wait sub process	サブプロセスの待ち処理に失敗した。
command terminated by error	コマンドがエラーで終了した。
open spool job store dir:エラー理由	保管先のスプールジョブディレクトリのオープンでエラーが発生した。エラー理由は errno から求めた文字列である。

(S)スプールジョブの保管処理を中断する。

(O)システム管理者に連絡する。

KAKC0202-E

Failed to create spool job store directory. (jobid=ジョブ識別子, error=理由)

保管先のスプールジョブディレクトリ作成処理でエラーが発生した。

ジョブ識別子：スプールジョブのジョブ識別子

理由：エラー内容を表す文字列

理由の表示内容と説明を次に示す。

理由	説明
failed to create sub process	サブプロセスの生成に失敗した。
failed to wait sub process	サブプロセスの待ち処理に失敗した。
command terminated by error	コマンドがエラーで終了した。
上記以外	errno から求めた文字列である。

(S)スプールジョブの保管処理を中断する。

(O)保管先に同一ディレクトリ名のスプールジョブが存在する場合は、そのスプールジョブを削除してジョブの再実行、または bjexchjob -C コマンドを入力する。上記以外の場合はシステム管理者に連絡する。

KAKC0210-E

I/O error occurred, EUR definition file. (jobid=ジョブ識別子,
step number=ジョブステップ番号, DDNAME=dd 名, EURFORM=EUR 定義ファイル名,
error=理由)

EUR 定義ファイルの読み込み処理でエラーが発生した。

理由：エラー内容を表す文字列

理由の表示内容と説明を次に示す。

理由	説明
not a file	通常のファイルではない。
file is swapped	オープン前後でファイルが同一ではない。
上記以外	errno から求めた文字列である。

(S)該当する SYSOUT データファイルの処理を中断する。

(O)EUR 定義ファイルが正しいかチェックする。

KAKC0211-E

Specified EURENCODE is not defined in configuration file. (jobid=ジョブ識別子, step number=ジョブステップ番号, DDNAME=dd 名, EURFORM=EUR 定義ファイル名, EURENCODE=EUR エンコード種別)

EUR 連携処理で DD 要素に定義した EUR エンコード種別が設定ファイルに定義されていない。

(S)該当する SYSOUT データファイルの処理を中断する。

(O)システム管理者に連絡する。

システム管理者は、ジョブ実行時の設定ファイルがジョブ実行時から変更されていないか確認する。確認した結果、設定ファイルを修正、またはジョブ定義 XML ファイルの指定を変更してジョブを再度実行する。

KAKC0212-E

Failed to open spool job control file. (jobid=ジョブ識別子)

EUR 連携処理でスプールジョブ管理ファイルのオープン処理でエラーが発生した。

ジョブの実行が正常に処理されないで、スプールジョブ管理ファイルが正しく作成されていないおそれがある。

(S)該当するスプールジョブの処理を中断する。

(O)ジョブが正常に終了しているか確認する。

KAKC0213-E

EUR command start error. (jobid=ジョブ識別子, step number=ジョブステップ番号,

DDNAME=dd 名, EURFORM=EUR 定義ファイル名, error=理由)

EUR の eurps コマンドの起動処理に失敗した。

理由：エラー内容を表す文字列

理由の表示内容と説明を次に示す。

理由	説明
over limits of command arguments	コマンドの引数の上限 (126) を超えた。
over limits of EUR definition file size	EUR 定義ファイルサイズの上限を超えた。

(S)該当する SYSOUT データファイルの処理を中断する。

(O)EUR 定義ファイルを見直す。

KAKC0214-E

EUR command error. (jobid=ジョブ識別子, step number=ジョブステップ番号,
DDNAME=dd 名, EURFORM=EUR 定義ファイル名, error=理由)

EUR の eurps コマンドがエラーで終了した。

理由：エラー内容を表す文字列

理由の表示内容と説明を次に示す。

理由	説明
failed to create sub process	EUR のサブプロセスの生成に失敗した。
failed to wait sub process	EUR のサブプロセスの待ち処理に失敗した。
command terminated by error	EUR の eurps コマンドがエラーで終了した。

(S)該当する SYSOUT データファイルの処理を中断する。

(O)EUR 定義ファイルを見直す。また、EUR の eurps コマンドのメッセージまたはログを調査する。原因が不明の場合はシステム管理者に連絡する。

KAKC0215-I

EUR_DEFINITION_FILE_DIR is not defined. EUR print process is skipped.

設定ファイルに EUR_DEFINITION_FILE_DIR の指定がない。

EUR の印刷処理をスキップした。

(S)処理を終了する。

(O)設定ファイルの指定を見直す。

KAKC0220-E

Environment variable "環境変数名" is not specified.

環境変数名で示す環境変数が指定されていない。

環境変数名が JP1_HOSTNAME である場合、JP1/AJS から起動されていないおそれがある。

(S)処理を終了する。

(O)JP1/AJS から起動されているか確認する。

KAKC0236-E

String size of environment variable "環境変数名" value is invalid.

環境変数名で示す環境変数の値が長過ぎる。

(S)処理を終了する。

(O)システム管理者に連絡して環境変数の値を見直す。

KAKC0238-E

Invalid char is included in environment variable "環境変数名" value.

環境変数名で示す環境変数の値に使用できない文字が含まれている。

(S)処理を終了する。

(O)システム管理者に連絡して環境変数の値を見直す。

KAKC0299-E

Internal error occurred. Detail=**保守情報**

メモリ確保で内部矛盾が発生した。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC0300-I

Usage: コマンド名 コマンド引数

コマンド名で示すコマンドの指定が誤っている。

(S)処理を終了する。

(O)コマンド名を正しく指定して実行する。

KAKC0301-E

Option "オプション名" value is not specified.

オプション名の指定値が誤っている。

(S)処理を中断する。

(O)オプション名を正しく指定する。

KAKC0302-E

Option "オプション名" was unknown.

不正なオプション名を指定している。

(S)処理を中断する。

(O)オプション名を正しく指定する。

KAKC0303-E

XML file name is not specified.

ジョブ定義 XML ファイルの指定がない。

(S)処理を中断する。

(O)ジョブ定義 XML ファイルを指定して再入力する。

KAKC0304-E

Option オプション名 is specified with other exclusive option.

オプション名に不当な指定値がある。

(S)処理を中断する。

(O)オプション名を正しく指定して再入力する。

KAKC0305-E

Option オプション名 is invalid.

不正なオプション名を指定している。

(S)処理を中断する。

(O)オプション名を正しく指定して再入力する。

KAKC0306-E

Option "オプション名" value is invalid.

オプション名の指定値が誤っている。

(S)処理を中断する。

(O)オプション名を正しく指定して再入力する。

KAKC0307-E

The option is not specified.

必要なオプションが指定されていない。

(S)処理を終了する。

(O)必要なオプションを指定して再入力する。

KAKC0309-I

プログラム名 version is バージョン文字列.

プログラム名で示すコマンドのバージョンをバージョン文字列に示す。

(S)処理を終了する。

(O)なし。

KAKC0310-E

Too many operands.

指定したオプションが多過ぎる。

(S)処理を中断する。

(O)コマンドとオプションを正しく指定する。

KAKC0311-E

One or more necessary options for コマンド名 are missing.

コマンド処理に必要なオプション，パラメータが不足している。

(S)処理を終了する。

(O)必要なオプション，パラメータを指定して再入力する。

KAKC0336-E

String size of option "オプション名" value is invalid.

オプション名の指定値が長過ぎる。

(S)処理を終了する。

(O)オプション名を正しく指定して再入力する。

KAKC0338-E

Invalid char is included in option "オプション名" value.

オプション名の指定値に使用できない文字が含まれている。

(S)処理を終了する。

(O)オプション名を正しく指定して再入力する。

KAKC0400-I

Parsing of config file "ファイル名" is started.

ファイル名で示す設定ファイルの解析を開始した。

(S)処理を続行する。

(O)なし。

KAKC0401-E

Config file open failed. reason="エラー詳細"

エラー詳細で示す原因によって，設定ファイルのオープンに失敗した。

(S)処理を終了する。

(O)エラー詳細を基に設定ファイルが読み込みできるように、権限などの問題がないか確認する。問題が解決しない場合は、システム管理者に連絡する。

KAKC0402-E

```
Config file read error. reason="エラー詳細"
```

エラー詳細で示す原因によって、設定ファイルの読み込みに失敗した。

(S)処理を終了する。

(O)エラー詳細を基に設定ファイルが読み込みできるように、権限などの問題がないか確認する。問題が解決しない場合は、システム管理者に連絡する。

KAKC0403-E

```
Config file name too long.
```

設定ファイルのファイル名が長過ぎる。

(S)処理を終了する。

(O)システム管理者に連絡して、/opt/hitachi/bjex/conf 配下の設定ファイルに問題ないか確認する。

KAKC0404-E

```
The required element "パラメータ" is missing in config file.
```

設定ファイルに記述が必須であるパラメータが定義されていない。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0405-E

```
パラメータ名 "クラス名" is not defined.
```

設定ファイルのパラメータ名で示すパラメータに記述された"クラス名"が SysoutClass 名として定義されていない。

パラメータ名には、次のどれかが出力される。

- MSGCLASS_DEFAULT：設定ファイルの MSGCLASS_DEFAULT パラメータ
- STDOUTCLASS_DEFAULT：設定ファイルの STDOUTCLASS_DEFAULT パラメータ
- STDERRCLASS_DEFAULT：設定ファイルの STDERRCLASS_DEFAULT パラメータ

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0406-E

```
Failed to get host name. reason="エラー詳細"
```

エラー詳細で示す原因によって、動作しているサーバのホスト名の取得時にエラーが発生した。

(S)処理を終了する。

(O)システム管理者に連絡して、ネットワーク上のホスト名を確認する。

KAKC0407-E

```
"ファイル名" is not an ordinary file.
```

ファイル名で示すファイルは通常のファイルではない。

(S)処理を終了する。

(O)ファイルを確認する。

KAKC0409-I

```
Parsing of config file "ファイル名" ended. code=終了コード
```

ファイル名で示す設定ファイルの解析を終了した。

終了コードが0の場合、設定ファイルの解析処理は正常に終了した。

終了コードが0以外の場合、設定ファイルに誤りがある、または、設定ファイルの解析でエラーが発生した。

(S)終了コードが0の場合、処理を続行する。

終了コードが0以外の場合、処理を終了する。

(O)設定ファイルの誤りを修正する。または、設定ファイルの解析で発生したエラーの原因を解決する。

KAKC0411-E

```
Line size exceeds limits. line=行番号
```

設定ファイルの行番号で示す行の解析中に改行コードを見つけられない。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0414-E

ETag "タグ名" is not specified.

設定ファイルの**タグ名**で示すパラメータの終了タグが見つからなかった。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0420-E

Tag name is not specified. line=**行番号**

設定ファイルの**行番号**で示す行にパラメータが見つからなかった。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0422-E

Unrecognized Tag name is specified. line=**行番号**

設定ファイルの**行番号**で示す行に認識できないパラメータが見つかった。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0425-E

Tag format is invalid. line=**行番号**

設定ファイルの**行番号**で示す行は、記述形式に誤りがある。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0426-E

There are extra words after ">". line=**行番号**

設定ファイルの**行番号**で示す行のパラメータの終端 (>) 以降に不当な文字が見つかった。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0427-E

ETag is not match with STag "Tag 名". line=行番号

設定ファイルの行番号で示す行の Tag 名で示すパラメータの先頭と終端が不一致である。なお、パラメータの先頭が省略されている場合、Tag 名は空文字列となる。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0429-E

Nested tags are found. line=行番号

設定ファイルの行番号で示す行で、パラメータがネストして定義されている。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0430-E

SysoutClass "クラス名" is specified more than once. line=行番号

設定ファイルの行番号で示す行の SysoutClass は、クラス名で示すクラス名が重複して指定されている。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0431-E

Parameter name is invalid. line=行番号

設定ファイルの行番号で示す行に不当なパラメータ名が見つかった。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0432-E

Parameter value of "パラメータ名" is invalid. line=行番号

設定ファイルの行番号で示す行のパラメータ名が認識できない。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0433-E

Parameter value of "パラメータ名" is missing. line=行番号

設定ファイルの行番号で示す行のパラメータ名のオペランドが存在しない。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0434-E

Parameter "パラメータ名" is specified multiple. line=行番号

設定ファイルの行番号で示す行のパラメータ名が、重複して定義されている。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0435-E

Number of parameter "パラメータ名" exceed limit. line=行番号

設定ファイルの行番号で示す行のパラメータ名が、定義できる上限を超えている。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0436-E

String size of parameter "パラメータ名" value is invalid. line=行番号

設定ファイルの行番号で示す行のパラメータ名で指定した値が長過ぎる。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0437-E

Value scope of parameter "パラメータ名" value is invalid. line=行番号

設定ファイルの行番号で示す行のパラメータ名で指定した値が範囲外である。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0438-E

Wrong char is included in parameter "パラメータ名" value. line=行番号

設定ファイルの行番号で示す行のパラメータ名で指定した行内に、不当な文字コードが含まれている。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0439-E

Directory "パラメータ名" must be specified in absolute path. line=行番号

設定ファイルの行番号で示す行のパラメータ名で指定したファイルパス名を、絶対パスで指定する。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0441-E

Directory of "パラメータ名" is not exist. line=行番号

設定ファイルの行番号で示す行のパラメータ名で指定したディレクトリは存在しない。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0442-E

Parameter value of "パラメータ名" is not directory. line=行番号

設定ファイルの行番号で示す行のパラメータ名で指定したディレクトリは、ディレクトリではない。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0444-E

There are too many value for "パラメータ名". line=行番号

設定ファイルの行番号で示す行のパラメータ名のオペランドが多く、不当な文字列が含まれている可能性がある。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0445-E

Default Directory of "パラメータ名" does not exist : "ディレクトリ名"

設定ファイルの定義がないためデフォルトを仮定したが、パラメータ名のディレクトリ名は存在しない。

(S)処理を終了する。

(O)システム管理者に連絡して、バッチジョブ実行システムを再インストールする。

KAKC0446-E

Default Directory of "パラメータ名" is not valid directory : "ディレクトリ名"

設定ファイルの定義がないためデフォルトを仮定したが、パラメータ名のディレクトリ名はディレクトリではない。

(S)処理を終了する。

(O)システム管理者に連絡して、ディレクトリ構成を確認する。

KAKC0449-E

The required directory "ディレクトリ名" does not exist.

バッチジョブ実行システムに必須であるディレクトリ名が存在しない。ディレクトリ名に"/opt/jp1_fts"が表示される場合は、JP1/FTP がインストールされていない。

(S)処理を終了する。

(O)システム管理者に連絡して、バッチジョブ実行システムを再インストールする。JP1/FTP がインストールされていない場合は、インストールする。

KAKC0450-E

The required directory "ディレクトリ名" is not valid directory.

バッチジョブ実行システムに必須である、ディレクトリ名はディレクトリではない。

(S)処理を終了する。

(O)システム管理者に連絡して、ディレクトリ構成を確認する。

KAKC0453-E

The required element "パラメータ名" is missing in SysoutClass "クラス名". line=行番号

クラス名で示す SysoutClass 内に、必須であるパラメータ名の指定がない。

(S)処理を終了する。

(O)システム管理者に連絡して、設定ファイルを見直す。

KAKC0456-E

Parameter value of "パラメータ名" is specified multiple. line=番号

パラメータ名で指定した値が二重に定義されている。

(S)処理を終了する。

(O)設定ファイルの指定を見直す。

KAKC0700-E

Could not create the job directory. reason=エラー詳細

スプールディレクトリ配下にジョブのためのディレクトリ作成に失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。エラー詳細の原因を取り除いて再実行する。

KAKC0701-E

"ファイル名" open failed. reason=エラー詳細

スプールジョブディレクトリ内のファイル名で表示されるファイルのオープンに失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。エラー詳細の原因を取り除いて再実行する。

KAKC0702-E

Joblog file I/O error. reason=エラー詳細

スプールジョブディレクトリ内のジョブログファイルの書き込みに失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。エラー詳細の原因を取り除いて再実行する。

KAKC0703-E

"ファイル名" does not exist.

スプールジョブディレクトリ内に**ファイル名**で表示されるファイルが存在しない。

(S)処理を終了する。

(O)ファイル名を見直す。

KAKC0704-E

Failed to get the current date.

日付の取得に失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC0705-E

Failed to get host name.

ホスト名の取得に失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC0706-E

Failed to create "ファイルパス". reason=エラー詳細

ファイルパスの作成に失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。エラー詳細の原因を取り除いて再実行する。

KAKC0707-I

MSGCLASS is "メッセージクラス名".

このジョブのメッセージクラスは"メッセージクラス名"である。

(S)処理を続行する。

(O)なし。

KAKC0709-E

Failed to create stdout file path.

ジョブステッププログラムの標準出力を格納するファイルのパス作成に失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC0710-E

Failed to create stderr file path.

ジョブステッププログラムの標準エラー出力を格納するファイルのパス作成に失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC0711-E

Failed to create stdout file "ファイル名". reason=エラー詳細

ジョブステッププログラムの標準出力を格納するファイルの作成に失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。エラー詳細の原因を取り除いて再実行する。

KAKC0712-E

Failed to create stderr file "ファイル名". reason=エラー詳細

ジョブステッププログラムの標準エラー出力を格納するファイルの作成に失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。エラー詳細の原因を取り除いて再実行する。

KAKC0719-I

STEP ステップ番号,ジョブステップ名[.プロシジャステップ名],出力先

このメッセージのあとに、ジョブステップの出力先の内容を出力する。

(S)処理を続行する。

(O)なし。

KAKC0720-E

Failed to open jobid file. reason=**エラー詳細**

ジョブ識別子管理ファイルのオープンに失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。**エラー詳細**の原因を取り除いて再実行する。

KAKC0721-E

Jobid file I/O error. reason=**エラー詳細**

ジョブ識別子管理ファイルの入出力に失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。**エラー詳細**の原因を取り除いて再実行する。

KAKC0722-E

Failed to allocate Jobid.

このジョブに固有のジョブ識別子を与えられなかった。

(S)処理を終了する。

(O)システム管理者に連絡する。前後に出力されたメッセージから原因を取り除く。

KAKC0723-E

Failed to lock jobid file. reason=**エラー詳細**

ジョブ識別子管理ファイルを排他しようとしたが失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。**エラー詳細**の原因を取り除いて再実行する。

KAKC0724-I

Jobid was assigned. Jobid=**ジョブ識別子**

ジョブ識別子を割り当てた。

(S)処理を続行する。

(O)なし。

KAKC0730-E

Failed to convert joblog. (rc=終了コード,errno=エラー番号,function=関数名)

ジョブログを標準エラー出力に出力する際、コード変換に失敗した。エラーが発生した Hitachi Code Converter 関数名およびエラー番号が出力される。

(S)コード変換前の内容を出力し、処理を続行する。

(O)エラーの原因を取り除く。

KAKC0800-E

Failed to create the .sysout. reason=エラー詳細

スプールジョブ管理ファイルを作成しようとしたが失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。エラー詳細の原因を取り除いて再実行する。

KAKC0801-E

The .sysout lock error. reason=エラー詳細

スプールジョブ管理ファイルを排他しようとしたが失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。エラー詳細の原因を取り除いて再実行する。

KAKC0802-E

The .sysout open failed. reason=エラー詳細

スプールジョブ管理ファイルをオープンしようとしたが失敗した。

(S)処理を終了する。

(O)システム管理者に連絡する。エラー詳細の原因を取り除いて再実行する。

KAKC0803-E

The .sysout I/O error. reason=エラー詳細

スプールジョブ管理ファイルで入出力エラーが発生した。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を終了する。

(O)システム管理者に連絡する。エラー詳細の原因を取り除いて再実行する。

KAKC0804-E

Failed to get the current time used for STARTTIME in the .sysout.

スプールジョブ管理ファイルの STARTTIME に使用する現在時刻の取得に失敗した。

(S)処理を続行する。

(O)システム管理者に連絡する。

KAKC0805-E

Failed to get the current time used for ENDTIME in the .sysout.

スプールジョブ管理ファイルの ENDTIME に使用する現在時刻の取得に失敗した。

(S)処理を続行する。

(O)システム管理者に連絡する。

KAKC0900-I

Parsing of xml file "ファイル名" started.

ファイル名で示すジョブ定義 XML ファイルの解析処理を開始した。

(S)処理を続行する。

(O)なし。

KAKC0901-E

Xml file "ファイル名" open failed. reason="エラー詳細"

エラー詳細で示す原因によって、ファイル名で示されるジョブ定義 XML ファイルのオープンに失敗した。

(S)処理を終了する。

(O)エラー詳細を基に、ジョブ定義 XML ファイルが読み込みできるよう、権限などの問題がないか確認する。問題が解決しない場合は、システム管理者に連絡する。

KAKC0902-E

Xml file "ファイル名" read error. reason="エラー詳細"

エラー詳細で示す原因によって、ファイル名で示されるジョブ定義 XML ファイルの読み込み中にエラーが発生した。

(S)処理を終了する。

(O)エラー詳細を基に、ジョブ定義 XML ファイルが読み込みできるよう、権限などの問題がないか確認する。問題が解決しない場合は、システム管理者に連絡する。

KAKC0904-E

Root Element must be "ルート要素名". elementNo=番号

番号で示すタグの解析中にルート要素を見つけたが、ルート要素名には"ルート要素名"を指定する必要がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを確認し、ルート要素名には"ルート要素名"を指定する。

KAKC0905-E

Root Element is specified multiple. elementNo=番号

番号で示すタグの解析中にルート要素を見つけたが、ルート要素が重複して定義されている。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、ルート要素が1つだけとなるように修正して、ジョブを再実行する。

KAKC0906-E

Specified encoding attribute value is invalid.

指定したエンコーディングは使用できない。

(S)ジョブ定義 XML ファイルの解析処理を終了する。

(O)ジョブ定義 XML ファイルの XML 宣言での encoding 属性値を正しく修正してジョブを再実行する。

KAKC0908-E

"ファイル名" is not an ordinary file.

ファイル名で示すファイルは特殊なファイルである。

(S)処理を終了する。

(O)ファイル名に通常のファイルを指定し、ジョブを再実行する。

KAKC0909-I

Parsing of xml file "ファイル名" ended. code=終了コード

ファイル名で示されるジョブ定義 XML ファイルの解析処理が完了した。

終了コードが 0 の場合、ジョブ定義 XML ファイルの解析は正常終了した。

終了コードが 0 以外の場合、ジョブ定義 XML ファイルに誤りがある、または、ジョブ定義 XML ファイルの解析処理でエラーが発生した。

(S)終了コードの値によって、次のとおり処理を続行または終了する。

- 終了コードが 0 の場合、処理を続行する。
- 終了コードが 0 以外の場合、処理を終了する。

(O)終了コードが 0 以外の場合、ジョブ定義 XML ファイルの誤りを修正する。または、ジョブ定義 XML ファイルの解析処理で発生したエラーの原因を解決する。

KAKC0911-E

End of file encountered in element/comment/CDATA.

ジョブ定義 XML ファイルの解析処理中に、要素、コメント、または CDATA セクションの途中でファイルの終端を検出した。それぞれのタグの終了文字列を、正しく記述する必要がある。

要因として、次のことが考えられる。

- "element"の場合
要素の終了を表す >, />, ?> が見つからない。要素の終了位置にこれらを記述する必要がある。
- "comment"の場合
コメントの終了を表す --> が見つからない。コメントの終了位置に --> を記述する必要がある。
- "CDATA"の場合
CDATA セクションの終了を表す]]> が見つからない。CDATA セクションの終了位置に]]> を記述する必要がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、要因に応じて対処してジョブを再実行する。

KAKC0912-E

Prolog is not Xml declaration. elementNo=番号

番号で示す prolog (ジョブ定義 XML ファイルの先頭部分) が XML 宣言 (<?xml・・・?>) ではない。

ジョブ定義 XML ファイルの先頭の<?で始まるタグは、XML 宣言である必要がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルの先頭の<?で始まるタグを記述する場合は、XML 宣言を記述する。修正後、ジョブを再実行する。

KAKC0913-E

The root element is required.

ジョブ定義 XML ファイルのルート要素がない。

このエラーは、ジョブ定義 XML ファイル内に開始タグ、または空要素タグが1つも存在しないときに発生する。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、ルート要素を定義してジョブを再実行する。

KAKC0918-E

Semicolon is required at the end of the EntityRef. elementNo=番号

番号で示すタグで実体参照をしているが、セミコロン (;) がない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、実体参照をしている個所の正しい位置に、セミコロンを記述してジョブを再実行する。

KAKC0919-E

Undefined entity is referred. elementNo=番号

番号で示すタグで実体参照をしているが、amp, lt, gt, apos, quot などで決められた名称以外である。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、実体参照の名称を amp, lt, gt, apos, quot などの決められた名称に修正してジョブを再実行する。

KAKC0920-E

Element name is not specified. elementNo=番号

番号で示すタグの要素名がない。

このエラーは、要素の開始記号<, </, <?の直後の文字が要素名として使用できる文字でない場合に発生する。要素名として使用できる文字は、英字、数字、'!', '-', '_', ':'である。

要因として、次のことが考えられる。

- 要素での開始記号の直後の文字が空白
< JOB NAME="JOB00">, </ JOB>, <? xml version="1.0" encoding="Shift-JIS" ?>
- 要素での開始記号の直後の文字が不当文字
<%JOB NAME="JOB00">, </%JOB>, <?%xml version="1.0" encoding="Shift-JIS" ?>

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、要因に応じて対処してジョブを再実行する。

KAKC0921-E

Element name is invalid. elementNo=番号

番号で示すタグの要素名に誤りがある。

このエラーは、要素名の直後の文字が空白でなく、かつ要素名として使用できる文字でない場合に発生する。要素名として使用できる文字は、英字、数字、'!', '-', '_', ':'である。

要因として、次のことが考えられる。

- 要素名の直後の文字が不当文字
<JOB% NAME="JOB00">, </JOB%>, <?xml% version="1.0" encoding="Shift-JIS" ?>

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、要因に応じて対処してジョブを再実行する。

KAKC0922-E

Unrecognized element "要素名" is specified in the element "要素名". elementNo=番号

番号で示すタグの要素名で表示された未サポートの要素名がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、正しい要素名を指定してジョブを再実行する。

KAKC0923-E

The required element "要素名 1" is missing from the element "要素名 2". elementNo=番号

番号で示すタグに、要素名 2 中に記述する必須の要素名 1 がない。

要素名 1 の出力には、次のようなケースがある。

- "要素 A"
要素 A を記述する必要がある。
- "要素 A | 要素 B"
要素 A, または要素 B のどちらかを記述する必要がある。'|'は選択を表す。
- "要素 A,要素 B"
要素 A, 要素 B の両方を, この順序で記述する必要がある。','は順序の定められた並びを示す。
- "要素 A,[要素 B]"
要素 A, 要素 B をこの順序で記述可能だが, 記述する必要があるのは要素 A だけであり, 要素 B は記述してもしなくてもよい。'[', ']'は指定が任意であることを示す。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し, 要素名 2 中に要素名 1 を記述してジョブを再実行する。

KAKC0924-E

The element "要素名 1" is specified multiple in the element "要素名 2". elementNo=番号

番号で示すタグの要素名 1 を見つけたが, 要素名 2 中には重複定義ができない。

要素名 1 の出力には、次のようなケースがある。

- "要素 A"
要素 A が重複定義されている。
- "要素 A | 要素 B"
要素 A, または要素 B が重複定義されている。'|'は選択を表す。
要素 A と要素 B が 1 つずつ定義されている場合もこのケースに該当するため, どれかを一回だけ定義する必要がある。
- "要素 A,要素 B"
要素 A, 要素 B の両方がこの順序で並んだものが, 重複定義されている。','は順序の定められた並びを示す。
要素 A と要素 B が 1 つずつ定義されている場合は正しい記述であり, このケースには該当しない。
- "要素 A,[要素 B]"
要素 A, 要素 B がこの順序で並んだものが重複定義されている。'[', ']'は指定が任意であることを示す。
要素 A が複数定義されていて, 要素 B が定義されていない場合も, このケースに該当する。要素 A を 1 回, 要素 B を 1 回以下定義する必要がある。
要素 A と要素 B が 1 つずつ定義されている場合は正しい記述であり, このケースには該当しない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、**要素名 2**中に**要素名 1**が重複しないよう修正してジョブを再実行する。

KAKC0925-E

E|EmptyElement/Prolog/Comment tag format is invalid. elementNo=**番号**

番号で示す要素、XML 宣言、またはコメントのタグ形式に誤りがある。

要因として、次のことが考えられる。

- "E | EmptyElement tag"の場合

タグの開始が</かつ終了が/>となっている。該当するタグが終了タグである場合は、タグの終了を>に、空要素タグである場合は、タグの開始を<に修正する必要がある。

- "Prolog tag"の場合

タグの開始が<?だが、タグの終了が?>ではない。該当するタグが XML 宣言、または処理命令 PI であるならば終了タグを?>に、それ以外のタグであるなら開始タグを<または</に修正する必要がある。

- "Comment tag"の場合

コメント中に文字列--があり、かつその後ろが>ではない。コメント終了を表す-->以外で、コメント中に文字列--を記述しないよう修正する必要がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、要因に応じて対処してジョブを再実行する。

KAKC0926-E

ETag of the element "**要素名**" is not specified.

要素名を解析したが、終了タグが指定されていない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、**要素名**の終了タグを指定してジョブを再実行する。

KAKC0927-E

ETag of the element "**要素名**" is not match with STag. elementNo=**番号**

番号で示すタグの**要素名**を解析したが、タグの開始と終了が不一致である。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、**要素名**のタグの開始と終了が一致するよう修正してジョブを再実行する。

KAKC0929-E

Extra element "**要素名 1**" is specified in the element "**要素名 2**". elementNo=**番号**

番号で示すタグの**要素名 1**を解析したが、**要素名 2**中には定義できない要素である。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、正しい要素を定義してジョブを再実行する。

KAKC0930-E

Attribute name is not specified in the element "**要素名**". elementNo=**番号**

番号で示すタグの**要素名**を解析中に、属性名が見つからなかった。

このエラーは、属性名を記述しなければいけない個所に、属性名として使用できない文字が出現したときに発生する。

要因として、次のことが考えられる。

- 属性名の先頭に不当な文字を記述している
<DD NAME="ERRDD" %DSN="INPUT.FILE" /> : 属性名の先頭に不当な文字
- 属性間などに不当な文字を記述している
<DD NAME="ERRDD" % DSN="INPUT.FILE" /> : 属性間に不当な文字
<DD % NAME="ERRDD" DSN="INPUT.FILE" /> : 要素名のあとに不当な文字
<DD NAME="ERRDD" DSN="INPUT.FILE" % /> : 要素内末尾に不当な文字

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、要因に応じて対処してジョブを再実行する。

KAKC0931-E

Attribute name "**属性名**" is invalid in the element "**要素名**". elementNo=**番号**

番号で示すタグの**要素名**を解析中に、**属性名**が見つかったが、属性名に誤りがある。

このエラーは、属性名の直後の文字が空白でも '='でもなく、かつ属性名として使用できる文字でない場合に発生する。属性名として使用できる文字は、英字、数字、 '.', '-', '_', ':'である。

要因として、次のことが考えられる。

- 属性名の直後の文字が不当文字

<JOB NAME%="JOB00">, <JOB NAME% = "JOB00">

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、要因に応じて対処してジョブを再実行する。

KAKC0932-E

Unrecognized attribute "属性名" is specified in the element "要素名". elementNo=番号

番号で示すタグの要素名を解析中に、未サポートの属性名が見つかった。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、正しい属性名を指定してジョブを再実行する。

KAKC0933-E

The required attribute "属性名" is missing in the element "要素名". elementNo=番号

番号で示すタグの要素名を解析中に、属性名で示す必須の属性が見つからなかった。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、属性名で示す必須の属性を指定してジョブを再実行する。

KAKC0934-E

The attribute "属性名" is specified multiple in the element "要素名". elementNo=番号

番号で示すタグの要素名を解析中に、複数回の属性名が見つかった。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、要素内に属性名が複数回指定されていないよう修正してジョブを再実行する。

KAKC0935-E

The attribute "属性名" format is invalid in the element "要素名". elementNo=番号

番号で示すタグの要素名を解析中に、属性名が見つかったが、属性の指定形式に誤りがある。

要因として、次のことが考えられる。

- 属性名の後ろに空白が続き、その後ろは要素の終了記号>, />, ?>である
<JOB NAME >, <EXEC PGM />, <?xml version ?>
- 属性名の後ろに空白が続き、その後ろが'='でないケース

<JOB NAME A >, <EXEC PGM % />, <?xml version 1?>

- 属性名の後ろに '=' があるが、その後ろが要素の終了記号 >, />, ?> である

<JOB NAME=>, <EXEC PGM =/>, <?xml version =?>

- 属性名の後ろに '=' があるが、その後ろが空白だけである

<JOB NAME= >, <EXEC PGM = />, <?xml version = ?>

- 属性名の後ろに '=' があるが、その後ろが引用符でない

<JOB NAME=JOB00" >, <EXEC PGM= UP00' />, <?xml version = 1.0" ?>

(S) ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O) ジョブ定義 XML ファイルを見直し、要因に応じて対処してジョブを再実行する。

KAKC0937-E

Delimiter space between some attributes is not specified in the element "要素名". elementNo=番号

番号で示すタグの要素名を解析中に、属性と属性の間がスペースで区切られていない属性を見つけた。

(S) ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O) ジョブ定義 XML ファイルを見直し、属性と属性の間をスペースで区切ってジョブを再実行する。

KAKC0940-E

The value of the attribute "属性名" is an invalid format. elementNo=番号

番号で示すタグの属性名の値が不当な形式である。

(S) ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O) ジョブ定義 XML ファイルを見直し、「9. ジョブ定義 XML ファイル」に従い、各属性で指定できる値を指定してジョブを再実行する。

KAKC0941-E

String size of the attribute "属性名" value is invalid. elementNo=番号

番号で示すタグの属性名の値が不当な長さの文字列である。

(S) ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O) ジョブ定義 XML ファイルを見直し、「9. ジョブ定義 XML ファイル」に従い、各属性の値として指定できる長さの文字列を指定してジョブを再実行する。

KAKC0942-E

Value scope of the attribute "属性名" value is invalid. elementNo=番号

番号で示すタグの属性名の値が範囲外数値である。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、「9. ジョブ定義 XML ファイル」に従い、各属性の値として指定できる範囲の数値を指定してジョブを再実行する。

KAKC0943-E

Invalid char is found in value of attribute "属性名". elementNo=番号

番号で示すタグの属性名の値が利用できない文字コードを含んでいる。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、「9. ジョブ定義 XML ファイル」に従い、各属性の値として利用できる文字コードを指定してジョブを再実行する。

KAKC0944-E

The attribute "属性名" value is invalid. elementNo=番号

番号で示すタグの属性名の値が利用できない文字列である。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、「9. ジョブ定義 XML ファイル」に従い、各属性の値として利用できる文字列を指定してジョブを再実行する。

KAKC0945-E

Too many positional parameters. attribute="属性名" elementNo=番号

番号で示すタグの属性名の値が、コンマ (,) などで必要以上に複数指定されている。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、「9. ジョブ定義 XML ファイル」に従い、各属性の値として指定できる個数を指定してジョブを再実行する。

KAKC0946-E

Positional parameter is missing. attribute="属性名" elementNo=番号

番号で示すタグの属性名の値にコンマ (,) などで指定する場合に、必須となる値が指定されていない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、「9. ジョブ定義 XML ファイル」に従い、各属性に必須の値を指定して、ジョブを再実行する。

KAKC0947-E

```
Invalid format for positional parameter. attribute="属性名" elementNo=番号
```

番号で示すタグの属性名の値が、コンマ (,) などで指定する場合に、括弧のフォーマットが不正である。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、「9. ジョブ定義 XML ファイル」に従い、各属性の正しい括弧のフォーマットで値を指定してジョブを再実行する。

KAKC1100-I

```
Parsing of proc file "ファイル名" started.
```

ファイル名で示されるカタログドプロシジャの解析処理を開始した。

(S)処理を続行する。

(O)なし。

KAKC1101-E

```
Proc "プロシジャ名" not found. elementNo=番号
```

番号で示すタグの解析中に、プロシジャ名で示すカタログドプロシジャが見つからなかった。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、ジョブ内に定義した入力ストリームプロシジャの PROC 要素の NAME 属性、またはプロシジャライブラリに存在するカタログドプロシジャファイル名 (".xml"は記述しない) を CALL 要素の NAME 属性に指定する。修正後、ジョブを再実行する。

KAKC1102-E

```
Step "ジョブステップ名" not found. elementNo=番号
```

番号で示すタグの解析中にジョブステップ名で示すジョブステップを置換しようとしたが、該当するジョブステップが見つからなかった。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、REPLACESTEP 要素の NAME 属性にプロシジャ中に存在するジョブステップ名を指定して、ジョブを再実行する。

KAKC1103-E

EXEC entity cannot be replaced. elementNo=番号

番号で示すタグの解析中に EXEC 要素の内容（シェルスクリプト）を置換しようとしたが、EXEC 要素の内容は置換できない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、EXEC 要素の内容（シェルスクリプト）を置換しないよう修正して、ジョブを再実行する。

KAKC1104-E

The DD element with NAME="NAME 属性値" is specified more than once. elementNo=番号

番号で示すタグの解析中に、NAME 属性値で示す名称の重複した DD 要素が見つかった。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを次に従って修正し、ジョブを再実行する。

- JOB 要素内の DD 要素の場合
JOB 要素内の DD 要素には NAME="JOB LIB"かつ TYPE="LIB"を指定しなければならないため、それ以外の DD を修正する。
- STEP 要素内の DD 要素の場合
同一ジョブステップ内では、NAME 属性が重複しないよう修正する。ただし、DD 要素を連結したい場合は、NAME 属性が同一の DD 要素を連続して定義する。間に他の NAME 属性を指定した DD 要素を定義してはならない。

KAKC1105-E

Cataloged procedure xml cannot execute direct.

カタログドプロシジャは直接実行できない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)bjexec コマンドの引数に指定するファイル名には、JOB 要素を含む通常のジョブ定義 XML ファイルを指定する。ただし、-c オプションを指定した場合は、引数にカタログドプロシジャのファイルを指定してもよい。この場合、カタログドプロシジャのシンタクスチェックだけ行って処理を終了する。

KAKC1106-E

procedure nest error. elementNo=番号

番号で示すタグの解析中にプロシジャをネストして定義しているのが見つかったが、プロシジャをネストして定義することはできない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、プロシジャ中のジョブステップには、CALL 要素を定義しないよう修正してジョブを再実行する。

KAKC1107-E

Step "ジョブステップ名" in cond attribute is not found. elementNo=番号

番号で示すタグの解析中に COND 属性が見つかったが、属性値に指定したジョブステップ名が見つからない。

ジョブステップ名：ジョブステップ名が表示される場合と、ジョブステップ名.プロシジャステップ名が表示される場合がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、COND 属性の値にジョブステップ名を指定する場合は、ジョブ中またはプロシジャ中に存在するジョブステップ名を指定して、ジョブを再実行する。

KAKC1109-I

Parsing of proc file "ファイル名" ended. code=終了コード

ファイル名で示されるカタログドプロシジャの解析処理を終了した。

終了コードが 0 の場合、カタログドプロシジャの解析は正常終了した。

終了コードが 0 以外の場合、カタログドプロシジャに誤りがある。または、カタログドプロシジャの解析処理でエラーが発生した。

(S)終了コードの値によって、次の処理を続行または終了する。

- 終了コードが 0 の場合、処理を続行する。
- 終了コードが 0 以外の場合、処理を終了する。

(O)終了コードが 0 以外の場合、カタログドプロシジャの誤りを修正する。または、カタログドプロシジャの解析処理で発生したエラーの原因を解決する。

KAKC1110-E

The number of steps exceeded ジョブステップ数上限値.

ジョブ中のプロシジャステップを含むジョブステップ数がジョブステップ数上限値を超えた

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、ジョブ中のプロシジャを含むジョブステップ数がジョブステップ数上限値を超えないように修正して、ジョブを再実行する。

KAKC1111-E

PGM attribute of EXEC element cannot be replaced. elementNo=番号

番号で示すタグの解析中に、PGM 属性を置換する指定があったが、PGM 属性は置換できない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、PGM 属性を置換しないように修正して、ジョブを再実行する。

KAKC1112-E

Directory ID format is invalid. elementNo=番号

番号で示すタグ内でパス名置換用識別子を利用しているが、指定形式に誤りがある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、パス名置換用識別子を正しい形式に修正してジョブを再実行する。

KAKC1113-E

Directory ID "パス名置換用識別子" includes invalid character. elementNo=番号

番号で示すタグ内でパス名置換用識別子を利用しているが、利用できない文字を含んでいる。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、設定ファイルで定義したパス名置換用識別子を記述してジョブを再実行する。

KAKC1114-E

Directory ID "パス名置換用識別子" is too long. elementNo=番号

番号で示すタグ内でパス名置換用識別子を利用しているが、長さが不当である。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、設定ファイルで定義したパス名置換用識別子を記述してジョブを再実行する。

KAKC1115-E

Directory ID "パス名置換用識別子" is not defined. elementNo=番号

番号で示すタグ内でパス名置換用識別子を利用しているが、パス名置換用識別子が定義されていない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、設定ファイルで定義したパス名置換用識別子を記述してジョブを再実行する。

KAKC1117-E

Attribute "属性名" must be specified with TYPE="ファイル種別". elementNo=番号

番号で示すタグ内の DD 要素に、ファイル種別で示す属性で必須の属性（属性名）が定義されていない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、属性名で示す必須の属性を指定、または TYPE 属性を正しく指定してジョブを再実行する。

KAKC1118-E

Attribute "属性名" must not be specified with TYPE="ファイル種別". elementNo=番号

番号で示すタグ内の DD 要素に、ファイル種別で示す属性で背反の属性（属性名）が定義されている。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、属性名で示す背反の属性を削除、または TYPE 属性を正しく指定してジョブを再実行する。

KAKC1119-E

Invalid back reference in the DSN attribute. elementNo=番号

番号で示すタグ内の DD 要素で DSN 属性の逆方向参照を解析したが、文法誤りによって正しく参照ができない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、逆方向参照の文法誤りを取り除いてジョブを再実行する。

KAKC1120-E

The step specified in back reference is not found. elementNo=番号

番号で示すタグ内の DD 要素で DSN 属性の逆方向参照を解析したが、対象となるジョブステップが見つからなかった。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、ジョブステップ名にはジョブ中、またはプロシジャ中に存在するジョブステップ名を指定して、ジョブを再実行する。

KAKC1121-E

DDNAME "DD 名" in back reference is not found. elementNo=番号

番号で示すタグ内の DD 要素で DSN 属性の逆方向参照を解析したが、DD 名で示す対象となる DD 要素が見つからなかった。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、DD 名には同一ジョブステップ、または指定したジョブステップに存在する DD 名を指定して、ジョブを再実行する。

KAKC1122-E

This DD must specify NAME="JOBLIB" and TYPE="LIB". elementNo=番号

番号で示すタグ内の DD 要素は、NAME="JOBLIB"および TYPE="LIB"を指定する必要がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、JOB 要素中に指定する DD 要素は、すべて NAME="JOBLIB"および TYPE="LIB"を指定するように修正して、ジョブを再実行する。

KAKC1123-E

NAME="JOBLIB" cannot be specified for this DD. elementNo=番号

番号で示すタグ内の JOBLIB 属性の DD 要素は、STEP 要素内には定義できない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、STEP 要素内に NAME="JOBLIB"を指定した DD 要素を定義しないように修正してジョブを再実行する。

KAKC1124-E

SysoutClass "クラス名" is not defined in Config File. elementNo=番号

番号で示すタグ内のクラス名で示す値は、SysoutClass として設定ファイルに定義されていない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、設定ファイルで定義した SysoutClass のクラス名を指定してジョブを再実行する。

KAKC1125-E

Sysin data is invalid. Delimiter is not found. elementNo=番号

番号で示すタグ内の SYSIN データの区切りが判断できない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、SYSIN データの開始と終了位置に！を記述するように修正してジョブを再実行する。

KAKC1126-E

Sysin data is invalid. End of delimiter is not found. elementNo=番号

番号で示すタグ内の SYSIN データの終端が判断できない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、SYSIN データの終了位置に！を記述するように修正してジョブを再実行する。

KAKC1127-E

Exec script is invalid. Delimiter is not found. elementNo=番号

番号で示す要素の内容（スクリプト）の区切りが判断できない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、スクリプトの開始と終了位置に！を記述するように修正してジョブを再実行する。

KAKC1128-E

Exec script is invalid. End of delimiter is not found. elementNo=番号

番号で示す要素の内容（スクリプト）の終端が判断できない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、スクリプトの終了位置に！を記述するように修正してジョブを再実行する。

KAKC1130-E

Value of Symbol Parameter "記号パラメータ名" is not specified. elementNo=番号

番号で示すタグの解析中に記号パラメータ名を処理しようとしたが、記号パラメータが定義されていない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、プロシジャ中に SYMBOL 要素で記号パラメータ名を定義する、または CALL 要素中に SETSYMBOL 要素で記号パラメータ名を定義する。

KAKC1131-E

Invalid char is found in the Symbol Parameter "記号パラメータ名". elementNo=番号

番号で示すタグの解析中に記号パラメータ名を処理しようとしたが、記号パラメータとして処理できない文字が含まれている。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、記号パラメータとして利用できる文字で記述するように修正してジョブを再実行する。

KAKC1132-E

Proc file "ファイル名" 関数名 error reason="エラー詳細" elementNo=番号

カタログドプロシジャ XML ファイルを関数名で示す関数で確認しようとしたが、エラー詳細で示すエラーが発生した。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)エラー詳細を基にカタログドプロシジャ XML ファイルが読み込みできるように、権限などの問題がないか確認する。問題が解決しない場合は、システム管理者に連絡する。

KAKC1133-E

Attribute "PGM" is not specified. elementNo=番号

番号で示すタグ内に PGM 属性が指定されていない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、PGM 属性を指定してジョブを再実行する。

KAKC1134-E

Specified step "ジョブステップ名" is already replaced. elementNo=番号

番号で示すタグの解析中にジョブステップ名で示すジョブステップが見つかったが、このジョブステップは置換済みのため置換できない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、一回のプロシジャ呼び出しで、特定のジョブステップを複数回置換することがないように修正してジョブを再実行する。

KAKC1135-E

TYPE="LIB" must be specified with NAME="JOBLIB" or "STEPLIB". elementNo=番号

番号で示すタグ内の DD 要素で、TYPE="LIB"は NAME="JOBLIB"または NAME="STEPLIB"の場合に指定する必要がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、NAME 属性を JOBLIB, STEPLIB に修正する、または TYPE を LIB 以外に修正して、ジョブを再実行する。

KAKC1136-E

"DD 名" cannot be specified for NAME. elementNo=番号

番号で示すタグ内の DD 要素で、DD 名で示す文字列は NAME 属性の値に指定できない。

DD 要素の NAME 属性に指定できない名称：AJOBLIB, ASTEPLIB, SYSUDUMP, SYSABEND, SYSNDDUMP, SYSCHK, SYSCKEOV, JOBCAT, STEPCAT, KANJLIB, IMGLIB, SYSMAPLB

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、DD 要素の NAME 属性に指定できない DD 名を指定しないように修正して、ジョブを再実行する。

KAKC1137-E

NAME="STEPLIB" must be specified with TYPE="LIB". elementNo=番号

番号で示すタグ内の DD 要素で、NAME="STEPLIB"は TYPE="LIB"の場合に指定する必要がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、NAME 属性を STEPLIB 以外に修正する、または TYPE を LIB に修正してジョブを再実行する。

KAKC1138-E

The SETENV element with NAME="NAME 属性値" is specified more than once. elementNo=番号

番号で示すタグの解析中に NAME 属性値で示す名称の重複した SETENV 要素が見つかった。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、JOB 要素内または STEP 要素内では、SETENV 要素の NAME 属性が重複しないように修正する。

KAKC1139-E

JOB element cannot be specified in cataloged procedure. elementNo=番号

番号で示すタグ内の解析中に JOB 要素が見つかったが、カタログドプロシジャ中に JOB 要素を指定することはできない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、カタログドプロシジャ中に JOB 要素を定義しないように修正して、ジョブを再実行する。

KAKC1140-E

TYPE is invalid in DD element of back reference. elementNo=番号

番号で示すタグの解析中に、逆方向参照を指定した DD 要素が見つかったが、TYPE 属性の値が不正である。

逆方向参照を指定した DD 要素は、TYPE 属性に"FILE"以外を指定してはならない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、逆方向参照を指定した DD 要素の TYPE 属性を修正してジョブを再実行する。

KAKC1141-E

TYPE is invalid in target DD element of back reference. elementNo=番号

番号で示すタグの解析中に、逆方向参照を指定した DD 要素が見つかったが、逆方向参照の対象となる DD 要素の TYPE 属性の値が不正である。

逆方向参照の対象となる DD 要素は、TYPE 属性に"FILE", "DATA", "TEMP"以外を指定してはならない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、逆方向参照の対象となる DD 要素の TYPE 属性を修正してジョブを再実行する。

KAKC1142-W

The DD with TYPE="TYPE 属性値" can not be concatenated. elementNo=番号

番号で示すタグで TYPE="TYPE 属性値"の DD 要素を連結指定しているが、TYPE="TYPE 属性値"の DD 要素は連結できない。

(S)処理を続行する。

(O)実行するプログラムの入出力仕様を確認し、TYPE="TYPE 属性値"を指定した DD 要素を連結しているのが誤りであれば修正する。

KAKC1143-E

Minimum record length cannot be specified for attribute "ISRECL". elementNo=番号

番号で示すタグの ISRECL 属性に最小レコード長は指定できない。

最小レコード長は、ISRECFM 属性で V を指定した場合にだけ指定できる。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、ISRECFM 属性の指定を V に修正するか、または ISRECL 属性に指定した最小レコード長を削除してジョブを再実行する。

KAKC1144-E

Minimum record length is not specified for attribute "ISRECL". elementNo=番号

番号で示すタグの ISRECL 属性に最小レコード長の指定がない。

ISRECFM 属性で V を指定した場合は、ISRECL 属性に最小レコード長を指定する必要がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、ISRECFM 属性を F に修正するか、または ISRECL 属性に最小レコード長を指定してジョブを再実行する。

KAKC1145-E

Minimum record length specified for attribute "ISRECL" is bigger than the record length.
elementNo=番号

番号で示すタグの ISRECL 属性で指定した最小レコード長がレコード長より大きい。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、ISRECL 属性で指定する最小レコード長がレコード長以下になるように修正してジョブを再実行する。

KAKC1146-E

Key position specified for attribute "ISKEY" is out of range. elementNo=番号

番号で示すタグの ISKEY 属性で指定したキー位置が範囲を超えている。

ISKEY 属性で指定するキー位置は、ISRECL 属性で指定するレコード長または最小レコード長より小さい値にする必要がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、ISKEY 属性で指定するキー位置が ISRECL 属性で指定するレコード長または最小レコード長より小さい値となるように修正してジョブを再実行する。

KAKC1147-E

Specified key extends beyond or is not within the record. attribute="ISKEY" elementNo=番号

番号で示すタグの ISKEY 属性で指定したキー位置+キー長が範囲を超えている。

ISKEY 属性で指定するキー位置とキー長の合計は、ISRECL 属性で指定するレコード長または最小レコード長以下にする必要がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、ISKEY 属性で指定するキー位置とキー長の合計が ISRECL 属性で指定するレコード長または最小レコード長以下の値となるように修正して、ジョブを再実行する。

KAKC1148-E

Specified DD element must contain all of attributes "ISKEY", "ISRECFM" and "ISRECL".
elementNo=番号

番号で示すタグの DD 要素に ISKEY, ISRECFM, および ISRECL 属性をすべて指定する必要がある。

一時 ISAM ファイルを作成する DD 要素の場合は、ISKEY、ISRECFM、および ISRECL 属性を必ず指定すること。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、ISKEY、ISRECFM、および ISRECL 属性を指定してジョブを再実行する。

KAKC1149-E

Attribute "属性名" must be specified with LANG="プログラム言語" of EXEC element.
elementNo=番号

番号で示すタグ内に属性名で示す属性が指定されているが、EXEC 要素の LANG="プログラム言語"と同時に指定する必要がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、属性名で示す属性、または LANG 属性を正しく指定してジョブを再実行する。

KAKC1151-E

DISP=",DELETE" cannot be specified for the same DSN attributes between parallelly-executed job steps. elementNo=番号

番号で示すタグの解析中に、DISP=",DELETE"を指定した DD 要素が見つかったが、並列実行するジョブステップ間の同一ファイルに DISP=",DELETE"の指定はできない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、並列実行するジョブステップ間の同一ファイルに DISP=",DELETE"を指定しないように修正して、ジョブを再実行する。

KAKC1152-E

DISP="SHR," must be specified for the same DSN attributes between parallelly-executed job steps. elementNo=番号

番号で示すタグの解析中に、DISP="SHR"以外を指定した DD 要素が見つかったが、並列実行するジョブステップ間の同一ファイルでは DISP="SHR,"を指定する必要がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、並列実行するジョブステップ間の同一ファイルでは DISP="SHR,"を指定するように修正する。または並列実行を終了して、ジョブを再実行する。

KAKC1153-E

TYPE="属性値" cannot be specified for sub step between parallely-executed job steps.
elementNo=番号

番号で示すタグの解析中に、TYPE="属性値"を指定した DD 要素が見つかったが、並列実行するジョブステップのサブステップには TYPE="属性値"の指定はできない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、並列実行するジョブステップのサブステップに TYPE="属性値"を指定しないよう修正して、ジョブを再実行する。

KAKC1154-W

COND attribute of sub step in parallely-executed job steps is ignored. elementNo=番号

番号で示すタグの解析中に、COND 属性を指定した EXEC 要素が見つかったが、無効な条件式を指定している可能性がある。

このメッセージは、並列実行するジョブステップの EXEC 要素の COND 属性に、次に示す条件式を指定した場合に出力される。

- この COND 属性を指定したジョブステップと並列に実行するジョブステップの名称を指定している。
- EVEN, ONLY, またはジョブステップの名称を指定しない条件式を指定している。

ただし、EVEN, ONLY, またはジョブステップの名称を指定しない条件式を指定した場合、この COND 属性を指定したジョブステップと並列実行しない先行ジョブステップが存在すれば、その条件式は無効とはならない。

(S)処理を続行する。

(O)ジョブ定義 XML ファイルを見直し、COND 属性に指定した条件式に誤りがないかどうか確認する。誤りがあればジョブ定義 XML ファイルを修正してジョブを再実行し、正しいジョブステップが実行されているかどうかを確認する。

KAKC1155-E

SYSIN record length is too long. elementNo=番号

番号で示すタグ内の SYSIN データのレコード長が長過ぎる。SYSIN データ中に、81 バイト目以降に空白・改行以外の文字が含まれている行がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、SYSIN データの各行の 81 バイト目以降に空白・改行以外の文字を含まないように修正する。修正後、ジョブを再実行する。

KAKC1156-E

属性名="属性値" can be specified only when パラメータのオペランド名 is specified for parameter パラメータ名. elementNo=番号

番号で示すタグ内に属性名="属性値"が指定されているが、属性名="属性値"は、設定ファイルのパラメータ名がパラメータのオペランド名の場合にしか指定できない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルまたは設定ファイルを見直し、ジョブを再実行する。

KAKC1157-E

Number of job steps using the PREST file exceeds the limit. elementNo=番号

番号で示すタグの解析中に PREST ファイルの組み合わせが上限を超えた。

ジョブ間 PREST ファイルは、入力側と出力側の関係が 1 対 1 になっている必要がある。ジョブ間 PREST ファイルを、並列実行するジョブステップ間に 3 つ以上指定しないこと。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、PREST ファイルの組み合わせが間違っていないか確認する。ジョブ間 PREST ファイルを並列実行するジョブステップ間に 3 つ以上指定しないように、ジョブ定義 XML ファイルを修正する。修正後、ジョブを再実行する。

KAKC1158-E

Number of job steps using the PREST file is not 2. elementNo=番号

番号で示すタグの解析中に、PREST ファイルの組み合わせの数が 2 でない PREST ファイルが存在した。

ジョブステップ間 PREST ファイルは、入力側と出力側の関係が 1 対 1 となっている必要がある。ジョブステップ間 PREST ファイルは、並列実行するジョブステップ間に 2 つ指定する必要がある。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、PREST ファイルの組み合わせが間違っていないか確認する。ジョブステップ間 PREST ファイルを並列実行するジョブステップ間に 2 つ指定するように、ジョブ定義 XML ファイルを修正する。修正後、ジョブを再実行する。

KAKC1159-E

The same temporary file cannot be used between parallely-executed job steps. elementNo=番号

番号で示すタグの一時ファイルは、並列実行するジョブステップに、一時ファイル識別名が同一の一時ファイル、一時 ISAM ファイルが指定されているため使用できない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、一時ファイル識別名を修正する。修正後、ジョブを再実行する。

KAKC1160-I

Specified DISP was changed to "SHR,KEEP,KEEP". elementNo=番号

番号で示すタグの DISP 属性値を"SHR,KEEP,KEEP"に変更した。

(S)処理を続行する。

(O)なし。

KAKC1161-E

PREST file is used by serially-executed job steps. elementNo=番号

番号で示すタグの PREST ファイルは、並列実行しないジョブステップ間で使用している。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、PREST ファイルの組み合わせ相手が間違っていないか確認する。必要な場合はジョブ定義 XML ファイルを修正する。修正後、ジョブを再実行する。

KAKC1162-E

PREST file is used within the same job step. elementNo=番号

番号で示すタグで表示している PREST ファイルは、同一ジョブステップ内で使用している。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、PREST ファイルの組み合わせ相手が間違っていないか確認する。必要な場合はジョブ定義 XML ファイルを修正する。修正後、ジョブを再実行する。

KAKC1164-E

Attribute "属性名" must be specified with LANG="LANG 属性値". elementNo=番号

番号で示すタグ内の解析中に属性名で示す属性が見つかったが、この属性は LANG="LANG 属性値"のときだけ指定できる。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、**属性名**で示す属性を削除、または LANG="LANG 属性値"を指定してジョブを再実行する。

KAKC1165-E

属性名="属性値" must not be specified with LANG="LANG 属性値". elementNo=番号

番号で示すタグ内の解析中に**属性名**="属性値"で示す属性が見つかったが、LANG="LANG 属性値"のときは指定することができない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、**属性名**="属性値", または LANG="LANG 属性値"を修正してジョブを再実行する。

KAKC1166-E

EURENCODE "エンコード種別" is not defined in Config File. elementNo=番号

指定された**エンコード種別**は設定ファイルで定義していない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルの指定が正しいかを見直す。

KAKC1167-W

Specified value "属性値" of attribute "属性名" is overridden to "適用属性値" for LANG="LANG 属性値". elementNo=番号

番号で示すタグ内の解析中に**属性名**="属性値"で示す属性値が見つかったが、LANG="LANG 属性値"の場合は属性値を指定しても**適用属性値**が仮定される。

(S)処理を続行する。

(O)ジョブ定義 XML ファイルを見直す。必要な場合は、**属性値**を変更するか、または正しい LANG="LANG 属性値"を指定してジョブを再実行する。

KAKC1168-E

DDNAME "DD 名" in SETENVFILE element is not found. elementNo=番号

番号で示す SETENVFILE 要素で指定した **DD 名**を定義する DD 要素が見つからない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、**DD 名**を正しく修正してジョブを再実行する。

KAKC1169-E

The TYPE attribute of the DD element to define the environment variable file is invalid.
elementNo=番号

番号で示す SETENVFILE 要素で指定した DD 名を定義する DD 要素の TYPE 属性が不当である。

(S)ジョブ定義 XML ファイルの解析処理後の処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、DD 要素を正しく修正してジョブを再実行する。

KAKC1175-I

Environment variable %{環境変数名} was replaced to "環境変数値".

環境変数名で示す XML ファイル環境変数の値を環境変数値に置換した。

(S)処理を続行する。

(O)なし。

KAKC1180-E

DISP="DISP 属性値" cannot be specified with TYPE="TYPE 属性値". elementNo=番号

番号で示すタグの DISP 属性値に DISP 属性値を指定しているが、TYPE="TYPE 属性値"と同時に指定することはできない。

(S)処理を続行する。

(O)ジョブ定義 XML ファイルを見直し、DISP 属性、または TYPE 属性に誤りがないかどうかを確認する。

KAKC1185-W

Specified procedure group is not defined in a configuration file, default group is assumed.

指定されたプロシジャグループは設定ファイルに定義されていないので、デフォルトのグループを仮定した。

(S)処理を続行する。

(O)デフォルトグループ名が仮定されて問題がある場合は、JOBPARM 要素の PROCGROUP 属性の値、または JP1 のキュー名が設定ファイルに定義されているか見直す。

KAKC1186-E

Specified step name is not found. "ジョブステップ名[.プロシジャステップ名]" elementNo=番号

番号で示すタグで指定したステップ名が見つからない、または指定できないジョブステップである。

ジョブステップ名[.プロシジャステップ名]：IF 要素の OP 属性で指定したジョブステップ名，プロシジャステップ名を表示する。カタログドプロシジャで指定したジョブステップ名はジョブステップ名.プロシジャステップ名の形式で表示する。

(S)処理を終了する。

(O)指定したジョブステップ名を見直す。

KAKC1187-E

Incorrect combination of the elements. elementNo=番号

番号で示すタグで指定した要素の組み合わせが正しくない。

(S)処理を終了する。

(O)対で指定が必要な要素の指定が不足していないかどうかを見直す。

KAKC1188-E

Incorrect order of the elements. elementNo=番号

番号で示すタグで指定した要素の指定順序が正しくない。

(S)処理を終了する。

(O)指定した要素の指定順序が正しいかどうかを見直す。

KAKC1189-E

This element can not be used in JSPEC. elementNo=番号

番号で示すタグで指定した要素は JSPEC 機能を使用した並列実行するジョブステップ間では使用できない。

(S)処理を終了する。

(O)指定した要素の指定位置が正しいかどうかを見直す。

KAKC1190-I

TYPE:"type 値" assumed in DD element of back reference. elementNo=番号

番号で示すタグの解析中に，逆方向参照を指定した DD 要素が見つかったため，TYPE 属性値を"type 値"に変更した。

(S)ジョブ定義 XML ファイルの解析処理を続行する。

(O)なし。

KAKC1191-E

TYPE is invalid in DD element of back reference between parallelly-executed job steps.
elementNo=番号

番号で示すタグの解析中、逆方向参照を指定した DD 要素が見つかったが、並列実行するジョブステップでは指定できない TYPE 属性値が指定された。

並列実行するジョブステップで逆方向参照する場合、参照先の TYPE 属性に"FILE"以外を指定してはならない。

(S)ジョブ定義 XML ファイルの解析処理後に処理を終了する。

(O)ジョブ定義 XML ファイルを見直し、逆方向参照の対象となる DD 要素の TYPE 属性を修正してジョブを再実行する。

KAKC1600-I

ジョブ名 Job allocated.

ジョブ名で示すジョブのファイル割り当てを開始した。

(S)処理を続行する。

(O)なし。

KAKC1601-I

ジョブ名.ジョブステップ名 Step allocated.

ジョブ名とジョブステップ名で示すジョブステップのファイル割り当てを開始した。

(S)処理を続行する。

(O)なし。

KAKC1602-I

DD 名, DISP 指定値, ファイル名

DD 名はファイル名で示すファイルを DISP 指定値として割り当てた。

(S)処理を続行する。

(O)なし。

KAKC1603-I

DD 名, DISP 指定値, ファイル名[, size= {ファイルサイズ | unknown}]

DD 名はファイル名で示すファイルを DISP 指定値として解放した。ファイルサイズを「ファイルサイズ」で示す (バイト単位)。ファイルサイズは DD 要素の TYPE 属性に"FILE", "TEMP", "GDG", "SYSOUT"を指定した場合に表示される。シンボリックリンクを指定した場合のファイルサイズは、シンボリックリンクに含まれるパス名の長さを表示する。DD 要素の DSN 属性に索引順編成ファイルを指定した場合や、ファイルが存在しないなどの理由でファイルサイズを求めることができない場合は、「unknown」を表示する。

(S)処理を続行する。

(O)なし。

KAKC1604-I

ファイルパス is deleted.

DISP 指定で"ファイルパス"に示すファイルを削除した。

(S)処理を続行する。

(O)なし。

KAKC1605-I

ファイルパス is not deleted due to "エラー詳細".

DISP 指定で"ファイルパス"を削除したが、"エラー詳細"に示すエラーが発生した。

(S)処理を続行する。

(O)なし。

KAKC1609-E

DD "DD 名" failed to {allocate | deallocate} temporary ISAM file.

DD 名で示す一時 ISAM ファイルの割り当てまたは割り当て解除に失敗した。

このメッセージは、設定ファイルに"ISAMLIB_MODE M64"を指定して一時 ISAM ファイルの割り当てまたは割り当て解除に失敗した場合に出力される。また、次のメッセージのどれかが stderr に出力される。

- KAKC0005-E
- KAKC1626-E
- KAKC1627-I

(S)メッセージテキストに"allocate"が表示されている場合、処理を終了する。メッセージテキストに"deallocate"が表示されている場合、処理を続行する。

(O)使用する環境で ISAM ライブラリの 64 ビット版の使用が適切であるか、実行環境は正しく設定されているかどうかを確認する。エラー要因を取り除いて必要ならばジョブを再実行する。

KAKC1610-E

```
DD "DD 名" failed to allocate.
```

DD 名で示すファイルの割り当てに失敗した。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルで指定したファイル名の指定を見直す。

KAKC1611-E

```
DD "DD 名" create failed. reason="エラー詳細"
```

DD 名で示すファイルを作成するためにオープンしたが、エラー詳細によって失敗した。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルを見直す。

KAKC1612-E

```
DD "DD 名" putenv failed.
```

DD 名で示す DD 要素または SETENV 要素の環境変数の作成に失敗した。

"DD 名"には、DD 要素または SETENV 要素で指定した NAME 属性に指定した名称が表示される。

DD 要素で TYPE="LIB"属性指定の場合は、"DD 名"に"PATH"と表示される。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルを見直す。

KAKC1613-E

```
DD "DD 名" file is already exists.
```

DD 名で示すファイルはすでに存在する。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルを見直す。

KAKC1614-E

DD "DD 名" file does not exist.

DD 名で示すファイルが存在しない。またはサブステップで PASS したファイルを後続ジョブステップへ引き継ごうとした。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルを見直す。

KAKC1615-E

DD "DD 名" stat error. reason="エラー詳細".

DD 名で示すファイルを stat 関数で確認しようとしたが、エラー詳細で示すエラーが発生した。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルを見直す。

KAKC1616-E

The DSN of DD "DD 名" is already allocated directory.

DD 名で示すディレクトリはすでに存在する。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルを見直す。

KAKC1617-E

The DSN of DD "DD 名" is already allocated file.

DD 名で示すディレクトリはすでに存在する。または指定したディレクトリではない可能性がある。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルを見直す。

KAKC1618-E

DD "DD 名" create error reason="エラー詳細".

DD 名で示すファイルを作成しようとしたが、エラー詳細で示すエラーが発生した。

DD 名で示すファイルがシンボリックリンクの場合、ファイルの実体が存在しないかマウントされていないと、エラー詳細に"File exists"が表示される。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルを見直す。

KAKC1619-E

DD "DD 名" environment variable PATH exceeded the maximum.

DD 名で示す JOBLIB, STEPLIB を PATH 環境変数に設定しようとしたが、設定する値が長過ぎ、または存在しないパス（ディレクトリ）を指定した可能性がある。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルを見直す。

KAKC1620-E

"RNW" is not specified.

ディレクトリは RNW 指定のため動作できない。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルを見直す。

KAKC1625-E

An error occurred during SYSIN data output processing. reason="エラー詳細".

入力データの書き込み処理で、"エラー詳細"に示すエラーが発生した。

(S)処理を終了する。

(O)エラー詳細の原因を取り除いて再実行する。

KAKC1626-E

Failed to create temporary ISAM file defined by DD "DD 名". (errno=エラー番号,function=ISAM 関数名,file name=一時 ISAM ファイル名)

DD 名で示す一時 ISAM ファイルの作成に失敗した。エラーが発生した ISAM 関数名, ISAM 関数のエラー番号 (iserrno) および一時 ISAM ファイル名が出力される。

(S)処理を終了する。

(O)マニュアル「索引順編成ファイル管理 ISAM」を参照し、ISAM 関数名, エラー番号からエラー原因を求め、取り除く。システムコールのエラー番号の場合はエラー情報を調べ、原因を取り除く。

KAKC1627-I

Failed to delete temporary ISAM file "一時 ISAM ファイル名". (errno=エラー番号,function=ISAM 関数名)

一時 ISAM ファイル名で示す一時 ISAM ファイルの削除に失敗した。エラーが発生した ISAM 関数名および ISAM 関数のエラー番号 (iserrno) が出力される。

(S)処理を続行する。

(O)ジョブ終了後、一時 ISAM ファイルを格納するディレクトリ内を確認し、一時 ISAM ファイル名で示す一時 ISAM ファイルが存在する場合は削除する。なお、一時 ISAM ファイルは次に示す物理ファイルで構成される。

- キー定義ファイル
一時 ISAM ファイル名で示す名称に拡張子".DEF"が付加された名称である。
- キーファイル
一時 ISAM ファイル名で示す名称に拡張子".K01"が付加された名称である。
- データファイル
一時 ISAM ファイル名で示す名称に拡張子".DAT"が付加された名称である。

KAKC1628-E

Failed to create temporary ISAM file. File already exists. (DD name=DD 名称,file name=一時 ISAM ファイル名)

一時 ISAM ファイル名で示す一時 ISAM ファイルがすでに存在しているため、一時 ISAM ファイルの作成に失敗した。

(S)処理を終了する。

(O)一時 ISAM ファイルを格納するディレクトリ内を確認し、一時 ISAM ファイル名で示す一時 ISAM ファイルが存在する場合は削除して再実行する。なお、一時 ISAM ファイルは次に示す物理ファイルで構成される。

- キー定義ファイル
一時 ISAM ファイル名で示す名称に拡張子".DEF"が付加された名称である。
- キーファイル
一時 ISAM ファイル名で示す名称に拡張子".K01"が付加された名称である。
- データファイル
一時 ISAM ファイル名で示す名称に拡張子".DAT"が付加された名称である。

KAKC1629-E

```
DD "DD 名" 属性名 | オペランド名" putenv failed.
```

DD 名で示す DD 要素の属性名で示す属性値、または設定ファイルの SysoutClass パラメータで指定したオペランド名で示すオペランドで指定した値を設定する環境変数の作成に失敗した。

"DD 名"には、DD 要素で指定した NAME 属性に指定した名称が表示される。

属性名またはオペランド名の表示を次に示す。

FTYPE：SYSOUT データファイルのファイル形式

RECFM：制御文字の種類

TRUNC：行末空白削除

(S)処理を終了する。

(O)ジョブ定義 XML ファイルまたは設定ファイルを見直す。

KAKC1630-E

```
DD "DD 名" truncate failed. reason="エラー詳細"
```

DD 名で示すファイルを 0 バイトにしようとしたが、エラー詳細によって失敗した。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルを見直す。

KAKC1640-I

```
DD "DD 名", DISP: "disp 変更値" assumed.
```

DISP 属性値が不当なため、disp 変更値に変更した。

(S)処理を続行する。

(O)なし。

KAKC1641-I

```
DD "DD 名", DISP: "disp 変更値" assumed because of "パラメータ名".
```

DISP 属性値をパラメータ名の設定によって disp 変更値に変更した。

(S)処理を続行する。

(O)なし。

KAKC1648-E

DD "DD 名" must contain all of attributes "ISKEY", "ISRECFM" and "ISRECL" because DISP "MOD" was changed to "NEW".

DD 名で示す一時 ISAM ファイルの DD 要素は、DISP 属性を"MOD"から"NEW"に変更したため、ISKEY、ISRECFM、および ISRECL 属性がすべて指定されている必要がある。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルを見直す。

KAKC1650-W

Waiting for locking a file.
ロック確保待ちが発生しました

ファイルのロック確保待ちが発生した。ロック確保待ちの状態は、bjexlslock コマンドで確認できる。

(S)ファイルのロックが確保されるまでジョブを待たせる。

(O)ロック待ち時間が長時間となっている場合、ジョブスケジュールの見直しをする。

KAKC1651-I

Quit waiting for locking a file.
ロック待ちは解除されました

KAKC1650-W で示すファイルのロック確保待ちは解除された。

(S)処理を続行する。

(O)なし。

KAKC1652-W

Waiting for locking a semaphore.
ジョブ実行待ちが発生しました

ジョブの実行数が設定ファイルの LOCK_JOBMPLEX で指定したジョブ多重度を超えたため、ジョブの実行待ちを行った。

(S)ジョブの実行数がジョブ多重度より小さくなるまでジョブを待たせる。

(O)ジョブの実行待ち時間が長時間となっている場合、ジョブスケジュールの見直しをする。

KAKC1653-I

Quit waiting for locking a semaphore.

ジョブ実行待ちは解除されました

KAKC1652-W で示すジョブ多重度によるジョブの実行待ちを解除した。

(S)処理を続行する。

(O)なし。

KAKC1654-W

Waiting for lock resource control table.

排他資源管理テーブルの空き待ちが発生しました

ロック獲得中またはロック待ち中のファイル合計数が、設定ファイルの LOCK_MAXFILE で指定したファイル数を超えたため、排他資源の空き待ちが発生した。

(S)排他資源管理テーブルに空きができるまでジョブを待たせる。

(O)設定ファイルの LOCK_MAXFILE の指定を見直す。

KAKC1655-I

Quit waiting for lock resource control table.

排他資源管理テーブルの空き待ちが解除されました

KAKC1654-W で示す排他資源管理テーブルの空き待ちは解除された。

(S)処理を続行する。

(O)なし。

KAKC1660-E

DD "DD 名" does not exist. file path=ファイルパス

DD 名で示す DD 要素で指定したファイルパスのファイルが存在しない。

(S)処理を終了する。

(O)ファイル名の指定を見直す。

KAKC1680-I

DD "DD 名" Specified DISP "MOD" was changed to "NEW" because file does not exist.

DD 名で示すファイルは DISP 属性値に"MOD"を指定しているが、ファイルが存在しないため NEW に変更した。

(S)処理を続行する。

(O)なし。

KAKC1690-I

DD "DD 名" was truncated to 0 byte.

DD 名で示すファイルを削除しないでファイルを 0 バイトにした。

(S)処理を続行する。

(O)なし。

KAKC1691-E

DD "DD 名" referenced DD element is not allocated.

DD 名で示す DD 要素で参照している DD 要素は割り当て未完了であるため参照できない。

(S)処理を終了する。

(O)参照先の DD 要素が正しいかを見直し、割り当て済みの DD 要素を参照するようにジョブ定義 XML ファイルを修正してジョブを再実行する。

KAKC1717-E

There is no running job.

実行中のジョブは存在しない。

(S)処理を終了する。

(O)指定を確認し、必要な場合はコマンドを再入力する。

KAKC1718-E

No resource is allocated with the specified file or path name.

指定されたファイルパス名での資源確保は行われていない。

(S)処理を終了する。

(O)指定を確認し、必要な場合はコマンドを再入力する。

KAKC1719-E

Specified file or path name is not found.

表示対象となるファイル名およびパス名が見つからない。

(S)処理を終了する。

(O)指定を確認し、必要な場合はコマンドを再入力する。

KAKC1720-E

Failed to acquire a semaphore set identifier. (code=保守情報,error message=エラー詳細)

ファイルの排他/共用制御または実行中のジョブ情報表示機能の処理中にセマフォ集合識別子の取得に失敗した。

このメッセージが出力された場合、次に示すことが考えられる。

bjexec 実行で表示した場合

bjexec 実行するのに先立ち、bjexpd の起動が行われていない。

bjexlslock 実行で表示した場合

bjexlslock 実行するのに先立ち、bjexpd の起動が行われていない。または排他制御機能を使用するための設定ファイルの指定 (LOCK_OPTION パラメータで ENABLE を指定) が行われていない。

bjexlsexjob 実行で表示した場合

bjexlsexjob 実行するのに先立ち、bjexpd の起動が行われていない。または実行中のジョブ情報表示機能を使用するための設定ファイルの指定 (JOB_STATUS_DISPLAY パラメータで ENABLE を指定) が行われていない。

(S)処理を中止する。

(O)コマンド実行に先立ち、bjexpd を起動しておく。

排他制御機能を使用する場合、設定ファイルの指定 (LOCK_OPTION パラメータで ENABLE を指定) をする。

実行中のジョブ情報表示機能を使用する場合、設定ファイルの指定 (JOB_STATUS_DISPLAY パラメータで ENABLE を指定) をする。

上記以外の要因の場合、システム管理者に連絡する。

KAKC1721-E

Failed to create a semaphore set. (code=保守情報,error message=エラー詳細)

ファイルの排他/共用制御または実行中のジョブ情報表示機能の処理中にセマフォ集合の生成に失敗した。

(S)処理を中止する。

(O)システム管理者に連絡する。

KAKC1722-E

An error occurred during initialization of a semaphore. (code=**保守情報**,error message=**エラー詳細**)

ファイルの排他/共用制御または実行中のジョブ情報表示機能の処理中にセマフォの初期化処理でエラーが発生した。

(S)処理を中止する。

(O)システム管理者に連絡する。

KAKC1723-E

An error occurred during initialization of the shared memory. (code=**保守情報**,error message=**エラー詳細**)

ファイルの排他/共用制御または実行中のジョブ情報表示機能の処理中に共用メモリの初期化処理でエラーが発生した。

(S)処理を中止する。

(O)システム管理者に連絡する。

KAKC1724-E

An error occurred while locking a file. (code=**保守情報**,error message=**エラー詳細**,elementNo=**番号**)

番号で示すタグの DD 要素で、ファイルのロック確保処理中にエラーが発生した。

(S)処理を中止する。

(O)DSN 属性で指定したパス名に誤りがないか見直す。

KAKC1725-E

An error occurred while unlocking a file. (code=**保守情報**,error message=**エラー詳細**)

ファイルのロック解除処理中にエラーが発生した。

(S)処理を中止する。

(O)システム管理者に連絡する。

KAKC1726-E

Access to the shared memory occurred while terminating the exclusion control function.

排他/共用制御の計画停止処理中のため、共用メモリへのアクセスは禁止されている。

(S)ジョブの実行を中止する。

(O)排他/共用制御が使用可能となるまでジョブの入力を延期するか、または排他/共用制御を使用可能にしてからジョブを入力する。

KAKC1727-E

More than two PREST files were used at the same time.

PREST ファイルの共用使用数が 2 を超えた。

(S)ジョブの実行処理を終了する。

(O)共用できるジョブの多重実行となるように、実行スケジュールを見直す。

KAKC1728-E

Number of files for exclusion control exceeds the limit.

この実行環境で、一度に管理できないファイル数を指定している。

(S)処理を中止する。

(O)一度に扱う排他対象ファイル数が多過ぎる。または環境設定値が小さ過ぎるため、指定値を見直して再実行するか、もしくは環境設定値を大きくする。

KAKC1729-E

Length of file path exceeds 1023 characters.

指定したファイルパスは 1,023 文字を超えている。

(S)コマンド処理を中止する。

(O)指定を見直し、必要な場合はコマンドを再入力する。

KAKC1731-E

Duplicate error was detected while checking host name of the shared memory. (ホスト名)

ファイルの排他/共用制御または実行中のジョブ情報表示機能の処理中、共用メモリのホスト名の確認で、他プロセスのものと重複しているのが検出された。物理ホストで実行したときは、ホスト名に物理ホスト名が表示される。

(S)処理を終了する。

(O)システム管理者に連絡する。

システム管理者は、共有メモリが重複しないよう LOCK_KEY_ID パラメータで、識別情報を設定する。

KAKC1732-E

Duplicate error was detected while checking identifier of the shared memory. (ホスト名)

ファイルの排他/共用制御または実行中のジョブ情報表示機能の処理中、共有メモリの識別情報の確認で、他プロセスのものと重複しているのが検出された。物理ホストで実行したときは、ホスト名に物理ホスト名が表示される。

(S)処理を終了する。

(O)システム管理者に連絡する。

システム管理者は、共有メモリが重複しないよう LOCK_KEY_ID パラメータで、識別情報を設定する。

KAKC1733-E

Duplicate error was detected while checking spool filepath of the shared memory.(ホスト名)

ファイルの排他/共用制御または実行中のジョブ情報表示機能の処理中、共有メモリのスプールのファイル名の確認で、他プロセスのものと重複しているのが検出された。物理ホストで実行したときは、ホスト名に物理ホスト名が表示される。

セマフォ、共有メモリの確保時に内部的に使用するキーが重複しないように、キー生成時に使用する識別情報をあらかじめ重複しないように指定する。

(S)処理を終了する。

(O)システム管理者に連絡する。

システム管理者は、共有メモリが重複しないよう LOCK_KEY_ID パラメータで、識別情報を設定する。

KAKC1734-E

An error occurred during lock release processing. (function=関数名, error="エラー詳細")

ジョブステップ終了時の排他/共用制御の解除処理で、エラーが発生した。

スプールジョブ転送デーモンが停止した可能性がある。

(S)ジョブステップ終了時の排他/共用制御の解除処理を行わず、以降のジョブステップの実行を打ち切る。

(O)システム管理者に連絡する。システム管理者は、スプールジョブ転送デーモンの状態を確認する。

KAKC1800-I

Environment variable file is applied. Enviroment variable count=**環境変数数**

SETENVFILE 要素で指定した環境変数格納ファイルを適用した。環境変数数で、適用した**環境変数数**を示す。

(S)処理を続行する。

(O)なし。

KAKC1801-E

An error occurred during environment variable file processing. (file="**ファイル名**", function=**関数名**, error=**エラー詳細**)

SETENVFILE 要素で指定した環境変数格納ファイル処理中にエラーが発生した。

(S)ジョブの実行を中止する。

(O)エラー詳細の原因を取り除いて再実行する。

KAKC1900-E

Step program is not found.

ジョブステップを実行しようとしたが、EXEC 要素で指定したプログラムが見つからなかった。

(S)処理を続行する。

(O)ジョブ定義 XML ファイルを確認する。

KAKC1901-E

Step program could not be executed. reason=**エラー詳細**

ジョブステップを実行しようとしたが、**エラー詳細**で示すエラーが発生し、EXEC 要素で指定したプログラムを実行できなかった。

(S)処理を続行する。

(O)**エラー詳細**の原因を取り除いて再実行する。

KAKC1902-W

Step program stopped by SIGSTOP.

ジョブステップで実行中のプログラムが、SIGSTOP シグナルによって休止した。

(S)処理を続行する。

(O)実行中のプログラムを確認する。

KAKC1903-W

Send SIGKILL to process group of step. step=ジョブステップ名[.プロシジャステップ名]

ジョブステップのプロセスグループに対して SIGKILL シグナルを送り、処理を打ち切った。

(S)次のどちらかの場合、ジョブを打ち切る。

- JOB 要素の COND 属性の条件を満たしたため、JOBPARM_JSPECAN パラメータによるジョブステップの打ち切りを行う場合
- JP1/AJS からジョブを強制終了した場合

上記以外の場合、ジョブステップの打ち切り後にジョブを続行する。

(O)なし。

KAKC1904-E

Failed to create step process. reason=エラー詳細

ジョブステップのプログラムを実行したとき、エラー詳細で示すエラーが発生し、プロセスを生成できなかった。

(S)処理を終了する。

(O)エラー詳細の原因を取り除いて再実行する。

KAKC1905-E

Failed to wait for step process. pid=ジョブステップ PID, reason=エラー詳細

ジョブステップのプログラムの終了コードを取得しようとしたとき、エラー詳細で示すエラーが発生し、プロセスの完了を待つことができなかった。

(S)処理を終了する。

(O)エラー詳細の原因を取り除いて再実行する。

KAKC1906-I

ジョブ名.ジョブステップ名[.プロシジャステップ名] The value of COND attribute of JOB element was met.

JOB 要素の COND 属性で指定した条件が成立したため、後続のすべてのジョブステップをスキップした。

(S)処理を続行する。

(O)なし。

KAKC1907-I

ジョブ名.ジョブステップ名[.プロシジャステップ名] Step was skipped because the value of COND attribute of EXEC element was met.

EXEC 要素の COND 属性で指定した条件が成立したため、ジョブステップをスキップした。ジョブステップで実行したユーザプログラムが異常終了すると、後続ジョブステップのうち COND 属性に EVEN または ONLY が指定されていないジョブステップをスキップする。この場合、スキップするジョブステップに対して、COND 属性の指定有無に関わらずこのメッセージが出力される。

(S)処理を続行する。

(O)なし。

KAKC1908-W

ジョブ名 The maximum value of return code exceeds 240.

ジョブステップの終了コードの最大値が 240 を超えた。

このメッセージは、JOB_RCDEFINE パラメータに STEPMAX を指定した場合、または JOB_RCDEFINE パラメータの指定を省略した場合にだけ出力される。

(S)処理を続行する。

(O)このコードで問題ないか確認する。

KAKC1909-I

ジョブ名 The maximum value of return code was changed by SETMAXRC element. rc=終了コード

終了コードが SETMAXRC 要素によって書き換えられた。

(S)処理を続行する。

(O)なし。

KAKC1910-E

Invalid C-Time.

C-Time が不正である。

(S)処理を続行する。

(O)システム管理者に連絡する。

KAKC1911-E

Invalid E-Time.

E-Time が不正である。

(S)処理を続行する。

(O)システム管理者に連絡する。

KAKC1912-W

Failed to copy a user message. (file=ファイル名, function=保守情報, error=エラー内容)

ジョブログに出力する処理で障害が発生したため、アプリケーションが出力したユーザからのメッセージをジョブログに出力できなかった。考えられる要因を次に示す。

- ディスクの容量が満杯になったため、ジョブログに出力できなかった。
- スプールディレクトリ内のファイルまたはディレクトリに対して不当な操作を行ったため、ジョブログに出力できなかった。

(S)処理を続行する。

(O)アプリケーションが出力したユーザからのメッセージをジョブログに出力する必要がある場合、ディスクの容量またはスプールディレクトリの書き込み権限などを確認し、ジョブを再実行する。

KAKC1915-I

Job ID of Java application was assigned. Job ID=Cosminexus の Java バッチアプリケーションのジョブ ID

バッチアプリケーションへジョブ ID を割り当てた。

(S)処理を続行する。

(O)なし。

KAKC1917-I

ジョブ名.ジョブステップ名[.プロシジャステップ名] Step was skipped because the condition of IF element(要素番号) was met.

IF 要素で指定した条件が成立したため、ジョブステップをスキップした。

要素番号：要因となった IF 要素の要素番号

(S)処理を続行する。

(O)なし。

KAKC1918-W

ジョブ名 The last value of return code exceeds 240.

最後に実行完了したメインステップの終了コードが 240 を超えた。

このメッセージは、JOB_RCDEFINE パラメータに STEPLAST を指定した場合にだけ出力される。

(S)処理を続行する。

(O)このコードで問題ないか確認する。

KAKC1922-E

Failed to terminate step process. (step=ジョブステップ名[.プロシジャステップ名], function=関数名, detail=詳細情報)

ジョブステップ名で示すジョブステップのプロセスを強制終了しようとしたが、エラーが発生したため強制終了できなかった。

(S)処理を終了する。

KAKC1923-W

Step process was terminated. (step=ジョブステップ名[.プロシジャステップ名])

ジョブステップ名で示すジョブステップのプロセスを強制終了した。

(S)処理を終了する。

(O)なし。

KAKC1932-I

ジョブ名.ジョブステップ名[.プロシジャステップ名] PGM_RCABEND parameter is applied. rc=終了コード

ジョブステップが終了コードで終了したため、PGM_RCABEND パラメータが適用された。ジョブステップは異常終了となる。

(S)処理を続行する。

(O)ジョブステップの実行結果に問題がないか確認する。

KAKC1942-W

Send シグナル名 to プログラム名 process group of step. step=ジョブステップ名[.プロシジャステップ名]

ジョブステップのプログラム名のプロセスグループに対してシグナル名で示すシグナルを送り、処理を打ち切った。

(S)次のどちらかの場合、ジョブを打ち切る。

- JOB 要素の COND 属性の条件を満たしたため、JOBPARM_JSPECAN パラメータの指定によってジョブステップを終了する場合
 - JP1/AJS からジョブを強制終了した場合
- 上記以外の場合、ジョブステップを終了してからジョブを続行する。

(O)なし。

KAKC1943-W

Parallely-executed steps are terminated. reason=理由

JOBPARM_JSPECAN パラメータによって理由で示す理由で実行中のジョブステップを終了させる。

理由	意味
error during preliminary processing	アロケートエラーやプロセス生成エラーなど、UAP プロセス生成前のエラー
abnormal termination	シグナルによるジョブステップの異常終了
COND attribute of JOB element in effect	JOB 要素の COND 属性によるジョブの打ち切り

(S)JOB 要素の COND 属性による打ち切りの場合、処理を終了する。それ以外の場合、処理を続行する。

(O)必要であれば、理由やこのメッセージの前後に実行ログに出力されたメッセージから原因を調べ、取り除き、ジョブを再実行する。

KAKC1950-E

bjexec support program is not found.

bjexec から起動指示されたサポートプログラムが見つからなかった。

(S)処理を続行する。

(O)バッチジョブ実行システムが正しくインストールされているかどうかを確認する。

KAKC1951-E

bjexec support program could not be executed. reason= エラー詳細

bjexec から起動指示されたサポートプログラムは、エラー詳細で示すエラーが発生し、実行できなかった。

(S)処理を続行する。

(O)エラー詳細の原因を取り除いて再実行する。実行環境が適切ではない場合（64 ビットアプリケーションを 32 ビット環境で実行しようとしたなど）は、設定ファイルの定義を見直す。

KAKC2201-E

The message is too long. msgid=メッセージ番号

メッセージ番号（KAKC に続く番号）に示すメッセージは、テキストが長いのですべてのテキストを出力できない。

(S)処理を続行する。

(O)システム管理者に連絡する。

KAKC2202-E

Output to JOBLOG failed. msgid=メッセージ ID

メッセージ ID に示すメッセージは、ジョブログへの出力を失敗した。

(S)処理を続行する。

(O)システム管理者に連絡し、設定ファイルのスパールディレクトリに対する指定を見直す。

KAKC2203-E

Output to SYSMSG failed. msgid=メッセージ ID

メッセージ ID に示すメッセージは、システムメッセージへの出力を失敗した。

(S)処理を続行する。

(O)システム管理者に連絡し、設定ファイルのスパールディレクトリに対する指定を見直す。

KAKC2204-E

Output to stdout failed. msgid=メッセージ ID

メッセージ ID に示すメッセージは、標準出力への出力を失敗した。

(S)処理を続行する。

(O)システム管理者に連絡し、設定ファイルのスプールディレクトリに対する指定を見直す。

KAKC2205-E

Output to stderr failed. msgid=メッセージ ID

メッセージ ID に示すメッセージは、標準エラー出力への出力を失敗した。

(S)処理を続行する。

(O)システム管理者に連絡し、設定ファイルのスプールディレクトリに対する指定を見直す。

KAKC2206-E

Output to log failed. msgid=メッセージ ID

メッセージ ID に示すメッセージは、実行ログへの出力を失敗した。

(S)処理を続行する。

(O)システム管理者に連絡し、設定ファイルの実行ログに対する指定を見直す。

KAKC2207-E

Output to trace failed. msgid=メッセージ ID

メッセージ ID に示すメッセージは、トレースへの出力を失敗した。

(S)処理を続行する。

(O)システム管理者に連絡し、設定ファイルのトレースに対する指定を見直す。

KAKC2208-E

Initialization of log failed. code=保守情報, reason=エラー原因

実行ログの初期化に失敗した。

(S)次のとおりとなる。

- ジョブコントローラの場合、処理を続行する。起動したジョブでは実行ログへメッセージを出力しない。
- スプールジョブ転送デーモンの場合、処理を終了する。

(O)コマンドやジョブを再実行する。何度か再実行しても同一現象になる場合は、システム管理者に連絡し、設定ファイルの実行ログに対する指定を見直す。

KAKC2209-E

Initialization of trace failed. code=**保守情報 1**, os_code=**保守情報 2**

トレースの初期化に失敗した。

(S)次のとおりとなる。

- ジョブコントローラの場合、処理を続行する。起動したジョブではトレースへメッセージを出力しない。
- スプールジョブ転送デーモンの場合、処理を終了する。

(O)コマンドやジョブを再実行する。何度か再実行しても同一現象になる場合は、システム管理者に連絡し、設定ファイルのトレースに対する指定を見直す。保守情報 2 が 11,13 の場合、別プロセスがトレースを排他していることが原因のため、排他しているプロセスを見直す。

KAKC2211-E

The file size or the number of files used by trace is invalid.

トレースの設定に誤りがある。

(S)次のとおりとなる。

- ジョブコントローラの場合、処理を続行する。起動したジョブではトレースへメッセージを出力しない。
- スプールジョブ転送デーモンの場合、処理を終了する。

(O)コマンドやジョブを再実行する。何度か再実行しても同一現象になる場合は、システム管理者に連絡し、設定ファイルのトレースに対する指定を見直す。

KAKC2213-E

The file path used by trace is invalid.

トレースの出力先の設定に誤りがある。

(S)次のとおりとなる。

- ジョブコントローラの場合、処理を続行する。起動したジョブではトレースへメッセージを出力しない。
- スプールジョブ転送デーモンの場合、処理を終了する。

(O)コマンドやジョブを再実行する。何度か再実行しても同一現象になる場合は、システム管理者に連絡し、設定ファイルのトレースに対する指定を見直す。

KAKC2214-E

Failed to get the current time.

現在時刻を求めることができない。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC2215-E

Failed to open log. code=保守情報 1, reason=エラー原因, os_code=保守情報 2

実行ログのオープンに失敗した。

(S)実行ログのオープンをリトライする。10回リトライしてもオープンできなかった場合、次のとおりとなる。

- ジョブコントローラの場合、処理を続行する。起動したジョブでは実行ログへメッセージを出力しない。
- スプールジョブ転送デーモンの場合、処理を終了する。

(O)コマンドやジョブを再実行する。何度か再実行しても同一現象になる場合は、システム管理者に連絡する。保守情報 2 が 11,13 の場合、別プロセスが実行ログを排他していることが原因のため、排他しているプロセスを見直す。

KAKC2300-E

Initialization of JP1Event failed. function=関数名()

JP1 イベントの初期化処理でエラーが発生した。

(S)処理を続行する。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2301-E

Send processing of JP1Event failed. function=関数名()

JP1 イベントの送信処理でエラーが発生した。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2302-E

```
Termination processing of JP1Event failed. function=関数名()
```

JP1 イベントの終了処理でエラーが発生した。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2303-E

```
The event server of JP1/Base is not running.
```

JP1 が稼働していないため、イベントが送れない。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2310-E

```
Parameter is invalid.
```

JP1 イベントの処理で不正な情報を設定した。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2311-E

```
The library of JP1Event is not found.
```

JP1 のイベントを発行しようとしたが、JP1/Base のプログラムが見つからない。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2312-E

Memory is insufficient.

JP1 のイベントを発行しようとしたが、メモリが不足している。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。

(O)システム管理者に連絡する。

KAKC2313-E

A system error occurred.

JP1 のイベントを発行しようとしたが、システムエラーが発生した。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2314-E

Initialization is not completed.

JP1 のイベントを発行しようとしたが、JP1 とのインタフェースが初期化されていない。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2315-E

Failed to connect to the event service.

JP1 のイベントを発行しようとしたが、イベントサービスに接続できない。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2316-E

An I/O error occurred.

JP1 のイベントを発行しようとしたが、入出力エラーが発生した。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2317-E

The number of opened files exceeds the maximum.

JP1 のイベントを発行しようとしたが、オープンしているファイルが多く、イベントが発行できない。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2318-E

The specified attribute has already been registered.

JP1 イベントに設定しようとした拡張属性が重複した。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。なお、JP1 イベントは発行するが、設定されない拡張属性がある。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2319-E

The specified attribute is not registered.

JP1 イベントに定義されていない拡張属性を設定した。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。JP1 イベントは発行するが、設定されない拡張属性がある。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2320-E

The number of specified extended attributes exceeds the maximum.

JP1 イベントに設定できる拡張属性数を超えている。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。JP1 イベントは発行するが、設定されない拡張属性がある。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2321-E

The total length of the extended attributes exceeds the maximum.

JP1 イベントに設定できる拡張属性の容量を超えている。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。JP1 イベントは発行するが、設定されない拡張属性がある。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2322-E

The sequence of functions is invalid.

JP1 のイベントを発行しようとしたが、イベント発行手順を誤っている。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を続行する。

(O)システム管理者に連絡する。JP1/Base 側に問題がないか確認する。

KAKC2400-E

```
Initialization is not completed. output=出力先 msgid=メッセージ ID dest=保守情報 1 setDest=保守情報 2
```

メッセージ ID に示すメッセージの出力を行ったが、初期化されていないため出力できない。

(S)処理を続行する。

(O)システム管理者に連絡する。

KAKC2499-E

```
Message id "メッセージ番号" is not defined.
```

メッセージ番号で示すメッセージはマニュアルに存在しない。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC3000-I

```
The bjexpd started. [The .hold file was initialized.] [The old pid was PID.]  
bjexpd が起動しました
```

スプールジョブ転送サービスが起動した。syslog への出力時には、KAKC3007-W, KAKC3008-W で示されるメッセージと同じ詳細メッセージが付与されることがある。

(S)処理を続行する。

(O)なし。

KAKC3001-I

```
The bjexpd stopped.  
bjexpd が終了しました
```

スプールジョブ転送サービスが終了した。

(S)処理を終了する。

(O)なし。

KAKC3002-E

An attempt to start bjexpd failed. [The bjexpd could not start because another one is already running.]

bjexpd が起動できません

スプールジョブ転送サービスの起動に失敗した。syslog への出力時には、KAKC3009-E で示されるメッセージと同じ詳細メッセージが付与されることがある。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生のタイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を終了する。

(O)原因を取り除き、再起動する。

KAKC3003-E

Error was detected while stopping the bjexpd.

bjexpd が異常終了しました

スプールジョブ転送サービスが異常終了した。または、終了処理中にエラーが発生した。

(S)処理を終了する。

(O)原因を取り除き、再起動する。

KAKC3004-E

An error occurred during JP1/FTP processing.

bjexpd でJP1/FTP エラーが発生しました

スプールジョブの転送時、転送エラーが発生した。

(S)JP1FTP_RETRY_INTERVAL パラメータで指定された時間を待ったあと、転送処理をリトライする。

処理を続行する。ただし、エラー状態から回復するまでの間、転送は行われない。

(O)このメッセージの前後に実行ログに出力されたメッセージから原因を調査して、エラー原因を取り除く。

KAKC3006-I

The bjexpd will now start. (pid=PID,uid=UID,gid=GID,username=ユーザ名)

bjexpd が起動を開始した。

(S)処理を続行する。

(O)なし。

KAKC3007-W

```
The .hold file was initialized.
```

スプール管理ファイルを再作成した。

(S)処理を続行する。

(O)bjexchjob コマンドを使用して、保存してあるスプールジョブの再登録をする。

KAKC3008-W

```
The old pid was PID.
```

前回 bjexpd が異常終了時の pid を表示する。

(S)処理を続行する。

(O)なし。

KAKC3009-E

```
The bjexpd could not start because another one is already running.
```

bjexpd はすでに起動している。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を終了する。

(O)必要であればすでに起動している bjexpd を止めてコマンドを再入力する。

KAKC3010-I

```
File transfer of spool job data file ended successfully. (転送開始日,転送開始時刻,転送終了日,転送終了時刻,ユーザ名,ジョブ名,ジョブ識別子,DD名,出力クラス,DEST名,サイズ,転送先ファイル名称)
```

SYSOUT データファイルの転送に成功した。

(S)処理を続行する。

(O)なし。

KAKC3011-I

The transfer file information: (転送開始日,転送開始時刻,転送終了日,転送終了時刻,trno=伝送番号,trcno=接続番号,size=サイズ,cmpsize=圧縮サイズ)

スプールジョブの転送に成功した。

(S)処理を続行する。

(O)なし。

KAKC3012-E

File transfer of spool job data file failed. (転送開始日,転送開始時刻,転送終了日,転送終了時刻,ユーザ名,ジョブ名,ジョブ識別子,DD名,出力クラス,DEST名,サイズ,転送先ファイル名称)

SYSOUT データファイルの転送に失敗した。

(S)処理を続行する。

(O)転送先サーバが起動されていない, またはビジー状態であるなどの可能性がある。エラー情報を調べ, 原因を取り除く。bjexchjob コマンドで再度印刷指示にする。

KAKC3013-E

The transfer error file information: (転送開始日,転送開始時刻,転送終了日,転送終了時刻,trno=伝送番号,trcno=接続番号,kind=エラー種別,place=場所,func=モジュール,system=システムコール名称,errno=エラー番号,msg=メッセージ)

スプールジョブの転送に失敗した。

(S)処理を続行する。

(O)転送先サーバが起動されていない, またはビジー状態であるなどの可能性がある。

伝送番号以下で示す転送に関するエラー情報と JPI/FTP の履歴情報 (ftshistory コマンドなどで表示可) からエラーの要因を調査し取り除く。JPI/FTP の履歴情報の取得, 表示方法や表示内容については, JPI のマニュアルを参照のこと。bjexchjob コマンドで再度印刷指示にする。

KAKC3019-E

The specified JPI/FTP card name was not found. (伝送カード名)

スプールジョブの転送に使用する伝送カードが登録されていない。

(S)処理を続行する。

(O)エラー情報を調べ, 原因を取り除く。bjexchjob コマンドで再度印刷指示にする。

KAKC3020-E

An error occurred during file processing. (機能名 error 対象名-エラー詳細)

スプールジョブ転送サービスでファイル操作エラーが発生した。

機能名, 対象名, およびエラー詳細はエラー情報を示す。

このメッセージは, エラー通知として出力できる出力先に出力するが, エラー発生タイミングによって, 一部の出力先にしか出力されないことがある。

(S)処理を終了する。

(O)エラー情報を調べ, 原因を取り除き, 再起動する。

KAKC3021-E

An error occurred during dynamic link processing. (機能名 error -エラー詳細)

スプールジョブ転送サービスでダイナミックリンクエラーが発生した。

機能名, およびエラー詳細はエラー情報を示す。

(S)処理を終了する。

(O)エラー情報を調べ, 原因を取り除き, 再起動する。

KAKC3022-E

An error occurred during message queue processing. (機能名 error -エラー詳細)

スプールジョブ転送サービスでメッセージキューエラーが発生した。

機能名, およびエラー詳細はエラー情報を示す。

(S)処理を終了する。

(O)エラー情報を調べ, 原因を取り除き, 再起動する。コマンド実行時に発生した場合, 必要ならコマンドを再投入する。

設定ファイルで定義するスプールディレクトリのパス名をメッセージキュー名として利用しており, 次に示す原因が考えられる。

- mq_open error -No such file or directory
スプールジョブ転送サービスが起動していない。
または, 起動時の設定ファイルと現時点の設定ファイルでスプールディレクトリのパス名に相違がある。
- その他

機能名はシステムの関数名であり、システム関数のエラーメッセージから原因を調査する。

KAKC3023-E

An error occurred during signal processing. (機能名 error 対象名-エラー詳細)

スプールジョブ転送サービスでシグナルエラーが発生した。

機能名, 対象名, およびエラー詳細はエラー情報を示す。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を終了する。

(O)エラー情報を調べ、原因を取り除き、再起動する。

KAKC3024-E

An error occurred during signal processing. (機能名 error 対象名)

スプールジョブ転送サービスでシグナルエラーが発生した。

機能名, および対象名はエラー情報を示す。

(S)処理を終了する。

(O)エラー情報を調べ、原因を取り除き、再起動する。

KAKC3025-E

An error occurred during signal processing. (機能名 error -エラー詳細)

スプールジョブ転送サービスでシグナルエラーが発生した。

機能名, およびエラー詳細はエラー情報を示す。

(S)処理を終了する。

(O)エラー情報を調べ、原因を取り除き、再起動する。

KAKC3026-E

An error occurred during signal processing. (機能名 error)

スプールジョブ転送サービスでシグナルエラーが発生した。

機能名はエラー情報を示す。

(S)処理を終了する。

(O)エラー情報を調べ、原因を取り除き、再起動する。

KAKC3027-E

An error occurred during process processing. (機能名 error 対象名-エラー詳細)

スプールジョブ転送サービスでプロセスエラーが発生した。

機能名, 対象名, およびエラー詳細はエラー情報を示す。

(S)処理を終了する。

(O)エラー情報を調べ、原因を取り除き、再起動する。コマンド実行時に発生した場合、必要ならコマンドを再投入する。

機能名はシステムの関数名であり、システム関数のエラーメッセージから原因を調査する。

KAKC3028-E

An error occurred during waitpid processing. (機能名 error waitpid()の戻り値-エラー詳細)

スプールジョブ転送サービスで waitpid エラーが発生した。

機能名, およびエラー詳細はエラー情報を示す。

(S)処理を終了する。

(O)エラー情報を調べ、原因を取り除き、再起動する。コマンド実行時に発生した場合、必要ならコマンドを再投入する。

機能名はシステムの関数名であり、システム関数のエラーメッセージから原因を調査する。

KAKC3029-E

An error occurred. (機能名 error -エラー詳細)

スプールジョブ転送サービスでシステム関数エラーが発生した。

機能名, およびエラー詳細はエラー情報を示す。

このメッセージは、エラー通知として出力できる出力先に出力するが、エラー発生タイミングによって、一部の出力先にしか出力されないことがある。

(S)処理を終了する。

(O)エラー原因を調べ、原因を取り除き、再起動する。

KAKC3031-E

The spool directory which bjexpd used was made again. (file:ファイル/ディレクトリ名 old:旧 inode 番号,new:新 inode 番号)

スプールジョブ転送デーモンで使用しているスプールディレクトリが再作成されたため、ファイルの排他制御機能、実行中のジョブ情報表示機能およびメッセージ制御機能が正しく動作できない。

"ファイル/ディレクトリ名"で、再作成されたファイルまたはディレクトリ名を示す。

(S)処理を続行する。

(O)システム管理者に連絡し、スプールジョブ転送デーモンを停止し、再起動する。

メッセージは、スプールディレクトリの再作成を検知後一度だけ出力される。以降、bjexpd を再起動するまでは出力されない。

KAKC3099-E

An internal error occurred. (機能名 error 発生場所-対象名)

スプールジョブ転送サービスで内部エラーが発生した。

機能名、発生場所、および対象名はエラー情報を示す。

(S)処理を終了する。

(O)エラー情報を調べ、原因を取り除き、再起動する。

KAKC3120-E

An error occurred during file processing. (機能名 error,jobid=ジョブ識別子,対象名-エラー詳細)

スプールジョブ転送サービスでファイルエラーが発生した。

機能名、ジョブ識別子、対象名、およびエラー詳細はエラー情報を示す。

(S)処理を続行する。

(O)エラー情報を調べ、原因を取り除く。bjexchjob コマンドで再度印刷指示をする。コマンド実行時に発生した場合、必要ならコマンドを再投入する。

機能名はシステムの関数名または処理名であり、エラーメッセージから原因を調査する。

KAKC3123-E

An error occurred during JP1/FTP API processing. (JP1/FTP API 名 error,jobid=ジョブ識別子,エラー番号-エラー詳細)

スプールジョブ転送サービスで JP1/FTP API エラーが発生した。

(S)処理を続行する。

(O)JP1/FTP API 名, エラー番号, またはエラー詳細から JP1/FTP のマニュアルを参照し, エラー情報を調べ, 原因を取り除く。ただし, リソース不足や論理矛盾など回復困難なエラーの場合は, スプールジョブ転送サービスを再起動したあとで, バッチジョブを再実行するか, bjextrjob コマンドで明示的に印刷ジョブの再転送を行うこと。

また, 次のエラー番号の場合はシステム管理者に連絡する。

- 805306124 (FTS_API_ERROR_NOTSOCK)
- 805306128 (FTS_API_ERROR_LOGIC)

KAKC3129-E

An error occurred. (機能名 error,jobid=ジョブ識別子,エラー詳細)

スプールジョブ転送サービスでシステム関数エラーが発生した。

機能名, ジョブ識別子, およびエラー詳細はエラー情報を示す。

(S)処理を続行する。

(O)エラー情報を調べ, 原因を取り除く。bjexchjob コマンドで再度印刷指示をする。

KAKC3130-I

Transfer retry times exceeded limit. (jobid=ジョブ識別子,file name=転送先ファイル名称)
スプールジョブの再転送回数が指定のリトライ回数を超えました。

ジョブ識別子で示すスプールジョブの転送時に転送先ファイル名称で示すファイルの転送エラーが発生し, 設定ファイルで定義したリトライ回数を超えた。

(S)該当するスプールジョブの転送を中断し, 別ジョブの処理を続行する。

(O)このメッセージの前後に実行ログに出力されたメッセージ (KAKC3013-E や KAKC3123-E) から原因を調査し, エラー原因を取り除く。bjextrjob コマンドで印刷ジョブの再転送を行う。

KAKC3199-E

An internal error occurred. (機能名 error,jobid=ジョブ識別子,発生場所-対象)

スプールジョブ転送サービスで内部矛盾エラーが発生した。

機能名, ジョブ識別子, 発生場所, および対象はエラー情報を示す。

(S)処理を続行する。

- (O)メモリ不足やスプールジョブ管理ファイルが壊れている可能性がある。システム管理者に連絡する。
また、設定ファイルがスプールジョブ転送サービス起動時、ジョブ実行時、または bjexchjob コマンド入力時で異なっている。この場合、スプールジョブ転送サービスを停止して、ジョブ実行時の設定ファイルでスプールジョブ転送サービスを再起動する。その後、必要ならコマンドを再入力する。

KAKC3204-I

File transfer of spool job was stopped.

スプールジョブのファイル転送が中断されました

スプールジョブの転送時、JP1/FTP でエラーが発生した。

(S)処理を続行する。転送中のスプールジョブの処理は中断される。

(O)このメッセージの前後に実行ログに出力されたメッセージから原因を調査し、エラー原因を取り除く。

KAKC3220-E

An error occurred during file processing. (機能名 error 対象名-エラー詳細)

スプールジョブ転送サービスでファイル操作エラーが発生した。

機能名、対象名、およびエラー詳細はエラー情報を示す。

(S)処理を続行する。

(O)エラー情報を調べ、原因を取り除く。bjexchjob コマンドで再度印刷指示をする。

機能名はシステムの関数名であり、システム関数のエラーメッセージから原因を調査する。

KAKC3240-E

A spool job was not deleted because the size of .sysout was not zero. (スプールジョブ管理
ファイル名称)

スプールジョブ管理ファイルのサイズが0ではないので、スプールジョブを削除しない。

(S)処理を続行する。

(O)システム管理者に連絡する。

KAKC3241-W

The size of .sysout is zero. (スプールジョブ管理ファイル名称)

スプールジョブ管理ファイルのサイズが0なので、スプールジョブを転送しない。

(S)処理を続行する。

(O)なし。

KAKC3243-W

The spool job files is not transfer. (スプールジョブ管理ファイル名称)

スプールジョブを転送しない設定のため、スプールジョブを転送しない。

SPOOL_JOB_TRANSFER パラメータでスプールジョブを転送しない指定がされているが、ジョブ定義 XML ファイルの指定もしくはクラスの定義で転送先が指定されている、または bjexchjob コマンドを入力した。

(S)処理を続行する。

(O)設定ファイル、ジョブ定義 XML ファイル、または運用を見直す。

KAKC3249-E

The .sysout name is invalid. (発生場所,詳細コード)

スプールジョブ管理ファイルの名称が不当である。

(S)処理を続行する。

(O)システム管理者に連絡する。

KAKC3250-I

The 対象名 process was created. (pid=PID)

子プロセスを生成した。対象名は、conv, trans である。

(S)処理を続行する。

(O)なし。

KAKC3251-I

The 対象名 process ended normally. (pid=PID, cause=要因)

子プロセスが終了した。対象名は、conv, trans, cp である。

(S)処理を続行する。

(O)なし。

KAKC3252-E

The 対象名 process ended abnormally. (pid=PID,signal no=シグナル番号)

シグナルによって、子プロセスが異常終了した。対象名は、conv, trans, cp である。

(S)処理を続行する。

(O)原因を取り除いたあと、必要ならば bjexchjob コマンドを入力して、スプールジョブを転送する。

KAKC3260-I

The bjexpd received a message from the message queue. (ジョブ ID,コード)

bjexpd に対する要求を受信した。

(S)処理を続行する。

(O)なし。

KAKC3261-I

The bjexpd received a signal. (コード)

bjexpd が終了要求のシグナルを受信した。

(S)処理を続行する。

(O)なし。

KAKC3400-I

The specified parameters in the configuration file were correct.

設定ファイルのチェックが正常に終了した。

(S)処理を続行する。

(O)なし。

KAKC3402-E

The specified parameters in the configuration file were incorrect.

設定ファイルのチェックでエラーが発生した。

(S)処理を終了する。

(O)エラー情報を調べ、原因を取り除く。

KAKC3412-E

An error occurred while bjexpd is writing log. (err code=エラーコード, err text=エラー発生時に出力しようとしていたメッセージ)

スプールジョブ転送デーモンの実行ログ情報出力処理でエラーが発生した。

このメッセージは、実行ログ出力エラー発生時に一度だけ出力される。以降、実行ログ出力エラーが発生しても、スプールジョブ転送デーモンを再起動するまでこのメッセージは出力されない。

(S)スプールジョブ転送デーモンの実行ログ出力処理を停止する。

(O)実行ログファイルのあるディスクの状態を確認し、容量不足が発生していれば不要なファイルを削除して空き領域を確保する。その後、bjexpdctl コマンドでスプールジョブ転送デーモンの停止、再起動を行う。

KAKC3415-E

The specified parameters in the configuration file were incorrect.

設定ファイルのチェックでエラーが発生した。

-h オペランドで指定した論理ホスト名に対応する設定ファイルが読み込めない場合も表示される。

(S)bjexpdctl コマンドを終了する。

(O)このメッセージとともに出力されているメッセージからエラー情報を調べ、原因を取り除く。

KAKC3416-E

An error occurred during shared memory processing. (エラー情報)

bjexpdctl コマンドの処理中に、共有メモリの操作でエラーが発生した。

(S)bjexpdctl コマンドを終了する。

(O)メッセージ中のエラー情報を調べ、原因を取り除く。

KAKC3508-I

Response request is cancelled. (ジョブ識別子, ステップ番号, JP1/AJS のホスト名)
応答要求をキャンセルしました。 (ジョブ識別子, ステップ番号, JP1/AJS のホスト名)

応答要求をキャンセルした。

(S)処理を続行する。

(O)なし。

KAKC3522-E

An error occurred during shared memory processing. (機能名 error -エラー情報)

スプールジョブ転送サービスで、共有メモリエラーが発生した。

機能名、およびエラー情報はエラー情報を示す。

(S)処理を終了する。

(O)エラー情報を調べ、原因を取り除き、再起動する。

機能名はシステムの関数名であり、システム関数のエラーメッセージから原因を調査する。

KAKC3542-W

対象名 is already exists.

スプールジョブ転送サービスが、共有メモリまたはセマフォを作成しようとしたが、すでに作成されている。対象名は、Shared memory object または Named semaphore である。

(S)すでに作成されている共有メモリまたはセマフォを使用して処理を続行する。

(O)スプールジョブ転送サービスが異常終了したあとの再起動であることを確認する。スプールジョブ転送サービスが異常終了していない場合、他機能の共有メモリまたはセマフォを使用しているため、バッチジョブ実行システムの運用を中止する。

KAKC3543-E

Duplicate error was detected while checking identifier of the shared memory. (ホスト名)

スプールジョブ転送サービスで、共有メモリの管理情報が他プロセスと重複している。

(S)スプールジョブ転送サービスは終了する。

(O)システム管理者に連絡する。

システム管理者は、共有メモリの重複が発生しないよう、MSG_KEY_ID パラメータで、識別情報を設定する。

KAKC3544-E

Duplicate error was detected while checking identifier of the shared memory. (ホスト名) key=キー情報

スプールジョブ転送サービス終了時、共有メモリの管理情報が他プロセスのものと重複している。

(S)共用メモリの管理情報を残したまま、スプールジョブ転送サービスは終了する。

(O)システム管理者に連絡する。

システム管理者は、共有メモリの重複が発生しないよう、MSG_KEY_ID パラメータで、識別情報を設定する。

KAKC3602-E

Directory "パラメータ名" is used by other bjexpd.

パラメータ名で指定したファイルパス名は、ほかのホストのスパールジョブ転送サービスが使用している。

(S)コマンドを終了する。

(O)設定ファイルを見直す。

KAKC3700-I

The bjexpd will now start.

スパールジョブ転送サービスを起動する。

(S)処理を続行する。

(O)なし。

KAKC3701-I

The bjexpd will now stop.

スパールジョブ転送サービスを終了する。

(S)処理を続行する。

(O)なし。

KAKC3703-E

The bjexpd is not running.

スパールジョブ転送サービスが動作していない。

(S)処理を終了する。

(O)スパールジョブ転送サービスを起動しているかを確認する。

KAKC3709-E

The bjexpd cannot start because another one is already running.

スプールジョブ転送サービスはすでに動作している。

(S)処理を終了する。

(O)必要であればすでに起動している bjexpd を止めてコマンドを再入力する。

KAKC3710-I

```
The bjexpd is running.
```

スプールジョブ転送デーモンは起動している。

スプールジョブ転送プロセスがリトライ待ちにならずに動作中であることを示す。

また、status_log オプションを指定した場合、実行ログ出力も動作中であることを示す。

(S)処理を終了する。

(O)メッセージを参照して、スプールジョブ転送デーモンの状況を把握する。

KAKC3711-I

```
The bjexpd is not running.
```

スプールジョブ転送デーモンは起動していない。

(S)処理を終了する。

(O)メッセージを参照して、スプールジョブ転送デーモンの状況を把握する。

KAKC3712-I

```
Logging by bjexpd is inactive.
```

スプールジョブ転送デーモンの実行ログ出力が停止している。

(S)処理を終了する。

(O)実行ログファイルのあるディスクの状態を確認し、容量不足が発生していれば不要なファイルを削除して空き領域を確保する。その後、bjexpdctl コマンドでスプールジョブ転送デーモンの停止、再起動を行う。

KAKC3713-E

```
The bjexpd could not stop because job is running.
```

ジョブ詳細内容

bjexpdctl コマンドでスプールジョブ転送デーモンの停止が指定されたが、実行中のジョブが存在するためスプールジョブ転送デーモンを停止できなかった。

2 行目以降に示す**ジョブ詳細内容**には、bjexlslock コマンドと bjexlsmmsg コマンドの実行結果を表示する。

(S)処理を終了する。

(O)実行中のジョブを終了させてから bjexpdctl コマンドを再実行する。

KAKC3714-W

```
The bjexpd will terminate.
```

ジョブ詳細内容

スプールジョブ転送デーモンを強制停止する。

2 行目以降に示す**ジョブ詳細内容**には、強制停止直前の bjexlslock コマンドと bjexlsmmsg コマンドの実行結果を表示する。実行中のジョブが存在しない場合は**ジョブ詳細内容**を表示しない。

(S)処理を終了する。

(O)なし。

KAKC3715-W

```
The bjexpd is waiting for JP1FTP_RETRY_INTERVAL time to expire. (伝送カード名,リトライ  
待ち開始日,リトライ待ち開始時刻)
```

bjexpd のスプールジョブ転送プロセスは、伝送カード名で示す伝送カードに登録された伝送先との接続確立でエラーになり、リトライ待ち開始日、リトライ待ち開始時刻から、JP1FTP_RETRY_INTERVAL で指定した時間が経過するのを待っている。

(S)処理を続行する。

(O)JP1/FTP を起動していない場合、JP1/FTP を起動し、bjexpdctl コマンドに ftp_connect オペランドを指定して実行し、スプールジョブ転送プロセスの待ち状態を解除する。

KAKC3716-I

```
A transmission process tried reconnection. (card name=伝送カード名)
```

bjexpdctl コマンドでスプールジョブ転送プロセスの伝送カード名で示す伝送先との接続待ちが解除され、再度接続を確立を試みた。

(S)スプールジョブ転送プロセスは待ち状態が解除され、伝送先との接続確立を試みる。

(O)status オペランドを指定して bjexpdctl コマンドを実行し、スプールジョブ転送プロセスの待ち状態が解除されていることを確認する。

KAKC3799-I

```
Usage: bjexpdctl [-h LogicalHostName] {start|stop[-f]|status|status_log|confest|ftp_connect|help}
```

bjexpdctl コマンドの usage 情報を表示する。

(S)処理を続行する。

(O)なし。

KAKC3820-E

An error occurred during file processing. (機能名 error,jobid=ジョブ識別子,対象名-エラー詳細)

スプールジョブ転送サービスで異常が発生した。

機能名, ジョブ識別子, 対象名, およびエラー詳細はエラー情報を示す。

(S)処理を終了する。

(O)機能名が「open」かつエラー詳細が「Device busy」の場合、スプールジョブ転送サービスによる処理中の場合がある。

この場合は、時間を置いて再度実行する。その他の場合はシステム管理者に連絡する。

KAKC3829-E

An error occurred. (機能名 error -エラー詳細)

スプールジョブ転送サービスで異常が発生した。

機能名, およびエラー詳細はエラー情報を示す。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC3998-E

An error occurred during bjexpd signal handler processing.

スプールジョブ転送サービスでシグナルハンドラが異常終了した。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC3999-E

The bjexpd ended abnormally because of an unexpected exception.

スプールジョブ転送サービスが異常終了した。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC4000-E

Gdg name is not specified.

コマンド引数に、世代データ群名の指定がない。

(S)処理を終了する。

(O)コマンド引数を見直す。

KAKC4001-E

Gdg name path is too long.

世代データ群パス名文字列長が最大長を超えている。

(S)処理を終了する。

(O)世代データ群パス名文字列長を見直す。

KAKC4002-E

Gdg name path is invalid.

世代データ群名の記述に誤りがある。

(S)処理を終了する。

(O)世代データ群名を見直す。

KAKC4003-E

Gdg relative number is invalid.

相対世代番号の記述に誤りがある。

(S)処理を終了する。

(O)相対世代番号を見直す。

KAKC4004-E

Gdg file of relative number 世代相対番号 is not exist.

世代相対番号の世代ファイルが存在しない。

(S)処理を終了する。

(O)相対世代番号を見直す。

KAKC4005-E

New gdg file could not be created because there is already a file with generation 9999.

9,999 を超えて世代ファイルを作成できない。

(S)処理を終了する。

(O)世代管理方法を見直す。

KAKC4010-I

"世代ファイル名" was registered successfully.

世代ファイル名の登録で,"世代ファイル名"の世代ファイル情報を登録した。

(S)処理を続行する。

(O)なし。

KAKC4011-I

"世代ファイル名" was unregistered successfully.

世代ファイル名の削除で,"世代ファイル名"の世代ファイル情報を削除した。

(S)処理を続行する。

(O)なし。

KAKC4020-E

世代群名 gdg file could not be added to a new generation because there is already a file with generation 9999.

世代群名で示す世代群に対して、新しいファイルを追加したが、世代番号が 9,999 を超えてしまうため、新しいファイルを追加できない。

(S)処理を終了する。

(O)世代群名で示す世代群は NOSCATCH が設定された世代群であるため、9,999 を超える世代番号の世代ファイルを生成できない。このため、世代群内のファイルの見直しをするか、または SCRATCH 設定 (-l 最大世代数の指定で設定される) に変更し、旧世代を自動削除できるようにする。

KAKC4021-E

世代群名[(相対世代番号 | 日付世代番号)] gdg file of the specified generation data group is not found.

世代群名[(相対世代番号 | 日付世代番号)]で示す世代ファイルを参照しようとしたが、該当する世代ファイルは見つからない。

または、指定した世代群には世代ファイルが 1 つも登録されていない。

(S)処理を終了する。

(O)参照する世代ファイルの参照指定が正しいか確認する。

または、指定した世代群に世代ファイルの登録が行われているか確認する。

KAKC4022-E

世代群名 could not be unregistered.

世代群名で示す世代群の削除をしようとしたが、次に示す理由で削除できない。

- 世代群として指定された世代ファイルには世代管理ファイルが存在しないため、世代群として扱えない。
- 指定された世代群に少なくとも 1 つ以上の世代ファイルの登録がある。
- 指定された世代群は存在しない。

(S)処理を終了する。

(O)

- 世代管理ファイルが存在しない場合
指定された世代群名が正しくないため、正しい世代群名を指定する。
- 世代ファイルが 1 つ以上登録された世代群の場合
登録されている世代ファイルをすべて削除してから、世代群の削除を実行する。
- 世代群が存在しない
世代群名の指定が正しいか確認する。

KAKC4023-E

Duplicate gdg name 世代群名.

世代群名で示す世代群の登録をしようとしたが、すでに同一の世代群名が存在している。

(S)処理を終了する。

(O)世代群名の指定が正しいか確認する。

KAKC4024-E

Specification of only gdg name with TYPE="GDG" is not supported.

TYPE=GDG を指定した"世代群名"単一指定の割り当てはサポートしていない。

(S)処理を終了する。

(O)世代ファイルの割り当て時は、相対世代番号を指定する。

KAKC4025-E

Specified generation number caused an internal error in management of generation file names.

指定した相対世代番号によって生成した世代ファイル名に世代管理上の矛盾が発生した。

(S)処理を終了する。

(O)世代管理ファイルに登録されていない世代ファイルが残存しているため、不要な世代ファイルを削除したあと再実行する。

KAKC4026-E

Duplicate generation file number for the specified DD "DD 名".

DD 名で指定した世代ファイル番号は重複している。

(S)処理を終了する。

(O)正しい世代ファイルの番号を指定し、再実行する。

KAKC4050-E

Gdg management file open failed. reason="エラー詳細"

世代管理ファイルをオープンしたが、エラー詳細によって失敗した。

(S)処理を終了する。

(O)エラー詳細の原因を取り除いて再実行する。

KAKC4051-E

```
Gdg management file I/O error. reason="エラー詳細"
```

世代管理ファイルの入出力が**エラー詳細**によって失敗した。

(S)処理を終了する。

(O)**エラー詳細**の原因を取り除いて再実行する。

KAKC4052-E

```
Gdg management file delete error. reason="エラー詳細"
```

世代管理ファイルの削除が**エラー詳細**によって失敗した。

(S)処理を終了する。

(O)**エラー詳細**の原因を取り除いて再実行する。

KAKC4053-E

```
Gdg management file not found this directory. "世代管理ファイル設定パス名"
```

世代管理ファイル設定パス名で示す世代管理ファイルがない。

世代管理ファイルには、次の2つがある。どちらも存在しないか、どちらかが存在しない場合にもこのメッセージは出力される。

- "世代管理ファイル名.gdg"
- ".世代管理ファイル名.gdg"

(S)処理を終了する。

(O)指定した世代群名、世代管理ファイル設定パス名が正しいかどうかを確認する。

指定が正しい場合、指定した世代群の世代管理ファイルの存在状態を確認し、世代管理ファイルを回復し、再実行する。

指定が誤っていた場合、正しい世代群名、世代管理ファイル設定パス名を指定して再実行する。

指定した世代群の生成を忘れていた場合、bjexgdg コマンドで生成後、再実行する。

KAKC4058-I

```
The gdg command processing ended. rc=終了コード
```

世代管理ファイルの処理が**終了コード**で終了した。

終了コード：bjexgdg コマンドまたは bjexhkgdg コマンドの終了コード

(S)処理を終了する。

(O)なし。

KAKC4059-E

Gdg management file syntax error. line=**行番号**

世代管理ファイルの**行番号**で示す行でエラーが発生した。

(S)処理を終了する。

(O)世代ファイルを初期化, または回復して再実行する。

KAKC4060-E

An error occurred by reading of a gdg management file. "**世代管理ファイルコード**" was not found.

世代管理ファイルの解析処理で, 管理情報として必要な"**世代管理ファイルコード**"が見つからなかった。

(S)処理を終了する。

(O)世代管理ファイルの内容を確認する。間違った世代データ群を指定している場合は, 正しい世代データ群名を指定し再実行する。

KAKC4061-E

Specification of "GDGFILEnnnn" is invalid, or number of registered gdg files specified by "INFILE" is larger than the actual number of registered gdg files.

世代管理ファイルの解析処理で, GDGFILEnnnn (nnnn は相対世代番号指定機能では 0001~0255, 日付世代番号指定機能では 0001~0511 の数値) の指定に誤りがある。または"INFILE"で設定されている登録数が実際の世代ファイル登録数より大きい値となっている。

(S)処理を終了する。

(O)世代管理ファイルの内容を確認する。

KAKC4062-E

By an error to show in a "**エラー詳細**", deletion of "**ファイル名**" was not possible.

"**エラー詳細**"に示すエラーによって, "**ファイル名**"の削除ができなかった。

(S)処理を終了する。

(O)エラー詳細の原因を取り除いたあとに, コマンドを再入力する。

KAKC4063-E

Renaming of the file "世代管理ファイル名" failed because of the following reason: "エラー詳細".

"世代管理ファイル名"のファイル名変更処理が"エラー詳細"に示す理由でエラーとなった。

(S)処理を終了する。

(O)エラー詳細の原因を取り除いたあとに、コマンドを再入力する。

KAKC4064-E

Check generation number. DSN=世代ファイル名

世代ファイル名の絶対世代番号を世代管理ファイルに登録することで、絶対世代番号が 0000～0999 の世代ファイル番号、1000～8999 の世代ファイル番号、および 9000～9999 の世代ファイル番号を持つようになるため、登録されている絶対世代番号の見直しが必要となった。

(S)処理を中止する。

(O)世代管理ファイルに登録されている絶対世代番号群の中で、最古の世代番号群に含まれるすべての世代番号を世代管理ファイルから削除し、必要に応じてジョブを再実行する。

KAKC4065-E

Check generation management file. FILE=世代管理ファイル名

世代管理ファイル名の読み出し処理で、読み出した世代管理ファイルには絶対世代番号が 0000～0999 の世代ファイル番号、1000～8999 の世代ファイル番号、および 9000～9999 の世代ファイル番号を持っているため、登録されている絶対世代番号の見直しが必要である。

(S)処理を中止する。

(O)世代管理ファイルに登録されている絶対世代番号群の中で、最古の世代番号群に含まれるすべての世代番号を世代管理ファイルから削除し、必要に応じてジョブを再実行する。

KAKC4066-E

Specified EXDATE relative number is invalid.

日付世代番号指定機能で DSN 属性の括弧内に EXD を指定して世代ファイルを作成する際、括弧内に -、+ を指定した。または、(EXD-n) と指定した場合に、n が指定上限値または有効桁数を超過している。または、EXD に負の整数以外を付けて指定した。

(S)ジョブステップを続行せずに処理を終了する。

(O)DSN 属性の括弧内に EXD を指定する場合、+ を指定しない。また、DISP="NEW" 指定のとき、- を指定しない。もしくは、EXD に付けた整数の指定を見直し、再実行する。

KAKC4067-E

DD "DD 名" gdg file of EXDATE is already exists.

DD 名で示す DD 要素で日付世代番号指定機能の世代ファイルの作成を指定したが、すでに存在する世代ファイルであるため、作成できない。

(S)ジョブステップを続行せずに処理を終了する。

(O)すでに存在するファイルを指定しないように、DSN 属性の括弧内の記述を見直す。または、DD 要素の DISP を"OLD"に記述を変更し、再実行する。

KAKC4068-E

Gdg file of EXDATE is specified by gdg relative number.

日付世代番号指定機能の世代ファイルに、相対世代番号を指定した。

(S)ジョブステップを続行せずに処理を終了する。

(O)DSN 属性の括弧内の指定を EXD, EXD-n, Dyyyymmdd のいずれかに修正し、再実行する。

KAKC4069-E

Gdg file of relative number is specified by EXDATE number.

相対世代番号指定機能の世代ファイルに EXD または Dyyyymmdd を指定した。

(S)ジョブステップを続行せずに処理を終了する。

(O)DSN 属性の括弧内の指定を相対世代番号に修正し、再実行する。

KAKC4070-E

Specified EXDATE absolute number is invalid.

日付世代番号指定機能で DSN 属性の括弧内に Dyyyymmdd を指定した際、日付の指定が誤っている。

(S)ジョブステップを続行せずに処理を終了する。

(O)DSN 属性の括弧内に Dyyyymmdd 指定する場合、不当な文字や日付を指定していないか確認する。

KAKC4071-E

世代群名(日付世代番号) gdg file is already exist, but gdg file of EXDATE is not registered.

世代群名（日付世代番号）で示す世代ファイルはすでに存在しているが、世代管理ファイルに登録されていない。

(S)ジョブステップを続行せずに処理を終了する。

(O)世代管理されていない世代ファイルを削除したあと再実行する。

KAKC4072-E

世代群名(日付世代番号) gdg file is registered, but gdg file of EXDATE is not exist.

世代群名（日付世代番号）で示す世代ファイルは世代管理ファイルに登録されているが、実体が存在しない。

(S)ジョブステップを続行せずに処理を終了する。

(O)ジョブ定義 XML ファイルの DD 要素で、DISP="DELETE"を指定してジョブを再実行し、世代管理ファイルから世代ファイル名の世代管理情報を削除する。

KAKC4073-E

Environment variable AJSEXDATE is not specified.

環境変数 AJSEXDATE が取得できない。

世代ファイル機能の日付世代番号指定機能を使用して、DSN 属性の括弧内に EXD を指定したが、JP1/AJS から起動されていないおそれがある。

(S)ジョブステップを続行せずに処理を終了する。

(O)JP1/AJS から起動されているか確認する。

KAKC4074-E

Invalid char is included in environment variable AJSEXDATE value.

環境変数 AJSEXDATE に使用できない文字が含まれている。

(S)ジョブステップを続行せずに処理を終了する。

(O)システム管理者に連絡して環境変数の値を見直す。

KAKC4075-E

The required function LOCK_OPTION is missing in function EXDATE.

日付世代番号指定機能を使用するためには LOCK_OPTION の機能の有効化指定が必要である。

(S)コマンド処理を中断する。

(O)パラメータの指定に矛盾が無いか指定値を見直す。

KAKC4076-E

Specified generation data group of EXDATE.

bjexhkgdg コマンドで日付世代番号指定機能による世代データ群を指定した。日付世代番号指定機能による世代データ群を指定することはできない。

(S)コマンド処理を中断する。

(O)相対世代番号指定機能の世代データ群を指定する。

KAKC4077-E

Specified gdg relative number for generation data group of EXDATE.

日付世代番号指定機能の世代データ群に相対世代番号を指定した。

(S)コマンド処理を中断する。

(O)相対世代番号または日付世代番号を指定しないで再入力する。

KAKC4210-E

The specified spool job does not exist. (jobid = ジョブ識別子)

指定したジョブ識別子のスプールジョブは存在しない。

(S)該当するスプールジョブの操作を中断し、別ジョブの処理を続行する。

(O)正しいジョブ識別子を指定して、コマンドを再入力する。

KAKC4211-E

The spool job is not ready for operation or display. (jobid = ジョブ識別子, status = 状態)

指定したジョブ識別子のスプールジョブは status で示す状態のため、表示および操作ができない。

status に表示する状態とその意味を次に示す。

状態	意味
HOLD	出力保留状態
NOHOLD	スプールジョブ転送デーモンで処理中の状態
EXEC	実行中
ERROR	エラー状態
NONE	ジョブは存在しない状態

状態	意味
UNKNOWN	管理状態が正しくない (スプール管理ファイルに異常がある状態)

(S)該当するスプールジョブの操作を中断し、別ジョブの処理を続行する。

(O)指定したジョブ識別子のスプールジョブがコマンドで処理できる状態のスプールジョブとして存在することを `bjexlsjob` コマンドで再確認し、正しいジョブ識別子を指定してコマンドを再入力する。

ただし、状態に UNKNOWN が表示される場合、スプール管理ファイルに異常があるため、運用を停止してスプール管理ファイルを再作成する必要がある。

KAKC4212-E

The specified spool job does not exist. (jobname = ジョブ名)

指定したジョブ名のスプールジョブは、スプールジョブが存在しなかった。または、スプールジョブは存在するが、コマンドで指定した処理条件に該当しないなどの理由で見つからなかった。

指定したジョブ名のすべてについてこのメッセージを出力した場合、KAKC4214-E メッセージも総括として出力する。

(S)処理を終了する。

(O)指定したジョブ名のスプールジョブが存在するか、処理条件 (ジョブ実行開始時刻が指定範囲内であるか) があるかを `bjexlsjob` コマンドで再確認して、正しいジョブ名を指定したコマンドを再入力する。

KAKC4214-E

No spool job exists.

コマンドで処理するスプールジョブは 1 つも存在しないため、コマンドは何も処理できていない。

(S)処理を終了する。

(O)処理できるスプールジョブが最低でも 1 つ存在することを `bjexlsjob` コマンドで確認して、コマンドを再入力する。

KAKC4215-E

No sysout file exists in spool job. (jobid = ジョブ識別子, jobname = ジョブ名)

表示できる sysout は 1 件も存在しない。

(S)該当するスプールジョブの処理を中断し、別ジョブの処理を続行する。

(O)該当ジョブに SYSOUT データが存在するか確認する。

KAKC4216-E

A spool job is incomplete. (jobid = ジョブ識別子, jobname = ジョブ名)

ジョブ識別子およびジョブ名で示すスプールジョブは不完全なジョブである (スプールジョブ管理ファイルが未完成の状態である)。

(S)該当するスプールジョブの処理を中断し、別ジョブの処理を続行する。

(O)該当ジョブの実行時に異常がないか調査する。最後に実行が完了したジョブステップまでの SYSOUT データは転送可能である。

KAKC4220-E

Spool job storage options are not enabled.

スプールジョブ保管機能が有効になっていないので、この機能は利用できない。

(S)コマンドを終了する。

(O)設定ファイルを変更してコマンドを再入力する。

KAKC4300-E

An error occurred while opening a spool control file. (error message = エラー詳細)

スプール管理ファイルのオープンに失敗した。

(S)処理を終了する。

(O)スプールディレクトリ内のファイル (.hold) に異常がないかエラー詳細を基に調査する。障害を取り除いたあとに、コマンドを再入力する。

KAKC4301-E

The file was not found. (file name = ファイル名)

スプール管理ファイルが存在しない。

(S)処理を終了する。

(O)スプールディレクトリ内のファイル (.hold) に異常がないか調査する。障害を取り除いたあとに、コマンドを再入力する。

KAKC4302-E

An error occurred while reading a spool control file. (error message = エラー詳細)

スプール管理ファイルの読み込みに失敗した。

(S)処理を終了する。

(O)スプールディレクトリ内のファイル (.hold) に異常がないかエラー詳細を基に調査する。障害を取り除いたあとに、コマンドを再入力する。

KAKC4303-E

The spool job file has already been removed. (jobid = ジョブ識別子, jobname = ジョブ名)

ジョブ識別子およびジョブ名で示すスプールジョブのファイルは削除済みである。

(S)処理を続行する。

(O)スプールジョブが削除されていることを確認する。削除されていない場合は、再度コマンドを入力する。

KAKC4304-E

An error occurred while removing a spool job. (jobid = ジョブ識別子, jobname = ジョブ名, process = 処理, error message = エラー詳細)

ジョブ識別子およびジョブ名で示すスプールジョブの削除処理でエラーが発生した。

処理はエラーになった関数名または処理と対象を表す。なお、対象がない場合もある。

(S)該当するスプールジョブの処理を中断し、別ジョブの処理を続行する。

(O)エラー詳細を基に原因を調査する。障害を取り除いたあとに、コマンドを再入力する。処理が lock spool job の場合、ほかで該当するスプールジョブを処理中の可能性がある。この場合は、必要ならほかの処理が終了するのを待ってから、コマンドを再投入する。

KAKC4305-E

A spool job file could not be removed. (jobid = ジョブ識別子, jobname = ジョブ名, file name = ファイル名)

ジョブ識別子およびジョブ名で示すスプールジョブのファイル削除でエラーが発生した。

引き続き、詳細を示すメッセージ KAKC4303-E, KAKC4304-E を出力する。

(S)処理を続行する。

(O)障害を取り除いたあとに、必要ならば、コマンドを再入力する。

KAKC4306-E

A spool job has been skipped. (jobid = ジョブ識別子, jobname = ジョブ名)

ジョブ識別子およびジョブ名で示すスプールジョブは処理対象の範囲外であり、処理をスキップした。

(S)該当するスプールジョブの処理を中断し、別ジョブの処理を続行する。

(O)指定したジョブが処理対象か確認する。必要ならば、コマンドを再入力する。

KAKC4307-E

The spool control file is too large.

スプール管理ファイルのサイズが大きい。スプール管理ファイルが破壊されている場合は再作成が必要である。

(S)処理を終了する。

(O)障害を取り除いたあとに、コマンドを再入力する。

KAKC4308-E

The spool control file is too small.

スプール管理ファイルのサイズが小さい。

ほかの処理と競合している場合に発生することがある。

(S)処理を終了する。

(O)コマンドを再入力する。コマンドを何度か再入力しても同一現象になる場合は、スプール管理ファイルが破壊されているおそれがあるため、再作成が必要である。

KAKC4309-E

An error occurred while writing to a spool control file. (error message = エラー詳細)

スプール管理ファイルの書き込みに失敗した。

(S)処理を終了する。

(O)エラー詳細を基に原因を調査する。

スプールジョブの管理が正しく行われていない可能性があるため、bjexchjob -S コマンドで回復処理を実行する。または、運用を停止してスプール管理ファイルの再作成をする必要がある。障害を取り除いたあとに、コマンドを再入力する。

KAKC4310-E

An error occurred while opening a spool job control file. (jobid = ジョブ識別子)

ジョブ識別子で指定したスプールジョブ管理ファイルのオープンに失敗した。

(S)該当するスプールジョブの処理を中断し、別ジョブの処理を続行する。

(O)該当ジョブのspoolジョブ管理ファイルに異常がないか、メモリは十分あるかを確認する。障害を取り除いたあと、コマンドを入力する。

KAKC4311-E

The spool job control file has no contents. (jobid = ジョブ識別子)

ジョブ識別子で指定したspoolジョブ管理ファイルは空のファイルである。

(S)該当するspoolジョブの処理を中断し、別ジョブの処理を続行する。

(O)該当ジョブのspoolジョブ管理ファイルに異常がないか確認する。障害を取り除いたあと、コマンドを入力する。また、spoolジョブ管理ファイルが破壊されている場合は、該当spoolジョブを強制削除して、ジョブを再実行する。

spoolジョブ管理ファイルが破壊されている場合、該当するspoolジョブを強制削除して、ジョブを再実行する。

KAKC4312-E

An error occurred while removing a sysout. (jobid = ジョブ識別子, jobname = ジョブ名, step number = ステップ番号, DD name = DD 名称, error message = エラー詳細)

ジョブ識別子およびジョブ名で指定したspoolジョブのSYSOUTデータファイルの削除に失敗した。

(S)処理を続行する。

(O)エラー詳細を基に調査して、障害を取り除いたあとに、コマンドを再入力する。

エラー詳細がunknown fileの場合、SYSOUTデータファイルは該当するspoolジョブのファイルではなく、spoolジョブ管理ファイルが破壊されているか、spoolディレクトリ名を変更したなどのおそれがある。

KAKC4313-E

The spool transfer directory is not accessible. (dir name = 転送ディレクトリ, error message = エラー詳細)

SYSOUTデータファイルの転送ディレクトリにアクセスできない。

(S)処理を終了する。

(O)エラー詳細を基に調査して、障害を取り除いたあとに、コマンドを再入力する。

KAKC4314-E

The spool job control file is not accessible. (jobid = ジョブ識別子, error message = エラー詳細)

spoolジョブ管理ファイルにアクセスできない。

(S)該当するスプールジョブの処理を中断し、別ジョブの処理を続行する。

(O)エラー詳細を基に調査して、障害を取り除いたあとに、コマンドを再入力する。

KAKC4315-E

An error occurred while changing attribute of a spool job. (jobid = ジョブ識別子, jobname = ジョブ名)

スプールジョブの属性変更処理に失敗した。

(S)該当するスプールジョブの処理を中断し、別ジョブの処理を続行する。

(O)メモリが不足していないか、スプールジョブ管理ファイルに問題がないか確認する。障害を取り除いたあとに、コマンドを再入力する。

KAKC4316-E

An error occurred while getting status of a spool job file. (jobid = ジョブ識別子, error message = エラー詳細)

スプールジョブ管理ファイル、またはスプールジョブディレクトリの情報取得でエラーが発生した。

(S)該当するスプールジョブの処理を中断、または続行する。その後、ほかのスプールジョブがあった場合は処理する。

(O)エラー詳細を基に調査して、障害を取り除いたあとに、コマンドを再入力する。

KAKC4317-E

An error occurred while updating a spool job control file. (jobid = ジョブ識別子, jobname = ジョブ名, process = 処理)

スプールジョブ管理ファイルでエラーが発生した。なお、削除処理で発生した場合は、スプールジョブは削除されているおそれがある。

(S)処理を続行する。

(O)スプールジョブ管理ファイルに問題がないか調査する。削除処理でスプールジョブが削除されずに残っている場合は、強制削除する。

KAKC4318-E

Spool job cannot be removed. (jobid = ジョブ識別子, jobname = ジョブ名, reason = 理由)

スプールジョブ削除処理でエラーが発生した。

理由	意味
probably alive	-a で指定した秒数以前の時刻以降に更新されているため、ジョブが実行中の可能性がある。

(S)該当するジョブの処理を中断し、別ジョブの処理を続行する。

(O)ジョブが実行中の場合は、ジョブが終了してからコマンドを入力する。ジョブが実行中でない場合は、一定時間待ったあとに、コマンドを再度入力する。

KAKC4319-E

The spool job is busy. (jobid = ジョブ識別子)

スプールジョブは、ほかで使用中のため処理できない。

(S)該当するジョブの処理を中断し、別ジョブの処理を続行する。

(O)一定時間待ってからコマンドを再度入力する。

KAKC4320-E

An error occurred while opening message queue. (error message = エラー詳細)

メッセージキューのオープンでエラーが発生した。

スプールジョブ転送デーモンが起動されていない可能性がある。

(S)処理を終了する。

(O)エラー詳細を基に調査して、障害を取り除いたあとに、コマンドを再入力する。エラー詳細が No such file or directory の場合は、スプールジョブ転送デーモンを起動していない。または、設定ファイルに指定したスプールディレクトリのパス名が、スプールジョブ転送デーモン起動時とコマンド投入時点で異なる。

KAKC4321-E

An error occurred while sending message queue. (jobid = ジョブ識別子, jobname = ジョブ名, error message = エラー詳細)

メッセージキューへのメッセージ送信でエラーが発生した。

(S)該当ジョブの処理を中断し、別ジョブの処理を続行する。

(O)エラー詳細を基に調査して、障害を取り除いたあとに、コマンドを再入力する。

KAKC4322-E

An error occurred while closing message queue. (error message = エラー詳細)

メッセージキューのクローズでエラーが発生した。

(S)処理を終了する。

(O)エラー詳細を基に調査して、障害を取り除いたあとに、コマンドを再入力する。

KAKC4360-E

An error occurred while changing a spool job status. (jobid = ジョブ識別子, jobname = ジョブ名, error code = エラーコード, error details = 詳細コード)

スプールジョブの印刷指示処理でエラーが発生した。

(S)該当ジョブの処理を中断し、別ジョブの処理を続行する。

(O)障害を取り除いたあとに、コマンドを再入力する。

エラーコードを基に次のことを調査する。

- スプールジョブ管理ファイルに異常がないか。
- スプールジョブ転送デーモンは起動しているか。
- ジョブ実行時の設定ファイルと現在の設定ファイルの指定が同一であるか。
- メモリは不足していないか。

KAKC4361-E

A spool job could not be recovered. (jobid = ジョブ識別子, reason = 理由)

スプールジョブの回復処理に失敗した。

理由：probably alive 処理中の可能性があり、回復処理を中断した。

(S)該当するジョブの処理を中断して、別ジョブの処理を続行する。

(O)ジョブが実行中の場合、回復処理は不要である。ジョブが実行中でない場合には、一定時間待ってからコマンドを再度入力する。

KAKC4362-E

The recovery process of spool job aborted. (jobid = ジョブ識別子, jobname = ジョブ名)

スプールジョブの回復処理のサブプロセスでエラーが発生した。

サブプロセスはシグナルなどで中断したおそれがある。

(S)該当ジョブの処理を中断し、別ジョブの処理を続行する。

(O)再度コマンドを入力する。

KAKC4363-E

An error occurred while recovering spool job. (jobid = ジョブ識別子, jobname = ジョブ名, process = 処理, error message = エラー詳細)

スプールジョブの回復処理でエラーが発生した。

処理	意味
fopen	スプールジョブ管理ファイルのオープン処理に失敗した。
lock spool job	スプールジョブのロック処理に失敗した (ジョブが実行中のおそれがある)。
mp-open	メッセージキューのオープンに失敗した (エラー詳細が No such file or directory の場合は、スプールジョブ転送デーモンが起動していない。または、スプールジョブ転送デーモン起動時の設定ファイルと現時点の設定ファイルで、スプールディレクトリのパス名が一致していない)。
mq-send	メッセージキューのメッセージ送信に失敗した。
mq-close	メッセージキューのクローズに失敗した。

(S)該当ジョブの処理を中断し、別ジョブの処理を続行する。

(O)エラー詳細を基に調査して、障害を取り除いたあとに、コマンドを再入力する。なお、ジョブ実行中の場合には、回復処理は不要である。

KAKC4430-E

The specified file name is incorrect.

関数にファイル名の指定がない。

(S)関数の処理を中断する。

(O)システム管理者に連絡する。

KAKC4431-E

The specified address of buffer is incorrect.

関数に指定されたバッファアドレスが不当である。

(S)関数の処理を中断する。

(O)システム管理者に連絡する。

KAKC4500-E

An error occurred in the time routine. (error message = エラー詳細)

時刻を求める処理でエラーが発生した。

(S)該当ジョブの処理を中断し、別ジョブの処理を続行、または処理を終了する。

(O)システム管理者に連絡する。

KAKC4501-E

An error occurred in the mktime routine.

時刻変換処理でエラーが発生した。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC4502-E

An error occurred in the setuid routine. (jobid = ジョブ識別子, jobname = ジョブ名, error message = エラー詳細)

ユーザ ID 設定処理でエラーが発生した。該当するスプールジョブを操作する権限がないおそれがある。

(S)該当ジョブの処理を中断し、別ジョブの処理を続行する。

(O)エラー詳細を基に調査して、障害を取り除いたあとに、コマンドを再入力する。

KAKC4504-E

An error occurred in the setgid routine. (jobid = ジョブ識別子, jobname = ジョブ名, error message = エラー詳細)

グループ ID 設定処理でエラーが発生した。

(S)該当ジョブの処理を中断し、別ジョブの処理を続行する。該当するスプールジョブを操作する権限がないおそれがある。

(O)エラー詳細を基に調査して、障害を取り除いたあとに、コマンドを再入力する。

KAKC4506-E

An error occurred in the fork routine of chjob command. (error message = エラー詳細)

プロセスの生成でエラーが発生した。

(S)該当ジョブの処理を中断し、別ジョブの処理を続行する。

(O)エラー詳細を基に調査して、障害を取り除いたあとに、コマンドを再入力する。

KAKC4507-E

An error occurred in the waitpid routine of chjob command. (error message = エラー詳細)

プロセスの終了待ちでエラーが発生した。

(S)該当ジョブの処理を中断し、別ジョブの処理を続行する。

(O)エラー詳細を基に調査して、障害を取り除いたあとに、コマンドを再入力する。

KAKC4508-E

An error occurred during the release process.

スプールジョブの印刷指示処理でエラーが発生した。シグナルなどでサブプロセスが中断したおそれがある。

(S)該当ジョブの処理を中断し、別ジョブの処理を続行する。

(O)コマンドを再入力する。

KAKC4509-E

Insufficient memory. (object = 対象)

対象で示す領域のメモリ確保に失敗した。

(S)関数処理を中断する。

(O)システム管理者に連絡する。

KAKC4560-E

The specified option is invalid. (option = オプションコード)

不当なオプションコードまたはオプションコード値が指定された。

(S)処理を終了する。

(O)正しい指定をする。

KAKC4561-E

The specified jobid is invalid. (jobid = ジョブ識別子)

不当なジョブ識別子が指定された。

ジョブ識別子を範囲指定した場合、その大小関係が正しくない。

(S)処理を終了する。

(O)正しい指定をする。

KAKC4562-E

A jobid is not specified.

ジョブ識別子の指定がない。

(S)処理を終了する。

(O)ジョブ識別子を指定する。

KAKC4563-E

The option is incomplete.

必要なオプションの指定がない。

(S)処理を終了する。

(O)正しい指定をする。

KAKC4564-E

The number of items exceeded limit. (item = 該当項目)

指定項目が上限を超えた。

該当項目はエラーになった項目を最大 8 文字表示する。

(S)処理を終了する。

(O)項目数を減らして、コマンドを再入力する。

KAKC4566-E

The specified jobname is invalid. (jobname = ジョブ名)

不当なジョブ名が指定された。

(S)処理を終了する。

(O)正しい指定をする。

KAKC4567-E

A jobname is not specified.

ジョブ名の指定がない。

(S)処理を終了する。

(O)ジョブ名を指定する。

KAKC4599-E

Unsupported value is specified for the LANG environment variable. (LANG = 環境変数値)

サポートされていない LANG 環境変数値が設定されている。

(S)処理を終了する。

(O)サポートされた LANG 環境変数値を設定する。

KAKC4600-I

EXECUTION JOB INFORMATION DATE=情報表示日付 TIME=情報表示時刻

bjexlsexjob コマンドで、実行中のジョブ情報を表示する日付と時刻を表示する。

(S)処理を続行する。

(O)なし。

KAKC4624-E

An error occurred during the access of the /proc file. pid=プロセス番号 (code=保守情報,error message=エラー詳細)

実行中のジョブ情報表示機能の処理中に pid で示すプロセス情報の取得中にエラーが発生した。

(S)エラーが発生したプロセスの情報取得を中止し、次のプロセスの情報を表示する。

(O)システム管理者に連絡する。

KAKC5000-E

The number of response message exceeds the limit.

応答待ちイベントの出力数が上限を超えた。

(S)出力できるまでリトライを繰り返す。リトライ中はこのメッセージを出力しない。

(O)MAX_RESPONSE_REQUEST パラメータの設定を見直す。

KAKC5001-I

応答 (ジョブ識別子, ジョブステップ番号, ホスト名, 応答者, 保守情報)

応答待ちイベントに対して応答が入力された。

応答：入力された応答

ジョブ識別子：応答対象プロセスのジョブ識別子

ジョブステップ番号：応答対象プロセスのジョブステップ番号

ホスト名：JP1_HOSTNAME パラメータで指定したホスト名

ただし、バッチジョブ実行システムのコマンドで-h オプションを指定した場合、設定ファイルの JP1_HOSTNAME パラメータは無視され、論理ホスト名が有効になる。詳細については、「[7. コマンド](#)」を参照すること。

応答者：JP1/IM - View から応答を入力したユーザの JP1 ユーザ ID

(S)処理を続行する。

(O)なし。

KAKC5002-I

```
Message library started.
```

メッセージ制御機能のライブラリが処理を開始した。

(S)処理を続行する。

(O)なし。

KAKC5003-I

```
Message library ended. (rc=終了コード)
```

メッセージ制御機能のライブラリが終了コードで処理を終了した。

終了コード：「[付録 D メッセージ制御機能のエラー情報](#)」に示す、戻り値の値

(S)ライブラリの呼び出し元に処理を返す。

(O)なし。

KAKC5005-E

```
An error occurred during shared memory processing. (errno=エラー番号,function=保守情報)
```

応答待ちイベントの処理中に、共有メモリの操作でエラーが発生した。

(S)処理を終了する。

(O)bjexpd を起動して、ジョブを再実行する。

KAKC5006-E

Duplicate error was detected while checking identifier of the shared memory.

メッセージ制御機能のライブラリで使用する共有メモリの管理情報が他プロセスのものと重複している。

(S)ライブラリの呼び出し元に処理を返す。

(O)システム管理者に連絡する。

システム管理者は、共有メモリの重複が発生しないよう、MSG_KEY_ID パラメータで、識別情報を設定する。

KAKC5008-I

Response request is cancelled. (ジョブ識別子, ステップ番号, ホスト名)

応答要求をキャンセルしました (ジョブ識別子, ステップ番号, ホスト名)

応答待ちイベントをキャンセルした。

ジョブ識別子：このジョブのジョブ識別子

ステップ番号：このジョブステップのステップ番号

ホスト名：JP1_HOSTNAME パラメータで指定したホスト名

ただし、バッチジョブ実行システムのコマンドで-h オプションを指定した場合、設定ファイルの JP1_HOSTNAME パラメータは無視され、論理ホスト名が有効になる。詳細については、「[7. コマンド](#)」を参照すること。

(S)処理を続行する。

(O)なし。

KAKC5009-E

Output to JOBLOG failed.

JOBLOG への出力に失敗した。

(S)処理を続行する。

(O)システム管理者に連絡し、設定ファイルのプールディレクトリに対する指定を見直す。

KAKC5010-E

Output to SYSMSG failed.

SYSMMSG への出力に失敗した。

(S)処理を続行する。

(O)システム管理者に連絡し、設定ファイルのプールディレクトリに対する指定を見直す。

KAKC5011-E

An error occurred during JP1Event processing. (errno=エラー番号,function=保守情報)

JP1 イベントの処理中にエラーが発生した。

(S)処理を終了する。

(O)JP1/Base が起動していることを確認する。問題が解決しない場合は、システム管理者に連絡する。

KAKC5012-I

メッセージ (ジョブ識別子, ステップ番号, ホスト名, メッセージ識別子, 保守情報)

メッセージ制御機能によって出力した JP1 イベントの内容を示す。

なお、このメッセージと同時に KAKC5011-E または KAKC5020-E が出力されている場合は JP1 イベントの出力が失敗しているが、出力しようとしていた内容をこのメッセージで示す。

メッセージ: メッセージ制御機能に入力されたメッセージ

ジョブ識別子: このジョブのジョブ識別子

ステップ番号: このジョブステップのステップ番号

ホスト名: JP1_HOSTNAME パラメータで指定したホスト名

ただし、バッチジョブ実行システムのコマンドで-h オプションを指定した場合、設定ファイルの JP1_HOSTNAME パラメータは無視され、論理ホスト名が有効になる。詳細については、「[7. コマンド](#)」を参照すること。

メッセージ識別子: メッセージごとに割り当てる識別子

(S)処理を続行する。

(O)KAKC5011-E または KAKC5020-E メッセージが出力されている場合は、それぞれの処置を行う。

KAKC5013-I

メッセージ (ジョブ識別子, ステップ番号, ホスト名, メッセージ識別子, 保守情報)

メッセージ制御機能によって応答待ちイベントを出力した。

メッセージ: メッセージ制御機能に入力されたメッセージ

ジョブ識別子：このジョブのジョブ識別子

ステップ番号：このジョブステップのステップ番号

ホスト名：JP1_HOSTNAME パラメータで指定したホスト名

ただし、バッチジョブ実行システムのコマンドで-h オプションを指定した場合、設定ファイルの JP1_HOSTNAME パラメータは無視され、論理ホスト名が有効になる。詳細については、「7. コマンド」を参照すること。

メッセージ識別子：メッセージごとに割り当てる識別子

(S)処理を続行する。

(O)なし。

KAKC5014-I

Descriptor of JP1Event was acquired successfully. (host=ホスト名)

JP1 イベントを発行するための JP1 イベントの記述子の取得に成功した。

ホスト名：ジョブコントローラが稼働するホスト名

(S)処理を続行する。

(O)なし。

KAKC5015-E

Failed to acquire the descriptor of JP1Event. (errno=保守情報,host=ホスト名)

JP1 イベントを発行するための JP1 イベントの記述子の取得に失敗した。

保守情報：JP1 イベント処理のエラー情報

ホスト名：ジョブコントローラが稼働するホスト名

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5016-I

JP1Event was sent to the event service successfully.

JP1/Base のイベントサービスへの JP1 イベントの送信に成功した。

(S)処理を続行する。

(O)なし。

KAKC5017-E

Failed to send the JP1Event to the event service. (errno=エラー番号)

JP1/Base のイベントサービスへの JP1 イベントの送信に失敗した。

(S)処理を終了する。

(O)エラー番号のエラーの原因を取り除き、ジョブを再実行する。

KAKC5018-I

Transmission of JP1Event was checked successfully. (dest=転送先ホスト名)

JP1/IM が稼働するホストへの JP1 イベントの転送確認に成功した。

転送先ホスト名：JP1IM_MANAGER_HOST パラメータで指定した転送先ホスト名

(S)処理を続行する。

(O)なし。

KAKC5019-E

Failed to check transmission of JP1Event. (errno=保守情報,dest=転送先ホスト名,seqno=発行元イベント DB 内通し番号)

JP1/IM が稼働するホストへの JP1 イベントの転送確認に失敗した。

保守情報：JP1 イベント処理のエラー情報

転送先ホスト名：JP1IM_MANAGER_HOST パラメータで指定した転送先ホスト名

発行元イベント DB 内通し番号：JP1/Base の発行元イベント DB 内通し番号

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5020-E

Timeout occurred while processing message library.

メッセージ制御機能でタイムアウトが発生した。

次に示す要因が考えられる。

- JP1/Base のイベントサービスが起動していない。
- 応答要求メッセージの出力数が MAX_RESPONSE_REQUEST パラメータの指定値を超えている。

(S)処理を終了する。

(O)問題を解決し、ジョブを再実行する。

KAKC5021-E

Initialization of log failed. code=**保守情報**, reason=**エラー原因**, os_code=**保守情報 2**

実行ログの初期化に失敗した。

(S)処理を続行する。実行したメッセージ制御機能での実行ログへメッセージを出力しない。

(O)ジョブを再実行する。何度か再実行しても同一現象になる場合は、システム管理者に連絡する。保守情報 2 が 11, 13 の場合、別プロセスが実行ログを排他していることが原因のため、排他しているプロセスを見直す。

KAKC5100-I

Successfully copied Adapter_HITACHI_BJEX_MESSAGE.conf

メッセージ制御機能で使用する Adapter_HITACHI_BJEX_MESSAGE.conf を、JP1/Base のディレクトリに格納した。

(S)処理を続行する。

(O)なし。

KAKC5101-E

JP1/Base is not installed

JP1/Base がインストールされていないため、メッセージ制御機能で使用する Adapter_HITACHI_BJEX_MESSAGE.conf を、JP1/Base のディレクトリに格納できなかった。

(S)処理を終了する。

(O)JP1/Base をインストールする。

KAKC5102-E

Failed to copy Adapter_HITACHI_BJEX_MESSAGE.conf

メッセージ制御機能で使用する Adapter_HITACHI_BJEX_MESSAGE.conf を、JP1/Base のディレクトリに格納できなかった。

(S)処理を終了する。

(O)root 権限で再実行する。

KAKC5205-E

The output of the message failed. rc=戻り値, error=エラー番号, place=場所

メッセージ出力に失敗した。

戻り値とエラー番号はマニュアルの「付録 D メッセージ制御機能のエラー情報」を参照すること。

場所は内部情報である。

(S)処理を終了する。

(O)原因を取り除いて再実行する。

KAKC5208-E

The response message is undefined. Please respond again. (ジョブ識別子, ステップ番号, ホスト名, メッセージ識別子, 保守情報)

応答されたメッセージは未定義です。再度応答してください。(ジョブ識別子, ステップ番号, ホスト名, メッセージ識別子, 保守情報)

bjexm_message コマンドに応答したメッセージは定義されていない。

(S)再度、応答要求メッセージを出力する。

(O)応答要求メッセージに対して、正しい応答を入力する。

KAKC5309-E

An error occurred. (機能名 error)

内部エラーが発生した。

機能名はエラー情報を示す。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5323-E

An error occurred during signal processing. (機能名 error 対象名-エラー詳細)

アダプタコマンドでシグナルエラーが発生した。

機能名, 対象名, およびエラー詳細はエラー情報を示す。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5362-E

Non-ASCII character is found in a response.

応答データ中に ASCII 文字以外が指定された。

(S)処理を終了する。

(O)ASCII 文字で応答する。

KAKC5370-I

Response message was accepted.

応答メッセージを受信した。

(S)処理を終了する。

(O)なし。

KAKC5371-E

The bjex message function is busy.

バッチジョブ実行システムの処理がビジー状態である。

(S)処理を終了する。

(O)時間を空けて, 再度応答メッセージに応答する。

KAKC5372-E

A message is not found.

応答メッセージは存在しない。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5380-I

Received data from JP1: 受信データ

メッセージ制御機能が JP1 からデータを受信した。

(S)処理を続行する。

(O)なし。

KAKC5381-I

```
Sent status info to JP1: 送信データ
```

メッセージ制御機能が JP1 にデータを送信した。

(S)処理を続行する。

(O)なし。

KAKC5390-I

```
bjexmessage.adapter completed because signal is detected.
```

アダプタコマンドが、終了シグナルを受信して終了した。

(S)処理を終了する。

(O)なし。

KAKC5397-I

```
Signal handler processing completed.
```

アダプタコマンドが、シグナルを受信した。

(S)処理を終了する。

(O)core が出力されているため、システム管理者に連絡する。

KAKC5398-E

```
An error occurred during bjexmessage.adapter signal handler processing.
```

アダプタコマンドがシグナルを受信したが、シグナルハンドラの処理でエラーが発生した。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5399-E

```
The bjexmessage.adapter ended abnormally because of an unexpected exception.
```

アダプタコマンドが、異常終了した。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5407-E

Non-ASCII character is found in a response.

bjexchmsg コマンドで、応答メッセージ中に ASCII 文字以外を指定した。

(S)処理を終了する。

(O)ASCII 文字で応答する。

KAKC5408-I

応答要求をキャンセルしました。 (ジョブ識別子, ステップ番号, JP1/AJS のホスト名)

bjexchmsg コマンドで応答が入力されたまたは取り消されたため、応答要求をキャンセルした。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5409-E

No valid response request was found.

bjexchmsg コマンドで、応答の入力または応答要求の取り消しが指定されたが、指定された応答要求が存在しない。

(S)処理を終了する。

(O)bjexlsmmsg コマンドで-n オプションに指定した番号に誤りがないか確認し、正しい番号を指定する。

KAKC5410-E

A system call error occurred. (機能名 error -エラー詳細)

システムコールエラーが発生した。

機能名, およびエラー詳細はエラー情報を示す。

(S)処理を終了する。

(O)エラー情報を調べ、原因を取り除き、再度実行する。

KAKC5423-E

An error occurred during signal processing. (機能名 error 対象名-エラー詳細)

bjexchmsg コマンドまたは bjexlsmmsg コマンドでシグナルエラーが発生した。

機能名, 対象名, およびエラー詳細はエラー情報を示す。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5424-E

An error occurred during signal processing. (機能名 error 対象名)

bjexchmsg コマンドまたは bjexlsmmsg コマンドでシグナルエラーが発生した。

機能名, および対象名はエラー情報を示す。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5425-E

An error occurred during signal processing. (機能名 error -エラー詳細)

bjexchmsg コマンドまたは bjexlsmmsg コマンドでシグナルエラーが発生した。

機能名, およびエラー詳細はエラー情報を示す。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5426-E

An error occurred during signal processing. (機能名 error)

bjexchmsg コマンドまたは bjexlsmmsg コマンドでシグナルエラーが発生した。

機能名はエラー情報を示す。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5429-E

An error occurred. (保守情報)

内部エラーが発生した。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5498-E

An error occurred during コマンド名 signal handler processing.

bjexchmsg または bjexlsmmsg コマンドのシグナルハンドラの処理で、エラーが発生した。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKC5499-E

The コマンド名 ended abnormally because of an unexpected exception.

bjexchmsg または bjexlsmmsg コマンドが異常終了した。

(S)処理を終了する。

(O)システム管理者に連絡する。

KAKD1001-E

Memory shortage.

必要とする作業領域や入出力バッファを割り当てることができない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)システム管理者に連絡する。システム管理者は、メモリ見積もりを見直す。

KAKD1002-E

Environment variable "環境変数名" is invalid.

環境変数名で示す環境変数の設定値が正しくない。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。

KAKD1003-E

File defined by "環境変数名" open failed. reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルのオープン中にエラー詳細で示すエラーが発生した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

DD 要素 SYSPRINT でエラーが発生した場合は、このメッセージに続いて KAKD1091-E メッセージを出力し、RC=4 で続行する。

(O)ジョブ定義 XML ファイルの DD 要素を見直す。またはファイルシステムをチェックする。

KAKD1004-E

File defined by "環境変数名" I/O error. reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルの入出力中にエラー詳細で示すエラーが発生した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。またはファイルシステムをチェックする。

KAKD1005-E

File "ファイルパス名" defined by "環境変数名" open failed. reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルパス名で示されるパス名またはファイル名のファイルのオープン中にエラー詳細で示すエラーが発生した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。またはファイルシステムをチェックする。

KAKD1006-E

File "ファイルパス名" defined by "環境変数名" I/O error. reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルパス名で示されるパス名またはファイル名のファイルの入出力中にエラー詳細で示すエラーが発生した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。またはファイルシステムをチェックする。

KAKD1009-I

Line=行番号 text="テキスト".

制御文の行番号とテキストを出力する。

(S)処理を継続する。(RC=0)

KAKD1010-E

Length of the previous line exceeded the limit.

制御文のうち、直前の KAKD1009-I メッセージが示す行の文字数が最大文字数を越えた。

(S)制御文の解析処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文の行を分割して最大文字数以下にしたのち再実行する。

KAKD1011-E

No following line after continuation mark.

制御文の直前の KAKD1009-I メッセージが示す行の継続行がない。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文を見直す。

KAKD1012-E

Unmatched parenthesis.

制御文の直前の KAKD1009-I メッセージが示す行のオペランドの括弧が閉じられていない。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文の括弧を閉じる。

同時に KAKD1013-E メッセージが出力されているときは、制御文の引用符の指定が原因の場合があるので、引用符の指定を見直す。

KAKD1013-E

Unmatched quotation.

制御文の直前の KAKD1009-I メッセージが示す行のオペランドの引用符が閉じられていない。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文の引用符を閉じる。

KAKD1021-E

Command name is invalid.

誤った制御文を直前の KAKD1009-I メッセージが示す行に指定した。または、制御文の使用方法に誤りがある。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文を見直す。

KAKD1031-E

Operand name "オペランド名" is invalid.

誤ったオペランド名を指定した。または、オペランドの使用方法に誤りがある。指定内容は、直前のコマンドの KAKD1009-I メッセージが示す制御文に出力される。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文を見直す。

KAKD1032-E

Operand name "オペランド名" is not specified.

オペランド名で示す必須オペランドが指定されていない。

SELECT 文のオペランドが指定されていないときは、オペランド名に"GROUP or MEMBER"が表示される。

指定内容は、直前のコマンドの KAKD1009-I メッセージが示す制御文に出力される。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文を見直す。

KAKD1033-E

Operand name "オペランド名" is inconsistent.

オペランド名で示すオペランドの指定に矛盾がある。指定内容は、直前のコマンドの KAKD1009-I メッセージが示す制御文に出力される。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文を見直す。

KAKD1041-E

Parameter value "パラメータ値" of operand name "オペランド名" is invalid.

オペランド名で示すオペランドに誤ったパラメータ値を指定した。指定内容は、直前のコマンドの KAKD1009-I メッセージが示す制御文に出力される。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文を見直す。

KAKD1042-E

Parameter value of operand name "オペランド名" is not specified.

オペランド名で示すオペランドの必須パラメータが指定されていない。指定内容は、直前のコマンドの KAKD1009-I メッセージが示す制御文に出力される。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文を見直す。

KAKD1043-E

The DD element with NAME="環境変数名" is not specified.

ジョブ定義 XML ファイルの DD 要素には、NAME="環境変数名"に対応する DD 要素が指定されていない。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)ジョブ定義 XML ファイルの DD 要素に NAME="環境変数名"で示す環境変数名に対応する DD 要素を指定する。

KAKD1044-E

The sequence of Command name "コマンド名" is invalid.

コマンド名が示すコマンドの指定順序が誤っている。指定内容は、直前のコマンドの KAKD1009-I メッセージが示す制御文に出力される。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文を見直す。

KAKD1045-E

Parameter value of operand name "オペランド名" is invalid.

オペランド名で示すオペランドに誤ったパラメータ値を指定した。INDD コマンドの指定誤りのときも、このメッセージが出力される。指定内容は、直前のコマンドの KAKD1009-I メッセージが示す制御文に出力される。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文を見直す。

KAKD1046-E

Pathname "パス名" defined by "環境変数名" is too long.

環境変数名で示す環境変数に設定されているパス名の長さが、指定できる上限値を超えている。

(S)処理を打ち切り、ジョブステップを終了する。(RC=8)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。または、ファイルシステムをチェックする。

KAKD1047-E

File name "ファイル名" defined by MEMBER is too long.

MEMBER オペランドで指定したファイル名で示すファイルの名称の長さが、指定できる上限値を超えている。

(S)処理を打ち切り、ジョブステップを終了する。(RC=8)

(O)MEMBER オペランドの指定値を見直す。

KAKD1048-E

Combined pathname "パス名" is too long.

ジョブ定義 XML ファイルの DD 要素で指定したパス名と MEMBER オペランドで指定したファイル名を連結したパス名で示す名称の長さが、指定できる上限値を超えている。

(S)処理を打ち切り、ジョブステップを終了する。(RC=8)

(O)ジョブ定義 XML ファイルの DD 要素と MEMBER オペランドの指定値を見直す。

KAKD1090-E

SYSPRINT DD element missing.

ジョブ定義 XML ファイルの DD 要素には、SYSPRINT の DD 要素が指定されていない。

(S)SYSPRINT にはメッセージを出力しないで処理を継続する。(RC=4)

(O)ジョブ定義 XML ファイルを見直す。

KAKD1091-E

```
SYSPRINT cannot be opened.
```

ジョブ定義 XML ファイルで指定した SYSPRINT の DD 要素をオープンできなかった。

(S)SYSPRINT にはメッセージを出力しないで処理を継続する。(RC=4)

直前のエラーメッセージのリターンコードを無効にし、RC=4 で処理を継続する。

(O)ジョブ定義 XML ファイルを見直す。

KAKD1100-I

```
File "ファイル名" is copied.[ -> "新ファイル名"]
```

ファイル名で示すファイルのコピー処理が終了した。

新ファイル名を指定してコピーしたときは、コピーした新ファイル名が表示される。

(S)処理を続行する。(RC=0)

KAKD1102-I

```
All files defined by "環境変数名 1" is copied to "環境変数名 2" or skipped.
```

環境変数名 1 で示す環境変数に設定されているディレクトリ配下のファイルを環境変数名 2 で示すディレクトリ配下に複製する処理が終了した。

コピーステップが終了すると、出力される。処理中にエラーを検出し、処理をスキップしていることがあるので、エラーが発生しているときは、内容を確認する。

(S)処理を続行する。(RC=0)

(O)エラーが発生しているときは、ジョブ定義 XML ファイルの DD 要素または制御文を見直す。

KAKD1105-I

```
File "ファイル名" is not copied because same name file found.
```

ファイル名で示すファイルは、コピー先のディレクトリに同一ファイル名のファイルが存在したためコピーしなかった。

(S)処理を続行する。(RC=0)

KAKD1110-E

Failed to copy file "ファイル名". reason="エラー詳細".

ファイル名で示すファイルのコピーがエラー詳細で示すエラーによって失敗した。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)ジョブ定義 XML ファイルの DD 要素または制御文を見直す。

KAKD1112-E

Some file defined by "環境変数名 1" cannot be copied to "環境変数名 2".

環境変数名 1 で示す環境変数に設定されているディレクトリ配下のファイルを環境変数名 2 で示す環境変数に設定されているディレクトリ配下に複製中に、幾つかのファイルの複製処理に失敗した。

(S)処理を続行する。(RC=4)

(O)ジョブ定義 XML ファイルの DD 要素または制御文を見直す。

KAKD1120-E

"環境変数名" is not directory.

環境変数名で示す環境変数に設定されているパス名は、ディレクトリでない。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。

KAKD1121-E

SELECT or EXCLUDE is not allowed.

SELECT 文または EXCLUDE 文の指定はできない。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文を見直す。

KAKD1122-E

OUTDD or INDD keyword not found.

OUTDD または INDD のキーワードの指定がない。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文を見直す。

KAKD1130-E

File "ファイル名" not found.

SELECT 文で指定された**ファイル名**が見つからなかった。

(S)処理を続行する。(RC=4)

(O)入力ファイルまたは SELECT 文の MEMBER オペランドを調べて見直す。

KAKD1131-I

No files copied from input directory defined by "環境変数名".

環境変数名で示す環境変数に設定されている入力ディレクトリからは、ファイルをコピーできなかった。

(S)処理を続行する。(RC=0)

KAKD1137-E

Unable to specify duplicate file names "ファイル名" in same copy step.

1つのコピーステップで**ファイル名**を重複して指定しているので、ファイルをコピーできなかった。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=8)

(O)制御文を見直す。

KAKD1147-I

End of job - return code is "終了コード".

ジョブステップの終了を示す。**終了コード**で示す値が、このジョブステップ内で最も大きい値である。

(S)ジョブステップを終了する。(RC=0)

KAKD1151-I

Job terminated (error occurred).

bjexu_pcopy を実行したジョブステップで何らかのエラーが発生したことを示す。

(S)ジョブステップを終了する。(RC=0)

(O)このメッセージに先行するエラーメッセージを調べる。

KAKD1153-I

All files were already in place.

すべてのファイルはすでに存在するので、コピー処理を終了した。

(S)コピーオペレーションを打ち切り、次のコピーオペレーションを実行する。(RC=0)

(O)制御文を見直す。

KAKD2001-E

Memory shortage.

必要とする作業領域や入出力バッファを割り当てることができない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)システム管理者に連絡する。システム管理者は、メモリ見積もりを見直す。

KAKD2003-E

File defined by "環境変数名" open failed. reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルのオープン中にエラー詳細で示すエラーが発生した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

DD 要素 SYSPRINT でエラーが発生した場合は、このメッセージに続いて KAKD2091-E メッセージを出力し、RC=4 で続行する。

(O)ジョブ定義 XML ファイルの DD 要素を見直す。またはファイルシステムをチェックする。

KAKD2004-E

File defined by "環境変数名" I/O error. reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルの入出力中にエラー詳細で示すエラーが発生した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。またはファイルシステムをチェックする。

KAKD2005-E

File "ファイルパス名" defined by "環境変数名" open failed. reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルパス名で示されるパス名またはファイル名のファイルのオープン中にエラー詳細で示すエラーが発生した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。またはファイルシステムをチェックする。

KAKD2009-I

```
Line=行番号 text="テキスト".
```

制御文の行番号とテキストを出力する。

(S)処理を継続する。(RC=0)

KAKD2010-E

```
Length of the previous line exceeded the limit.
```

制御文の文字数が最大文字数を越えた。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文の行を分割して最大文字数以下にしたあと再実行する。

KAKD2011-E

```
No following line after continuation mark.
```

直前のコマンドの KAKD2009-I メッセージが示す制御文の継続行がない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD2012-E

```
Unmatched parenthesis.
```

直前のコマンドの KAKD2009-I メッセージが示す制御文のオペランドの括弧が閉じられていない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文の括弧を閉じる。

同時に KAKD2013-E メッセージが出力されているときは、制御文の引用符の指定が原因の場合があるので、引用符の指定を見直す。

KAKD2013-E

```
Unmatched quotation.
```

直前のコマンドの KAKD2009-I メッセージが示す制御文のオペランドの引用符が閉じられていない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文の引用符を閉じる。

KAKD2021-E

Command name is invalid.

誤った制御文を直前のコマンドの KAKD2009-I メッセージが示す制御文に指定した。または、コマンドの使用方法に誤りがある。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD2031-E

Operand name "オペランド名" is invalid.

誤ったオペランド名を指定した。または、オペランドの使用方法に誤りがある。指定内容は、直前のコマンドの KAKD2009-I メッセージが示す制御文に出力される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD2032-E

Operand name "オペランド名" is not specified.

オペランド名で示す必須オペランドが指定されていない。指定内容は、直前のコマンドの KAKD2009-I メッセージが示す制御文に出力される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD2041-E

Parameter value "パラメータ値" of operand name "オペランド名" is invalid.

オペランド名で示すオペランドに誤ったパラメータ値を指定した。指定内容は、直前のコマンドの KAKD2009-I メッセージが示す制御文に出力される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD2042-E

Parameter value of operand name "オペランド名" is not specified.

オペランド名で示すオペランドの必須パラメータが指定されていない。指定内容は、直前のコマンドの KAKD2009-I メッセージが示す制御文に出力される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD2043-E

The DD element with NAME="環境変数名" is not specified.

ジョブ定義 XML ファイルの DD 要素には、NAME="環境変数名"に対応する DD 要素が指定されていない。

(S)コマンド処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素に NAME="環境変数名"に対応する DD 要素を指定する。

KAKD2044-E

The sequence of Command name "コマンド名" is invalid.

コマンド名が示すコマンドの指定順序が誤っている。指定内容は、直前のコマンドの KAKD2009-I メッセージが示す制御文に出力される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

INFILE 文を制御文の最初に指定する必要がある。

OUTFILE 文は、INFILE 文の次に指定する必要がある。

GENERATE 文は、INFILE 文または OUTFILE 文のあとに指定する必要がある。

KAKD2045-E

Parameter value of operand name "オペランド名 1"[or "オペランド名 2"] is invalid.

"オペランド名 1"または"オペランド名 2"で示すオペランドのどちらかに誤ったパラメータ値を指定した。指定内容は、直前のコマンドの KAKD2009-I メッセージが示す制御文に出力される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD2046-E

Pathname "パス名" defined by "環境変数名" is too long.

環境変数名で示す環境変数に設定されているパス名の長さが指定できる上限値を超えている。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。または、ファイルシステムをチェックする。

KAKD2047-E

File name "ファイル名" defined by MEMBER is too long.

MEMBER オペランドで指定したファイル名で示すファイルの名称の長さが指定できる上限値を超えている。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)MEMBER オペランドの指定値を見直す。

KAKD2048-E

Combined pathname "パス名" is too long.

ジョブ定義 XML ファイルの DD 要素で指定したパス名と MEMBER オペランドで指定したファイル名を連結したパス名で示す名称の長さが指定できる上限値を超えている。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素と MEMBER オペランドの指定値を見直す。

KAKD2090-E

SYSPRINT DD element missing.

ジョブ定義 XML ファイルの DD 要素には、SYSPRINT の DD 要素が指定されていない。

(S)SYSPRINT にはメッセージを出力しないで処理を継続する。(RC=4)

(O)ジョブ定義 XML ファイルを見直す。

KAKD2091-E

SYSPRINT cannot be opened.

ジョブ定義 XML ファイルで指定した SYSPRINT の DD 要素をオープンできなかった。

(S)SYSPRINT にはメッセージを出力しないで処理を継続する。(RC=4)

(O)ジョブ定義 XML ファイルを見直す。

KAKD2100-I

```
Copied "ファイル名 1"->"ファイル名 2".
```

ファイル名 1 で示すファイルをファイル名 2 で示すファイルにコピーする処理が完了した。

(S)処理を継続する。(RC=0)

KAKD2101-E

```
Copied failed "ファイル名 1"->"ファイル名 2".
```

ファイル名 1 で示すファイルをファイル名 2 で示すファイルにコピーする処理に失敗した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素または制御文を見直す。

KAKD2102-I

```
Processing ended after last IDENT operand found.
```

RECORD 文の IDENT オペランドで最後に指定したグループ識別文字列を検出したので、入力データセットを最後まで入力しないで処理を終了した。

(S)処理を終了する。(RC=0)

KAKD2103-E

```
Processing member needs IDENT operand.
```

MEMBER 文と MEMBER 文との間に IDENT オペランドを指定した RECORD 文がない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD2104-E

```
Same member name is specified more than once.
```

MEMBER 文の NAME オペランド値に、同じ名称を複数回指定した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD2105-E

There is an error in specification order of "MEMBER" and "RECORD".

MEMBER 文と RECORD 文の指定順序に誤りがある。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD2106-E

File "ファイルパス名" defined by "環境変数名" stat error. reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルパス名で示されるファイルのステータス情報取得時、エラー詳細で示されるエラーが発生した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。またはファイルシステムをチェックする。

KAKD2107-E

File "ファイル名" is not copied because same name file was found.

ファイル名で示すファイルと同じ名称のファイルがすでに存在するので、コピーできなかった。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルを見直す。

KAKD2108-E

Unable to specify duplicate filepath "ファイルパス名" for SYSUT1 and SYSUT2.

ジョブ定義 XML ファイルの DD 要素 SYSUT2 で指定したファイルパス名が、SYSUT1 で指定したファイルと実体と同じである。SYSUT1 と SYSUT2 に同じファイルを指定することはできない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルを見直す。

KAKD2109-E

Missing IDENT operand for nonterminal RECORD statement.

最後の RECORD 文でないにもかかわらず、IDENT オペランドの指定がない。

(S)コマンド処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD2120-E

"ファイルパス名" specified by SYSUT1 is not a file.

ジョブ定義 XML ファイルの DD 要素 SYSUT1 で指定したファイルパス名で示す実体がファイルではない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルを見直す。

KAKD2121-E

"ファイルパス名" specified by SYSUT2 is not a file.

ジョブ定義 XML ファイルの DD 要素 SYSUT2 で指定したファイルパス名で示す実体がファイルではない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルを見直す。

KAKD2130-I

Copied last file "ファイル名 1"-"ファイル名 2".

ジョブ定義 XML ファイルの DD 要素 SYSUT1 に連結された最後のファイル名 1 で示すファイルをファイル名 2 で示すファイルにコピーする処理が終了した。

(S)処理を終了する。(RC=0)

KAKD2131-E

Output record length exceeded max record length of file specified by SYSUT2.

ジョブ定義 XML ファイルの DD 要素 SYSUT1 で指定したファイルのレコード長と SYSUT2 で指定した出力ファイルのレコード長の関連に矛盾を検出した。

入力ファイルのレコード長より出力ファイルのレコード長を大きくする必要がある。SYSUT1 で指定した入力ファイルの指定誤りか、OUTFILE 文の RECLLEN オペランドで指定した最大レコード長の誤りが考えられる。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルと制御文を見直す。

KAKD2132-E

Output record length does not reach the min record length of file specified by SYSUT2.

ジョブ定義 XML ファイルの DD 要素 SYSUT1 で指定したファイルから入力したレコード長が SYSUT2 で指定した出力ファイルの最小レコード長より小さい。

入力ファイルのレコード長は、出力ファイルの最小レコード長以上の長さでなければならない。SYSUT1 で指定した入力ファイルの指定誤りか、OUTFILE 文の MINLEN オペランドで指定した最小レコード長の誤りが考えられる。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルと制御文を見直す。

KAKD2133-E

Input position and/or length specified by "オペランド名" is invalid.

オペランド名で示すオペランドで指定した入力位置と入力する情報の長さで示すフィールドが、入力レコードまたは入力論理レコードのフィールド外を指定している。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルと制御文を見直す。

KAKD2134-E

Output position and/or length specified by "オペランド名" is invalid.

オペランド名で示すオペランドで指定した出力位置と出力する情報の長さで示すフィールドが、出力レコードの範囲外を指定している。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルと制御文を見直す。

KAKD2135-E

An error occurred while reading the file specified by SYSUT1.

ジョブ定義 XML ファイルの DD 要素 SYSUT1 で指定したファイルを読み込み中にエラーを検出した。入力ファイルの指定誤りか、INPFIL 文の指定誤りが考えられる。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルと制御文を見直す。

入力ファイルに空（ゼロバイト）ファイル（ダミーファイルを含む）を指定し、ジョブ定義 XML ファイルで入力ファイルに可変長順編成ファイルを指定して形式変換しようとしたときも出力される。単純複写するか、ジョブ定義 XML ファイルで入力ファイルに固定長順編成ファイルを指定する。

KAKD2136-E

DD concatenation is not allowed for PREST file.

PREST ファイルを DD 連結することはできない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルを見直す。

KAKD2137-E

Field length and field information length do not match in FIELD operand.

FIELD オペランドの指定で、フィールド長とフィールド情報の長さが異なる。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD2138-I

End of file. output records=出力レコード数.

ファイルの終わりを検出した。出力レコード数で示す、レコードを出力した。

単純複写の場合は、65537 バイト/レコードで出力したレコード数が表示される。

(S)処理を打ち切る。(RC=0)

KAKD2139-I

"仮定値" is assumed as output record length.

出力レコード長として、仮定値を仮定した。

(S)処理を継続する。(RC=0)

KAKD2140-I

Truncate input record to 80 bytes.

RECORD 文による複写で、入力レコードの長さを 80 バイトに切り詰めた。

(S)処理を継続する。(RC=0)

KAKD2150-E

An error occurred while reading the file specified by SYSUT2.

ジョブ定義 XML ファイルの DD 要素 SYSUT2 で指定したファイルを読み込み中にエラーを検出した。出力ファイルの指定誤りか、OUTFILE 文の指定誤りが考えられる。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルと制御文を見直す。

KAKD3001-E

Memory shortage.

必要とする作業領域や入出力バッファを割り当てることができない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)システム管理者に連絡する。システム管理者は、メモリ見積もりを見直す。

KAKD3002-E

Environment variable "環境変数名" is invalid.

環境変数名で示す環境変数の設定値が正しくない。次の原因が考えられる。

- REPRO 文で指定した入力または出力ファイルの DD 要素が定義されていない。
- REPRO 文で指定した入力ファイルのファイル種別は DD 連結できない。
- REPRO 文で指定した出力ファイルを DD 連結している。
- REPRO 文で指定した出力ファイルの DD 要素のファイル種別が不正である。
- REPRO 文の OUTDD オペランドと COPYDD オペランドで指定したファイルの種別が一致していない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。

KAKD3003-E

File defined by "環境変数名" open failed. reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルのオープン中にエラー詳細で示すエラーが発生した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。または、ファイルシステムをチェックする。

KAKD3004-E

File defined by "環境変数名" I/O error. reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルの入出力中にエラー詳細で示すエラーが発生した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

SYSPRINT の DD 要素でエラーが発生した場合は、このメッセージに続いて KAKD3091-E メッセージを出力し、RC=4 で続行する。

(O)ジョブ定義 XML ファイルの DD 要素を見直す。または、ファイルシステムをチェックする。

KAKD3005-E

File "ファイルパス名" defined by "環境変数名" open failed. reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルパス名で示されるパス名またはファイル名のファイルのオープン中にエラー詳細で示すエラーが発生した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=8)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。または、ファイルシステムをチェックする。

KAKD3007-E

File defined by "環境変数名" I/O error. reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルの入出力中にエラー詳細で示すエラーが発生した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=8)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。または、ファイルシステムをチェックする。

KAKD3009-I

Line=行番号 text="テキスト".

制御文の行番号とテキストを出力する。

(S)処理を継続する。(RC=0)

KAKD3010-E

Length of the previous line exceeded the limit.

制御文の文字数が最大文字数を越えた。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文の行を分割して最大文字数以下にしたあと再実行する。

KAKD3011-E

No following line after continuation mark.

直前のコマンドの KAKD3009-I が示す制御文の継続行がない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD3012-E

Unmatched parenthesis.

直前のコマンドの KAKD3009-I が示す制御文のオペランドの括弧が閉じていない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文の括弧を閉じる。

同時に KAKD3013-E メッセージが出力されている場合は、制御文の引用符が原因のときがあるため、引用符の指定を見直す。

KAKD3013-E

Unmatched quotation.

直前のコマンドの KAKD3009-I が示す制御文のオペランドの引用符が閉じていない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文の引用符を閉じる。

KAKD3021-E

Command name is invalid.

誤ったコマンド名を直前のコマンドの KAKD3009-I が示す制御文に指定した。または、コマンドの使用方法に誤りがある。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD3031-E

Operand name "オペランド名 1"[[, "オペランド名 2"] or "オペランド名 3"] is invalid.

"オペランド名 1", "オペランド名 2", または"オペランド名 3"で示すオペランドのどれかに誤ったオペランドを指定した。または、オペランドの使用方法に誤りがある。指定内容は、KAKD3009-I が示す制御文に出力される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD3032-E

Operand name "オペランド名 1"[or "オペランド名 2"] is not specified.

"オペランド名 1"または"オペランド名 2"で示す必須オペランドが指定されていない。指定内容は、直前のコマンドの KAKD3009-I メッセージが示す制御文に出力される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD3041-E

Parameter value "パラメータ値" of operand name "オペランド名" is invalid.

オペランド名で示すオペランドに誤ったパラメータ値を指定した。指定内容は、直前のコマンドの KAKD3009-I メッセージが示す制御文に出力される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD3042-E

Parameter value of operand name "オペランド名" is not specified.

オペランド名で示すオペランドの必須パラメータが指定されていない。指定内容は、直前のコマンドの KAKD3009-I メッセージが示す制御文に出力される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD3043-E

The DD element with NAME="環境変数名" is not specified.

ジョブ定義 XML ファイルの DD 要素には、NAME="環境変数名"に対応する DD 要素が指定されていない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素に NAME="環境変数名"の DD 要素を指定する。

KAKD3044-E

The sequence of Command name "コマンド名" is invalid.

コマンド名が示すコマンドの指定順序が誤っている。指定内容は、直前のコマンドの KAKD3009-I メッセージが示す制御文に出力される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

INFILE 文を制御文の最初に指定する必要がある。

OUTFILE 文は、INFILE 文の次に指定する必要がある。

REPRO 文は、OUTFILE 文のあとに指定する必要がある。

KAKD3045-E

Parameter value of operand name "オペランド名 1"[or "オペランド名 2"] is invalid.

"オペランド名 1"または"オペランド名 2"で示すオペランドのどちらかに誤ったパラメータ値を指定した。指定内容は、直前のコマンドの KAKD3009-I メッセージが示す制御文に出力される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)制御文を見直す。

KAKD3046-E

Pathname "パス名" defined by "環境変数名" is too long.

環境変数名で示す環境変数に設定されているパス名の長さが指定できる上限値を超えている。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。または、ファイルシステムをチェックする。

KAKD3048-E

Combined pathname "パス名" is too long.

ジョブ定義 XML ファイルの DD 要素で指定したパス名と複写元のファイル名を連結したパス名で示す名称の長さが指定できる上限値を超えている。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素の指定値を見直す。

KAKD3090-E

SYSPRINT DD element missing.

ジョブ定義 XML ファイルの DD 要素には、SYSPRINT の DD 要素が指定されていない。

(S)SYSPRINT にはメッセージを出力しないで処理を継続する。(RC=0)

(O)ジョブ定義 XML ファイルを見直す。

KAKD3091-E

SYSPRINT cannot be opened.

ジョブ定義 XML ファイルで指定した SYSPRINT の DD 要素をオープンできなかった。

(S)SYSPRINT にはメッセージを出力しないで処理を継続する。(RC=0)

(O)ジョブ定義 XML ファイルを見直す。

KAKD3092-E

SYSPRINT I/O error. reason="エラー詳細"

SYSPRINT への出力中に**エラー詳細**で示すエラーが発生した。

(S)SYSPRINT にはメッセージを出力しないで処理を継続する。(RC=4)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。または、ファイルシステムをチェックする。

KAKD3093-E

Required statement is not specified in SYSIN.

SYSIN に必須の制御文が指定されていない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルと制御文を見直す。

KAKD3100-I

{Copied | Converted | Merged} "ファイル名 1"->"ファイル名 2".

ファイル名1 で示すファイルをファイル名2 で示すファイルにコピーする処理が完了した。複写時は"Copied", 変換時は"Converted", また併合時は"Merged"が表示される。

(S)処理を継続する。(RC=0)

KAKD3101-I

ISAM file "ファイル名" was created successfully.

ファイル名で示す索引順編成ファイルの作成が完了した。

(S)処理を継続する。(RC=0)

KAKD3102-I

ISAM file "ファイル名" was erased successfully.

ファイル名で示す索引順編成ファイルの削除が完了した。

(S)処理を継続する。(RC=0)

KAKD3105-E

Attributes of files specified by COPYDD operand and OUTDD operand do not match.

REPRO コマンドの COPYDD オペランドで指定したファイルの属性が、OUTDD オペランドで指定したファイルの属性と異なる。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルと制御文を見直す。

KAKD3106-E

File "ファイルパス名" defined by "環境変数名" stat error. reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルパス名で示されるファイルのステータス情報の取得時、エラー詳細で示されるエラーが発生した。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。または、ファイルシステムをチェックする。

KAKD3107-E

File "ファイル名" is not copied because same name file was found.

ファイル名で示すファイルと同じ名称のファイルがすでにあるので、コピーできなかった。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルを見直す。

KAKD3108-E

Unable to specify duplicate filepath "ファイルパス名" for "環境変数名 1" and "環境変数名 2".

ジョブ定義 XML ファイルの DD 要素の環境変数名 2 で指定したファイルパス名が、環境変数名 1 で指定したファイルと実体と同じである。環境変数名 1 と環境変数名 2 に同じファイルを指定できない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルを見直す。

KAKD3109-E

An error occurred while {copying | converting | merging} file: "ファイル名 1"->"ファイル名 2".

ファイル名 1 で示すファイルをファイル名 2 で示すファイルにコピーする処理中にエラーが発生し、コピー処理に失敗した。複写時は"copying"、変換時は"converting"、また併合時は"merging"が表示される。レコードの途中までしか出力できなかったときに出力される。その他の要因があるときは、エラーの内容を示すメッセージが続けて出力される。

(S)コピー処理を打ち切り、ジョブステップを終了する。(RC=8)

(O)ジョブ定義 XML ファイルの DD 要素または制御文を見直す。または、ファイルシステムをチェックする。

KAKD3120-E

"ファイルパス名" specified by "環境変数名" is not a file.

環境変数名で示す環境変数に設定されているファイルパス名で示す実体がファイルではない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルを見直す。

KAKD3121-E

Error occurred on function("関数名") while processing file "ファイルパス名" defined by "環境変数名". reason="エラー詳細"

環境変数名で示す環境変数に設定されているファイルパス名で示されるファイルを使用中に関数名で示す関数でエラーが発生した。関数の戻り値にエラー詳細があるときは、エラーの内容がエラー詳細に表示される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=8)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。または、ファイルシステムをチェックする。

KAKD3122-E

```
Error occurred on function("関数名") before processing file "ファイルパス名" defined by "環境変数名". reason="エラー詳細"
```

環境変数名で示す環境変数に設定されているファイルパス名で示されるファイルを使用中に関数名で示す関数でエラーが発生した。関数の戻り値にエラー詳細があるときは、ISAM のエラー番号またはシステムコールのエラー番号がエラー詳細に表示される。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。または、ファイルシステムをチェックする。

KAKD3131-E

```
Input record length is out of range of file specified by "環境変数名".
```

入力ファイルのレコード長が環境変数名で示す環境変数に設定されている出力ファイルのレコード長の範囲外である。入力ファイルの指定誤りか、または環境変数名で示す出力ファイルの最小レコード長または最大レコード長の誤りが考えられる。

(S)処理を打ち切り、ジョブステップを終了する。(RC=8)

(O)ジョブ定義 XML ファイルと制御文を見直す。

KAKD3135-E

```
Inconsistent file attribute information in "環境変数名".
```

環境変数名で示す環境変数で指定したファイルの属性は、INPFILE 文または OUTFILE 文で指定したファイルの属性と矛盾している。次の原因が考えられる。

- INPFILE 文で FILEORG=F 指定の場合
入力ファイルのサイズが INPFILE 文の RECLEN オペランドで指定したレコード長の整数倍の大きさではない。
- INPFILE 文で FILEORG=V 指定の場合
入力ファイルのヘッダレコード長またはヘッダレコードの形式が不正である。
- OUTFILE 文で FILEORG=V 指定の場合

出力ファイルの DD 要素で DISP="MOD"指定のとき、出力ファイルのヘッダレコード長またはヘッダレコードの形式が不正である。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルと制御文を見直す。

KAKD3136-E

An error occurred while reading the file specified by "環境変数名".

環境変数名で示す環境変数に設定されているファイルを読み込み中にエラーを検出した。入力ファイルや出力ファイルの指定誤りか、または INPFILE 文や OUTFILE 文の指定誤りが考えられる。次の原因が考えられる。

- 入力が固定長順編成ファイルまたは可変長順編成ファイルの場合、入力を要求したレコード長と入力したレコードの長さが異なっている。
- 入力が可変長順編成ファイルの場合、入力したレコードの長さが、最小レコード長と最大レコード長の範囲外である。
- 出力が固定長順編成ファイルまたは可変長順編成ファイルの場合、出力を要求したデータ長と出力したレコードの長さが異なっている。
- 入力が可変長順編成ファイルの場合、ヘッダレコードだけがあり、データがなかった。

(S)処理を打ち切り、ジョブステップを終了する。(RC=8)

(O)ジョブ定義 XML ファイルと制御文を見直す。

KAKD3137-W

File "ファイルパス名" defined by "環境変数名" is not deleted, the file does not exist.

環境変数名で示す環境変数に設定されているファイルパス名で示されるファイルが存在しないので、削除できなかった。

(S)処理を続行する。(RC=4)

(O)ジョブ定義 XML ファイルの DD 要素を見直す。

KAKD3138-I

End of file. output records=出力レコード数,replace=変更レコード数,skip=スキップレコード数.

出力処理を終了した。出力レコード数で示すレコードを出力した。変更レコード数で示すレコードを更新した。また、スキップレコード数で示すレコードの処理をスキップした。表示されたレコード数の合計が処理したレコード数を表す。

(S)処理を継続する。(RC=0)

KAKD3139-E

The length of the input record is different from the output file record length.

入力ファイルと出力ファイルが固定長の順編成ファイル，または入力ファイルと出力ファイルが固定長の索引順編成ファイルのとき，入力レコードの長さとお出力ファイルのレコードの長さが異なっている。

(S)処理を打ち切る。(RC=12)

(O)ジョブ定義 XML ファイルと制御文を見直す。

KAKD3140-E

Attribute of ISAM file specified by "環境変数名" is not allowed.

環境変数名で示す環境変数で指定した ISAM ファイルの属性は，許可されていない属性である。

主キーを持たないファイルや主キーの重複を許可しているファイルを指定した。

キーのデータ型が整数型と文字型以外のファイルを指定した。

(S)処理を打ち切り，ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルを見直す。

KAKD3141-E

The data type of key specified by "オペランド名" is invalid.

オペランド名で示すオペランドで指定したキーのデータ型が入力ファイルまたは出力ファイルのデータ型と合わない。次の原因が考えられる。

- オペランドで指定したキーのデータ型とファイルに定義されているデータ型が合わない。
- オペランドで指定したキーのデータ型が文字型の場合，ファイルに定義されたキーの長さより長いキーの値を指定した。
- オペランドで指定したキーのデータ型が整数型の場合でファイルに定義されたキーのデータ型が 2 バイト整数型のとき，指定できる値の範囲 ($-32768 \leq \text{値} \leq 32767$) 以外の値を指定した。
- オペランド名で示すオペランドは，キーの要素が降順型のときは指定できない。

(S)処理を打ち切り，ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルと制御文を見直す。

KAKD3142-E

Key value is duplicated in file name "ファイル名".

ファイル名で示すファイルへの併合処理でキーの重複が見つかった。

(S)処理を打ち切り、ジョブステップを終了する。(RC=8)

(O)ジョブ定義 XML ファイルと制御文を見直す。

REPRO 文で NOREPLACE オペランドを指定したときは、更新が必要なら、REPLACE オペランドを指定して再実行する。

REPRO 文で REPLACE オペランドを指定していたときは、副キーの重複があるので、入出力ファイルの内容を確認する。

KAKD3143-E

Failed to load ISAM library. reason="エラー詳細"

索引順編成ファイル管理 ISAM のライブラリがロードできない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)エラー詳細を調べ、ISAM 製品が正しくインストールされているかどうかを確認する。

KAKD3144-E

The output file "ファイル名" specified by "環境変数名" is not empty, or "MOD" is specified to DISP attribute of DD element.

REPRO 文で COPYDD オペランド指定時、"環境変数名"で示す環境変数に設定されている"ファイル名"で示すファイルはすでにデータが格納されている。または、ジョブ定義 XML ファイルの DD 要素で、DISP="MOD"が指定されている。

なお、COPYDD オペランドを指定したときは、併合できない。

(S)処理を打ち切り、ジョブステップを終了する。(RC=12)

(O)ジョブ定義 XML ファイルと制御文を見直す。

KAKD3145-E

Failed to execute a command. (error=理由).

コマンド実行処理でエラーが発生した。

理由：エラー内容を表す文字列

理由の表示内容と説明を次に示す。

理由	説明
コマンド名:create sub process:エラー理由	サブプロセスの生成に失敗した。エラー理由は errno から求めた文字列である。
コマンド名:wait process:エラー理由	サブプロセスの終了待ちに失敗した。エラー理由は errno から求めた文字列である。
コマンド名:command:リターンコード	コマンドがリターンコードで終了した。
コマンド名:sub process:abnormal terminated	サブプロセスがシグナルなどで終了した。
コマンド名:sub process(execv):エラー理由	サブプロセスがコマンドの起動に失敗した。エラー理由は errno から求めた文字列である。

(S)コマンド実行処理を中断する。(RC=12)

(O)エラーになった理由を調べ、原因を取り除き、ジョブを再実行する。

10.6 JP1 イベント一覧

JP1 イベント一覧を次の表に示す。

表 10-8 JP1 イベント一覧

イベント ID	イベント発行契機	メッセージ
00005C00	ジョブコントローラ起動エラー時	KAKC0100-E
00005C01	ジョブ開始時	KAKC0091-I
00005C02	ジョブ終了時	KAKC0098-I
00005C03	ジョブ実行エラー時	KAKC0101-E
00005C04	ジョブステップ開始時	KAKC0092-I
00005C05	ジョブステップ終了時	KAKC0097-I
00005C06	ジョブステップ異常終了時	KAKC0096-E
00005C07	スプールジョブ転送サービスへの通知エラー時	KAKC0013-E
00005C08	ファイルおよびディレクトリのロック待ち発生時	KAKC1650-W
00005C09	ファイルおよびディレクトリのロック待ち解除時	KAKC1651-I
00005C0A	ジョブ多重度によるジョブ実行待ち発生時	KAKC1652-W
00005C0B	ジョブ多重度によるジョブ実行待ち解除時	KAKC1653-I
00005C0D	排他資源管理テーブルの空き待ち解除時	KAKC1655-I
00005C0E	排他資源管理テーブルの空き待ち発生時	KAKC1654-W
00005C10	スプールジョブ転送サービス起動時	KAKC3000-I
00005C11	スプールジョブ転送サービス起動エラー時	KAKC3002-E
00005C12	スプールジョブ転送サービス正常終了時	KAKC3001-I
00005C13	スプールジョブ転送サービス異常終了時	KAKC3003-E
00005C14	転送障害発生時	KAKC3004-E
00005C16	スプールジョブの転送エラー時	KAKC3204-I
00005C17	スプールジョブ転送サービスの実行ログ出力エラー発生時	KAKC3412-E
00005C18	スプールジョブ転送デーモンでスプールディレクトリの再作成検知時	KAKC3031-E
00005C19	スプールジョブの再転送回数が指定のリトライ回数を越えた時	KAKC3130-I
00005C20	メッセージ制御機能によるメッセージ出力時	COBOL アプリケーション またはメッセージ出力コマンド (bjexm_message コマンド) が出力するメッセージ

イベント ID	イベント発行契機	メッセージ
00005C21	メッセージ制御機能による応答要求メッセージ出力時	COBOL アプリケーション またはメッセージ出力コマ ンド (bjexm_message コ マンド) が出力するメッ セージ
00005C22	ジョブコントローラによる応答待ちイベントのキャンセル時	KAKC5008-I
00005C23	スプールジョブ転送サービスによる応答待ちイベントのキャンセル時	KAKC3508-I
00005C24	bjexchmsg コマンドによる応答待ちイベントのキャンセル時	KAKC5408-I
00005C25	メッセージ制御機能による応答要求メッセージに対する応答入力時	KAKC5001-I

10.7 JP1 イベントの詳細

JP1 イベントの詳細をイベント ID 別に次の表に示す。

なお、表中の JP1/AJS のジョブ情報は、次のように表示されることを示す。

UNIX ジョブの場合

設定ファイルの JP1EVENT_JOBINFO パラメータを未指定、または JP1AJS を指定した場合
次の環境変数の値を順番にコロン（:）で連結した文字列となる。

- AJS_AJSCONF
- AJSJOBNAME
- AJSEXECID

設定ファイルの JP1EVENT_JOBINFO パラメータに BJEX を指定した場合
bjexec となる。

サブミットジョブの場合

bjexec となる。

表 10-9 イベント ID (00005C00) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	—	00005C00
		メッセージ	—	KAKC0100-E bjexec の起動処理でエラーが発生しました
拡張属性	共通情報	重大度	SEVERITY	Error
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOB
		オブジェクト名	OBJECT_NAME	<ul style="list-style-type: none"> • UNIX ジョブの場合 JP1/AJS のジョブ情報:bjexec • サブミットジョブの場合 bjexec
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報
		事象種別	OCCURRENCE	NOTSTART
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		エラー詳細	ERROR_INFO	エラー詳細

(凡例)

－：なし。

表 10-10 イベント ID (00005C01) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C01
		メッセージ	－	KAKC0091-I ジョブ (JP1/AJS のジョブ情報：ジョブ名) を開始します
拡張属性	共通情報	重大度	SEVERITY	Information
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOB
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報
		事象種別	OCCURRENCE	START
		開始時刻	START_TIME	ジョブ開始時刻
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		ジョブ通し番号	JOB_ID	ジョブ通し番号
ジョブ名		JOB_NAME	ジョブ名	

(凡例)

－：なし。

表 10-11 イベント ID (00005C02) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C02
		メッセージ	－	KAKC0098-I ジョブ (JP1/AJS のジョブ情報：ジョブ名) が終了しました
拡張属性	共通情報	重大度	SEVERITY	Information
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOB
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名

属性種別		項目	属性名	内容
拡張属性	共通情報	登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報
		事象種別	OCCURRENCE	END
		開始時刻	START_TIME	ジョブ開始時刻
		終了時刻	END_TIME	ジョブ終了時刻
		終了コード	RESULT_CODE	終了コード
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		ジョブ通し番号	JOB_ID	ジョブ通し番号
		ジョブ名	JOB_NAME	ジョブ名

(凡例)

－：なし。

表 10-12 イベント ID (00005C03) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C03
		メッセージ	－	KAKC0101-E ジョブ (JP1/AJS のジョブ情報：ジョブ名) の実行中にエラーが発生しました
拡張属性	共通情報	重大度	SEVERITY	Error
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOB
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報
		事象種別	OCCURRENCE	END
		開始時刻	START_TIME	ジョブ開始時刻
		終了時刻	END_TIME	ジョブ終了時刻
		終了コード	RESULT_CODE	終了コード
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		ジョブ通し番号	JOB_ID	ジョブ通し番号

属性種別		項目	属性名	内容
拡張属性	固有情報	ジョブ名	JOB_NAME	ジョブ名
		エラー詳細	ERROR_INFO	エラー詳細

(凡例)

－：なし。

表 10-13 イベント ID (00005C04) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C04
		メッセージ	－	KAKC0092-I ジョブステップ (JP1/AJS のジョブ情報：ジョブ名、ジョブステップ名 [.プロシジャステップ名]) を開始します
拡張属性	共通情報	重大度	SEVERITY	Information
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOBSTEP
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名、ジョブステップ名 [.プロシジャステップ名]
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		事象種別	OCCURRENCE	START
		開始時刻	START_TIME	ジョブステップ開始時刻
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		ジョブ通し番号	JOB_ID	ジョブ通し番号
		ジョブ名	JOB_NAME	ジョブ名
		ジョブステップ番号	STEP_ID	ジョブステップ番号
		ジョブステップ名	STEP_NAME	ジョブステップ名 [.プロシジャステップ名]

(凡例)

－：なし。

表 10-14 イベント ID (00005C05) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C05

属性種別		項目	属性名	内容
基本属性		メッセージ	—	KAKC0097-I ジョブステップ (JP1/AJS のジョブ情報：ジョブ名.ジョブステップ名 [.プロシジャステップ名]) が終了しました
拡張属性	共通情報	重大度	SEVERITY	Information
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOBSTEP
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名.ジョブステップ名 [.プロシジャステップ名]
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		事象種別	OCCURRENCE	END
		開始時刻	START_TIME	ジョブステップ開始時刻
		終了時刻	END_TIME	ジョブステップ終了時刻
		終了コード	RESULT_CODE	終了コード
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		ジョブ通し番号	JOB_ID	ジョブ通し番号
		ジョブ名	JOB_NAME	ジョブ名
		ジョブステップ番号	STEP_ID	ジョブステップ番号
		ジョブステップ名	STEP_NAME	ジョブステップ名 [.プロシジャステップ名]

(凡例)

—：なし。

表 10-15 イベント ID (00005C06 の詳細)

属性種別		項目	属性名	内容
基本属性		イベント ID	—	00005C06
		メッセージ	—	KAKC0096-E ジョブステップ (JP1/AJS のジョブ情報：ジョブ名.ジョブステップ名 [.プロシジャステップ名]) が異常終了しました
拡張属性	共通情報	重大度	SEVERITY	Error
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名

属性種別		項目	属性名	内容
拡張属性	共通情報	プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOBSTEP
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名.ジョブステップ名 [.プロシジャステップ名]
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		事象種別	OCCURRENCE	END
		開始時刻	START_TIME	ジョブステップ開始時刻
		終了時刻	END_TIME	ジョブステップ終了時刻
		終了コード	RESULT_CODE	終了コード
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		ジョブ通し番号	JOB_ID	ジョブ通し番号
		ジョブ名	JOB_NAME	ジョブ名
		ジョブステップ番号	STEP_ID	ジョブステップ番号
		ジョブステップ名	STEP_NAME	ジョブステップ名 [.プロシジャステップ名]
		エラー詳細	ERROR_INFO	エラー詳細

(凡例)

－：なし。

表 10-16 イベント ID (00005C07 の詳細)

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C07
		メッセージ	－	KAKC0013-E スプールジョブ転送サービスへの通知に失敗しました
拡張属性	共通情報	重大度	SEVERITY	Error
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOB
		オブジェクト名	OBJECT_NAME	<ul style="list-style-type: none"> UNIX ジョブの場合 JP1/AJS のジョブ情報： bjexec

属性種別		項目	属性名	内容
拡張属性	共通情報	オブジェクト名	OBJECT_NAME	• サブミットジョブの場合 bjexec
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報
		事象種別	OCCURRENCE	EXCEPTION
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		エラー詳細	ERROR_INFO	エラー詳細

(凡例)

－：なし。

表 10-17 イベント ID (00005C08 の詳細)

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C08
		メッセージ	－	KAKC1650-W ロック確保待ちが発生しました
拡張属性	共通情報	重大度	SEVERITY	Warning
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOB
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報
		事象種別	OCCURRENCE	PAUSE
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		ジョブ通し番号	JOB_ID	ジョブ通し番号
		ジョブ名	JOB_NAME	ジョブ名

(凡例)

－：なし。

表 10-18 イベント ID (00005C09 の詳細)

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C09

属性種別		項目	属性名	内容
基本属性		メッセージ	—	KAKC1651-I ロック待ちは解除されました
拡張属性	共通情報	重大度	SEVERITY	Information
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOB
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報
		事象種別	OCCURRENCE	RELEASE
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		ジョブ通し番号	JOB_ID	ジョブ通し番号
ジョブ名		JOB_NAME	ジョブ名	

(凡例)

—：なし。

表 10-19 イベント ID (00005C0A の詳細)

属性種別		項目	属性名	内容
基本属性		イベント ID	—	00005C0A
		メッセージ	—	KAKC1652-W ジョブ実行待ちが発生しました
拡張属性	共通情報	重大度	SEVERITY	Warning
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOB
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報
		事象種別	OCCURRENCE	PAUSE
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		ジョブ通し番号	JOB_ID	ジョブ通し番号

属性種別		項目	属性名	内容
拡張属性	固有情報	ジョブ名	JOB_NAME	ジョブ名

(凡例)

－：なし。

表 10-20 イベント ID (00005C0B の詳細)

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C0B
		メッセージ	－	KAKC1653-I ジョブ実行待ちは解除されました
拡張属性	共通情報	重大度	SEVERITY	Information
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOB
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報
	事象種別	OCCURRENCE	RELEASE	
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		ジョブ通し番号	JOB_ID	ジョブ通し番号
ジョブ名		JOB_NAME	ジョブ名	

(凡例)

－：なし。

表 10-21 イベント ID (00005C0D の詳細)

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C0D
		メッセージ	－	KAKC1655-I 排他資源管理テーブルの空き待ちが解除されました
拡張属性	共通情報	重大度	SEVERITY	Information
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOB
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名

属性種別		項目	属性名	内容
拡張属性	共通情報	登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報
		事象種別	OCCURRENCE	RELEASE
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		ジョブ通し番号	JOB_ID	ジョブ通し番号
		ジョブ名	JOB_NAME	ジョブ名

(凡例)

－：なし。

表 10-22 イベント ID (00005C0E の詳細)

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C0E
		メッセージ	－	KAKC1654-W 排他資源管理 テーブルの空き待ちが発生しました
拡張属性	共通情報	重大度	SEVERITY	Warning
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOB
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報
		事象種別	OCCURRENCE	PAUSE
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	ジョブ定義 XML ファイル名
		ジョブ通し番号	JOB_ID	ジョブ通し番号
		ジョブ名	JOB_NAME	ジョブ名

(凡例)

－：なし。

表 10-23 イベント ID (00005C10 の詳細)

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C10

属性種別		項目	属性名	内容
基本属性		メッセージ	—	KAKC3000-I bjexpd が起動しました。
拡張属性	共通情報	重大度	SEVERITY	Information
		ユーザ名	USER_NAME	実行ユーザ
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	SERVICE
		オブジェクト名	OBJECT_NAME	BJEX-Base Print Job Transfer Service
		登録名タイプ	ROOT_OBJECT_TYPE	SERVICE
		登録名	ROOT_OBJECT_NAME	BJEX-Base Print Job Transfer Service
		事象種別	OCCURRENCE	START
		開始時刻	START_TIME	開始時刻
	固有情報	エラー詳細	ERROR_INFO	エラー詳細

(凡例)

—：なし。

表 10-24 イベント ID (00005C11 の詳細)

属性種別		項目	属性名	内容
基本属性		イベント ID	—	00005C11
		メッセージ	—	KAKC3002-E bjexpd が起動できません。
拡張属性	共通情報	重大度	SEVERITY	Error
		ユーザ名	USER_NAME	実行ユーザ
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	SERVICE
		オブジェクト名	OBJECT_NAME	BJEX-Base Print Job Transfer Service
		登録名タイプ	ROOT_OBJECT_TYPE	SERVICE
		登録名	ROOT_OBJECT_NAME	BJEX-Base Print Job Transfer Service
		事象種別	OCCURRENCE	NOTSTART
			終了コード	RESULT_CODE
		固有情報	エラー詳細	ERROR_INFO

(凡例)

－：なし。

表 10-25 イベント ID (00005C12) の詳細

属性種別	項目	属性名	内容	
基本属性	イベント ID	－	00005C12	
	メッセージ	－	KAKC3001-I bjexpd が終了しました。	
拡張属性	共通情報	重大度	SEVERITY	Information
		ユーザ名	USER_NAME	実行ユーザ
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	SERVICE
		オブジェクト名	OBJECT_NAME	BJEX-Base Print Job Transfer Service
		登録名タイプ	ROOT_OBJECT_TYPE	SERVICE
		登録名	ROOT_OBJECT_NAME	BJEX-Base Print Job Transfer Service
		事象種別	OCCURRENCE	END
		開始時刻	START_TIME	開始時刻
		終了時刻	END_TIME	終了時刻

(凡例)

－：なし。

表 10-26 イベント ID (00005C13) の詳細

属性種別	項目	属性名	内容	
基本属性	イベント ID	－	00005C13	
	メッセージ	－	KAKC3003-E bjexpd が異常終了しました。	
拡張属性	共通情報	重大度	SEVERITY	Error
		ユーザ名	USER_NAME	実行ユーザ
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	SERVICE
		オブジェクト名	OBJECT_NAME	BJEX-Base Print Job Transfer Service
		登録名タイプ	ROOT_OBJECT_TYPE	SERVICE

属性種別		項目	属性名	内容
拡張属性	共通情報	登録名	ROOT_OBJECT_NAME	BJEX-Base Print Job Transfer Service
		事象種別	OCCURRENCE	END
		開始時刻	START_TIME	開始時刻
		終了時刻	END_TIME	終了時刻
		終了コード	RESULT_CODE	終了コード
	固有情報	エラー詳細	ERROR_INFO	エラー詳細

(凡例)

－：なし。

表 10-27 イベント ID (00005C14) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C14
		メッセージ	－	KAKC3004-E bjexpd で JP1/FTP エラーが発生しました。
拡張属性	共通情報	重大度	SEVERITY	Error
		ユーザ名	USER_NAME	実行ユーザ
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	SERVICE
		オブジェクト名	OBJECT_NAME	BJEX-Base Print Job Transfer Service
		登録名タイプ	ROOT_OBJECT_TYPE	SERVICE
		登録名	ROOT_OBJECT_NAME	BJEX-Base Print Job Transfer Service
	事象種別	OCCURRENCE	NOTICE	

(凡例)

－：なし。

表 10-28 イベント ID (00005C16) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C16
		メッセージ	－	KAKC3204-I スプールジョブの ファイル転送が中断されました。
拡張属性	共通情報	重大度	SEVERITY	Information
		ユーザ名	USER_NAME	実行ユーザ

属性種別		項目	属性名	内容
拡張属性	共通情報	プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	SERVICE
		オブジェクト名	OBJECT_NAME	BJEX-Base Print Job Transfer Service
		登録名タイプ	ROOT_OBJECT_TYPE	SERVICE
		登録名	ROOT_OBJECT_NAME	BJEX-Base Print Job Transfer Service
	事象種別	OCCURRENCE	NOTICE	
	固有情報	エラー詳細	ERROR_INFO	エラー詳細

(凡例)

－：なし。

表 10-29 イベント ID (00005C17) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C17
		メッセージ	－	KAKC3412-E bjexpd の実行ログ出力処理でエラーが発生しました。
拡張属性	共通情報	重大度	SEVERITY	Error
		ユーザ名	USER_NAME	root
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	SERVICE
		オブジェクト名	OBJECT_NAME	BJEX-Base Print Job Transfer Service
		登録名タイプ	ROOT_OBJECT_TYPE	SERVICE
		登録名	ROOT_OBJECT_NAME	BJEX-Base Print Job Transfer Service
	事象種別	OCCURRENCE	NOTICE	
	固有情報	エラー詳細	ERROR_INFO	エラー発生時に出力しようとしていたメッセージ

(凡例)

－：なし。

表 10-30 イベント ID (00005C18) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C18

属性種別		項目	属性名	内容
基本属性		メッセージ	—	KAKC3031-E bjexpd が使用しているスプールディレクトリが再作成されました。
拡張属性	共通情報	重大度	SEVERITY	Error
		ユーザ名	USER_NAME	root
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	SERVICE
		オブジェクト名	OBJECT_NAME	BJEX-Base Print Job Transfer Service
		登録名タイプ	ROOT_OBJECT_TYPE	SERVICE
		登録名	ROOT_OBJECT_NAME	BJEX-Base Print Job Transfer Service
	事象種別	OCCURRENCE	NOTICE	
	固有情報	エラー詳細	ERROR_INFO	エラー情報（新旧 inode 番号およびファイル/ディレクトリ名が含まれる）

(凡例)

—：なし。

表 10-31 イベント ID (00005C19) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	—	00005C19
		メッセージ	—	KAKC3130-I スプールジョブの再転送回数が指定のリトライ回数を超えました。
拡張属性	共通情報	重大度	SEVERITY	Information
		ユーザ名	USER_NAME	実行ユーザ
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	SERVICE
		オブジェクト名	OBJECT_NAME	BJEX-Base Print Job Transfer Service
		登録名タイプ	ROOT_OBJECT_TYPE	SERVICE
		登録名	ROOT_OBJECT_NAME	BJEX-Base Print Job Transfer Service
	事象種別	OCCURRENCE	NOTICE	

(凡例)

－：なし。

表 10-32 イベント ID (00005C20) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C20
		発行元イベントサーバ名	－	バッチジョブ実行システムが稼働するホスト名
		メッセージ	－	メッセージ (ジョブ識別子, ステップ番号, JP1/AJS のホスト名, メッセージ識別子, 保守情報)
拡張属性	共通情報	重大度	SEVERITY	Notice
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOBSTEP
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		事象種別	OCCURRENCE	NOTICE
	固有情報	ジョブ通し番号	JOB_ID	ジョブ通し番号
		ジョブステップ番号	STEP_ID	ジョブステップ番号
		ジョブ名	JOB_NAME	ジョブ名
		ジョブステップ名	STEP_NAME	ジョブステップ名 [.プロシジャステップ名]

(凡例)

－：なし。

表 10-33 イベント ID (00005C21) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C21
		発行元イベントサーバ名	－	バッチジョブ実行システムが稼働するホスト名
		メッセージ	－	応答要求メッセージ (ジョブ識別子, ステップ番号, JP1/AJS のホスト名, メッセージ識別子, 保守情報)
拡張属性	共通情報	重大度	SEVERITY	Notice

属性種別		項目	属性名	内容
拡張属性	共通情報	ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOBSTEP
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		事象種別	OCCURRENCE	RESPONSE
	固有情報	ジョブ通し番号	JOB_ID	ジョブ通し番号
		ジョブステップ番号	STEP_ID	ジョブステップ番号
		ジョブ名	JOB_NAME	ジョブ名
		ジョブステップ名	STEP_NAME	ジョブステップ名 [.プロシジャステップ名]

(凡例)

－：なし。

表 10-34 イベント ID (00005C22) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C22
		発行元イベントサーバ名	－	バッチジョブ実行システムが稼働するホスト名
		メッセージ	－	KAKC5008-I 応答要求をキャンセルしました (ジョブ識別子, ステップ番号, JP1/AJS のホスト名)
拡張属性	共通情報	重大度	SEVERITY	Notice
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOBSTEP
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報
		事象種別	OCCURRENCE	CANCEL
	固有情報	ジョブ通し番号	JOB_ID	ジョブ通し番号
		ジョブステップ番号	STEP_ID	ジョブステップ番号

(凡例)

－：なし。

表 10-35 イベント ID (00005C23) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C23
		発行元イベントサーバ名	－	バッチジョブ実行システムが稼働するホスト名
		メッセージ	－	KAKC3508-I 応答要求をキャンセルしました (ジョブ識別子, ステップ番号, JP1/AJS のホスト名)
拡張属性	共通情報	重大度	SEVERITY	Notice
		ユーザ名	USER_NAME	実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	SERVICE
		オブジェクト名	OBJECT_NAME	BJEX-Base Print Job Transfer Service
		登録名タイプ	ROOT_OBJECT_TYPE	SERVICE
		登録名	ROOT_OBJECT_NAME	BJEX-Base Print Job Transfer Service
		事象種別	OCCURRENCE	CANCEL
	固有情報	ジョブ通し番号	JOB_ID	ジョブ通し番号
		ジョブステップ番号	STEP_ID	ジョブステップ番号

(凡例)

－：なし。

表 10-36 イベント ID (00005C24) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C24
		発行元イベントサーバ名	－	バッチジョブ実行システムが稼働するホスト名
		メッセージ	－	KAKC5408-I 応答要求をキャンセルしました (ジョブ識別子, ステップ番号, JP1/AJS のホスト名)
拡張属性	共通情報	重大度	SEVERITY	Notice
		ユーザ名	USER_NAME	実行ユーザ名

属性種別		項目	属性名	内容
拡張属性	共通情報	プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	COMMAND
		オブジェクト名	OBJECT_NAME	コマンド名
		登録名タイプ	ROOT_OBJECT_TYPE	COMMAND
		登録名	ROOT_OBJECT_NAME	コマンド名
		事象種別	OCCURRENCE	CANCEL
	固有情報	ジョブ通し番号	JOB_ID	ジョブ通し番号
		ジョブステップ番号	STEP_ID	ジョブステップ番号

(凡例)

－：なし。

表 10-37 イベント ID (00005C25) の詳細

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C25
		発行元イベントサーバ名	－	バッチジョブ実行システムが稼働するホスト名
		メッセージ	－	KAKC5001-I 応答メッセージ (ジョブ識別子, ジョブステップ番号, ホスト名, 応答者, 保守情報)
拡張属性	共通情報	重大度	SEVERITY	Notice
		ユーザ名	USER_NAME	JP1 ユーザ名または実行ユーザ名
		プロダクト名	PRODUCT_NAME	/HITACHI/BJEX
		オブジェクトタイプ	OBJECT_TYPE	JOBSTEP
		オブジェクト名	OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		登録名タイプ	ROOT_OBJECT_TYPE	JOB
		登録名	ROOT_OBJECT_NAME	JP1/AJS のジョブ情報：ジョブ名
		事象種別	OCCURRENCE	NOTICE
	固有情報	ジョブ通し番号	JOB_ID	ジョブ通し番号
		ジョブステップ番号	STEP_ID	ジョブステップ番号
		ジョブ名	JOB_NAME	ジョブ名
		ジョブステップ名	STEP_NAME	ジョブステップ名 [.プロシジャステップ名]

(凡例)

－：なし。

10.8 終了コード

10.8.1 ジョブステップの終了コード

ジョブステップの終了コードは、ジョブ定義 XML ファイルの EXEC 要素に記述したプログラムの終了コードである。

ジョブコントローラが EXEC 要素に記述したプログラムの終了コードを取得できた場合は、終了コードを KAKC0097-I メッセージに出力する。EXEC 要素に記述したプログラムが異常終了した場合は、終了コードを KAKC0096-E メッセージに出力する。

ジョブステップの終了コード一覧を次の表に示す。

表 10-38 ジョブステップの終了コード一覧

終了コード	意味
0~125	EXEC 要素に記述したプログラムの終了コードが 0~125 である。
126	次のどちらかである。 <ul style="list-style-type: none">EXEC 要素に記述したプログラムの終了コードが 126 である。EXEC 要素に記述したプログラムが存在するが実行できない。
127	次のどちらかである。 <ul style="list-style-type: none">EXEC 要素に記述したプログラムの終了コードが 127 である。EXEC 要素に記述したプログラムが存在しない。
128+N*	次のどちらかである。 <ul style="list-style-type: none">EXEC 要素に記述したプログラムの終了コードが 128+N である。EXEC 要素に記述したプログラムがシグナルコード N で異常終了した。
240~255	EXEC 要素に記述したプログラムの終了コードが 240~255 である。

注※ シグナルで打ち切られた場合のシグナルコード

10.8.2 ジョブコントローラの終了コード

(1) 終了コードの定義

JOB_RCDEFINE パラメータで STEPMAX を指定した場合

ジョブコントローラの終了コードは、ジョブステップの最大の終了コードである。

JOB_RCDEFINE パラメータで STEPLAST を指定した場合

ジョブコントローラの終了コードは、最後に実行完了したメインステップの終了コードである。

ただし、終了コードが 241 以上の場合は、ジョブコントローラでエラーが発生したことを示す。ジョブステップの最大の終了コード、または最終ジョブステップの終了コードが 240 以上の場合は、ジョブコントローラの終了コードを 240 とする。

ジョブコントローラの終了コード一覧を次の表に示す。

表 10-39 ジョブコントローラの終了コード一覧

終了コード	意味
0	次のどれかである。 <ul style="list-style-type: none"> • -v オプションを指定した場合、正常終了した。 • -c オプションを指定した場合、ジョブ定義 XML ファイルの誤りを検出することなく正常終了した。 • JOB_RCDEFINE パラメータで STEPMAX を指定した場合 ジョブを実行した場合、ジョブステップの最大の終了コードが 0 である。 • JOB_RCDEFINE パラメータで STEPLAST を指定した場合 ジョブを実行した場合、最終ジョブステップの終了コードが 0 である。 • コマンドは正常に処理を終了した。
1～239	<ul style="list-style-type: none"> • JOB_RCDEFINE パラメータで STEPMAX を指定した場合 ジョブステップの最大の終了コードが 1～239 である。 • JOB_RCDEFINE パラメータで STEPLAST を指定した場合 最終ジョブステップの終了コードが 1～239 である。
240	<ul style="list-style-type: none"> • JOB_RCDEFINE パラメータで STEPMAX を指定した場合 ジョブステップの最大の終了コードが 240～255 である。 • JOB_RCDEFINE パラメータで STEPLAST を指定した場合 最終ジョブステップの終了コードが 240～255 である。
242	次に示すエラーを検出したため、処理を終了した。 <ul style="list-style-type: none"> • ファイルの割り当てに失敗した。 • ディレクトリの割り当てに失敗した。 • SETENVFILE 要素で示す環境変数格納ファイル中の環境変数設定に失敗した。
244	次に示すエラーを検出したため、ジョブステップを実行しないで処理を終了した。 <ul style="list-style-type: none"> • 入力パラメータの文法エラー • 設定ファイルの文法エラー • ジョブ定義 XML ファイルの文法エラー • 環境変数の設定エラー • ユーザ名の取得エラー • ステップリスタートが指示されたがリスタートできないジョブステップが指定された。または、ジョブステップが見つからない。
	次に示すエラーを検出したため、処理を終了した。 <ul style="list-style-type: none"> • DD 要素で指定した一時ファイルが存在しない。 • DD 要素で指定した一時 ISAM ファイルが存在しない。 • 世代ファイルの割り当てエラー

終了コード	意味
245	次に示すエラーを検出したため、ジョブステップを実行しないで処理を終了した。 <ul style="list-style-type: none"> 設定ファイルのオープンエラー、入出力エラー ジョブ定義 XML ファイルのオープンエラー、入出力エラー ジョブログファイルのオープンエラー、入出力エラー ISAM ライブラリのロードエラー
246	次に示すエラーを検出したため、処理を終了した。 <ul style="list-style-type: none"> ジョブステッププロセス生成エラー ジョブステッププロセス終了待ちエラー
250	メモリ割り当てエラーなど、続行できないエラーを検出したため、処理を終了した。
253	メッセージを出力できなかったため、処理を終了した。
254	プログラムの内部矛盾検出など、内部エラーを検出したため、処理を中断した。
255	強制終了を指示されたため、ジョブの実行を中止し、処理を終了した。

注 ジョブコントローラの終了コードからジョブステップの実行結果を把握するために、EXEC 要素に記述するプログラムの終了コードは、125 以下とすることを推奨する。また、SETMAXRC 要素でジョブステップの最大の終了コード、または最終ジョブステップの終了コードを変更する場合も、125 以下とすることを推奨する。

(2) 終了コードの例

次のジョブについて、JOB_RCDEFINE パラメータで STEPMAX を指定した場合と STEPLAST を指定した場合のそれぞれについて、ジョブコントローラの終了コードを示す。

ジョブ定義 XML ファイル

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="JOB_RC">
    <STEP NAME="STEP1"> <!-- 終了コード4で終了するステップ -->
      <EXEC PGM="RTN" PARM="4" />
    </STEP>
    <STEP NAME="STEP2"> <!-- 終了コード8で終了するステップ -->
      <EXEC PGM="RTN" PARM="8" />
    </STEP>
    <STEP NAME="STEP3"> <!-- 終了コード0で終了するステップ -->
      <EXEC PGM="RTN" PARM="0" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

表 10-40 ジョブコントローラの終了コード (例 1)

ジョブステップ終了順序	ステップ終了コード	ジョブコントローラの終了コード	
		STEPMAX	STEPLAST
STEP1	4	—	—
STEP2	8	○	—
STEP3	0	—	○

(凡例)

- ：ジョブコントローラの終了コードとなる。
- ：ジョブコントローラの終了コードとならない。

JOB_RCDEFINE パラメータで STEPMAX を指定した場合、すべてのジョブステップの終了コードの最大値が、ジョブコントローラの終了コードとなる。終了コードの最大値は STEP2 の終了コード 8 であるため、ジョブコントローラの終了コードは 8 となる。

JOB_RCDEFINE パラメータで STEPLAST を指定した場合、最後に実行完了したメインステップの終了コードが、ジョブコントローラの終了コードとなる。最後に実行完了したメインステップの終了コードは STEP3 の終了コード 0 であるため、ジョブコントローラの終了コードは 0 となる。

(3) サブステップを含むジョブの終了コードの例

サブステップを含む次のジョブについて、JOB_RCDEFINE パラメータで STEPMAX を指定した場合と STEPLAST の場合それぞれについて、ジョブコントローラの終了コードの決定方法を示す。

ジョブ定義 XML ファイル

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<HitachiBatchJobExec version="1.3" os="unix">
  <JOB NAME="JOB_RC_SUBSTEP">
    <STEP NAME="STEP1_SUB"> <!-- 終了コード4で終了するサブステップ 終了順序3 -->
      <EXEC PGM="CBLRTN" PARM="004" LANG="COBOL" JSPEC="YES" />
    </STEP>
    <STEP NAME="STEP2_SUB"> <!-- 終了コード8で終了するサブステップ 終了順序1 -->
      <EXEC PGM="CBLRTN" PARM="008" LANG="COBOL" JSPEC="YES" />
    </STEP>
    <STEP NAME="STEP3_MAIN"> <!-- 終了コード0で終了するメインステップ 終了順序2 -->
      <EXEC PGM="CBLRTN" PARM="000" LANG="COBOL" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

表 10-41 ジョブコントローラの終了コード (例 2)

ジョブステップ終了順序	ジョブステップ終了コード	ジョブコントローラの終了コード	
		STEPMAX	STEPLAST
STEP2_SUB*	8	○	—
STEP3_MAIN	0	—	○
STEP1_SUB*	4	—	—

注※ サブステップであるため、STEPLAST の場合はジョブコントローラの終了コードに影響を与えない。

(凡例)

- ：ジョブコントローラの終了コードとなる。
- ：ジョブコントローラの終了コードとならない。

JOB_RCDEFINE パラメータで STEPMAX を指定した場合、サブステップを含むすべてのジョブステップの終了コードの最大値が、ジョブコントローラの終了コードとなる。終了コードの最大値は STEP2_SUB の終了コード 8 であるため、ジョブコントローラの終了コードは 8 となる。

JOB_RCDEFINE パラメータで STEPLAST を指定した場合、最後に実行完了したメインステップの終了コードが、ジョブコントローラの終了コードとなる。最後に実行完了したメインステップの終了コードは STEP3_MAIN の終了コード 0 であるため、ジョブコントローラの終了コードは 0 となる。

10.8.3 スプールジョブ転送サービスの終了コード

スプールジョブ転送サービスの終了コードについて次の表に示す。

表 10-42 スプールジョブ転送サービスの終了コード一覧

戻り値	意味
0	正常終了
0 以外	異常終了

10.8.4 世代コマンドの終了コード

世代コマンド (bjexgdg および bjexhkgdg) は、「表 10-39 ジョブコントローラの終了コード一覧」の終了コードと同じく、0 および 244~255 を出力する。

付録

付録 A ジョブ定義 XML ファイル仕様

ジョブ定義 XML ファイルの XML 仕様は、W3C 勧告「拡張可能なマーク付け言語 (XML) 1.0」REC-xml-980210 のサブセットに従う。サポートする定義方法を EBNF 表記方法で次に示す。この文法に従い、バッチジョブ実行システムでは XML を 1 バイトずつ解析する。

```
[ 1] document ::= prolog element Misc*
[ 2] Char ::= #x9 | #xA | #xD | [#x20-#xD7FF] |
             [#xE000-#xFFFD] | [#x10000-#x10FFFF]
[ 3] S ::= (#x20 | #x9 | #xD | #xA)+
[ 4] NameChar ::= Letter | Digit | '.' | '-' | '_' | ':'
[ 5] Name ::= (Letter | '_' | ':') (NameChar)*
[10] AttValue ::= '"' ([^&"] | Reference)* '"' |
              "'" ([^&' ] | Reference)* "'"

[14] CharData ::= [^&]* - ([^&]* ' ')]> [^&]*
[15] Comment ::= <!-- ((Char - '-') | ('-' (Char - '-')))* '--->
[18] CDsect ::= CDstart CData CDEnd
[19] CDstart ::= <![CDATA[
[20] CData ::= (Char* - (Char* ' ')]> Char*)
[21] CDEnd ::= ']]>'
[22] prolog ::= XMLDecl? Misc*
[23] XMLDecl ::= <?xml? VersionInfo EncodingDecl? SDDDecl? S? '?'>
[24] VersionInfo ::= S 'version' Eq (' VersionNum ' | " VersionNum ")
[25] Eq ::= S? '=' S?
[26] VersionNum ::= ([a-zA-Z0-9_.:] | '-')+
[27] Misc ::= Comment | S
[32] SDDDecl ::= S 'standalone' Eq
              (('yes' | 'no') '"' | ('yes' | 'no') "'")

[39] element ::= EmptyElemTag | STag content ETag
[40] STag ::= < Name (S Attribute)* S? '>'
[41] Attribute ::= Name Eq AttValue
[42] ETag ::= </ Name S? '>'
[43] content ::= (element | CharData | Reference | CDsect | Comment)*
[44] EmptyElemTag ::= < Name (S Attribute)* S? '/>'
[67] Reference ::= EntityRef
[68] EntityRef ::= '&' ('lt' | 'gt' | 'amp' | 'apos' | 'quot') ';'
[80] EncodingDecl ::= S 'encoding' Eq ('"' EncName '"' | "'" EncName "'")
[81] EncName ::= [A-Za-z] ([A-Za-z0-9_.:] | '-)*
[84] Letter ::= BaseChar
[85] BaseChar ::= [#x41-#x5A] | [#x61-#x7A]
[88] Digit ::= [#x30-#x39]
```

付録 A.1 ジョブ定義 XML ファイルの DTD

各 OS のジョブ定義 XML ファイルの DTD を次に示す。

- AIX の場合

```

<!ELEMENT HitachiBatchJobExec (JOB|PROC)>
<!ELEMENT JOB (JOBPARM?, PERMISSION?, (DD|SETENV)*, PROC*, (STEP, SETMAXRC?, (IF|ELSE|ENDIF)*)+)>
<!ELEMENT JOBPARM EMPTY>
<!ELEMENT PERMISSION EMPTY>
<!ELEMENT DD (#PCDATA)>
<!ELEMENT SETENV EMPTY>
<!ELEMENT PROC (SYMBOL*, (STEP, SETMAXRC?, (IF|ELSE|ENDIF)*)+)>
<!ELEMENT STEP ((EXEC|CALL), (DD|SETENV|SETENVFILE)*)>
<!ELEMENT SETMAXRC EMPTY>
<!ELEMENT IF EMPTY>
<!ELEMENT ELSE EMPTY>
<!ELEMENT ENDIF EMPTY>
<!ELEMENT SYMBOL EMPTY>
<!ELEMENT EXEC (#PCDATA)>
<!ELEMENT CALL (SETSYMBOL*, REPLACE?)>
<!ELEMENT SETSYMBOL EMPTY>
<!ELEMENT REPLACE (EXEC?, DD*, REPLACESTEP*)>
<!ELEMENT REPLACESTEP (EXEC?, DD*)>
<!ELEMENT SETENVFILE EMPTY>

<!ATTLIST HitachiBatchJobExec
  version CDATA #IMPLIED
  os CDATA #IMPLIED
>
<!ATTLIST JOB
  NAME CDATA #REQUIRED
  COND CDATA #IMPLIED
  MSGCLASS CDATA #IMPLIED
  STDOUTCLASS CDATA #IMPLIED
  STDERRCLASS CDATA #IMPLIED
>
<!ATTLIST JOBPARM
  FORMS CDATA #IMPLIED
  FCB CDATA #IMPLIED
  PROCLIB CDATA #IMPLIED
  ROOM CDATA #IMPLIED
>
<!ATTLIST PERMISSION
  FILE CDATA #IMPLIED
  DIR CDATA #IMPLIED
>
<!ATTLIST DD
  NAME CDATA #REQUIRED
  TYPE (DATA|DUMMY|TEMP|TEMPISAM|PREST|TEMPPREST|GDG|FILE|LIB|SYSOUT|DIR) #IMPLIED
  DSN CDATA #IMPLIED
  DISP CDATA #IMPLIED
  FPERM CDATA #IMPLIED
  CHARS CDATA #IMPLIED
  COPIES CDATA #IMPLIED
  COPYG CDATA #IMPLIED
  FCDFLT CDATA #IMPLIED
  DEST CDATA #IMPLIED
  FCB CDATA #IMPLIED
  OVERLAY CDATA #IMPLIED
  OVERC CDATA #IMPLIED
  HOLD (YES|NO) #IMPLIED
  KEEP (YES|NO) #IMPLIED
  SYSOUT CDATA #IMPLIED
  CMT CDATA #IMPLIED
  MODE (AN|K) #IMPLIED

```

```

OPTCDJ (YES|NO) #IMPLIED
RECFM (A|M|N|X) #IMPLIED
FORMS CDATA #IMPLIED
ISKEY CDATA #IMPLIED
ISRECFM (F|V) #IMPLIED
ISRECL CDATA #IMPLIED
RENAME CDATA #IMPLIED
SAMF (YES|NO) #IMPLIED
FTYPE (1|2) #IMPLIED
GAIJI1 CDATA #IMPLIED
ENCODE (SJIS|UTF8) #IMPLIED
EURFORM CDATA #IMPLIED
EURENCODE CDATA #IMPLIED
>
<!ATTLIST SETENV
  NAME CDATA #REQUIRED
  VALUE CDATA #REQUIRED
>
<!ATTLIST PROC
  NAME CDATA #REQUIRED
>
<!ATTLIST STEP
  NAME CDATA #IMPLIED
>
<!ATTLIST SETMAXRC
  VALUE CDATA #REQUIRED
>
<!ATTLIST IF
  OP CDATA #REQUIRED
>
<!ATTLIST SYMBOL
  NAME CDATA #REQUIRED
  VALUE CDATA #IMPLIED
>
<!ATTLIST EXEC
  PGM CDATA #IMPLIED
  PARM CDATA #IMPLIED
  COND CDATA #IMPLIED
  LANG (COBOL|Java|JavaScript) #IMPLIED
  PARMSPLIT (YES|NO) #IMPLIED
  JSPEC (YES|NO) "NO"
  Group CDATA #IMPLIED
  Option CDATA #IMPLIED
  STDOUTCLASS CDATA #IMPLIED
  STDERRCLASS CDATA #IMPLIED
>
<!ATTLIST CALL
  NAME CDATA #REQUIRED
>
<!ATTLIST SETSYMBOL
  NAME CDATA #REQUIRED
  VALUE CDATA #REQUIRED
>
<!ATTLIST REPLACESTEP
  NAME CDATA #REQUIRED
>
<!ATTLIST SETENVFILE
  DDNAME CDATA #REQUIRED
>

```

- Linux の場合

```

<!ELEMENT HitachiBatchJobExec (JOB|PROC)>
<!ELEMENT JOB (JOBPARM?, PERMISSION?, (DD|SETENV)*, PROC*, (STEP, SETMAXRC?, (IF|ELSE|ENDIF)*)+)>
<!ELEMENT JOBPARM EMPTY>
<!ELEMENT PERMISSION EMPTY>
<!ELEMENT DD (#PCDATA)>
<!ELEMENT SETENV EMPTY>
<!ELEMENT PROC (SYMBOL*, (STEP, SETMAXRC?, (IF|ELSE|ENDIF)*)+)>
<!ELEMENT STEP ((EXEC|CALL), (DD|SETENV|SETENVFILE)*)>
<!ELEMENT SETMAXRC EMPTY>
<!ELEMENT IF EMPTY>
<!ELEMENT ELSE EMPTY>
<!ELEMENT ENDIF EMPTY>
<!ELEMENT SYMBOL EMPTY>
<!ELEMENT EXEC (#PCDATA)>
<!ELEMENT CALL (SETSYMBOL*, REPLACE?)>
<!ELEMENT SETSYMBOL EMPTY>
<!ELEMENT REPLACE (EXEC?, DD*, REPLACESTEP*)>
<!ELEMENT REPLACESTEP (EXEC?, DD*)>
<!ELEMENT SETENVFILE EMPTY>

<!ATTLIST HitachiBatchJobExec
  version CDATA #IMPLIED
  os      CDATA #IMPLIED
>
<!ATTLIST JOB
  NAME      CDATA #REQUIRED
  COND      CDATA #IMPLIED
  MSGCLASS CDATA #IMPLIED
  STDOUTCLASS CDATA #IMPLIED
  STDERRCLASS CDATA #IMPLIED
>
<!ATTLIST JOBPARM
  FORMS CDATA #IMPLIED
  FCB   CDATA #IMPLIED
  PROCLIB CDATA #IMPLIED
  ROOM  CDATA #IMPLIED
>
<!ATTLIST PERMISSION
  FILE CDATA #IMPLIED
  DIR  CDATA #IMPLIED
>
<!ATTLIST DD
  NAME      CDATA #REQUIRED
  TYPE      (DATA|DUMMY|TEMP|TEMPISAM|PREST|TEMPPREST|GDG|FILE|LIB|SYSOUT|DIR) #IMPLIED
  DSN       CDATA #IMPLIED
  DISP      CDATA #IMPLIED
  FPERM     CDATA #IMPLIED
  CHARS     CDATA #IMPLIED
  COPIES    CDATA #IMPLIED
  COPYG     CDATA #IMPLIED
  FCDFLT    CDATA #IMPLIED
  DEST      CDATA #IMPLIED
  FCB       CDATA #IMPLIED
  OVERLAY   CDATA #IMPLIED
  OVERC     CDATA #IMPLIED
  HOLD      (YES|NO) #IMPLIED
  KEEP      (YES|NO) #IMPLIED
  SYSOUT    CDATA #IMPLIED
  CMT       CDATA #IMPLIED
  MODE      (AN|K) #IMPLIED

```

```

OPTCDJ (YES|NO) #IMPLIED
RECFM (A|M|N|X) #IMPLIED
FORMS CDATA #IMPLIED
ISKEY CDATA #IMPLIED
ISRECFM (F|V) #IMPLIED
ISRECL CDATA #IMPLIED
RENAME CDATA #IMPLIED
SAMF (YES|NO) #IMPLIED
GAIJ11 CDATA #IMPLIED
EURFORM CDATA #IMPLIED
EURENCODE CDATA #IMPLIED
>
<!ATTLIST SETENV
NAME CDATA #REQUIRED
VALUE CDATA #REQUIRED
>
<!ATTLIST PROC
NAME CDATA #REQUIRED
>
<!ATTLIST STEP
NAME CDATA #IMPLIED
>
<!ATTLIST SETMAXRC
VALUE CDATA #REQUIRED
>
<!ATTLIST IF
OP CDATA #REQUIRED
>
<!ATTLIST SYMBOL
NAME CDATA #REQUIRED
VALUE CDATA #IMPLIED
>
<!ATTLIST EXEC
PGM CDATA #IMPLIED
PARAM CDATA #IMPLIED
COND CDATA #IMPLIED
LANG (COBOL|Java|JScript) #IMPLIED
PARMSPLIT (YES|NO) #IMPLIED
JSPEC (YES|NO) "NO"
Group CDATA #IMPLIED
Option CDATA #IMPLIED
STDOUTCLASS CDATA #IMPLIED
STDERRCLASS CDATA #IMPLIED
>
<!ATTLIST CALL
NAME CDATA #REQUIRED
>
<!ATTLIST SETSYMBOL
NAME CDATA #REQUIRED
VALUE CDATA #REQUIRED
>
<!ATTLIST REPLACESTEP
NAME CDATA #REQUIRED
>
<!ATTLIST SETENVFILE
DDNAME CDATA #REQUIRED
>

```

付録 B プロセス一覧

バッチジョブ実行システムのプロセス一覧を記載する。

UNIX の ps コマンドを使って、プロセスの動作状態を確認できる。正常に動作している場合に表示されるプロセスを次の表に示す。なお、表内の括弧内の数値は、同時に実行できるプロセス数である。

表 B-1 プロセス一覧

プロセス名	機能	ユーザ	親プロセス名	機能の詳細
bjexec (任意)	ジョブコントローラ	ユーザ	任意 (jqagtchild や シェルなど)	ジョブの制御を行う。
bjexpd (1)	スプールジョブ転送デー モン	root	—※1	スプールの管理、転送処理を行 う。
bjexpd (0~伝送カー ド数+1)	スプールジョブ転送プロ セス	ユーザ	bjexpd※2	転送処理を実施する。転送処理 が完了すると、これらのプロセ スは消滅する。
cp (0~伝送カード数)	スプールジョブ転送プロ セス	ユーザ	bjexpd※3	転送処理の過程で、さらに cp プ ロセスを生成する。転送処理が 完了すると、これらのプロセス は消滅する。

(凡例)

—：なし。

注※1 デーモンのため、親プロセスの pid は 1 になる。

注※2 デーモンが親プロセスとなる。

注※3 スプールジョブ転送プロセスの bjexpd が親プロセスとなる。

付録 C bjexcollect コマンド (障害調査のための資料の一括収集)

付録 C.1 形式

bjexcollect	収集情報出力先ディレクトリ [-f 定義ファイル名]
-------------	-------------------------------

付録 C.2 機能

バッチジョブ実行システムが異常終了および無応答となった場合などの障害調査を実施するための資料を一括して収集できる。

このコマンドは、root 権限で実行する必要がある。

付録 C.3 オプション

- 収集情報出力先ディレクトリ

収集した情報を tar のアーカイブファイルとして出力する出力先のディレクトリを指定する。bjexcollect コマンドの内部処理で一時ファイルを使用する場合にも、このディレクトリに作成する。

アーカイブファイルの名称は、BJEXyyyymmddhhmmss.tar となる。ここで yyyymmdd は bjexcollect を起動した日付、hhmmss は bjexcollect を起動した 24 時間制のローカルタイムでの時刻である。

- -f 定義ファイル名<</opt/hitachi/bjex/maintenance/bjexcollect.conf>>

bjexcollect.conf ファイルを指定する。絶対パス、またはカレントディレクトリからの相対パスで指定する。

付録 C.4 使用方法

バッチジョブ実行システムを標準的な構成でインストールした場合の使用方法を次に示す。

1. bjexcollect.conf ファイルを運用環境に合わせて書き換える。詳細については、「[2.3.3 bjexcollect コマンドの設定](#)」を参照のこと。
2. /tmp など、バッチジョブ実行システムの運用で使用しないディレクトリを収集情報出力先ディレクトリに指定して bjexcollect コマンドを実行する。これによって、「(3) オプション」で示した名称の、tar のアーカイブファイルが作成される。/tmp ディレクトリを使用した例を次に示す。

```
/opt/hitachi/bjex/maintenance/bjexcollect△/tmp
```

(凡例)

△：半角空白を示す。

付録 C.5 定義ファイルの設定と指定例

(1) 定義ファイルの設定

bjexcollect.conf で bjexcollect の動作を定義する。bjexcollect.conf は、キーワードと値をスペースで区切って記述する。行の最初に#を付けるとコメント行になる。ファイル名はすべて絶対パスで指定する。定義ファイルのキーワードと指定の関係を次の表に示す。キーワードは、大文字小文字を区別しない。

表 C-1 定義ファイルのキーワードと指定の関係

キーワード	指定する値	指定	複数指定	ワイルドカード
SPOOL_DIR	bjex.conf の SPOOL_DIR と同じスプールディレクトリ名称を指定する。	必須	×	×
LOG_DIR	bjex.conf の LOG_DIR と同じディレクトリ名称を指定する。			
TRACE_DIR	bjex.conf の TRACE_DIR と同じディレクトリ名称を指定する。			
LOGICAL_HOST	バッチジョブ実行システムが論理ホストで稼働している場合、論理ホスト名を指定する。	任意		
CORE	その他任意で採取する core ファイル名を指定する。	任意	○	○
FILES	その他任意で採取するファイル名を指定する。			

(凡例)

○：指定できる。

×：指定できない。

(2) 定義ファイルの指定例

定義ファイルの指定例を次に示す。

SPOOL_DIR	/var/opt/bjex/spool
LOG_DIR	/opt/hitachi/bjex/log
TRACE_DIR	/opt/hitachi/bjex/trace
FILES	/opt/hitachi/bjex/conf/*

- 一時的に使用するファイル
10MB
- 収集した情報を入力する tar ファイル

SPOOL_DIR 下の core ファイルのすべての容量 + 実行ログ, トレースログの容量 + 一時的に使用するファイルの容量 + CORE, FILES で指定したファイルの容量

付録 C.6 注意事項

- 収集情報出力先ディレクトリには, core ファイルを含む保守情報のアーカイブファイルが作成されるため, 空き領域を確保しておく。
- 収集情報出力先ディレクトリに出力ファイル, および一時ファイルを作成する。このため, 収集情報出力先ディレクトリは書き込み可能とする。
- CORE, FILES にディレクトリを指定した場合, 指定されたディレクトリ下のファイルのすべてを採取する。このため, ルートディレクトリなど上位ディレクトリを指定しないこと。
- LOGICAL_HOST に値を指定しなかった場合, 物理ホストを仮定する。
- このコマンドは root で実行すること。
- bjexcollect コマンドの収集情報出力先ディレクトリとして, スプールディレクトリやインストールディレクトリなど, バッチジョブ実行システムに関するディレクトリを指定しないこと。指定すると bjexcollect コマンドが異常終了することがある。
/tmp など, バッチジョブ実行システムとは関係ないディレクトリを指定すること。
- bjexcollect コマンドを実行中に, バッチジョブ (bjexec コマンド) やほかの BJEX のコマンドは実行しないでください。

付録 C.7 戻り値

- 0
正常終了。
- 0 以外の値
異常終了。

付録 C.8 メッセージ

メッセージ	意味, オペレーターのとる処置
bjexcollect:RAS completed collection of tar ファイル名 採取したファイル名 採取したファイル名 ...	採取したファイル名をまとめて tar ファイル (ファイル名) を作成した。
Usage: bjexcollect directory [-f filename] directory : Specify output directory	オプションの設定が誤っている。正しく設定し, 再度実行する。

メッセージ	意味, オペレーターのとる処置
-f filename : Specify a config file	オプションの設定が誤っている。正しく設定し, 再度実行する。
bjexcollect:RAS error:出力先ディレクトリ(Permission denied).	出力先ディレクトリにアクセス権限がない。アクセス権限を設定するか, または別のディレクトリを指定して再度実行する。
bjexcollect:RAS error:出力先ディレクトリ(not found or not a directory).	出力先ディレクトリは存在しない, またはディレクトリではない。正しい出力先を指定し, 再度実行する。
bjexcollect:RAS error:設定ファイル名(not found or not a file).	設定ファイルは存在しない, またはファイルではない。正しい設定ファイルを指定し, 再度実行する。
bjexcollect:RAS error:設定ファイル名(Permission denied).	設定ファイルにアクセス権限がない。アクセス権限を設定し, 再度実行する。
bjexcollect:RAS error:設定ファイル名(キーワード Syntax Error).	設定ファイル中に不正なキーワードが指定された。設定ファイルを正しく設定し, 再度実行する。
bjexcollect:RAS error:設定ファイル名(キーワード not found).	必須キーワードの指定がない。必須キーワードを指定し, 再度実行する。
bjexcollect:RAS error:指定値(not found or not a directory).	設定ファイルで指定された指定値は存在しない, またはディレクトリではない。ディレクトリを指定し, 再度実行する。

付録 D メッセージ制御機能のエラー情報

メッセージ制御機能のエラーが発生した場合、COBOL アプリケーションは KCCC2060R-S メッセージを出力する。メッセージ出力コマンド (bjexm_message コマンド) は KAKC5205-E メッセージを出力する。

これらのメッセージの詳細情報に表示されるエラー情報の意味および対処方法を次の表に示す。

表 D-1 メッセージの詳細情報に表示されるエラー情報の意味および対処方法

戻り値	エラー番号	意味	対処
1	10	メッセージを送信しようとしたが、メッセージに表示できない文字が存在する、またはインタフェース上のパラメータが不正である。メッセージに NULL 文字が含まれている、またはメッセージを出力したプログラムがこの製品のサポートしていない機能を使用しているおそれがある。	データ置き換え機能を使用する。またはプログラムを見直す。問題が解決しない場合、システム管理者に連絡する。システム管理者は「 6.2 トラブルシューティング 」に従って資料を採取し、トラブルの回避および対処をする。
	11	環境変数が不正である。 JP1/AJS, またはバッチジョブ実行システムを利用して呼び出していない可能性がある。	該当するプログラムを JP1/AJS からバッチジョブ実行システムを通して呼び出すように変更して再実行する。
2	1	未実装の API を呼び出した。	システム管理者に連絡する。システム管理者は「 6.2 トラブルシューティング 」に従って資料を採取し、トラブルの回避および対処をする。
	2	共有メモリの参照に失敗した。 スプールジョブ転送デーモンが起動していない可能性がある、またはスプールジョブ転送デーモンの起動中に使用するスプールディレクトリを変更した可能性がある。	スプールジョブ転送デーモンが起動していることを確認する。スプールジョブ転送デーモンが起動中であれば、使用中のスプールディレクトリを変更していないか確認する。スプールディレクトリを変更した場合は、スプールジョブ転送デーモンを停止して再度起動する。 問題が解決しない場合は、システム管理者に連絡する。システム管理者は「 6.2 トラブルシューティング 」に従って資料を採取し、トラブルの回避および対処をする。
	3	空きメモリ領域が不足している。	システム管理者に連絡する。システム管理者はメモリ見直しが必要がある。
	4	不正な引数が渡された。	システム管理者に連絡する。システム管理者は「 6.2 トラブルシューティング 」に従って資料を採取し、トラブルの回避および対処をする。
	5	データの不整合を検出した。	システム管理者に連絡する。システム管理者は「 6.2 トラブルシューティング 」に従って資料を採取し、トラブルの回避および対処をする。

戻り値	エラー番号	意味	対処
2	6	サポートしていない文字エンコーディングが指定された。	システム管理者に連絡する。システム管理者は「6.2 トラブルシューティング」に従って資料を採取し、トラブルの回避および対処をする。
	7	アンダーフローが発生した。	システム管理者に連絡する。システム管理者は「6.2 トラブルシューティング」に従って資料を採取し、トラブルの回避および対処をする。
	8	オーバーフローが発生した。	システム管理者に連絡する。システム管理者は「6.2 トラブルシューティング」に従って資料を採取し、トラブルの回避および対処をする。
	9	API の呼び出し順序が間違っている。	システム管理者に連絡する。システム管理者は「6.2 トラブルシューティング」に従って資料を採取し、トラブルの回避および対処をする。
	10	オブジェクトなどの内部状態に不整合が発生した。	システム管理者に連絡する。システム管理者は「6.2 トラブルシューティング」に従って資料を採取し、トラブルの回避および対処をする。
	11	リソースへのアクセスが拒否された。	バッチジョブ実行システムの設定ファイルに指定したログのパス (LOG_DIR) で指定したディレクトリおよびその配下のログファイルに、書き込み権限が付いていることを確認する。
	12	指定されたファイルが存在しない。	バッチジョブ実行システムの設定ファイルに指定したログのパス (LOG_DIR) で指定したディレクトリおよびその配下のログファイルに、書き込み権限が付いていることを確認する。
	13	ファイルをオープンできない。	バッチジョブ実行システムの設定ファイルに指定したログのパス (LOG_DIR) で指定したディレクトリおよびその配下のログファイルに、書き込み権限が付いていることを確認する。
	14	ファイルに対するメモリマッピングを作成できない。	バッチジョブ実行システムの設定ファイルに指定したログのパス (LOG_DIR) で指定したディレクトリおよびその配下のログファイルに、書き込み権限が付いていることを確認する。
	15	ファイルからの読み込み処理でエラーが発生した。	バッチジョブ実行システムの設定ファイルに指定したログのパス (LOG_DIR) で指定したディレクトリおよびその配下のログファイルに、書き込み権限が付いていることを確認する。
	16	ファイルへの書き込み処理でエラーが発生した。	バッチジョブ実行システムの設定ファイルに指定したログのパス (LOG_DIR) で指定したディレクトリおよびその配下のログファイルに、書き込み権限が付いていることを確認する、または上記のログを格納するディスクに空きがあることを確認する。
17	ファイルのシーク処理でエラーが発生した。	バッチジョブ実行システムの設定ファイルに指定したログのパス (LOG_DIR) で指定したディレクトリおよび	

戻り値	エラー番号	意味	対処
2	17	ファイルのシーク処理でエラーが発生した。	その配下のログファイルに、書き込み権限が付いていることを確認する。
	18	ファイルへのフラッシュ処理でエラーが発生した。	バッチジョブ実行システムの設定ファイルに指定したログのパス (LOG_DIR) で指定したディレクトリおよびその配下のログファイルに、書き込み権限が付いていることを確認する。
	19	ファイルのリネーム処理でエラーが発生した。	バッチジョブ実行システムの設定ファイルに指定したログのパス (LOG_DIR) で指定したディレクトリおよびその配下のログファイルに、書き込み権限が付いていることを確認する。
	20	ファイルのコピー処理でエラーが発生した。	バッチジョブ実行システムの設定ファイルに指定したログのパス (LOG_DIR) で指定したディレクトリおよびその配下のログファイルに、書き込み権限が付いていることを確認する。
	21	ファイルの削除処理でエラーが発生した。	バッチジョブ実行システムの設定ファイルに指定したログのパス (LOG_DIR) で指定したディレクトリおよびその配下のログファイルに、書き込み権限が付いていることを確認する。
	22	ディレクトリの作成に失敗した。	バッチジョブ実行システムの設定ファイルに指定したログのパス (LOG_DIR) で指定したディレクトリおよびその配下のログファイルに、書き込み権限が付いていることを確認する。
	23	プロセス間のロックに失敗した。	システム管理者に連絡する。システム管理者は「 6.2 トラブルシューティング 」に従って資料を採取し、トラブルの回避および対処をする。
	24	ユーザプログラムによって無効にされた機能を利用しようとした。	システム管理者に連絡する。システム管理者は「 6.2 トラブルシューティング 」に従って資料を採取し、トラブルの回避および対処をする。
	25	ユーザプログラムによって禁止されているため、上書きできない。	システム管理者に連絡する。システム管理者は「 6.2 トラブルシューティング 」に従って資料を採取し、トラブルの回避および対処をする。
	400	パラメータが不正である。	システム管理者に連絡する。システム管理者は「 6.2 トラブルシューティング 」に従って資料を採取し、トラブルの回避および対処をする。
	404	bjexchmsg コマンドによって応答要求メッセージが削除された。または共有メモリを参照できない。	bjexchmsg コマンドの-d オプションによって応答要求メッセージを削除した場合、ジョブを正常に終了させたい場合はジョブを再実行する。 bjexchmsg コマンドを使用していない場合、スプールジョブ転送デーモンの起動でエラーが発生していないか確認する。
409	同一 PID から応答要求メッセージが出力された。	アプリケーションを見直す。	

戻り値	エラー番号	意味	対処
2	409	マルチスレッドで同時に応答要求メッセージを出力した可能性がある。	アプリケーションを見直す。
3	「10.4 システムコールのエラー情報」を参照のこと。		
4 および 5	10	パラメータが不正である。	システム管理者に連絡する。システム管理者は「6.2 トラブルシューティング」に従って資料を採取し、トラブルの回避および対処をする。
	11	関数発行の順序が不正である。	システム管理者に連絡する。システム管理者は「6.2 トラブルシューティング」に従って資料を採取し、トラブルの回避および対処をする。
	30	指定属性はすでに登録済みである。	システム管理者に連絡する。システム管理者は「6.2 トラブルシューティング」に従って資料を採取し、トラブルの回避および対処をする。
	32	登録可能な拡張属性数を超えている。	システム管理者に連絡する。システム管理者は「6.2 トラブルシューティング」に従って資料を採取し、トラブルの回避および対処をする。
	33	登録可能な拡張属性の合計サイズを超えている。	システム管理者に連絡する。システム管理者は「6.2 トラブルシューティング」に従って資料を採取し、トラブルの回避および対処をする。
	40	イベントサービスに接続できない。	自ホストおよび JP1/IM - Manager が稼働するホストで、JP1/Base のイベントサービスの起動を確認する。
	43	入出力エラーである。	システム管理者に連絡する。システム管理者は「6.2 トラブルシューティング」に従って資料を採取し、トラブルの回避および対処をする。
	50	JP1/Base のライブラリが見つからない。	JP1/Base をインストールして、ジョブを再実行する。
	51	メモリが不足している。	システム管理者に連絡する。システム管理者はメモリ見積もりを見直す必要がある。
	52	ファイルオープン数が限界である。	システム管理者に連絡する。システム管理者はオープンできる fd 数の見積もりを見直す必要がある。
	60	JP1 イベントの初期化処理が行われていない。	システム管理者に連絡する。システム管理者は「6.2 トラブルシューティング」に従って資料を採取し、トラブルの回避および対処をする。
	70	システムエラーである。	システム管理者に連絡する。システム管理者は「6.2 トラブルシューティング」に従って資料を採取し、トラブルの回避および対処をする。
6	10	設定ファイルの MAX_RESPONSE_REQUEST パラメータを超えて、メッセージ制御機能を利用しようとしたが、登録待ち中にタイムアウトとなった。	応答要求メッセージの出力数が MAX_RESPONSE_REQUEST パラメータの指定値以内となるようにする。

戻り値	エラー番号	意味	対処
6	11	設定ファイルの JP1IM_MANAGER_HOST パラメータに指定したホストに JP1 イベントを送信しようとしたが、処理のリトライ中にタイムアウトとなった。	JP1IM_MANAGER_HOST パラメータの指定を確認する。自ホストおよび JP1/IM - Manager が稼働するホストで、JP1/Base のイベントサービスの起動を確認する。

付録 E このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 E.1 関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

JP1/AJS 関連

- JP1 Version 12 ジョブ管理 基本ガイド (ジョブスケジューラー編) (3021-3-D20)
- JP1 Version 12 JP1/Automatic Job Management System 3 導入ガイド (3021-3-D21)
- JP1 Version 12 JP1/Automatic Job Management System 3 設計ガイド (システム構築編) (3021-3-D22)
- JP1 Version 12 JP1/Automatic Job Management System 3 設計ガイド (業務設計編) (3021-3-D23)
- JP1 Version 12 JP1/Automatic Job Management System 3 構築ガイド (3021-3-D24)
- JP1 Version 12 JP1/Automatic Job Management System 3 運用ガイド (3021-3-D25)
- JP1 Version 12 JP1/Automatic Job Management System 3 トラブルシューティング (3021-3-D26)
- JP1 Version 12 JP1/Automatic Job Management System 3 操作ガイド (3021-3-D27)
- JP1 Version 12 JP1/Automatic Job Management System 3 コマンドリファレンス (3021-3-D28)
- JP1 Version 12 JP1/Automatic Job Management System 3 連携ガイド (3021-3-D29)
- JP1 Version 12 JP1/Automatic Job Management System 3 メッセージ (3021-3-D30)
- JP1 Version 13 ジョブ管理 基本ガイド (ジョブスケジューラー編) (3021-3-L41)
- JP1 Version 13 JP1/Automatic Job Management System 3 導入ガイド (3021-3-L42)
- JP1 Version 13 JP1/Automatic Job Management System 3 設計ガイド (システム構築編) (3021-3-L43)
- JP1 Version 13 JP1/Automatic Job Management System 3 設計ガイド (業務設計編) (3021-3-L44)
- JP1 Version 13 JP1/Automatic Job Management System 3 構築ガイド (3021-3-L45)
- JP1 Version 13 JP1/Automatic Job Management System 3 運用ガイド (3021-3-L46)
- JP1 Version 13 JP1/Automatic Job Management System 3 トラブルシューティング (3021-3-L47)
- JP1 Version 13 JP1/Automatic Job Management System 3 操作ガイド (3021-3-L48)

- JP1 Version 13 JP1/Automatic Job Management System 3 コマンドリファレンス (3021-3-L49)
- JP1 Version 13 JP1/Automatic Job Management System 3 連携ガイド (3021-3-L50)
- JP1 Version 13 JP1/Automatic Job Management System 3 メッセージ (3021-3-L51)

JP1/IM 関連

- JP1 Version 12 統合管理 基本ガイド (3021-3-D50)
- JP1 Version 12 JP1/Integrated Management 2 - Manager 導入・設計ガイド (3021-3-D51)
- JP1 Version 12 JP1/Integrated Management 2 - Manager 構築ガイド (3021-3-D52)
- JP1 Version 12 JP1/Integrated Management 2 - Manager 運用ガイド (3021-3-D53)
- JP1 Version 12 JP1/Integrated Management 2 - Manager 画面リファレンス (3021-3-D54)
- JP1 Version 10 JP1/Integrated Management 2 - Manager コマンド・定義ファイルリファレンス (3021-3-D55)
- JP1 Version 12 JP1/Integrated Management 2 - Manager メッセージ (3021-3-D56)
- JP1 Version 13 統合管理 基本ガイド (3021-3-L01)
- JP1 Version 13 JP1/Integrated Management 3 - Manager 導入・設計ガイド (3021-3-L02)
- JP1 Version 13 JP1/Integrated Management 3 - Manager 構築ガイド (3021-3-L03)
- JP1 Version 13 JP1/Integrated Management 3 - Manager 運用ガイド (3021-3-L04)
- JP1 Version 13 JP1/Integrated Management 3 - Manager 画面リファレンス (3021-3-L05)
- JP1 Version 13 JP1/Integrated Management 3 - Manager コマンド・定義ファイル API リファレンス (3021-3-L06)
- JP1 Version 13 JP1/Integrated Management 3 - Manager メッセージ (3021-3-L07)

JP1/Base 関連

- JP1 Version 12 JP1/Base 運用ガイド (3021-3-D65)
- JP1 Version 12 JP1/Base メッセージ (3021-3-D66)
- JP1 Version 12 JP1/Base 関数リファレンス (3021-3-D67)
- JP1 Version 13 JP1/Base 運用ガイド (3021-3-L09)
- JP1 Version 13 JP1/Base メッセージ (3021-3-L10)
- JP1 Version 13 JP1/Base 関数リファレンス (3021-3-L11)

JP1/File Transmission Server/FTP 関連

- JP1 Version 12 JP1/File Transmission Server/FTP (Windows(R)用) (3021-3-D38)
- JP1 Version 12 JP1/File Transmission Server/FTP (UNIX(R)用) (3021-3-D39)
- JP1 Version 13 JP1/File Transmission Server/FTP (Windows(R)用) (3021-3-L59)
- JP1 Version 13 JP1/File Transmission Server/FTP (UNIX(R)用) (3021-3-L60)

COBOL 関連

- COBOL2002 使用の手引 手引編 (3021-3-602)

SORT 関連

- ソートマージ (3020-3-N73)

PRINT DATA EXCHANGE 関連

- PRINT DATA EXCHANGE for Open (3020-3-P02)
- PRINT DATA EXCHANGE - Form Designer (3020-3-P03)

EUR 関連

- uCosminexus EUR サーバ帳票出力 (3020-7-536)
- uCosminexus EUR Print Manager 帳票出力 (3020-7-538)

その他

- 索引順編成ファイル管理 ISAM (3000-3-169)
- Cosminexus アプリケーションサーバ V8 リファレンス コマンド編 (3020-3-U14)
- Hitachi Code Converter (UNIX 編) (3020-7-358)

付録 E.2 このマニュアルでの表記

このマニュアルでは、製品名を次のように表記しています。

正式名称	このマニュアルでの表記		
AIX 7.1, AIX 7.2, AIX 7.3	UNIX	AIX	
Red Hat Enterprise Linux Server 7 (64-bit x86_64)		Linux	Linux7
Red Hat Enterprise Linux Server 8 (64-bit x86_64)			Linux8
Red Hat Enterprise Linux Server 9 (64-bit x86_64)			Linux9
COBOL2002 Net Server Runtime	COBOL2002		
COBOL2002 Net Server Suite			
COBOL2002 Net Server Runtime(64)			
COBOL2002 Net Server Suite(64)			
Hitachi Code Converter - Runtime for C/COBOL(64)	Hitachi Code Converter		
ISAM Version7	ISAM	ISAM (32)	
ISAM Version7(64)		ISAM (64)	
JPI/Advanced Shell	Advanced Shell		
Advanced Shell			

正式名称	このマニュアルでの表記	
JP1/Automatic Job Management System 3 - Agent	JP1/AJS - Agent または JP1/AJS3 - Agent	JP1/AJS または JP1/AJS3
JP1/Automatic Job Management System 3 - Manager	JP1/AJS - Manager または JP1/AJS3 - Manager	
JP1/Automatic Job Management System 3 - View	JP1/AJS - View または JP1/AJS3 - View	
JP1/File Transmission Server/FTP	JP1/FTP	
JP1/Integrated Management 2 - Manager	JP1/IM または JP1/IM - Manager	
JP1/Integrated Management 3 - Manager		
JP1/Integrated Management 2 - View	JP1/IM - View	
JP1/Integrated Management 3 - View		
Network File System	NFS	
PRINT DATA EXCHANGE - Electronic Document Option	PDE	
PRINT DATA EXCHANGE - Enterprise Printer Option		
PRINT DATA EXCHANGE - Form Designer		
PRINT DATA EXCHANGE for Open		
SORT Version9	SORT EE	
SORT Version9 - Extended Edition		
SORT Version9(64)		
SORT Version9 - Extended Edition(64)		
User Application Program	UAP	
uCosminexus Batch Job Execution Server	バッチジョブ実行システムまたは BJEX	
uCosminexus Grid Processing Server	uGPS	
Virtual-storage Operating System 1	VOS1	
Virtual-storage Operating System 3	VOS3	

付録 E.3 KB (キロバイト) などの単位表記について

1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ $1,024$ バイト, $1,024^2$ バイト, $1,024^3$ バイト, $1,024^4$ バイトです。

(英字)

BOM (Byte Order Mark)

Unicode の特定のコード列を文書の先頭に挿入することで、その文書がビッグエンディアンとリトルエンディアンのどちらでエンコードされたものかを識別できるようにするためのものである。

DTD (Document Type Definition)

XML での最もベーシックなスキーマ言語である。

Java バッチアプリケーション

Java で記載されたバッチジョブの業務アプリケーションプログラムのことである。バッチジョブ実行システムでは、Cosminexus と連携して Java バッチアプリケーションを実行できる。

JSPEC 機能

ジョブステップを並列実行して処理時間を短縮する機能である。

PREST 機能

逐次実行していたジョブおよびジョブステップを並列に実行する機能。処理時間を短縮できる。ジョブステップおよびジョブステップ間で使用するファイルに PREST ファイルという特別のファイルを使用する必要がある。

SYSIN データ

ジョブ定義 XML ファイル中に置かれた、ジョブ実行時に使用するデータである。DD 要素の TYPE 属性値が DATA である場合が SYSIN データとなる。

SYSOUT

ジョブステップのプログラムが出力したデータを格納するファイルである。DD 要素の TYPE 属性に "SYSOUT" を指定したファイルが SYSOUT データファイルとして扱われる。このファイルに出力されたデータは、いったんスプール内に置かれたあとで、DD 要素に指定された各種属性および出力クラスの内容に従って出力装置に出力される。

UTF-8

Unicode で定義される文字集合を用いて記述された文字列を、バイト列（数値の列）に変換する方式の 1 つである。

XML (eXtensible Markup Language)

拡張できるマーク付け言語文書やデータの意味や構造を記述するためのマークアップ言語の1つである。マークアップ言語とは、「タグ」と呼ばれる特定の文字列で、文に構造を埋め込んでいく言語のことである。ジョブ定義 XML ファイルは、XML で記述される。

(ア行)

一時 ISAM ファイル

ジョブ実行時に使用する索引順編成ファイル管理 ISAM による索引順編成のファイルで、ジョブまたはジョブステップ開始時に作成し、ジョブ終了時には削除するようなファイルである。DD 要素の TYPE 属性に TEMPISAM を指定することで定義できる。

一時ファイル

ジョブ実行時に使用するファイルで、ジョブまたはジョブステップ開始時に作成し、ジョブ終了時には削除するようなファイルである。DD 要素の TYPE 属性に TEMP を指定することで定義できる。

引用符

バックスラッシュまたは円記号、単一引用符、二重引用符である。

応答要求メッセージ

ユーザからの応答入力が必要なメッセージ。

(カ行)

カタログドプロシジャ

一連のジョブ制御要素のことをいい、EXEC 要素の PROC 属性値にプロシジャ名を指定することで内容が展開され、指定したものと同等に扱われる。頻繁に使用される一連のジョブ制御要素を毎回指定することの手間を省くことと、ジョブ制御要素の記述量を減少させ誤りを少なくするために用いる。

環境変数

ユーザが設定できるシステムの各種の設定を格納した変数。

記号パラメータ

カタログドプロシジャまたは入力ストリームプロシジャ内のオペランドまたはパラメータを#{と} で囲んだものを記号パラメータという。このプロシジャを呼び出す CALL 要素内の SYMBOL 要素に、記号パラメータに任意の値を置換するよう指定できる。

部分的にプロシジャを修正でき、プロシジャに汎用性を持たせるための機能である。

逆方向参照

ジョブ定義 XML に備わっている逆方向に参照する機能。ジョブ内の DD 要素にある情報を取得するようにシステムに指示できる。

クラスタ運用

障害が発生した場合などのクラスタシステムで、実行系サーバから待機系サーバに処理を引き継いで業務の停止を防いだり、障害が回復した場合に待機系サーバから実行系サーバに処理を移して、通常の稼働に戻したりしている。これらの運用のことをクラスタ運用と呼ぶ。

クラスタシステム

複数のサーバシステムを連携して 1 つのシステムとして運用するシステムで、1 つのサーバで障害が発生しても、別のサーバで業務を継続できるシステム。業務を実行している実行系サーバと業務を引き継ぐ待機系サーバから構成される。

恒久ファイル

ジョブ終了後にジョブ結果として残すファイルである。設定ファイルの「PERMANENT_FILE_DIR」パラメータで出力先を設定できる。

(サ行)

実行ログ

システム管理者がジョブの実行を監視するためのジョブの履歴。

実体 (Entity)

XML 文書に含まれる 1 つの情報の単位である。

出力クラス

ジョブ定義 XML の DD 要素の SYSOUT 属性で指定する出力のためのクラス。出力クラスを使用して、出力情報をグループに分ける。

出力後保存

スプールジョブ転送サービスによって出力装置にいったん出力された SYSOUT データファイル（ジョブログを含む）の出力が完了した場合に、スプール内に保存することである。

出力保留

SYSOUT データファイルがスプール内に格納されてからスプールジョブ転送サービスに取り出されたあとに、印刷出力を保留すること。DD 要素の HOLD 属性で YES を指定して出力を保留する。

条件付き実行

先行するジョブステップの実行結果によって、後続するジョブステップを実行しないでそのジョブを終了させたり、または特定のジョブステップの実行へ移したりすることである。ジョブ定義 XML の JOB 要素または EXEC 要素の COND 属性を指定して条件付き実行を実施する。

ジョブ間 PREST ファイル

ジョブ間で並列実行する場合に必要な PREST ファイルを使用するときに使うファイルである。

ジョブコントローラ

ジョブ実行時にジョブをコントロールするためのプログラム。

ジョブ識別子

ジョブ実行時にバッチジョブ実行システムが与える 000001 から 999999 の識別番号である。

各ジョブには別々の識別子が与えられ、バッチジョブ実行システムではジョブ識別子によって一意にジョブを特定する。

ジョブ識別子を 999999 まで使用すると、ラップアラウンドして 000001 以降の未使用のジョブ識別子を使用する。

ジョブステップ

ある業務（仕事）を行うための最少単位であり STEP 要素で定義される。ジョブステップの集まりがジョブである。

ジョブステップ間 PREST ファイル

ジョブステップ間で並列実行する場合に必要な PREST ファイルを使用するときに使うファイルである。

ジョブ定義 XML ファイル

バッチジョブ実行システムの入力ファイルである。ジョブやジョブステップの定義を記述した、XML 形式のファイルである。

ジョブライブラリ

特定のジョブでだけ使用するプログラムライブラリであり、JOB 要素直後で、NAME 属性値が JOBLIB である DD 要素で定義される。JOBLIB の DD 要素が定義されると、ジョブ内のプログラムサーチパス環境変数の先頭に、DD 要素に指定したパスが追加され、ほかのディレクトリより優先してプログラムサーチとローディングが実行される。

ジョブログ

ジョブやジョブステップの開始終了メッセージや、COBOL プログラムからの CONSOLE 指定出力メッセージの集まりである。

ジョブログ出力機能

ジョブコントローラが出力する、ジョブ実行に付随したログ情報を出力する機能である。処理プログラムの実行結果（実行ログ）は、除く。

ステップライブラリ

特定のジョブステップでだけ使用するプログラムライブラリであり、STEP 要素直後で、NAME 属性値が STEPLIB である DD 要素で定義される。STEPLIB の DD 要素が定義されると、ジョブステップ内のプログラムサーチパス環境変数の先頭に、DD 要素に指定したパスが追加され、ほかのディレクトリより優先してプログラムサーチとローディングが実行される。

ステップリスタート

ジョブが異常終了して障害要因を取り除いてからジョブを再実行する場合、正常に終了したジョブステップを再度実行しないで、途中のジョブステップから実行を再開すること。途中のジョブステップから実行を開始するには、bjexec コマンドで再実行するジョブステップ名を指定する。

スプール

バッチジョブ実行システムでジョブの実行結果（実行ログ）やジョブログを管理するものである。

スプールジョブ

実行が終了したジョブの結果をジョブごとにまとめたデータである。

スプールジョブ転送サービス

スプールを管理してジョブの出力する帳票をプリントサーバに転送するサービスのことである。

スプールジョブ転送デーモン

スプールジョブ転送サービスを実行するためにバックグラウンドで動作するプログラムのことである。

スプールジョブ保管機能

ジョブ終了時にスプールジョブを転送する前に、任意のディレクトリにスプールジョブを保管する機能である。

世代データ群

一定期間ごとに更新されるファイルで、旧ファイルと新ファイル、旧バージョンと新バージョンのように互いに時間的または歴史的な関連を持った一連のファイルグループである。

世代ファイル

世代データ群中の各ファイルである。作成時期に従って相対的な番号を付け、世代データ群名とその番号で参照ができる。

世代ファイルの世代管理ファイルへの登録タイミング変更機能

世代ファイルを割り当てるジョブで、世代ファイルを作成するジョブステップは正常に実行できたが、後続のジョブステップで異常終了した場合でも、世代ファイルを新規に生成した世代データ群の世代管理情報は更新されるという機能である。世代管理情報の更新タイミングを世代ファイルを生成したジョブステップ単位ではなくジョブ終了時に変更し、ジョブの終了状態によって、世代管理情報の更新を行うかどうかを判定し、登録の可否を決定する機能である。

世代ファイルの保存日数超過による一括削除機能

世代管理ファイルの登録状態に関係なく、ユーザが指定する保存日数の範囲で、保存日数を超過した世代ファイルの削除と世代管理ファイルに登録されている情報を削除する機能である。

絶対世代番号

各世代ファイルに対して、そのグループ内で一意の番号のことである。絶対世代番号の形式は、"Gnnnn"である。nnnn は、0000～9999 の数値の範囲となる。

設定ファイル

バッチジョブ実行システムのインストール後に設定するファイル。バッチジョブ実行システムの環境を設定する。

相対世代番号

最新の世代ファイルの世代番号を0とし、これよりも古い世代ファイルの世代番号を負の整数、新しく登録する世代ファイルの世代番号を正の整数で表した相対的な世代番号である。各世代ファイルは、世代データ群名のあとに相対世代番号を付加したファイル名で識別される。

属性 (Attribute)

XML 文書の中で、要素に対して付加的な情報を加えるために使用される。

(タ行)

ダミー出力クラス

設定ファイルの出力クラスの属性値定義 (SysoutClass パラメータ) で、DUMMY オペランドに YES を指定した出力クラスである。

ダミーファイル

実際には実体が存在しない架空のファイルであり、プログラムの変更なしにあたかも入出力動作を行っているかのように処理するために用いるファイルである。DD 要素の TYPE 属性値に DUMMY を指定することで定義される。なお、設定ファイルに DUMMY_FILE_TYPE パラメータで TEMP を指定した場合、ダミーファイルとして一時ファイルを割り当てる。

統合コンソール

JP1/IM で使用するシステムを監視するためのコンソール。バッチジョブで発生した事象を JP1 イベントとして集め、時系列に表示することで、システムを集中的に監視する。

トレース

トラブルが発生した場合にトラブル発生の経緯を調査するために採取するログ情報。

(ナ行)

入力ストリームプロシジャ

定型化したジョブ制御要素の集まりを PROC 要素内に置き、入力ストリーム中に定義したものの (プロシジャ) である。カタログドプロシジャと同様の使用ができる。

(ハ行)

排他/共用制御機能

ジョブで指定した DD 要素の DISP 属性値に従った排他/共用制御を、ジョブ実行単位に行う機能である。

バッチジョブ実行システム

一定期間のデータを集めて、一括処理をするバッチ処理を対象としたシステムである。今までメインフレームなどでバッチジョブの実行をしていた場合でも、ジョブ制御言語によるバッチジョブイメージを継承したジョブ制御を実現できる。

日付世代番号

EXD, EXD-n, または Dyyyymmdd で指定する、世代ファイルを日付で管理するための世代番号である。EXD にはルートジョブネットの実行開始予定日が設定され、その日より古い世代を指定する場合は EXD-n と指定する。また、Dyyyymmdd と指定すると直接日付を指定する。各世代ファイルは、世代データ群名のあとに yyyymmdd を付加したファイル名で識別される。

ファイル操作ユティリティ

バッチジョブ実行システムで提供するファイルを操作するためのユティリティのことである。

ファイルの連結処理

COBOL2002 で作成したユーザアプリケーションおよび SORT EE と連携して、複数のファイルを論理的に連結した 1 つのファイルとして扱う処理のことである。

プログラムへのファイル割り当て機能

ジョブステップで実行するプログラムが利用するファイルを、あとから自由に変更できるように、ファイル名をジョブ定義 XML ファイルで指定しておく機能である。

プロシジャ

カタログドプロシジャと入力ストリームプロシジャとの総称である。

プロシジャライブラリ

カタログドプロシジャファイルを格納したディレクトリパスを設定ファイルに定義しておき、ライブラリとして使用できるようにしたものである。プロシジャライブラリでは、複数のディレクトリを定義して連結されたライブラリとして処理できる。

併合

ファイルの併合の意味。併合元ファイルと併合先ファイル（併合先ファイルにもレコードが入っている）を合わせて1つのファイルにすることである。併合元ファイルには、索引順編成ファイルおよび順編成ファイルのどちらかを指定できる。

(マ行)

メタキャラクタ

ジョブ定義 XML ファイル内でそれぞれに特別な意味を持つキャラクタ（文字列）のこと。メタキャラクタと見なされる文字を次に示す。

- パイプ：|
- アンパーサント：&
- セミコロン：;
- より小符号：<
- より大符号：>
- 左括弧：(
- 右括弧：)
- ドル記号：\$
- バック・クウォート：`
- 単一引用符：'
- バックスラントまたは円記号
- 二重引用符："
- 改行キャラクタ
- スペース文字

- タブ・キャラクタ

メッセージ制御機能

JP1/IM - View の画面を使ってメッセージを出力し、応答を入力する機能である。

(ヤ行)

要素 (Element)

XML 文書での情報の単位である。

索引

記号

/opt/eurpm_adp/bin/eurpmc [EUR_PROGRAM
パラメータ] 492

/opt/eurps/bin/eurps [EUR_PROGRAM パラメー
タ] 492

-a [bjexchjob コマンド] 413

-A [bjexchjob コマンド] 413

-a [bjexrmjob コマンド] 464

-C [bjexchjob コマンド] 413

-c [bjexec コマンド] 420

-c [bjexm_message コマンド] 454

-c [bjexpd コマンド] 459

-D LT [bjexlsjob コマンド] 436

-D UTC [bjexlsjob コマンド] 436

-d [bjexchmsg コマンド] 417

-d [bjexgdg コマンド] 421

-d [bjexhkgdg コマンド] 427

-d [bjextrjob コマンド] 468

-e [bjexgdg コマンド] 422

-e [bjextrjob コマンド] 468

-F [bjexchjob コマンド] 413

-f [bjexcollect コマンド] 835

-f [bjexlsjob コマンド] 436

-f [bjexpdctl コマンド] 461

-f [bjexrmjob コマンド] 464

-h [bjexchjob コマンド] 412

-h [bjexchmsg コマンド] 417

-h [bjexec コマンド] 419

-h [bjexlsjob コマンド] 435

-h [bjexlslock コマンド] 443

-h [bjexlsmsg コマンド] 452

-h [bjexpdctl コマンド] 461

-h [bjexpd コマンド] 459

-H [bjexrmjob コマンド] 464

-h [bjexrmjob コマンド] 463

-h [bjextrjob コマンド] 468

-i [bjexchjob コマンド] 413

-i [bjexgdg コマンド] 422

-J [bjexchjob コマンド] 414

-J [bjexlsjob コマンド] 436

-j [bjexlslock コマンド] 443

-J [bjexrmjob コマンド] 464

-K [bjexchjob コマンド] 413

-l [bjexgdg コマンド] 421

-l [bjexlsjob コマンド] 435

-l [bjexlslock コマンド] 443

-l [bjextrjob コマンド] 468

-m [bjexgdg コマンド] 421

-m [bjexm_message コマンド] 454

-n [bjexchmsg コマンド] 417

-n [bjexec コマンド] 419

-n [bjexlsjob コマンド] 435

-n [bjexlsmsg コマンド] 452

-p [bjexm_message コマンド] 454

-p [bjexpd コマンド] 459

-q [bjexgdg コマンド] 422

-R [bjexchjob コマンド] 412

-r [bjexchmsg コマンド] 417

-r [bjexec コマンド] 419

-r [bjexm_message コマンド] 454

-r [bjextrjob コマンド] 468

-S [bjexchjob コマンド] 413

-s [bjexlsjob コマンド] 435

-T [bjexchjob コマンド] 413

-t [bjexlsjob コマンド] 436

-T [bjexlsjob コマンド] 436

-t [bjexlslock コマンド] 443

-T [bjexrmjob コマンド] 464

-t [bjextrjob コマンド] 468

-u [bjexgdg コマンド] 421

-v [bjexec コマンド] 420

-v [bjexgdg コマンド] 422, 427

-v [bjexpd コマンド] 459

-w [bjexlslock コマンド] 443

数字

0 バイトの SYSOUT ファイル転送機能 168

B

BJEX_DIR_任意名の指定方法 110
BJEX_DIR_任意名 [パラメータ] 478
bjex_msg_setup [コマンド] 411
bjex_PAUSE カタログドプロシジャ 199
bjex_WRITERX カタログドプロシジャ 201
bjex_WRITER カタログドプロシジャ 200
bjex_WRITE カタログドプロシジャ 199
bjexchjob [コマンド] 412
bjexchmsg [コマンド] 417
bjexcollect コマンド (障害調査のための資料の一括収集) 835
bjexec コマンドでのステップリスタートの実行 137
bjexec [コマンド] 419
bjexgdg [コマンド] 421
bjexhkgdg [コマンド] 427
bjexlsexjob [コマンド] 429
bjexlsjob [コマンド] 435
bjexlslock [コマンド] 443
bjexlsmg [コマンド] 452
bjexm_message コマンド 198
bjexm_message [コマンド] 454
bjexpdctl [コマンド] 461
bjexpd [コマンド] 459
bjexrmjob [コマンド] 463
bjextrjob [コマンド] 468
bjexu_dummy (ファイルおよびディレクトリ操作用ダミー処理の実行) 225
bjexu_dummy プログラム 220
bjexu_ismut および bjexu_ismut64 (索引順編成ファイルの作成, 削除, 索引順編成ファイルと順編成ファイルの複写, 変換および併合) 262
bjexu_ismut および bjexu_ismut64 プログラム 220
bjexu_pcopy (ファイルの複写) 226
bjexu_pcopy プログラム 220
bjexu_scopy (ファイルのレコード単位の複写) 237

bjexu_scopy プログラム 220

BOM [用語解説] 848

C

CALL 要素 (プロシジャの呼び出し) 566
CHARS [DD 要素] 567
CHARS [SysoutClass パラメータ] 542
CMT [DD 要素] 568
COBOL_LANG [パラメータ] 479
COBOL2002 で作成した COBOL アプリケーションの使用 198
COBOL プログラムで使用する文字コードを変更する場合の設定 366
COND [EXEC 要素] 586
COND [JOB 要素] 599
conftest [bjexpdctl コマンド] 461
COPIES [DD 要素] 568
COPIES [SysoutClass パラメータ] 542
COPYG [DD 要素] 568
COPYG [SysoutClass パラメータ] 542
Cosminexus との連携によるバッチアプリケーション実行 298

D

DD_DEALLOC_ON_ALLOCERR 480
DD_DEALLOC_ON_ALLOCERR [パラメータ] 480
DD_EXISTCHECK_COBOL [パラメータ] 483
DD_EXISTCHECK [パラメータ] 481
DD_SAMF [パラメータ] 485
DDNAME [SETENVFILE 要素] 610
DD 要素 (ファイルの定義) 567
DD 要素の書き方 109
DD 要素を修正するための指定方法 81
DD 要素をプログラムから使用する方法 111
DEST [DD 要素] 568
DEST [SysoutClass パラメータ] 543
DIR [PERMISSION 要素] 603
DISP [DD 要素] 568
DISPLAY_REPLY_MESSAGE
[DISPLAY_REPLY_MESSAGE パラメータ] 486

DISPLAY_REPLY_MESSAGE [パラメータ] 486
DISPMOD_NOFILE [パラメータ] 487
DSN [DD 要素] 570
DSN 属性の指定方法 110
DTD [用語解説] 848
DUMMY_FILE_TYPE [DUMMY_FILE_TYPE パラメータ] 488
DUMMY_FILE_TYPE [パラメータ] 488
DUMMY [SysoutClass パラメータ] 543

E

ELSE 要素 (IF 要素に対する分岐) 584
ENCODE [DD 要素] 572
ENCODE [SysoutClass パラメータ] 543
encoding [XML 宣言] 562
ENDIF 要素 (ジョブステップの実行制御の終了) 585
EUR_DEFINITION_FILE_DIR [パラメータ] 490
EUR_LANG [パラメータ] 491
EUR_PROGRAM [パラメータ] 492
EURENCODE [DD 要素] 572
EURFORM [DD 要素] 573
EXEC_COND_NOSTEPNAME
[EXEC_COND_NOSTEPNAME パラメータ] 493
EXEC_COND_NOSTEPNAME [パラメータ] 493
EXEC_COND_STEPSEARCH
[EXEC_COND_STEPSEARCH パラメータ] 494
EXEC_COND_STEPSEARCH [パラメータ] 494
EXEC_PARMSPPLIT [EXEC_PARMSPPLIT パラメータ]
495
EXEC_PARMSPPLIT [パラメータ] 495
EXEC 要素 (プログラムの実行) 586

F

FCB [DD 要素] 573
FCB [JOBPARM 要素] 602
FCB [SysoutClass パラメータ] 543
FCDFLT [DD 要素] 573
FCDFLT [SysoutClass パラメータ] 543
FILE [PERMISSION 要素] 603
FORMS [DD 要素] 573

FORMS [JOBPARM 要素] 602
FORMS [SysoutClass パラメータ] 543
FPERM [DD 要素] 573
ftp_connect [bjexpdctl コマンド] 462
FTPCTL [SysoutClass パラメータ] 543
FTPDATA [SysoutClass パラメータ] 543
FTYPE [DD 要素] 573
FTYPE [SysoutClass パラメータ] 543

G

GAIJ1 [DD 要素] 574
GAIJ1 [SysoutClass パラメータ] 544
GDG_REGISTER_JOBRC_THRESHOLD [パラメータ]
496
Group [EXEC 要素] 587

H

help [bjexpdctl コマンド] 462
HitachiBatchJobExec 要素 (ジョブ定義 XML ファイルの宣言) 593
HOLD [DD 要素] 574
HOLD [SysoutClass パラメータ] 544

I

IF 要素 (ジョブステップの実行制御の開始) 595
ISAMLIB_MODE [ISAMLIB_MODE パラメータ]
497
ISAMLIB_MODE [パラメータ] 497
ISKEY [DD 要素] 574
ISRECFM [DD 要素] 575
ISRECL [DD 要素] 575

J

Java バッチアプリケーション [用語解説] 848
JOB_RCDEFINE [JOB_RCDEFINE パラメータ]
498
JOB_RCDEFINE [パラメータ] 498
JOB_STATUS_DISPLAY [JOB_STATUS_DISPLAY
パラメータ] 500
JOB_STATUS_DISPLAY [パラメータ] 500

JOBLIB, STEPLIB, および PATH 環境変数の使用方法 88

JOBLIB DD 要素 87

JOBLOG_CONVERT [JOBLOG_CONVERT パラメータ] 501

JOBLOG_CONVERT [パラメータ] 501

JOBPARM_JSPECAN [パラメータ] 503

JOBPARM 要素 (ジョブ単位の制御情報の定義) 602

JOB 要素 (ジョブの属性と終了条件の定義) 599

JP1_HOSTNAME [パラメータ] 504

JP1/AJS からバッチジョブを実行する場合のジョブ定義 374

JP1/AJS 環境設定 54

JP1/AJS によるジョブスケジュール 326

JP1/AJS を使用したバッチジョブ業務の自動化の概要 327

JP1/AJS - View でのステップリスタートの実行 134

JP1/FTP 環境設定 55

JP1/FTP を使った印刷データ転送 379

JP1/FTP を使った印刷データ転送での障害対応 405

JP1/IM による実行状態の監視と実行結果の確認 340

JP1/IM - Manager が稼働するホスト名 [JP1IM_MANAGER_HOST パラメータ] 510

JP1/IM - Manager との連携による応答要求メッセージへの応答入力 388

JP1/Script の実行 304

JP1/Script を実行する方法 304

JP1EVENT_JOBINFO [JP1EVENT_JOBINFO パラメータ] 505

JP1EVENT_JOBINFO [パラメータ] 505

JP1EVENT_SUPPRESS_ID [パラメータ] 506

JP1FTP_RETRY_INTERVAL [パラメータ] 507

JP1FTP_RETRY_TIMES [パラメータ] 508

JP1FTPCARD [SysoutClass パラメータ] 544

JP1IM_MANAGER_CHKDELAY [パラメータ] 509

JP1IM_MANAGER_HOST [パラメータ] 510

JP1 イベント一覧 801

JP1 イベントの詳細 803

JSPEC [EXEC 要素] 587

JSPEC 機能 169

JSPEC 機能 [用語解説] 848

K

KEEP [DD 要素] 575

KEEP [SysoutClass パラメータ] 544

L

LANG="Java"を指定した場合の Java バッチアプリケーションを実行する方法 299

LANG [EXEC 要素] 588

LANG 環境変数値 [COBOL_LANG パラメータ] 479

LANG 環境変数値 [EUR_LANG パラメータ] 491

LOCK_DIR_RECURSIVE [LOCK_DIR_RECURSIVE パラメータ] 511

LOCK_DIR_RECURSIVE [パラメータ] 511

LOCK_JOBIMPLX [パラメータ] 512

LOCK_KEY_ID [パラメータ] 515

LOCK_MAXFILE [パラメータ] 513

LOCK_OPTION [LOCK_OPTION パラメータ] 514

LOCK_OPTION [パラメータ] 514

LOCK_RELEASE [パラメータ] 516

LOG_DIR [パラメータ] 517

LOG_FILE_CNT [パラメータ] 518

LOG_FILE_SIZE [パラメータ] 519

M

MAX_RESPONSE_REQUEST [パラメータ] 520

MESSAGE_REPLACE_SPACE [パラメータ] 521

MESSAGE_RETRY_TIMEOUT [パラメータ] 522

MODE [DD 要素] 575

MODE [SysoutClass パラメータ] 544

MSG_KEY_ID [パラメータ] 523

MSGCLASS_DEFAULT [パラメータ] 524

MSGCLASS [JOB 要素] 600

N

NAME [CALL 要素] 566

NAME [DD 要素] 576

NAME [JOB 要素] 599
NAME [PROC 要素] 604
NAME [REPLACESTEP 要素] 608
NAME [SETENV 要素] 609
NAME [SETSYMBOL 要素] 613
NAME [STEP 要素] 614
NAME [SYMBOL 要素] 615
NFS のサポート内容 32

O

OP [IF 要素] 595
OPTCDJ [DD 要素] 576
Option [EXEC 要素] 588
os [HitachiBatchJobExec 要素] 593
OVERC [DD 要素] 576
OVERC [SysoutClass パラメータ] 544
OVERLAY [DD 要素] 576
OVERLAY [SysoutClass パラメータ] 544

P

PARM [EXEC 要素] 589
PARMSPLIT [EXEC 要素] 589
PDE との連携による帳票出力 27
PERMANENT_FILE_DIR の指定方法 110
PERMANENT_FILE_DIR [パラメータ] 525
PERMISSION_DIR [パラメータ] 526
PERMISSION_FILE [パラメータ] 527
PERMISSION_SPOOL_STORE_JOB [パラメータ] 528
PERMISSION 要素 (ファイルまたはディレクトリのパーミションの定義) 603
PGM_RCABEND [パラメータ] 529
PGM [EXEC 要素] 589
PREST 機能 176
PREST 機能に関する注意事項 180
PREST 機能を使用するジョブの定義 365
PREST 機能 [用語解説] 848
PREST ファイル 176
PREST ファイルが存在して、新たに PREST ファイルを使用するジョブを実行できない場合の対処方法 401

PREST ファイルを使用できるプログラムとファイル 178
Prinfina MANAGER SP との連携による高速プリンタへの印刷の流れ 335
PROCEDURE_DIR [パラメータ] 530
PROCGROUP_QUEUE_NAME
[PROCGROUP_QUEUE_NAME パラメータ] 531
PROCGROUP_QUEUE_NAME [パラメータ] 531
PROCLIB [JOBPARM 要素] 602
PROC 要素 (プロシジャの定義) 604
prolog (XML の宣言) 562

R

RECFM [DD 要素] 576
RECFM [SysoutClass パラメータ] 544
RENAME [DD 要素] 577
REPLACESTEP 要素 (DD 要素または EXEC 要素の属性を変更対象とするジョブステップの定義) 608
REPLACE 要素 (DD 要素または EXEC 要素の属性の変更内容の定義) 606
RNW_FILE_METHOD [RNW_FILE_METHOD パラメータ] 532
RNW_FILE_METHOD [パラメータ] 532
ROOM [JOBPARM 要素] 602

S

SAMF [DD 要素] 577
SETENVFILE 要素 (環境変数格納ファイルの定義) 610
SETENV 要素 (環境変数の定義) 609
SETMAXRC 要素 (ジョブステップの終了コードの最大値, または最終ジョブステップの終了コードの変更) 611
SETSYMBOL 要素 (記号パラメータの値の定義) 613
SPOOL_DIR [パラメータ] 533
SPOOL_JOB_STORE [SPOOL_JOB_STORE パラメータ] 534
SPOOL_JOB_STORE [パラメータ] 534
SPOOL_JOB_TRANSFER
[SPOOL_JOB_TRANSFER パラメータ] 535
SPOOL_JOB_TRANSFER [パラメータ] 535

SPOOL_STORE_DIRNAME 書式
 [SPOOL_STORE_DIRNAME パラメータ] 537
 SPOOL_STORE_DIRNAME [パラメータ] 537
 SPOOL_STORE_DIR [パラメータ] 536
 SPOOL_STORE_SUBDIR
 [SPOOL_JOB_TRANSFER パラメータ] 538
 SPOOL_STORE_SUBDIR [パラメータ] 538
 sptxe [スクリプト実行プログラム] 304
 standalone [XML 宣言] 562
 start [bjexpdctl コマンド] 461
 status_log 461
 status [bjexpdctl コマンド] 461
 STDERRCLASS_DEFAULT [パラメータ] 539
 STDERRCLASS [EXEC 要素] 590
 STDERRCLASS [JOB 要素] 600
 STDOUTCLASS_DEFAULT [パラメータ] 540
 STDOUTCLASS [EXEC 要素] 590
 STDOUTCLASS [JOB 要素] 600
 STEPLIB DD 要素 87
 STEP 要素 (ジョブステップの定義) 614
 stop [bjexpdctl コマンド] 461
 SYMBOL 要素 (記号パラメータの定義) 615
 SYSIN データとなる文字列 159
 SYSIN データの書き方 159
 SYSIN データの読み込み 159
 SYSIN データを順編成固定長ファイルとしてプログラムに渡す方法 161
 SYSIN データ [用語解説] 848
 syslog に出力するメッセージの一覧 627
 SYSOUT_APPEND_WRITE
 [SYSOUT_APPEND_WRITE パラメータ] 541
 SYSOUT_APPEND_WRITE [パラメータ] 541
 SysoutClass [パラメータ] 542
 SYSOUT [DD 要素] 577
 SYSOUT の機能 164
 SYSOUT [用語解説] 848

T

TEMP_FILE_DIR_XML_USE
 [TEMP_FILE_DIR_XML_USE パラメータ] 547

TEMP_FILE_DIR_XML_USE [パラメータ] 547
 TEMP_FILE_DIR [パラメータ] 546
 TRACE_DIR [パラメータ] 548
 TRACE_FILE_CNT [パラメータ] 549
 TRACE_FILE_SIZE [パラメータ] 550
 TRACE_LEVEL [パラメータ] 551
 TRANSFER_EMPTY_FILE [SysoutClass パラメータ] 545
 TRUNC [SysoutClass パラメータ] 545
 TYPE [DD 要素] 578

U

UTF-8 [用語解説] 848

V

VALUE [SETENV 要素] 609
 VALUE [SETMAXRC 要素] 611
 VALUE [SETSymbOL 要素] 613
 VALUE [SYMBOL 要素] 615
 version [HitachiBatchJobExec 要素] 593

X

XML ジョブ制御要素一覧 564
 XML の仕様の版番号 562
 XML ファイルへの環境変数指定機能 308
 XML [用語解説] 849

い

一時 ISAM ファイル 104
 一時 ISAM ファイルの指定方法 104
 一時 ISAM ファイルのファイル名 106
 一時 ISAM ファイルを構成するファイル 105
 一時 ISAM ファイルを構成するファイル名 106
 一時 ISAM ファイルを作成するディレクトリ 105
 一時 ISAM ファイルを利用する場合の注意事項 107
 一時 ISAM ファイル [用語解説] 849
 一時ファイル 102, 319
 一時ファイルの指定方法 102
 一時ファイルを作成するディレクトリ 102

一時ファイルを利用する場合の注意事項 103
一時ファイル〔用語解説〕 849
一般ユーザ 316
イベント ID〔JP1EVENT_SUPPRESS_ID パラメータ〕 506
イベント到達確認までの待ち時間〔JP1IM_MANAGER_CHKDELAY パラメータ〕 509
印刷ジョブ転送のリトライ回数〔JP1FTP_RETRY_TIMES パラメータ〕 508
印刷ジョブ転送のリトライ間隔〔JP1FTP_RETRY_INTERVAL パラメータ〕 507
印刷ジョブの削除 338
印刷データの転送契機 386
インストール 35, 39
インストール後の設定 46
引用符〔用語解説〕 849

う

運用 314

え

エンコード種別〔EUR_LANG パラメータ〕 491
演算符号 121

お

応答 n〔bjexm_message コマンド〕 455
応答の入力 388
応答待ちメッセージ 205
応答要求メッセージ 205
応答要求メッセージの監視 341
応答要求メッセージの同時出力数 388
応答要求メッセージ〔bjexm_message コマンド〕 455
応答要求メッセージ〔用語解説〕 849

か

概要 315
カタログドプロシジャ 75
カタログドプロシジャと入力ストリームプロシジャ 75
カタログドプロシジャの機能 199

カタログドプロシジャの使用 199
カタログドプロシジャの使用方法 76
カタログドプロシジャ〔用語解説〕 849
環境変数 306
環境変数一括設定機能 310
環境変数に設定される内容 111
環境変数の優先順位 313
環境変数名を任意の名称に設定する方法 112
環境変数〔用語解説〕 849

き

記号パラメータに対する値の割り当て 84
記号パラメータの定義 84
記号パラメータ〔用語解説〕 849
機能 58
逆方向参照 114
逆方向参照〔用語解説〕 850
共用制御 182

く

クラスタ運用〔用語解説〕 850
クラスタシステム〔用語解説〕 850
クラス名〔SysoutClass パラメータ〕 542

こ

恒久ファイル 319
恒久ファイル〔用語解説〕 850
コード〔MESSAGE_REPLACE_SPACE パラメータ〕 521
コマンド 408
コマンド一覧 410
コマンドの記述形式 409

さ

最大応答要求メッセージ数〔MAX_RESPONSE_REQUEST パラメータ〕 520
索引順編成ファイルおよび順編成ファイルの複写, 変換および併合 (REPRO 文) 262
作成する一時 ISAM ファイルの定義内容 107
サブステップ 169

し

- しきい値 [GDG_REGISTER_JOBRC_THRESHOLD パラメータ] 496
- 識別情報 [LOCK_KEY_ID パラメータ] 515
- 識別情報 [MSG_KEY_ID パラメータ] 523
- システム管理者 316
- システム管理者への指示 619
- システムコールのエラー情報 628
- システムのプログラム使用方法 87
- 実行結果の参照 377
- 実行中のジョブ情報表示機能 196
- 実行ログ 393
- 実行ログ出力用ディスクの容量不足の対処方法 406
- 実行ログ [用語解説] 850
- 実体 [用語解説] 850
- 自動消去機能 (SCRATCH) が設定された世代データ群の注意事項 155
- 収集情報出力先ディレクトリ [bjexcollect コマンド] 835
- 終了コード 823
- 終了コード [PGM_RCABEND パラメータ] 529
- 受信待ちメッセージ 205
- 出力クラス 165
- 出力クラスによる制御 336
- 出力クラスの設定 321
- 出力クラス名 [MSGCLASS_DEFAULT パラメータ] 524
- 出力クラス名 [STDERRCLASS_DEFAULT パラメータ] 539
- 出力クラス名 [STDOUTCLASS_DEFAULT パラメータ] 540
- 出力クラス [用語解説] 850
- 出力形式 165
- 出力後保存 165
- 出力後保存 [用語解説] 850
- 出力情報の複写 165
- 出力保留 165
- 出力保留属性および出力非保留属性 337
- 出力保留 [用語解説] 850
- 障害対策 390
- 条件付き実行 122
- 条件付き実行の指定方法 121
- 条件付き実行 [用語解説] 851
- ジョブ 316
- ジョブが長時間待ち状態となった場合の対処方法 402
- ジョブ間 PREST ファイル 176
- ジョブ間 PREST ファイルの使用状況の表示 178
- ジョブ間 PREST ファイル [用語解説] 851
- ジョブコントローラ 61
- ジョブコントローラが使用する環境変数 306
- ジョブコントローラから実行されるプログラムに設定される環境変数 306
- ジョブコントローラの終了コード一覧 824
- ジョブコントローラ [用語解説] 851
- ジョブ識別子管理ファイル 72
- ジョブ識別子 [bjexchjob コマンド] 414
- ジョブ識別子 [bjexlsjob コマンド] 436
- ジョブ識別子 [bjexrmjob コマンド] 464
- ジョブ識別子 [用語解説] 851
- ジョブ実行のスケジュール 318
- ジョブステップ 117, 316
- ジョブステップ異常終了時の処理 171
- ジョブステップ間 PREST ファイル 177
- ジョブステップ間 PREST ファイル [用語解説] 851
- ジョブステップの実行制御 (IF 要素) 131
- ジョブステップの終了コード一覧 823
- ジョブステップの条件付き実行 (COND 属性) 121
- ジョブステップを条件付きで実行するときのステップ名の検索範囲 127
- ジョブステップを並列実行する機能 169
- ジョブステップ [用語解説] 851
- ジョブ多重度数 [LOCK_JOBMPLEX パラメータ] 512
- ジョブ単位のファイルおよびディレクトリに対する排他/共用制御機能 182
- ジョブ定義 XML ファイル 59, 553
- ジョブ定義 XML ファイル仕様 829
- ジョブ定義 XML ファイルのパス名 [bjexec コマンド] 420
- ジョブ定義 XML ファイルの文法 554
- ジョブ定義 XML ファイル [用語解説] 851

ジョブとジョブステップの概念 316
ジョブとジョブステップの実行順序の関係 317
ジョブの構成 60
ジョブのシンタクスチェック機能 61
ジョブ名などの情報をイベント一覧に表示 341
ジョブ名ワイルドカード [bjexchjob コマンド] 414
ジョブ名ワイルドカード [bjexlsjob コマンド] 436
ジョブ名ワイルドカード [bjexrmjob コマンド] 465
ジョブ名 [bjexchjob コマンド] 414
ジョブ名 [bjexlsjob コマンド] 436
ジョブ名 [bjexrmjob コマンド] 465
ジョブライブラリ [用語解説] 851
ジョブログ 393
ジョブログ出力機能 62
ジョブログ出力機能 [用語解説] 852
ジョブログ [用語解説] 851
ジョブを構成するためのジョブ定義 XML ファイル 60
資料の採取方法 400

す

スクリプト実行プログラム 304
ステップライブラリ [用語解説] 852
ステップリスタート 134
ステップリスタートでの注意事項 138
ステップリスタートを実行する上でのファイルの扱い 137
ステップリスタートを実行するための条件 134
ステップリスタート [用語解説] 852
ストレージ設計 57
スプール 71
スプール管理ファイル 72
スプールジョブ管理ファイル 72
スプールジョブディレクトリ 72
スプールジョブ転送サービス 320
スプールジョブ転送サービスの運用 334
スプールジョブ転送サービスの終了コード一覧 827
スプールジョブ転送サービス [用語解説] 852
スプールジョブ転送デーモンの運用 334
スプールジョブ転送デーモンの障害対策 407

スプールジョブ転送デーモン [用語解説] 852
スプールジョブの回復 339
スプールジョブの構造 73
スプールジョブの削除 338
スプールジョブのディレクトリの構造 73
スプールジョブのデータの流れ 74
スプールジョブ保管機能の設定方法 215
スプールジョブ保管機能 [用語解説] 852
スプールジョブ [用語解説] 852
スプールディレクトリ 72
スプールの構造 71
スプールの制御 71
スプール [用語解説] 852

せ

セキュリティ情報の設定 322
世代ディレクトリ名 [bjexgdg コマンド] 421
世代ディレクトリ名 [bjexhkgdg コマンド] 427
世代データ群 139
世代データ群の削除 147
世代データ群の作成 142
世代データ群の配置 143
世代データ群名 [bjexgdg コマンド] 421
世代データ群名 [bjexhkgdg コマンド] 427
世代データ群を構成するファイル 141
世代データ群 [用語解説] 852
世代登録可能数以上の世代ファイル番号の登録が発生した場合の注意事項 154
世代ファイル 139
世代ファイルおよび世代データ群の削除 146
世代ファイルの概要 139
世代ファイルの削除 146
世代ファイルの作成と参照 143
世代ファイルの参照 144
世代ファイルの参照 (世代データ群名指定) 145
世代ファイルの世代管理ファイルへの登録タイミング変更機能 [用語解説] 853
世代ファイルの注意事項 154
世代ファイルの保存日数超過による一括削除機能 [用語解説] 853

世代ファイルを削除する場合の注意事項 156
世代ファイルを作成する場合の DD 要素の注意事項 155
世代ファイルを参照する場合の DD 要素の注意事項 155
世代ファイル〔用語解説〕 852
絶対世代番号 139
絶対世代番号〔用語解説〕 853
設定ファイル 472
設定ファイルの使用例 552
設定ファイルのパラメーター一覧 474
設定ファイル〔用語解説〕 853
セットアップ前の作業 51
前提条件 28
前提プログラムおよび関連プログラム 28

そ

操作 370
相対世代番号 139
相対世代番号 [bjexgdg コマンド] 422
相対世代番号〔用語解説〕 853
属性〔用語解説〕 853

た

タイプコード 617
ダミー出力クラス 853
ダミーファイル 108
ダミーファイルの主な用途 108
ダミーファイルの指定方法 108
ダミーファイル〔用語解説〕 853

ち

帳票印刷 379
帳票処理の概要 335

て

定義ファイル 319
ディレクトリのパーミッション [PERMISSION_DIR パラメータ] 526

ディレクトリパス名 [PROCEDURE_DIR パラメータ] 530
ディレクトリ割り当て時の排他/共用要求の範囲 188
電子帳票システムとの連携による PDF ファイル変換の流れ 336
伝送カード [bjextrjob コマンド] 468
転送ディレクトリ 72

と

統合コンソールによる集中監視 340
統合コンソール〔用語解説〕 854
動作環境の確認 319
トラブルシューティング 392
トラブル発生時に採取が必要な資料 395
トレースレベル [TRACE_LEVEL パラメータ] 551
トレースログと起動ログ 395
トレース〔用語解説〕 854

に

入力ストリームプロシジャ 75
入力ストリームプロシジャの作成方法 75
入力ストリームプロシジャの使用方法 77
入力ストリームプロシジャ〔用語解説〕 854

は

パーミッション
[PERMISSION_SPOOL_STORE_JOB パラメータ] 528
排他/共用制御 182
排他/共用制御機能の設定 323
排他/共用制御機能〔用語解説〕 854
排他/共用制御に関する注意事項 195
排他/共用制御による実行待ち状態の確認方法 192
排他/共用制御の指定方法と適用範囲 186
排他制御 182
パス名 [BJEX_DIR_任意名パラメータ] 478
パス名 [bjexlsock コマンド] 443
パス名 [EUR_DEFINITION_FILE_DIR パラメータ] 490
パス名 [LOG_DIR パラメータ] 517

パス名 [PERMANENT_FILE_DIR パラメータ] 525
パス名 [SPOOL_DIR パラメータ] 533
パス名 [SPOOL_STORE_DIR パラメータ] 536
パス名 [TEMP_FILE_DIR パラメータ] 546
パス名 [TRACE_DIR パラメータ] 548
バックアップ 391
バックアップとリカバリ 391
バッチジョブ業務と実行順序での定義スケジュールの定義開始の契機登録 329
バッチジョブ業務と実行順序の定義 328
バッチジョブ業務と実行順序の定義スケジュールの定義 329
バッチジョブ業務の自動化 326
バッチジョブ業務の自動化の検討 323
バッチジョブ実行システム実行ユーザ管理 332
バッチジョブ実行システムと JP1/AJS でバッチジョブ業務と実行順序の定義を行う場合のジョブネット 328
バッチジョブ実行システムと Prinfina MANAGER SP および PDE の連携による印刷の流れ 337
バッチジョブ実行システムとは 22
バッチジョブ実行システムにおける SystemV セマフォ、共有メモリの管理 368
バッチジョブ実行システムのアンインストール 42
バッチジョブ実行システムのインストール 39
バッチジョブ実行システムの環境を複数構築する場合の設定 352
バッチジョブ実行システムの関連プログラム【AIX の場合】 28
バッチジョブ実行システムの関連プログラム【Linux の場合】 29
バッチジョブ実行システムの機能 59
バッチジョブ実行システムの機能の特長と処理手順 59
バッチジョブ実行システムのコマンド一覧（一般ユーザ用） 410
バッチジョブ実行システムのコマンド一覧（管理者用） 410
バッチジョブ実行システムのシステム設計 319
バッチジョブ実行システムの使用概要 371
バッチジョブ実行システムの処理の流れ 23

バッチジョブ実行システムの設定で使用するコマンド一覧 410
バッチジョブ実行システムのセットアップ手順の概要 36
バッチジョブ実行システムの全体構成 34
バッチジョブ実行システムの前提プログラム【AIX の場合】 28
バッチジョブ実行システムの前提プログラム【Linux の場合】 28
バッチジョブ実行システムの特長 24
バッチジョブ実行システムのバージョン情報の表示 46
バッチジョブ実行システムを使用するためのシステム構成 371
バッチジョブ実行システム【用語解説】 854
バッチジョブ定義の記述方法 330
バッチジョブの起動 372
バッチジョブの強制終了 377
バッチジョブの実行 372
バッチジョブの実行環境 329
バッチジョブの実行結果の一元管理 26

ひ

日付世代番号 140
日付世代番号【用語解説】 854
日付や時間ごとのサブディレクトリへの保管 216

ふ

ファイルサイズ [LOG_FILE_SIZE パラメータ] 519
ファイルサイズ [TRACE_FILE_SIZE パラメータ] 550
ファイルシステム 31
ファイル数 [LOCK_MAXFILE パラメータ] 513
ファイル操作ユティリティ 220
ファイル操作ユティリティの機能を定義する環境変数 307
ファイル操作ユティリティ【用語解説】 854
ファイルの後処理の指定 93
ファイルの作成 (DEFINE 文) 262
ファイルの準備 90
ファイルの状態と後処理 91

ファイルの状態と後処理の指定省略 98
ファイルの状態の指定 91
ファイルのパーミッション [PERMISSION_FILE パラメータ] 527
ファイルの複写 264
ファイルの併合 266
ファイルの変換 265
ファイルの連結処理 99
ファイルの連結処理 [用語解説] 854
ファイルの割り当て 90
ファイルの割り当て時に付与されるアクセス権限 113
複数のプログラム要素への属性定義 25
プリンタへの出力要求 166
プログラムとファイルの組み合わせによるバッチジョブの一括処理 24
プログラムのシステムへの登録と使用 87
プログラムへのファイル割り当て機能 109
プログラムへのファイル割り当て機能 [用語解説] 855
プロシジャグループ名 [PROCEDURE_DIR パラメータ] 530
プロシジャ内の DD 要素の修正方法 81
プロシジャ内の EXEC 要素の修正方法 78
プロシジャ内への DD 要素の追加方法 83
プロシジャの概要 75
プロシジャの記号パラメータによる修正方法 84
プロシジャの修正方法 78
プロシジャライブラリ [用語解説] 855
プロシジャ [用語解説] 855
プロセス一覧 834

へ

併合 [用語解説] 855
並列実行するジョブステップの指定 169
並列実行するジョブステップのジョブログ 172
並列実行するジョブステップの制限事項 170

ほ

保管先ディレクトリの名称 216
保存日数 [bjexhkgdg コマンド] 427

め

メインステップ 169
メタキャラクタ [用語解説] 855
メッセージ一覧 630
メッセージ出力コマンド 198
メッセージ制御機能 198
メッセージ制御機能での障害対応 403
メッセージ制御機能に関する注意事項 210
メッセージ制御機能のエラー情報 839
メッセージ制御機能の種類 198
メッセージ制御機能の設定 324
メッセージ制御機能を使用する場合の環境設定 55
メッセージ制御機能 [用語解説] 856
メッセージと終了コード 616
メッセージの記載形式 617
メッセージの形式 617
メッセージの出力 619
メッセージの出力形式 617
メッセージの出力先一覧 620
メッセージ番号の割り当て 618
メッセージ [bjexm_message コマンド] 455
面数 [LOG_FILE_CNT パラメータ] 518
面数 [TRACE_FILE_CNT パラメータ] 549

ゆ

ユーザプログラム 319
ユーザプログラムの使用方法 87

よ

要素 [用語解説] 856

り

リカバリー 391
リトライ打ち切り時間
[MESSAGE_RETRY_TIMEOUT パラメータ] 522
利用者の役割 316

れ

レベル 1 外字名指定 168

ろ

ログ情報の種類 [392](#)

論理ホスト名 [JP1_HOSTNAME パラメータ] [504](#)