

## uCosminexus Batch Job Execution Server 使用の手引 (Windows(R)用)

解説・手引・文法・操作書

3020-3-N64-60

---

## 前書き

### ■ 対象製品

P-2A12-A294 uCosminexus Batch Job Execution Server 09-10 (適用 OS : Windows Server 2016, Windows Server 2019, Windows Server 2022)

### ■ 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。

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。

UNIX は、The Open Group の登録商標です。

Win32 は、マイクロソフト 企業グループの商標です。

Windows は、マイクロソフト 企業グループの商標です。

Windows Server 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。

### ■ マイクロソフト製品の表記について

このマニュアルでは、マイクロソフト製品の名称を次のように表記しています。

製品名称	表記
Microsoft(R) Windows Server(R) 2016 Datacenter 日本語版	Windows Server 2016
Microsoft(R) Windows Server(R) 2016 Standard 日本語版	
Microsoft(R) Windows Server(R) 2019 Datacenter 日本語版	Windows Server 2019
Microsoft(R) Windows Server(R) 2019 Standard 日本語版	
Microsoft(R) Windows Server(R) 2022 Datacenter 日本語版	Windows Server 2022
Microsoft(R) Windows Server(R) 2022 Standard 日本語版	

なお、Windows Server 2016、Windows Server 2019 および Windows Server 2022 を総称する場合は、Windows と表記します。

## ■ 発行

2023 年 12 月 3020-3-N64-60

## ■ 著作権

All Rights Reserved. Copyright (C) 2009, 2023, Hitachi, Ltd.

## 変更内容

### 変更内容 (3020-3-N64-60) 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"><li>Windows Server 2016</li><li>Windows Server 2019</li><li>Windows Server 2022</li></ul> また、適用 OS から次の OS を削除した。 <ul style="list-style-type: none"><li>Windows Server 2008 R2</li><li>Windows Server 2012</li><li>Windows Server 2012 R2</li></ul>	—

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 (3020-3-N64-50) uCosminexus Batch Job Execution Server 09-00

追加・変更内容
メッセージ制御機能を使用できるようにした。
スプールジョブを削除できるようにした。
bjexu_scopy (ファイルのレコード単位の複写) で出力ファイルに追加書きできるようにした。
ISAM ライブラリ 64bit 版に対応できるようにした。
日付や時間ごとにスプールジョブを保管できるようにした。
ファイル操作ユーティリティとして bjexu_ismut64 プログラムを追加した。
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 JP1/IM のイベント一覧に表示する設定を追加した。
スプールジョブの一覧を表示できるようにした。
メッセージを追加した。 KAKC1609-E, KAKC1950-E, KAKC1951-E, KAKC4210-E, KAKC4211-E, KAKC4214-E, KAKC4215-E, KAKC4216-E, KAKC4303-E, KAKC4304-E, KAKC4305-E, KAKC4306-E, KAKC4310-E, KAKC4311-E, KAKC4312-E, KAKC4314-E, KAKC4316-E, KAKC4317-E, KAKC4318-E, KAKC4319-E, KAKC4364-E, KAKC4501-E, KAKC4560-E, KAKC4562-E, KAKC4567-E, KAKC5000-E, KAKC5001-I, KAKC5002-I, KAKC5003-I, KAKC5005-E, KAKC5009-E, KAKC5010-E, KAKC5011-E, KAKC5012-I, KAKC5013-I, KAKC5014-I, KAKC5015-E, KAKC5016-I, KAKC5017-E, KAKC5018-I, KAKC5019-E, KAKC5020-E, KAKC5021-E, KAKC5407-E, KAKC5408-I, KAKC5409-E, KAKC5410-E, KAKC5429-E, KAKD2150-E
メッセージを変更した。 KAKC1603-I
JP1 イベントを追加した



追加・変更内容
00005C20, 00005C21, 00005C22, 00005C23, 00005C24, 00005C25
ジョブ定義 XML ファイルの DTD を追加した。
メッセージ制御機能のエラー情報を追加した。

## 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

Windows および JP1/AJS の知識がある方を前提とし、バッチジョブ実行システムを使用して、メインフレーム相当の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。

#### 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 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 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 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。

## ■ 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



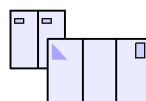
●プログラム



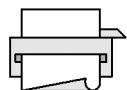
●コンピュータ



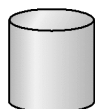
●ホストセンタ



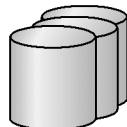
●端末プリンタ



●ファイル



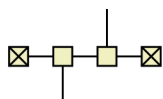
●ファイル(複数)



●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
[ ]	[ ] 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。 例 • [A] 「何も指定しない」か「Aを指定する」ことを示す。 • [B   C] 「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} * 「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。

記号	意味
… (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
<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>	Windowsのファイルパス名規則に従った文字列
<クラス名>	<記号名称>
<任意文字列>	任意の文字による文字列 <ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul>

シンタクス記号	指定できる文字の内容
<任意文字列>	<ul style="list-style-type: none"> <li>記号名称の範囲での利用を推奨する。</li> </ul> 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 指定できる文字の内容として JIS 2004 で追加された第 3 水準漢字, 第 4 水準漢字は使用できません。

# 目次

前書き	2
変更内容	4
はじめに	6

## 1 概要 20

1.1	バッチジョブ実行システムとは	21
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	30
1.3.3	2GB 超えのファイル (ラージファイル) の扱い	31
1.4	バッチジョブ実行システムの全体構成	32

## 2 インストール 33

2.1	インストールおよびアンインストール	34
2.1.1	インストール	34
2.1.2	アンインストール	35
2.1.3	インストールおよびアンインストール時の注意事項	35
2.1.4	バージョン間の移行時の注意事項	36
2.2	バッチジョブ実行システムのセットアップ手順の概要	37
2.2.1	基本システム構成の概要とセットアップ手順	37
2.3	セットアップ	39
2.3.1	セットアップ	39
2.3.2	JP1/AJS 環境設定	42
2.3.3	JP1/IM の設定	42
2.3.4	メッセージ制御機能を使用する場合の環境設定	44
2.4	ストレージ設計	46

## 3 機能 47

3.1	バッチジョブ実行システムの機能	48
3.1.1	バッチジョブ実行システムの特長	48
3.1.2	ジョブ定義 XML ファイル	48

3.1.3	ジョブコントローラ	50
3.1.4	スプール	59
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	64
3.2.1	プロシジャの概要	64
3.2.2	入力ストリームプロシジャの作成方法	64
3.2.3	カタログドプロシジャの使用方法	65
3.2.4	入力ストリームプロシジャの使用方法	66
3.2.5	プロシジャの修正方法	67
3.2.6	プロシジャ内の EXEC 要素の修正方法	67
3.2.7	プロシジャ内の DD 要素の修正方法	69
3.2.8	プロシジャ内への DD 要素の追加方法	72
3.2.9	プロシジャの記号パラメータによる修正方法	72
3.3	プログラムのシステムへの登録と使用	74
3.3.1	システムのプログラム使用方法	74
3.3.2	ユーザプログラムの使用方法	74
3.4	ファイルの割り当て	77
3.4.1	ファイルの準備	77
3.4.2	ファイルの状態と後処理	78
3.4.3	ファイルの連結処理	86
3.4.4	一時ファイル	87
3.4.5	一時 ISAM ファイル	88
3.4.6	ダミーファイル	91
3.4.7	プログラムへのファイル割り当て機能	92
3.4.8	逆方向参照	97
3.5	ジョブステップ	99
3.5.1	プログラムまたはコマンドを実行	99
3.5.2	スクリプトを実行	99
3.5.3	PARM 属性値が空白を含む場合	100
3.5.4	ジョブステップの正常終了／異常終了	101
3.5.5	注意事項	102
3.6	ジョブステップの条件付実行	103
3.6.1	条件付実行の指定方法	103
3.6.2	条件付実行	104
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	108
3.7	世代ファイル	113
3.7.1	世代ファイルの概要	113
3.7.2	世代データ群の作成	114
3.7.3	世代ファイルの作成と参照	115
3.7.4	世代ファイルおよび世代データ群の削除	117



- 3.7.5 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 118
- 3.7.6 世代ファイルの注意事項 122
- 3.8 SYSIN データの読み込み 126
  - 3.8.1 SYSIN データの書き方 126
  - 3.8.2 SYSIN データとなる文字列 126
  - 3.8.3 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 128
- 3.9 SYSOUT の機能 131
  - 3.9.1 出力クラス 132
  - 3.9.2 出力情報の複写 132
  - 3.9.3 出力形式 132
  - 3.9.4 出力保留 132
  - 3.9.5 出力後保存 132
  - 3.9.6 プリンタへの出力要求 133
  - 3.9.7 PDE ファイル形式 135
  - 3.9.8 レベル1 外字名指定 136
  - 3.9.9 SYSOUT 機能での注意事項 136
- 3.10 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 137
  - 3.10.1 排他/共用制御の指定方法と適用範囲 140
  - 3.10.2 排他/共用制御による実行待ち状態の確認方法 142
  - 3.10.3 排他/共用制御に関する注意事項 145
- 3.11 実行中のジョブ情報表示機能 146
  - 3.11.1 表示されるジョブ情報 146
  - 3.11.2 セットアップ時の設定 147
  - 3.11.3 設定ファイルの指定 147
  - 3.11.4 ジョブの多重度制御 147
- 3.12 メッセージ制御機能 149
  - 3.12.1 メッセージ制御機能の概要 149
  - 3.12.2 応答要求メッセージ 149
  - 3.12.3 JP1/IM - View との関係 150
  - 3.12.4 JP1/IM - Manager との関係 150
  - 3.12.5 JP1/IM - View からの応答入力 151
  - 3.12.6 メッセージ制御機能でのデータ置き換え機能 152
  - 3.12.7 DISPLAY 文による JP1 イベント出力の抑止機能 153
  - 3.12.8 メッセージ制御機能でのイベント到達確認 154
  - 3.12.9 メッセージ制御機能に関する注意事項 155
- 3.13 スプールジョブ保管機能 158
  - 3.13.1 スプールジョブ保管機能の概要 158
  - 3.13.2 スプールジョブ保管機能の設定方法 159
  - 3.13.3 スプールジョブ保管機能使用時の注意事項 160

- 3.14 ファイル操作ユーティリティ 161
  - 3.14.1 ファイル操作ユーティリティの概要 161
  - 3.14.2 ファイル操作ユーティリティのサポートするファイル 161
  - 3.14.3 ファイル操作ユーティリティの実行方法 163
  - 3.14.4 ファイル操作ユーティリティの制御文の形式 163
  - 3.14.5 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) 164
  - 3.14.6 bjexu\_pcopy (ファイルの複写) 165
  - 3.14.7 bjexu\_scopy (ファイルのレコード単位の複写) 177
  - 3.14.8 bjexu\_ismut および bjexu\_ismut64 (索引順編成ファイルの作成および削除, 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) 201
  - 3.14.9 メッセージの出力先 233
- 3.15 Cosminexus との連携によるバッチアプリケーション実行 234
  - 3.15.1 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 235
  - 3.15.2 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 239
  - 3.15.3 Java バッチアプリケーションの強制終了 239
- 3.16 XML ファイルへの環境変数指定機能 240
  - 3.16.1 指定方法 240
  - 3.16.2 使用例 241
- 3.17 環境変数 242
  - 3.17.1 ジョブコントローラが使用する環境変数 242
  - 3.17.2 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 242
  - 3.17.3 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 243

## 4 運用 244

- 4.1 概要 245
- 4.2 利用者の役割 246
  - 4.2.1 システム管理者 246
  - 4.2.2 一般ユーザ 246
- 4.3 バッチジョブ実行システムのシステム設計 249
  - 4.3.1 動作環境の確認 249
  - 4.3.2 出力クラスの設定 250
  - 4.3.3 セキュリティ情報の設定 250
  - 4.3.4 バッチジョブ業務の自動化の検討 251
  - 4.3.5 排他/共用制御機能の設定 251
  - 4.3.6 メッセージ制御機能の設定 252
- 4.4 JP1/AJS によるジョブスケジュール 254
  - 4.4.1 バッチジョブ業務の自動化 254
  - 4.4.2 バッチジョブの実行環境 257
  - 4.4.3 バッチジョブ定義の記述方法 258
  - 4.4.4 バッチジョブ実行システム実行ユーザ管理 260

4.4.5	バッチジョブおよびジョブネットの強制終了	260
4.4.6	注意事項	262
4.5	スプールジョブ転送の運用	264
4.5.1	スプールジョブ転送先の設定	264
4.5.2	帳票処理の概要	264
4.5.3	出力クラスによる制御	266
4.5.4	出力保留属性および出力非保留属性	267
4.5.5	スプールジョブの削除	268
4.5.6	スプールジョブを転送しない運用	268
4.5.7	スプールジョブ転送失敗時の注意事項	268
4.6	JP1/IM による実行状態の監視と実行結果の確認	269
4.6.1	統合コンソールによる集中監視	269
4.6.2	ジョブ名などの情報をイベント一覧に表示	270
4.6.3	応答要求メッセージの監視	270
4.7	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	272
4.7.1	複数のバッチジョブ実行システムの運用の前提条件	272
4.7.2	論理ホストのバッチジョブ実行システム的环境設定	274
4.7.3	複数のバッチジョブ実行システムの運用の注意事項	275
4.8	クラスタシステムで運用する場合の設定	276
4.8.1	クラスタシステムおよびクラスタ運用の概要	276
4.8.2	クラスタ運用の前提条件とサポート範囲	278
4.8.3	クラスタ運用の環境設定	279
4.8.4	クラスタ運用に関する注意事項	280
4.9	バッチジョブ実行システムの運用時の注意事項	282
4.9.1	システム停止時の注意事項	282
4.9.2	ウイルス対策ソフト実行時の注意事項	282
<b>5</b>	<b>操作</b>	<b>283</b>
5.1	バッチジョブ実行システムの使用概要	284
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	284
5.2	JP1/AJS によるジョブスケジュール	285
5.2.1	概要	285
5.2.2	バッチジョブ実行システムのバッチジョブの実行	285
5.2.3	実行結果の参照	289
5.3	Prinfina MANAGER SP および PDE との連携による帳票印刷	290
5.3.1	概要	290
5.3.2	印刷データ転送	290
5.3.3	SYSOUT 属性の PDE および Prinfina MANAGER SP での扱い	292
5.3.4	印刷データの転送契機	295

- 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 296
- 5.4.1 応答の入力 296
- 5.4.2 JP1/IM - Manager との連携 296
- 5.4.3 応答要求メッセージの同時出力数 296
- 5.4.4 応答入力 297

## 6 障害対策 299

- 6.1 バックアップとリカバリー 300
  - 6.1.1 バックアップ 300
  - 6.1.2 リカバリー 300
- 6.2 トラブルシューティング 301
  - 6.2.1 対処の方法 301
  - 6.2.2 ログ情報の種類 301
  - 6.2.3 トラブル発生時に採取が必要な資料 303
- 6.3 ジョブが長時間待ち状態となった場合の対処方法 306
- 6.4 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た環境でバッチジョブが異常終了する場合の確認手順 307
- 6.5 メッセージ制御機能での障害対応 308

## 7 コマンド 310

- コマンドの記述形式 311
- コマンド一覧 312
- bjexchmsg (応答待ちメッセージへの応答入力 (一般ユーザ用)) 313
- bjexec (バッチジョブの実行 (一般ユーザ用)) 315
- bjexgdg (世代データ群の生成および更新 (一般ユーザ用)) 317
- bjxlsexjob (実行中のジョブ情報の表示 (一般ユーザ用)) 320
- bjxlsljob スプールジョブの一覧表示 (一般ユーザ用)) 326
- bjxlsllock (ファイルの排他状態の表示 (一般ユーザ用)) 333
- bjxlslmsg (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 (一般ユーザ用)) 340
- bjexrmjob (スプールジョブの削除 (一般ユーザ用)) 342
- bjexsvc (バッチジョブ実行システムのサービスの登録およびサービス起動環境の設定 (管理者用)) 346
- bjex\_setup (バッチジョブ実行システムを実行する環境の設定 (管理者用)) 348

## 8 設定ファイル 350

- 設定ファイルの記述形式 351
- 設定ファイルのパラメーター一覧 353
- 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) 356
- 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) 357
- 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) 358
- 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) 360

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)	362
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)	363
DD_SAMF (DD 要素の SAMF 属性での省略値定義)	364
DSN_USE_ENV (DD 要素の DSN 属性への環境変数指定機能)	365
EXEC_COND_NOSTEPNAME (COND 属性ステップ名未指定時の演算対象定義)	366
EXEC_PARMSPLIT (EXEC 要素の PARMSPLIT 属性での省略値定義)	367
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)	368
ISAMLIB_MODE (ISAM ライブラリ (32 ビット版または 64 ビット版) の定義)	369
JOB_RCDEFINE (ジョブコントローラの終了コードの決定方法定義)	370
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)	372
JP1_HOSTNAME (バッチジョブ実行システムが稼働するホスト名定義)	373
JP1EVENT_SUPPRESS_ID (出力抑止する JP1 イベントのイベント ID 定義)	374
JP1IM_MANAGER_CHKDELAY (イベント到達確認までの待ち時間の指定)	375
JP1IM_MANAGER_HOST (JP1/IM - Manager が稼働するホスト名定義)	376
LOCK_JOBMPLEX (同時に実行できるジョブ数の定義)	377
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)	378
LOCK_OPTION (排他/共用制御を使用するかの定義)	379
LOCK_RELEASE (排他制御機能適用時の排他解除契機定義)	380
LOG_DIR (実行ログ出力ディレクトリのパス名定義)	381
LOG_FILE_CNT (実行ログをバックアップする面数定義)	382
LOG_FILE_SIZE (実行ログを出力するファイルサイズ定義)	383
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)	384
MESSAGE_REPLACE_SPACE (半角空白に置き換える 1 バイトコードの定義)	385
MESSAGE_RETRY_TIMEOUT (メッセージ制御機能でリトライを中断する定義)	386
MSGCLASS_DEFAULT (JOB 要素の MSGCLASS 属性での省略値定義)	387
PERMANENT_FILE_DIR (恒久ファイル用ディレクトリのパス名定義)	388
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)	389
PROCEDURE_DIR (プロシジャライブラリのパス名定義)	390
SET_PGM_PRIORITY (ユーザプログラムへの実行優先順位引き継ぎ)	391
SPOOL_DIR (スプールディレクトリのパス名定義)	392
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)	393
SPOOL_JOB_TRANSFER (スプールジョブの転送定義)	394
SPOOL_STORE_DIR (スプールジョブの保管先の定義)	395
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)	396
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)	397
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)	398
SYSOUT_APPEND_WRITE (SYSOUT データファイルへの書き込みモード定義)	399
SysoutClass (出力クラスの属性値定義)	400
TEMP_FILE_DIR (一時ファイルディレクトリのパス名定義)	404
TRACE_DIR (トレースを出力するディレクトリのパス名定義)	405
TRACE_FILE_CNT (トレース面数の定義)	406
TRACE_FILE_SIZE (トレースファイルサイズの定義)	407

TRACE\_LEVEL (トレース出力レベルの定義) 408  
設定ファイルの使用例 409

## 9 ジョブ定義 XML ファイル 410

ジョブ定義 XML ファイルの文法 411  
prolog (XML の宣言) 419  
XML ジョブ制御要素一覧 421  
CALL 要素 (プロシジャの呼び出し) 422  
DD 要素 (ファイルの定義) 423  
EXEC 要素 (プログラムの実行) 437  
HitachiBatchJobExec 要素 (ジョブ定義 XML ファイルの宣言) 443  
JOB 要素 (ジョブの属性と終了条件の定義) 445  
JOBPARM 要素 (ジョブ単位の制御情報の定義) 448  
PROC 要素 (プロシジャの定義) 449  
REPLACE 要素 (DD 要素または EXEC 要素の属性の変更内容の定義) 451  
REPLACESTEP 要素 (DD 要素または EXEC 要素の属性を変更対象のジョブステップの定義) 453  
SETENV 要素 (環境変数の定義) 454  
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) 455  
SETSymbOL 要素 (記号パラメータの値の定義) 457  
STEP 要素 (ジョブステップの定義) 458  
SYMBOL 要素 (記号パラメータの定義) 459

## 10 メッセージと終了コード 460

10.1 メッセージの形式 461  
10.1.1 メッセージの出力形式 461  
10.1.2 メッセージの記載形式 461  
10.1.3 メッセージ番号の割り当て 462  
10.1.4 システム管理者への指示 463  
10.1.5 メッセージの出力 463  
10.2 メッセージの出力先一覧 464  
10.2.1 KAKC で始まるメッセージの出力先一覧 464  
10.2.2 KAKD で始まるメッセージの出力先一覧 467  
10.3 メッセージに含まれるエラー情報 470  
10.4 メッセージ一覧 472  
10.5 JP1 イベント一覧 599  
10.6 JP1 イベントの詳細 600  
10.7 終了コード 615  
10.7.1 ジョブステップの終了コード 615  
10.7.2 ジョブコントローラの終了コード 616  
10.7.3 注意事項 619

## 付録 620

- 付録 A ジョブ定義 XML ファイル仕様 621
- 付録 A.1 ジョブ定義 XML ファイルの DTD 621
- 付録 B プロセス一覧 624
- 付録 C このマニュアルの参考情報 625
- 付録 C.1 関連マニュアル 625
- 付録 C.2 このマニュアルでの表記 627
- 付録 C.3 KB (キロバイト) などの単位表記について 628
- 付録 D メッセージ制御機能のエラー情報 629
- 付録 E 用語解説 632

## 索引 638

# 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スクリプトやバッチファイルなどの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定義できる。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、シェルスクリプトやバッチファイルなどの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。

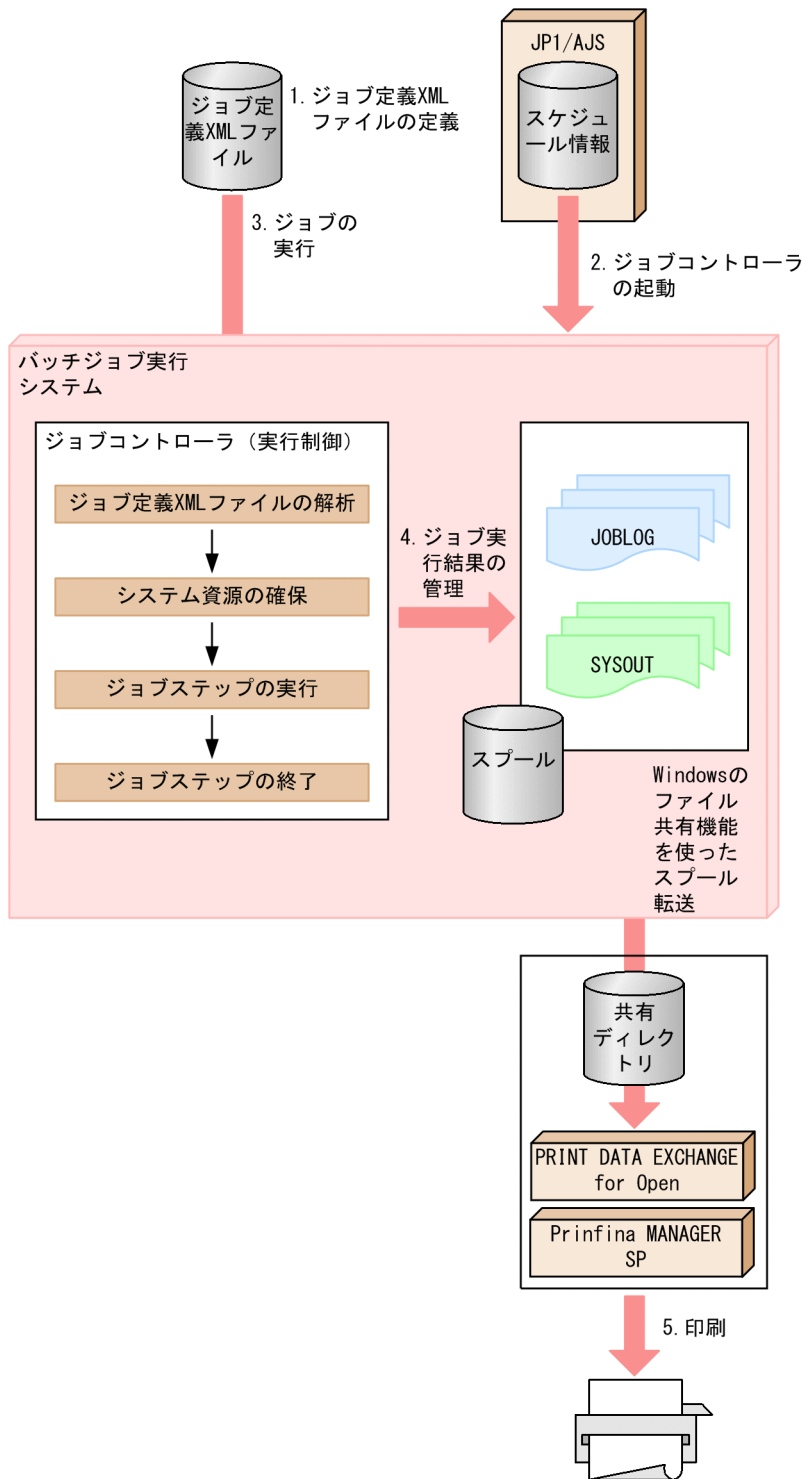
定義されたジョブ定義XMLファイルは、ジョブコントローラで解析される。ジョブコントローラは、入出力装置や各種システム資源の割り当て、および解放処理をして、ジョブの実行・終了を制御する。

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 と連携することで、次に示す運用が実現できる。

- JP1/AJS と連携することで、ジョブを実行するスケジュールを登録できる。
- PDE と連携することで、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プールに集めて、一元管理する。
5. 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

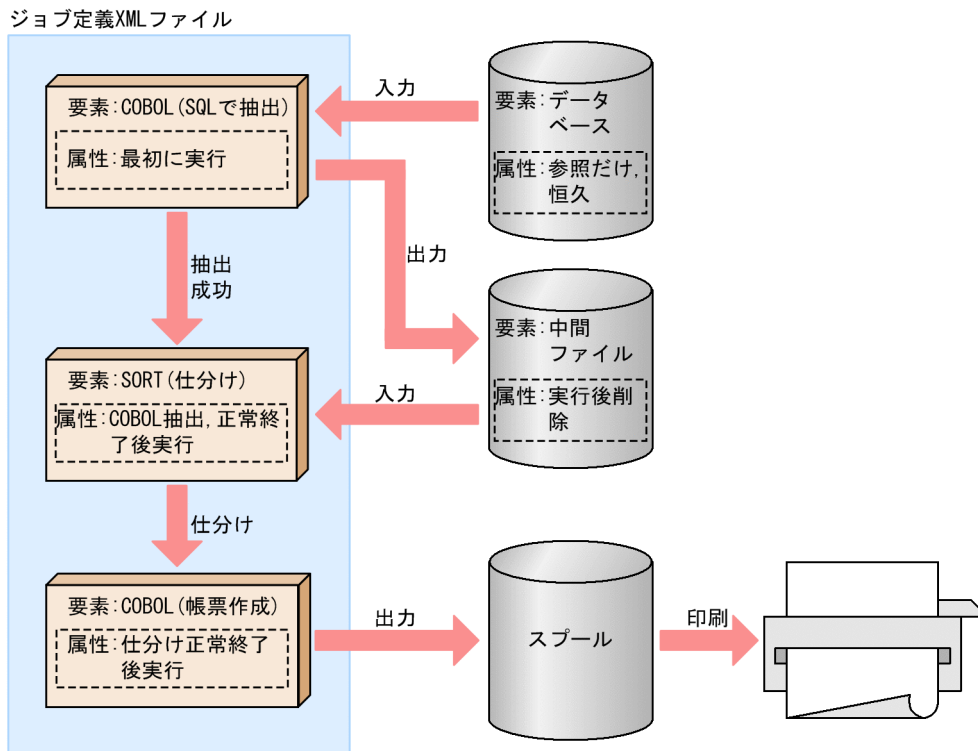
バッチジョブ実行システムの特長を次に示す。

- 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、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）で定義することで、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

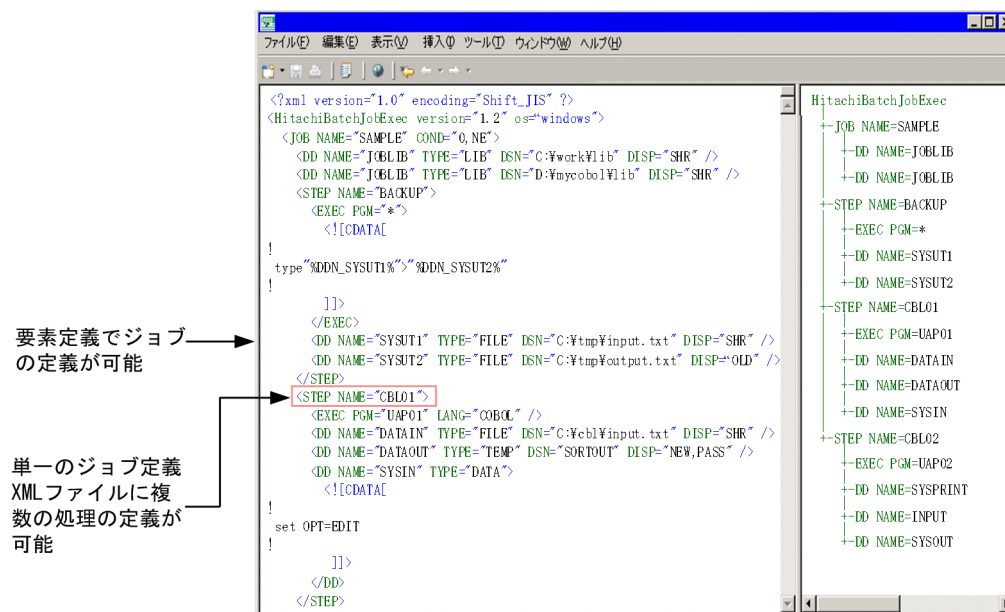
ファイルの要素に対して、世代ファイルおよび一時ファイルなどの属性を持たせることができる。実行終了後のファイルの削除なども要素の定義だけで実行できるため、スクリプトの記述も必要なく、運用の変更やメンテナンスが容易となる。

- 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

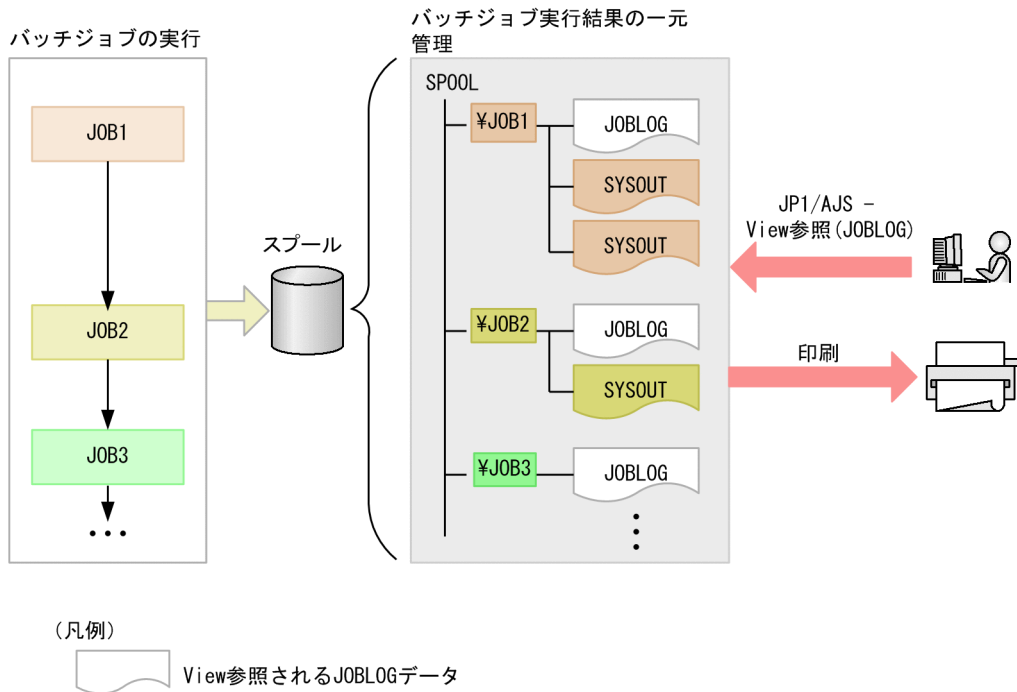


## 1.2.2 バッチジョブの実行結果の一元管理

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ない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



### 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。

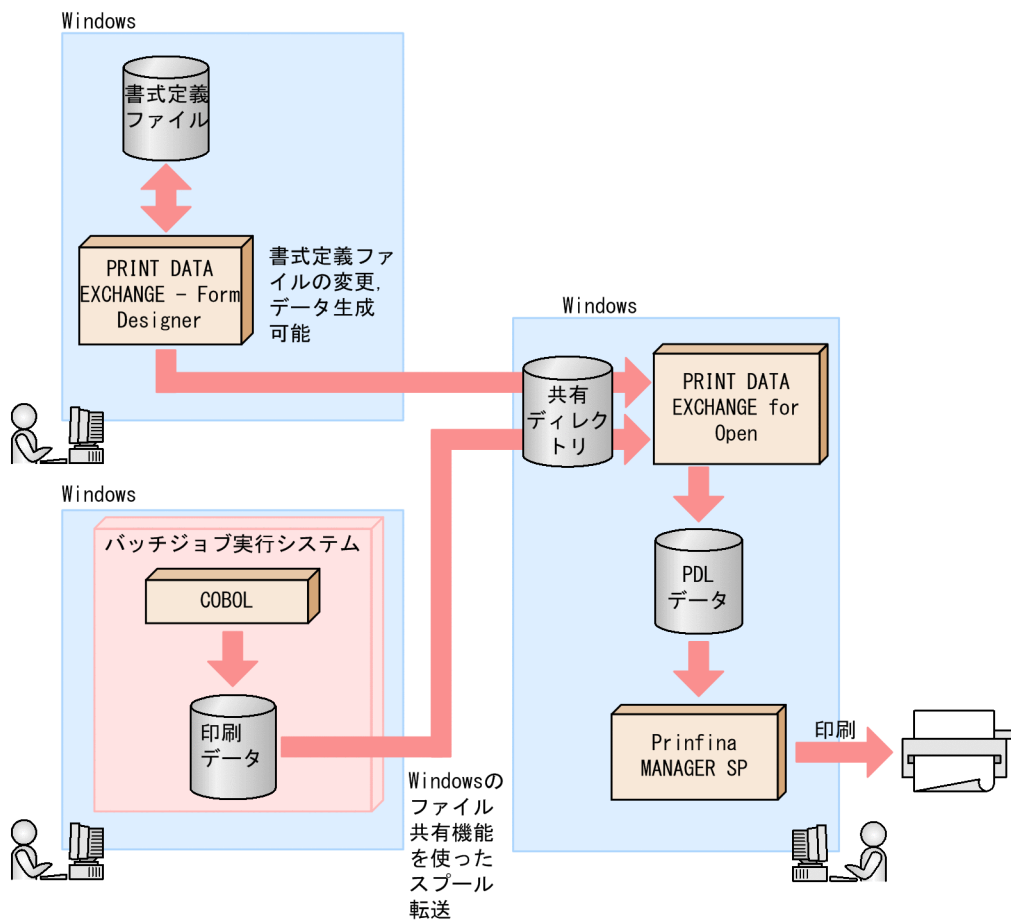
PRINT DATA EXCHANGE - Form Designer で書式や FCB を定義する書式定義ファイルを作成<sup>※</sup>し、PRINT DATA EXCHANGE for Open が PRINT DATA EXCHANGE - Form Designer で作成した書式定義ファイルとバッチジョブ実行システムの実行結果である印刷データを結合する。結合されたデータを PDL (Page Description Language) データに変換することで、プリンタ管理ソフトウェアである Prinfina MANAGER SP を通して帳票出力できる。

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的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的前提プログラム

プログラム	組み込むシステム
Windows	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関連プログラム

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)*	
uCosminexus Application Server Enterprise	
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)	
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 - Manager または JP1/Integrated Management 3 - Manager	JP1 イベント管理サーバ
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	

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。

- **Windows**  
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。
- **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。
- **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/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け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。バッチジョブ実行システムを実行する環境ではNTFSを使用すること。

バッチジョブ実行システムではFAT32はサポートしない。

また、バッチジョブ実行システムは、ネットワーク上のファイルシステムに配置できるファイルが限られている。ネットワーク上のファイルシステム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 1-3 ネットワーク上のファイルシステムのサポート内容

項目	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	×	×	All Users のドキュメントディレクトリ固定のため
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	×	×	
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	△	×	共用するファイルではないため指定を推奨しない。
PRTCNTL パラメータ、および PRTPDATA パラメータ	○	○	
SPOOL_STORE_DIR パラメータ	○	○	
TEMP_FILE_DIR パラメータ	△	×	共用するファイルではないため指定を推奨しない。
TRACE_DIR パラメータ、およびトレースファイル	×	×	

(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

×：設置または指定できない

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

×：複数のバッチジョブ実行システムで共用できない

注※ 恒久ファイル、世代ファイルをネットワーク上のファイルシステムに設置する場合、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

ファイル名には、ネットワークドライブ名から始まるパスではなく、「**¥¥コンピュータ名¥共有フォルダ¥ファイル名**」のようにコンピュータ名から始まるパスを指定してください。ネットワークドライブは、ユーザが Windows にログオンしたあとに再接続されるため、Windows にログオンしていない場合には、ファイルのパス名を正しく認識できない場合があります。

### 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使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<sup>※1</sup>（TYPE=TEMPISAM, DATA 以外）
- SYSOUT データファイル<sup>※2</sup>

注※1 UAP がラージファイルを扱えることが前提である。

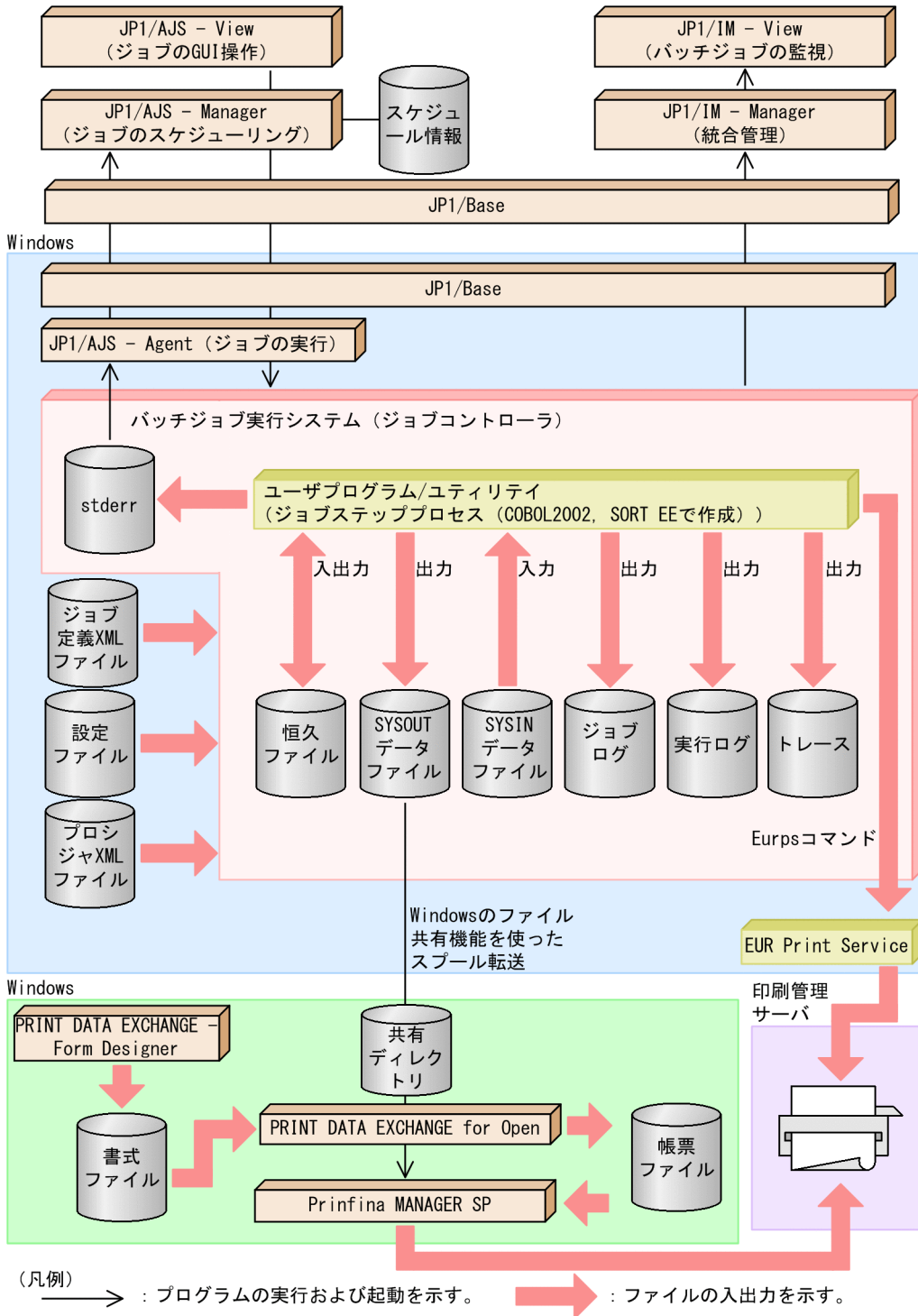
注※2 UAP がラージファイルを扱えることが前提である。また、ラージファイルの場合は転送しない。

## 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、ジョブコントローラによってジョブを実行する。

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 インストールおよびアンインストール

バッチジョブ実行システムのインストール、アンインストール、および注意事項について説明する。詳細については、リリースノートを参照のこと。

### 2.1.1 インストール

バッチジョブ実行システムを新規インストールする手順を次に示す。

1. バッチジョブ実行システムをインストールするサーバに管理者権限を持つユーザでログオンする。
2. すべてのプログラムを終了する。
3. バッチジョブ実行システムの CD-ROM 媒体を CD-ROM ドライブに入れる。
4. 起動したインストーラの指示に従って必要な情報を入力し、インストールする。

インストール時に定義する情報を次に示す。

- ユーザ情報
- インストール先ディレクトリ  
デフォルトではシステムドライブの「<プログラムファイルディレクトリ>¥Hitachi¥BJEX」以下にインストールされる。

5. OS を再起動する。

インストーラは PATH 環境変数にバッチジョブ実行システムで使用するコマンドへのパスを追加する。PATH 環境変数の変更を有効にするため、インストール後にいったん OS を再起動すること。

6. 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、次の設定をする。

- バッチジョブ実行システムのセットアップをしたあと、設定情報を有効にするため、OS の再起動をする。
- [スタート] メニューから [コントロールパネル] の [管理ツール] の [サービス] を開き、表示された [サービス] 管理ツールで [BjexSvc] サービスを起動する。もし、サービスが開始されなかった場合、イベントログに出力されているエラー情報を確認する。エラー情報に KAKC7004-E または KAKC7005-E メッセージが出力されている場合、設定ファイルの見直しを行う。設定ファイルに修正を加えた場合は、bjex\_setup -k コマンドを実行して、設定ファイルの形式が正しいことを確認したあと、[BjexSvc] サービスを起動する。

なお、バッチジョブ実行システムの論理ホスト環境で使用する場合、[BjexSvc\_論理ホスト名] サービスを登録する必要がある。詳細については、[\[4.7.2 論理ホストのバッチジョブ実行システムの環境設定\]](#)を参照のこと。また、バッチジョブ実行システムの論理ホストで運用する場合は「BjexSvc サービス」を「BjexSvc\_論理ホストサービス」に適宜読み替えること。

バッチジョブ実行システムをインストールすることで、ジョブ定義 XML ファイルの拡張子である".bjx"が、ジョブコントローラである bjexec コマンドに関連づけられる。

バッチジョブ実行システムのセットアップについては「[2.3 セットアップ](#)」を参照のこと。

## 2.1.2 アンインストール

バッチジョブ実行システムをアンインストールする手順を次に示す。

1. バッチジョブ実行システムをインストールするサーバに管理者権限を持つユーザでログオンする。
2. すべてのプログラムを終了する。
3. Windows の「コントロールパネル」の「プログラムの追加と削除」からバッチジョブ実行システム (uCosminexus Batch Job Execution Server) を削除する。

## 2.1.3 インストールおよびアンインストール時の注意事項

バッチジョブ実行システムのインストール、アンインストール時の注意事項を次に示す。

### (1) インストールについて

- 日立ネットワークオブジェクトプラザトレース共通ライブラリ (HNTRLib2) がサーバにインストールされていない場合、HNTRLib2 もインストールする。新規インストール先ディレクトリは、システムドライブの「<プログラムファイルディレクトリ>%Hitachi%HNTRLib2」および「<プログラムファイルディレクトリ>%Common Files%Hitachi」以下である。
- PATH 環境変数にインストール先ディレクトリ以下の bin, sbin ディレクトリへのパスを追加する。HNTRLib2 をインストールした場合は、システムドライブの「<プログラムファイルディレクトリ>%Common Files%Hitachi」へのパスも追加する。
- インストール先の templates ディレクトリに設定ファイル (bjex.conf) が存在する場合は、インストール時に上書きされない。
- すでにバッチジョブ実行システムがインストールされている環境に対してインストールを実施すると、修復セットアップとなり、設定ファイル (bjex.conf) 以外のファイルを上書きできる。新規にインストールし直す場合は、インストール済みのバッチジョブ実行システムを一度アンインストールしてから設定ファイル (bjex.conf) を削除して、新規インストールを実施すること。

### (2) アンインストールについて

- 設定ファイル (bjex.conf) および設定ファイルのあるディレクトリはアンインストール時に削除されない。不要な場合は削除すること。
- PATH 環境変数からインストール先ディレクトリ以下の bin, sbin ディレクトリへのパスも削除する。
- バッチジョブ実行システムをアンインストールすると、HNTRLib2 が自動的にアンインストールされる。ただし、HNTRLib2 を使用するほかのプログラムが存在する場合は、HNTRLib2 はアンインストールされない。HNTRLib2 を使用するプログラムがすべてアンインストールされるとき、HNTRLib2

もアンインストールされる。このとき PATH 環境変数の「<プログラムファイルディレクトリ>¥Common Files¥Hitachi」へのパスも削除される。

- BjexSvc\_論理ホスト名サービスを追加登録した場合、BjexSvc\_論理ホスト名サービスはアンインストール時に削除されない。不要な場合は Windows の sc コマンドなどで削除すること。

## 2.1.4 バージョン間の移行時の注意事項

- 次に示す条件を一つでも満たす場合には、既存のスプールディレクトリの内容を削除することなく、そのまま引き継いで使用できる。
  - 使用中のバッチジョブ実行システムよりも新しいバージョンに移行する場合。
  - 使用中のバージョンで修復セットアップをした場合。
  - 一度アンインストールしたバージョンと同じバージョンのバッチジョブ実行システムを再インストールした場合。
- 使用中のバッチジョブ実行システムよりも古いバージョンにバージョンダウンする場合は、スプールディレクトリの内容を引き継ぐことはできない。使用中のバッチジョブ実行システムをアンインストールしたあとでスプールディレクトリの内容を削除してから、古いバージョンのバッチジョブ実行システムをインストールすること。
- 移行後のバッチジョブ実行システムと異なるバージョンの BjexSvc\_論理ホスト名サービスは、正常に動作しない場合がある。異なるバージョンの BjexSvc\_論理ホスト名サービスは、バッチジョブ実行システムのアンインストール時に削除し、インストール後に新規に作成し直すこと。



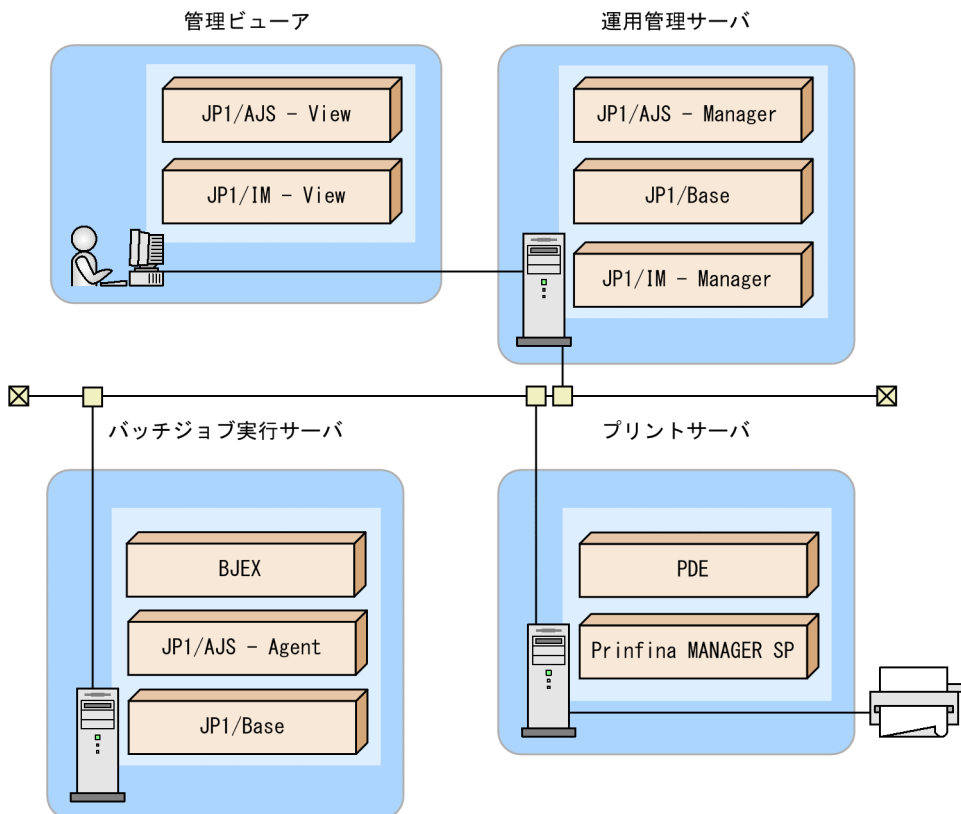
## 2.2 バッチジョブ実行システムのセットアップ手順の概要

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。

### 2.2.1 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/AJS - Manager, JP1/Base および JP1/IM - Manager のインストールとセットアップをする。

JP1/AJS - Manager, JP1/Base 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/AJS - Agent, JP1/Base のインストールとセットアップをする。

JP1/AJS - Agent, および JP1/Base 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 BJEX のセットアップをする。

BJEX のセットアップについては, 「[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. スプールジョブ管理ファイルの転送先を共有ディレクトリとして作成し, バッチジョブ実行システムのサーバから書き込み可能とする。
3. SYSOUT データファイルの転送先を共有ディレクトリとして作成し, バッチジョブ実行システムのサーバから書き込み可能とする。
4. 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. スプールジョブ管理ファイルの転送先を共有ディレクトリとして作成し, バッチジョブ実行システムのサーバから書き込み可能とする。
3. SYSOUT データファイルの転送先を共有ディレクトリとして作成し, バッチジョブ実行システムのサーバから書き込み可能とする。
4. 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。

## 2.3 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

バッチジョブ実行システムをインストール後に、次の手順でバッチジョブ実行システムを動作させるための環境を設定する。

#### 1. バッチジョブ実行システムの動作に必要なディレクトリと設定ファイルのひな形の作成

bjex\_setup コマンドでバッチジョブ実行システムが動作するために必要な標準的なディレクトリ作成と設定ファイルのひな形を作成する。

#### 2. 設定ファイルの変更

bjex\_setup コマンドで生成するディレクトリと設定ファイルのひな形はデフォルトの場所に作成される。設定ファイルのひな形は不完全であり、システム環境に合わせて SysoutClass, JP1\_HOSTNAMEなどを追加、変更する必要がある。

また、デフォルト以外のディレクトリを使用する場合はディレクトリの定義も変更する。ただし、設定ファイルの格納場所に変更できない。

バッチジョブ実行システムを動作させるための設定手順を次に示す。

### (1) bjex\_setup コマンドの実行

バッチジョブ実行システムを利用するユーザでログオンして、bjex\_setup コマンドでバッチジョブ実行システムの実行に必要な標準的なディレクトリと設定ファイルのひな形を作成する。

bjex\_setup コマンドは、バッチジョブ実行システムが動作するために必要なディレクトリをデフォルトの場所に作成する。また、インストール時に格納されたテンプレートの設定ファイルを入力として、設定ファイルのひな形を生成する。設定ファイルのテンプレートの内容を次に示す。

```
JP1_HOSTNAME dummy
MSGCLASS_DEFAULT A
<SysoutClass A>
DUMMY YES
MODE K
KEEP NO
</SysoutClass>
```

bjex\_setup コマンドが生成する設定ファイルは、関連 PP の設定に合わせて定義の追加、変更が必要である。詳細については「(2) 設定ファイルのカスタマイズ」を参照のこと。

- 生成するディレクトリ

<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥spool

<All Users のプロファイルのドキュメントディレクトリの場所>%Hitachi%BJEX%conf  
 <All Users のプロファイルのドキュメントディレクトリの場所>%Hitachi%BJEX%log  
 <All Users のプロファイルのドキュメントディレクトリの場所>%Hitachi%BJEX%perm  
 <All Users のプロファイルのドキュメントディレクトリの場所>%Hitachi%BJEX%proc  
 <All Users のプロファイルのアプリケーションデータディレクトリの場所>%Hitachi%BJEX%trace  
 <All Users のプロファイルのドキュメントディレクトリの場所>%Hitachi%BJEX%temp

- 生成するファイル

<All Users のプロファイルのドキュメントディレクトリの場所>%Hitachi%BJEX%conf%bjex.conf

## (2) 設定ファイルのカスタマイズ

bjex\_setup コマンドで生成した設定ファイルのひな形は、関連 PP の設定に合わせて定義を追加、変更する必要がある。また、スプールディレクトリなどをデフォルト以外のディレクトリにしたい場合は、任意のディレクトリに定義し直す必要がある。追加、変更は任意のテキストエディタで行う。設定ファイルのひな形は、bjex\_setup コマンドが次に示す場所に格納する。

<All Users のプロファイルのドキュメントディレクトリの場所>%Hitachi%BJEX%conf

bjex\_setup コマンドを実行する場合、設定ファイルがすでに存在しているときは bjex\_setup は異常終了する。再度 bjex\_setup コマンドを実行する場合は、既存の設定ファイルを削除してから実行すること。

## (3) バッチジョブ実行システム用のディレクトリ作成とファイルの権限設定

バッチジョブ実行システムを使用するには、次の表に示すディレクトリが必要である（通常は bjex\_setup コマンドがデフォルトのディレクトリを自動的に作成、および設定をする。ただし、バッチジョブ実行ユーザに対して、ディレクトリのアクセス権限の設定が必要である）。これらをデフォルト以外の場所にしたい場合は、システム管理者が手作業で作成する必要がある。また、次の表に示す設定ファイルのディレクトリの定義を変更しておく必要がある。

表 2-1 バッチジョブ実行システムを使用時に必要なファイルディレクトリ

ディレクトリの種類	説明
カタログドプロシジャ用	定形のジョブ定義 XML ファイルを登録するディレクトリ
一時ファイル用	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
スプール用	ジョブログや SYSOUT データを格納するディレクトリ
恒久ファイル用	恒久ファイルの指定で絶対パスの指定をしなかった場合に利用するカレントディレクトリ
実行ログ用	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
トレース用	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

これらのディレクトリには、適切なアクセス権を設定する必要がある。

また、バッチジョブ実行システムはスプール用ディレクトリ内にジョブ識別子管理ファイル (bjex.jobid) を作成する。このファイルは同一ホスト上でバッチジョブ実行システムを使用する全ジョブ実行ユーザによって共有され、バッチジョブ実行システムを実行するたびに参照・更新される。バッチジョブ実行システムのインストール直後の設定では、ジョブ識別子管理ファイルは最初にバッチジョブ実行システムを実行したジョブ実行ユーザを所有者として作成されるため、複数 OS ユーザによる運用を行う場合、ほかのジョブ実行ユーザが同ファイルを変更できるようにジョブ識別子管理ファイルのアクセス権を変更すること。ジョブ識別子管理ファイルの詳細については「[3.1.4\(1\)\(a\) スプールの構造](#)」を参照のこと。

ジョブ実行ユーザに必要な権限を次の表に示す。

表 2-2 ジョブ実行ユーザに必要な権限

ディレクトリ/ファイル	読み込み許可	書き込み許可	作成権限	削除権限
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	○	○	○	○
ジョブ識別子管理ファイル (bjex.jobid)	○	○*	○	△

(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。

注※ 複数の OS ユーザによる運用を行う場合は、最初にバッチジョブ実行システムを実行したあと、そのユーザ以外のジョブ実行ユーザもジョブ識別子管理ファイルを変更できるようにアクセス権を変更すること。

#### (4) バージョンの異なるバッチジョブ実行システムをインストールした場合の注意事項

従来の設定ファイルが残っている場合は、定義内容を確認するとともに、bjex\_setup -k で設定ファイルの形式が正しいことを検証すること。

#### (5) 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ための設定

バッチジョブ実行システムで排他/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には、次に示す設定が必要となる。

- 排他/共用制御機能を使用するには、設定ファイルの LOCK\_OPTION パラメータで ENABLE を指定する。また、必要に応じて LOCK\_JOBMPLEX パラメータ、LOCK\_MAXFILE パラメータの設定をする。
- 実行中のジョブ情報表示機能を使用するには、設定ファイルの JOB\_STATUS\_DISPLAY パラメータで ENABLE を指定する。また、必要に応じて LOCK\_JOBMPLEX パラメータを設定する。
- bjexsvc -setup コマンドでローカルグループの生成と BjexSvc サービスに通知する情報の登録を行う。生成するローカルグループは、"BJEX File Exclusive Control Group"である。すでにローカルグループ"BJEX File Exclusive Control Group"が生成されている場合、BjexSvc サービスに通知する情報の登録だけを行う。
- 生成したローカルグループに、バッチジョブ実行システムを使用するローカルユーザを登録する。bjexec コマンド、bjexlsock コマンド、および bjexlsexjob コマンドを使用するローカルユーザ、ならびに [BjexSvc] サービスを起動するローカルユーザを、ローカルグループに登録しておく必要がある。
- BjexSvc サービスのプロパティを開き、ログオンタグでログオンするアカウントとパスワードを指定する。ここで指定するアカウントは、バッチジョブ実行システムをインストールしたサーバの管理者権限を持つユーザとする。

## 2.3.2 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

```
<JP1/AJS-Managerのインストール先ディレクトリ>
¥log¥jppagent¥jppagt_{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。

## 2.3.3 JP1/IM の設定

バッチジョブ実行システムが発行する JP1 イベントを JP1/IM で表示させるために、事前に環境を設定する必要がある。ここでは、JP1/IM の環境設定の留意点について説明する。

## (1) JP1 イベントの発行先ホスト

バッチジョブ実行システムが発行する JP1 イベントは、メッセージ制御機能が発行する JP1 イベントだけ発行先ホストが異なる。それぞれの JP1 イベントの発行先ホストを次の表に示す。メッセージ制御機能の詳細については、「3.12 メッセージ制御機能」を参照のこと。

表 2-3 JP1 イベントの発行先ホスト

JP1 イベント	イベント ID	発行先ホスト
メッセージ制御機能を使用することで発行される JP1 イベント	<ul style="list-style-type: none"><li>• 00005C20</li><li>• 00005C21</li><li>• 00005C22</li><li>• 00005C23</li><li>• 00005C24</li><li>• 00005C25</li></ul>	JP1IM_MANAGER_HOST に指定したホストに直接発行
バッチジョブ実行システムが発行する上記以外の JP1 イベント	上記以外のイベント ID	JP1/Base の JP1 イベントの転送経路を使用して発行

JP1/Base の JP1 イベントの転送経路の詳細は、JP1 のマニュアルを参照のこと。

## (2) 拡張属性定義ファイルのコピー

バッチジョブ実行システムが発行する JP1 イベントの詳細情報を JP1/IM - View で表示させるには、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する必要がある。

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 templates 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。

- 拡張属性定義ファイルのコピー手順
  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。また、Linux に格納する場合は拡張属性定義ファイルを UTF-8 に変換すること。

- UNIX の場合  
/etc/opt/jp1cons/conf/console/attribute/ または  
共有ディレクトリ/jp1cons/conf/console/attribute/
- Windows の場合



Console パス¥conf¥console¥attribute¥ または  
共有ディレクトリ¥jplcons¥conf¥console¥attribute¥

## 注意事項

複数バージョンのバッチジョブ実行システムが混在するシステムを構築する場合は、最新バージョンの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使用すること。

## 2.3.4 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、「4.3.6 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) アダプタコマンドの設定

JP1/IM - Manager との連携を有効にするために、メッセージ応答用アダプタコマンド設定ファイルを作成する。

設定ファイルの作成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. JP1/IM - View から応答するためのメッセージ応答用アダプタコマンド設定ファイルの「cmdpath」に指定されているパスを確認し、バッチジョブ実行システムのインストールフォルダと異なる場合は、インストールフォルダへ修正する。

メッセージ応答用アダプタコマンド設定ファイルの格納ディレクトリを次に示す。

```
<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>¥sample¥Adapter_HITACHI_BJEX_MESSAGE.conf
```

3. JP1/Base のインストール先に、手順 2 のメッセージ応答用アダプタコマンド設定ファイルをコピーする。

コピー先ディレクトリを次に示す。

```
<JP1/Baseのインストール先ディレクトリ>¥plugin¥conf
```

### (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。有効にしている場合、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) JP1IM\_MANAGER\_HOST パラメータの設定

メッセージ制御機能を使用するためには、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、次に示すホスト名を仮定する。

- 物理ホストの場合  
JP1\_HOSTNAME パラメータに指定したホスト名
- 論理ホストの場合  
JP1 の論理ホスト名

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。

## 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

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。
- JP1/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。

次に主な内容について説明する。

#### (1) 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 形式でその要求を記述する。ファイルの拡張子は".bjx"とすること。

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。

ジョブコントローラプロセスは、bjexec.exe プロセスとその子プロセスの bjexecsub.exe プロセスの 2 プロセスから成る。

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 SYSMSG 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 PC 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多い場合、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 SYSMSG の出力データが多くなる場合も考えられるため、JP1/AJS の PC 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。

### (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"><li>• ジョブ識別子</li><li>• スプールジョブディレクトリ</li><li>• 実行ホスト名</li><li>• JP1 情報</li></ul>

出力契機	出力内容
ジョブ開始時	<ul style="list-style-type: none"> <li>• ジョブの開始時刻</li> <li>• ジョブ識別子</li> <li>• ジョブ名</li> </ul>
ジョブステップ開始時	<ul style="list-style-type: none"> <li>• ジョブステップの開始時刻</li> <li>• ジョブ識別子</li> <li>• ジョブ名</li> <li>• ジョブステップ名</li> </ul>
ジョブステップ終了時	<ul style="list-style-type: none"> <li>• ジョブステップの終了時刻</li> <li>• ジョブ識別子</li> <li>• ジョブ名</li> <li>• ジョブステップ名</li> <li>• ジョブステッププロセスの終了コード</li> <li>• C-TIME</li> <li>• E-TIME</li> </ul>
ジョブ終了時	<ul style="list-style-type: none"> <li>• ジョブの終了時刻</li> <li>• ジョブ識別子</li> <li>• ジョブ名</li> <li>• ジョブの終了コード</li> <li>• ジョブステップの C-TIME の合計</li> <li>• ジョブステップの E-TIME の合計</li> </ul>

注※ ジョブログヘッダはJOBLOG ファイルの先頭に出力する。

JOBLOG の出力例を次に示す（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]
Jobid : 001047 ... (1)
Spool Directory : C:\Users\Public\Documents\Hitachi\BJEX\spool\001047 ... (2)
Date : 2023/10/01 ... (3)
Configure File : C:\Users\Public\Documents\Hitachi\BJEX\conf\bjex.conf ... (4)
HostName : nvm014349 ... (5)
[JP1 Parameter]
JP1JobName : SAMPLE_JCL.bjx ... (6)
JP1JobID : 0000000206 ... (7)
JP1_USERNAME : bjex ... (8)
JP1UNCName : nvm014349 ... (9)
JP1NBQQueueName: ¥¥nvm014349¥¥SYSTEM ... (10)
JP1Priority : 32 ... (11)
AJSEXECID : @A105 ... (12)
-----

***** BJEX MESSAGE *****
14:14:46 001047 KAKC0091-I SAMPLE_JCL Job started.
14:14:46 001047 KAKC0092-I SAMPLE_JCL.STEP1 Step started.
14:14:46 001047 KAKC0097-I SAMPLE_JCL.STEP1 Step ended. rc=0 E-Time=0.036s C-Time=0.015s
14:14:46 001047 KAKC0092-I SAMPLE_JCL.STEP2 Step started.
14:14:46 001047 KAKC0097-I SAMPLE_JCL.STEP2 Step ended. rc=0 E-Time=0.049s C-Time=0.015s
14:14:46 001047 KAKC0092-I SAMPLE_JCL.STEP3 Step started.
14:14:46 001047 KAKC0097-I SAMPLE_JCL.STEP3 Step ended. rc=0 E-Time=0.026s C-Time=0.000s
14:14:46 001047 KAKC0092-I SAMPLE_JCL.STEP4 Step started.
14:14:46 001047 KAKC0097-I SAMPLE_JCL.STEP4 Step ended. rc=0 E-Time=0.027s C-Time=0.000s
14:14:46 001047 KAKC0092-I SAMPLE_JCL.STEP5.SAMPLE_PROC.STEP1 Step started.
14:14:46 001047 KAKC0097-I SAMPLE_JCL.STEP5.SAMPLE_PROC.STEP1 Step ended. rc=0 E-Time=0.028s C-Time=0.015s
14:14:46 001047 KAKC0098-I SAMPLE_JCL Job ended. rc=0 E-Time=0.166s C-Time=0.045s

```



(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。

ジョブ実行中のスプールジョブディレクトリ名を表示する。

詳細は「[3.1.4\(1\)\(a\)\(2\) スプールジョブディレクトリ](#)」を参照のこと。

(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 JP1\_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 JP1\_HOSTNAME パラメータは無視され、JP1 の論理ホスト名となる。詳細については、「[7. コマンド](#)」の「[bjexec \(バッチジョブの実行 \(一般ユーザ用\)\)](#)」を参照のこと。

(7) JP1 の環境変数

(8) JP1 ジョブ名

JP1 で定義した実行ファイル名、またはジョブ名。

(9) JP1 ジョブ番号

JP1 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。

## (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 桁の 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 桁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  
+/: 修正された可能性のある入力ストリームプロシジャ  
XX：カタログドプロシジャ  
X/: 修正された可能性のあるカタログドプロシジャ  
\*\*：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="windows" >
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" />
000010 <DD NAME="DD2_WRITE" TYPE="TEMP" DSN="DD2FILE" DISP="NEW,PASS" />
000011 </STEP>
000012 <STEP NAME="STEP3" >
000013 <EXEC PGM="UAP3" PARM="%%SAMPLE_JCL_STEP3%%" LANG="COBOL" />
000014 <DD NAME="DD2_READ" TYPE="TEMP" DSN="DD2FILE" DISP="OLD,DELETE" />
000015 </STEP>
000016 <STEP NAME="STEP4" >
000017 <EXEC PGM="UAP4" PARM="%%SAMPLE_JCL_STEP4%%" LANG="COBOL" />
000018 <DD NAME="DD4_READ" TYPE="DATA" SAMF="YES" >
<![CDATA[
!
INPUT DATA 1
INPUT DATA 2
!
]]>

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

>>>> C:\Users\Public\Documents\Hitachi\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="windows" >
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>
<<<<< C:\Users\Public\Documents\Hitachi\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 SYSMMSG 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"> <li>ジョブ定義 XML ファイルの誤り</li> <li>ジョブのメッセージクラス</li> </ul>
ファイル割り当て時	<ul style="list-style-type: none"> <li>割り当てたファイルの DD 名</li> <li>パス名</li> </ul>
ファイル割り当て解除時	<ul style="list-style-type: none"> <li>割り当て解除したファイルの DD 名</li> <li>パス名</li> </ul>
ジョブ実行時	<ul style="list-style-type: none"> <li>ジョブコントローラが作成したワークファイルのパス名</li> <li>ジョブコントローラが削除したワークファイルのパス名</li> </ul>

SYSMMSG の表示例を次に示す。

```

***** SYSTEM MESSAGE *****
14:14:46 001047 KAKC0900-I Parsing of xml file "C:\bjex_xml\SAMPLE_JCL.bjx" started.
14:14:46 001047 KAKC1100-I Parsing of proc file "C:\Users\Public\Documents\Hitachi\BJEX\proc\SAMPLE_PROC.xml"
started.
14:14:46 001047 KAKC1109-I Parsing of proc file "C:\Users\Public\Documents\Hitachi\BJEX\proc\SAMPLE_PROC.xml"
ended. code=0
14:14:46 001047 KAKC0909-I Parsing of xml file "C:\bjex_xml\SAMPLE_JCL.bjx" ended. code=0
14:14:46 001047 KAKC0707-I MSGCLASS is "A".
14:14:46 001047 KAKC1600-I SAMPLE_JCL Job allocated.
14:14:46 001047 KAKC1601-I SAMPLE_JCL.STEP1 Step allocated.
14:14:46 001047 KAKC1602-I DD1, .
C:\Users\Public\Documents\Hitachi\BJEX\spool\001047\001_STEP1_001_DD1.sysout
14:14:46 001047 KAKC1603-I DD1, .
C:\Users\Public\Documents\Hitachi\BJEX\spool\001047\001_STEP1_001_DD1.sysout, size=532
14:14:46 001047 KAKC1601-I SAMPLE_JCL.STEP2 Step allocated.
14:14:46 001047 KAKC1602-I DD2_WRITE, NEW, C:\Users\Public\Documents\Hitachi\BJEX\tmp\BJX95FC.tmp
14:14:46 001047 KAKC1603-I DD2_WRITE, PASS, C:\Users\Public\Documents\Hitachi\BJEX\tmp\BJX95FC.tmp,
size=1039
14:14:46 001047 KAKC1601-I SAMPLE_JCL.STEP3 Step allocated.
14:14:46 001047 KAKC1602-I DD2_READ, OLD, C:\Users\Public\Documents\Hitachi\BJEX\tmp\BJX95FC.tmp
14:14:46 001047 KAKC1603-I DD2_READ, DELETE, C:\Users\Public\Documents\Hitachi\BJEX\tmp\BJX95FC.tmp,
size=1039
14:14:46 001047 KAKC1604-I C:\Users\Public\Documents\Hitachi\BJEX\tmp\BJX95FC.tmp is deleted.
14:14:46 001047 KAKC1601-I SAMPLE_JCL.STEP4 Step allocated.
14:14:46 001047 KAKC1602-I DD4_READ, . C:\Users\Public\Documents\Hitachi\BJEX\tmp\BJX964B.tmp
14:14:46 001047 KAKC1603-I DD4_READ, . C:\Users\Public\Documents\Hitachi\BJEX\tmp\BJX964B.tmp
14:14:46 001047 KAKC1604-I C:\Users\Public\Documents\Hitachi\BJEX\tmp\BJX964B.tmp is deleted.
14:14:46 001047 KAKC1601-I SAMPLE_JCL.STEP5, SAMPLE_PROC.STEP1 Step allocated.
14:14:46 001047 KAKC1602-I PROC_DD1, . C:\Users\Public\Documents\Hitachi\BJEX\tmp\BJX966C.tmp
14:14:46 001047 KAKC1603-I PROC_DD1, . C:\Users\Public\Documents\Hitachi\BJEX\tmp\BJX966C.tmp
14:14:46 001047 KAKC1604-I C:\Users\Public\Documents\Hitachi\BJEX\tmp\BJX966C.tmp 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 "C:\Users\Public\Documents\Hitachi\BJEX\conf\bjex.conf" started.
KAKC0409-I Parsing of config file "C:\Users\Public\Documents\Hitachi\BJEX\conf\bjex.conf" ended. code=0
KAKC0724-I Jobid was assigned. Jobid=001047
KAKC0097-I SAMPLE_JCL_STEP1 Step ended. rc=0 E-Time=0.036s C-Time=0.015s
KAKC0097-I SAMPLE_JCL_STEP2 Step ended. rc=0 E-Time=0.049s C-Time=0.015s
KAKC0097-I SAMPLE_JCL_STEP3 Step ended. rc=0 E-Time=0.026s C-Time=0.000s
KAKC0097-I SAMPLE_JCL_STEP4 Step ended. rc=0 E-Time=0.027s C-Time=0.000s
KAKC0097-I SAMPLE_JCL_STEP5, SAMPLE_PROC_STEP1 Step ended. rc=0 E-Time=0.028s C-Time=0.015s
KAKC0098-I SAMPLE_JCL Job ended. rc=0 E-Time=0.166s C-Time=0.045s

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

[Information]
Jobid      : 001047
Spool Directory : C:\Users\Public\Documents\Hitachi\BJEX\spool\001047\
Date       : 2023/10/01
Configure File : C:\Users\Public\Documents\Hitachi\BJEX\conf\bjex.conf
HostName    : nvm014349
[JP1 Parameter]
JP1JobName  : SAMPLE_JCL.bjx
JP1JobID    : 0000000206
JP1_USERNAME : bjex
JP1UNCName  : nvm014349
JP1NBQueueName: ¥¥nvm014349¥¥@SYSTEM
JP1Priority  : 32
AJSEXECID   : @A105

-----
***** BJEX MESSAGE *****
14:14:46 001047 KAKC0091-I SAMPLE_JCL Job started.
14:14:46 001047 KAKC0092-I SAMPLE_JCL_STEP1 Step started.
14:14:46 001047 KAKC0097-I SAMPLE_JCL_STEP1 Step ended. rc=0 E-Time=0.036s C-Time=0.015s
14:14:46 001047 KAKC0092-I SAMPLE_JCL_STEP2 Step started.
14:14:46 001047 KAKC0097-I SAMPLE_JCL_STEP2 Step ended. rc=0 E-Time=0.049s C-Time=0.015s
14:14:46 001047 KAKC0092-I SAMPLE_JCL_STEP3 Step started.
14:14:46 001047 KAKC0097-I SAMPLE_JCL_STEP3 Step ended. rc=0 E-Time=0.026s C-Time=0.000s
14:14:46 001047 KAKC0092-I SAMPLE_JCL_STEP4 Step started.
14:14:46 001047 KAKC0097-I SAMPLE_JCL_STEP4 Step ended. rc=0 E-Time=0.027s C-Time=0.000s
14:14:46 001047 KAKC0092-I SAMPLE_JCL_STEP5, SAMPLE_PROC_STEP1 Step started.
14:14:46 001047 KAKC0097-I SAMPLE_JCL_STEP5, SAMPLE_PROC_STEP1 Step ended. rc=0 E-Time=0.028s C-Time=0.015s
14:14:46 001047 KAKC0098-I SAMPLE_JCL Job ended. rc=0 E-Time=0.166s C-Time=0.045s

```

```

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

000002 <HitachiBatchJobExec version="1.3" os="windows" >
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" />
000010 <DD NAME="DD2_WRITE" TYPE="TEMP" DSN="DD2FILE" DISP="NEW,PASS" />
000011 </STEP>
000012 <STEP NAME="STEP3" >
000013 <EXEC PGM="UAP3" PARM="%%SAMPLE_JCL_STEP3%%" LANG="COBOL" />
000014 <DD NAME="DD2_READ" TYPE="TEMP" DSN="DD2FILE" DISP="OLD,DELETE" />
000015 </STEP>
000016 <STEP NAME="STEP4" >
000017 <EXEC PGM="UAP4" PARM="%%SAMPLE_JCL_STEP4%%" LANG="COBOL" />
000018 <DD NAME="DD4_READ" TYPE="DATA" SAMF="YES" >
<![CDATA[
!
INPUT DATA 1
INPUT DATA 2
!
]]>

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

>>>>> C:\Users\Public\Documents\Hitachi\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="windows" >
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>
<<<<< C:\Users\Public\Documents\Hitachi\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 *****
14:14:46 001047 KAKC0900-I Parsing of xml file "C:\bjex_xml\SAMPLE_JCL.bjx" started.
14:14:46 001047 KAKC1100-I Parsing of proc file "C:\Users\Public\Documents\Hitachi\BJEX\proc\SAMPLE_PROC.xml"
started.
14:14:46 001047 KAKC1109-I Parsing of proc file "C:\Users\Public\Documents\Hitachi\BJEX\proc\SAMPLE_PROC.xml"
ended. code=0
14:14:46 001047 KAKC0909-I Parsing of xml file "C:\bjex_xml\SAMPLE_JCL.bjx" ended. code=0
14:14:46 001047 KAKC0707-I MSGCLASS is "A".
14:14:46 001047 KAKC1600-I SAMPLE_JCL Job allocated.
14:14:46 001047 KAKC1601-I SAMPLE_JCL.STEP1 Step allocated.
14:14:46 001047 KAKC1602-I DD1, .
C:\Users\Public\Documents\Hitachi\BJEX\spool\001047\001_STEP1_001_DD1.sysout
14:14:46 001047 KAKC1603-I DD1, .
C:\Users\Public\Documents\Hitachi\BJEX\spool\001047\001_STEP1_001_DD1.sysout, size=532
14:14:46 001047 KAKC1601-I SAMPLE_JCL.STEP2 Step allocated.
14:14:46 001047 KAKC1602-I DD2_WRITE, NEW, C:\Users\Public\Documents\Hitachi\BJEX\temp\BJX95FC.tmp
14:14:46 001047 KAKC1603-I DD2_WRITE, PASS, C:\Users\Public\Documents\Hitachi\BJEX\temp\BJX95FC.tmp,
size=1039
14:14:46 001047 KAKC1601-I SAMPLE_JCL.STEP3 Step allocated.
14:14:46 001047 KAKC1602-I DD2_READ, OLD, C:\Users\Public\Documents\Hitachi\BJEX\temp\BJX95FC.tmp
14:14:46 001047 KAKC1603-I DD2_READ, DELETE, C:\Users\Public\Documents\Hitachi\BJEX\temp\BJX95FC.tmp,
size=1039
14:14:46 001047 KAKC1604-I C:\Users\Public\Documents\Hitachi\BJEX\temp\BJX95FC.tmp is deleted.
14:14:46 001047 KAKC1601-I SAMPLE_JCL.STEP4 Step allocated.
14:14:46 001047 KAKC1602-I DD4_READ, . C:\Users\Public\Documents\Hitachi\BJEX\temp\BJX964B.tmp
14:14:46 001047 KAKC1603-I DD4_READ, . C:\Users\Public\Documents\Hitachi\BJEX\temp\BJX964B.tmp
14:14:46 001047 KAKC1604-I C:\Users\Public\Documents\Hitachi\BJEX\temp\BJX964B.tmp is deleted.
14:14:46 001047 KAKC1601-I SAMPLE_JCL.STEP5.SAMPLE_PROC.STEP1 Step allocated.
14:14:46 001047 KAKC1602-I PROC_DD1, . C:\Users\Public\Documents\Hitachi\BJEX\temp\BJX966C.tmp
14:14:46 001047 KAKC1603-I PROC_DD1, . C:\Users\Public\Documents\Hitachi\BJEX\temp\BJX966C.tmp
14:14:46 001047 KAKC1604-I C:\Users\Public\Documents\Hitachi\BJEX\temp\BJX966C.tmp 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、ジョブ単位にディレクトリを作成して管理する。



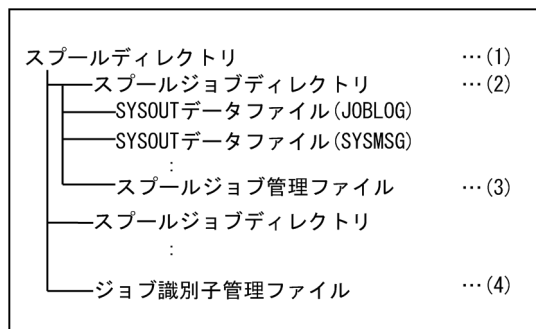
## (1) スプールの制御

### (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。

ディレクトリ名は次の 2 種類がある。

- ジョブ識別子※1
- ジョブ識別子-ジョブ名※2

注※1 ジョブ識別子だけのディレクトリ名称の場合は、実行中のジョブであり、スプールジョブディレクトリは隠しファイルとなっているため、参照や操作をしないこと。

注※2 ジョブ識別子とジョブ名を "-" で接続したディレクトリ名称の場合は、実行済みのジョブである。ジョブ実行中に隠しファイルとなっていたスプールジョブディレクトリは、ジョブ実行後に通常のファイル属性となる。

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、ジョブ終了時に印刷属性に従い処理される。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.ini



#### (4) ジョブ識別子管理ファイル

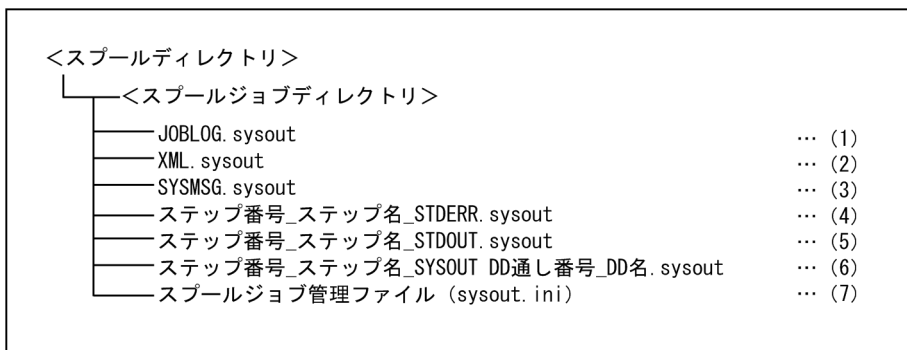
このスプールのジョブ識別子を管理するファイルである。このファイルは隠しファイルであり、バッチジョブ実行システムが使用するため、参照や操作をしないこと。

ファイル名：bjex.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.sysout

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.sysout

実行時のジョブ定義 XML ファイル。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.sysout

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メッセージとして格納する。

#### (4) ステップ番号\_ステップ名\_STDERR.sysout

このステップで実行したプログラムの標準エラー出力。

#### (5) ステップ番号\_ステップ名\_STDOUT.sysout

このステップで実行したプログラムの標準出力。

#### (6) ステップ番号\_ステップ名\_SYSOUT DD 通し番号\_DD 名.sysout

DD 要素の TYPE 属性に SYSOUT を指定したファイル。

#### (7) スプールジョブ管理ファイル (sysout.ini)

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。

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 桁の 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 桁の 10 進数とする。
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。

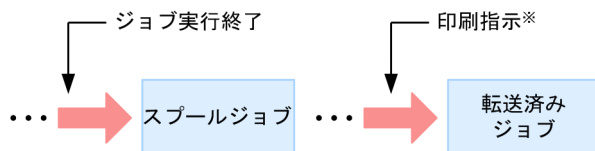
### (c) スプールジョブのデータの流れ

ジョブ終了時にスプールディレクトリに生成されたスプールジョブは、スプールジョブの属性に従って、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に転送される。転送後はスプールジョブは削除される。ただし、出力後保存属性の SYSOUT、または転送しない属性の SYSOUT が存在する場合は、スプールジョブは削除されないでスプールディレクトリに残る。

ジョブ実行中のスプールジョブディレクトリ名は"ジョブ識別子"であるが、ジョブ終了後は"ジョブ識別子-ジョブ名"に変更される。

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指定することを示す。

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	印刷後のスプールジョブ削除、または印刷後にいったん保存したが、長期保存が不要な場合の bjexrmjob コマンドによるスプールジョブ削除指示または手作業 (del コマンド) での削除	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。

データ種別 名称	生成契機	削除契機	格納場所	意味
転送済みジョブ	印刷指示	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。

#### (d) スプールに関する注意事項

スプールディレクトリおよびスプールジョブディレクトリ配下にある管理ファイルはバッチジョブ実行システムが使用するファイルであるため、手作業による変更はしないこと。

## 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。

カタログドプロシジャの拡張子は".xml"とすること。

#### (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, SETSYMBOL 要素, REPLACE 要素, および REPLACESTEP 要素
- NAME 属性"JOBLIB"を持つ 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" の拡張子を付け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="1.2" os="windows">
  <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="1.2" os="windows">
  <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="1.2" os="windows"> ... (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 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>
```

```
[</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="1.2" os="windows">  
  <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="1.2" os="windows">  
  <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="1.2" os="windows">
  <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="1.2" os="windows">
  <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は変更できない。

## (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="1.2" os="windows">
  <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="1.2" os="windows">
  <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くるようにすると便利である。
- 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要素とDD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属性およびDD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。

## 3.3 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ります。

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、システムの設定に従い、ログオン時に 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、[\[3.3.2 ユーザプログラムの使用方法\]](#)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"/>
```

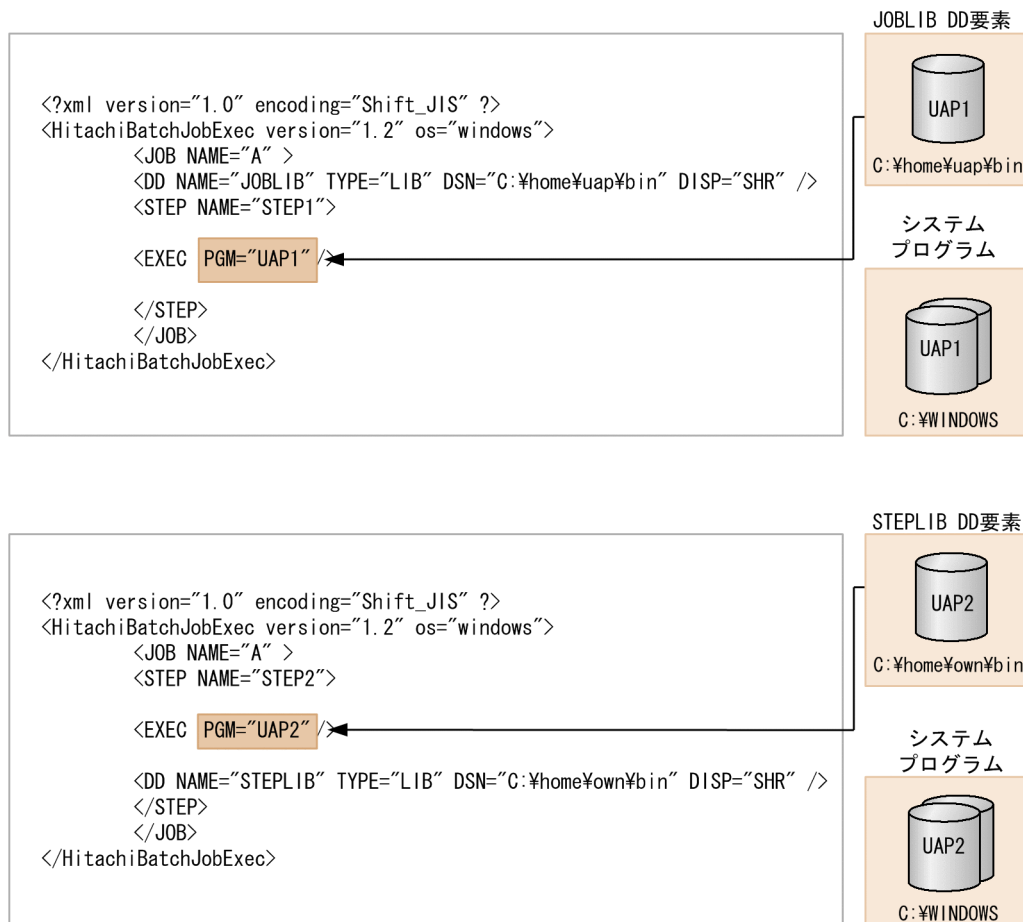
## (1) JOBLIB, STEPLIB, および PATH 環境変数の使用方法

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。
- JOBLIB DD 要素, STEPLIB DD 要素によって PATH 環境変数に設定される値の長さが 16,383 を超える場合、KAKC1632-E を出力し、ジョブを終了する。PATH 環境変数の値の長さが 16,383 を超えないよう、システムで設定している PATH 環境変数の値、または JOBLIB/STEPLIB DD 要素によって指定したパスの長さを修正すること。



## 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。	既存ファイルがあれば割り当てエラーとなる。	0 バイトファイルを作成する。
RNW パラメータ※1		既存ファイルがあれば削除する。	
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、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、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登録されていない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する。	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、

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する。	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 <ul style="list-style-type: none"> <li>• DD 要素でファイル連結として指定している。</li> <li>• DD_EXISTCHECK_COBOL パラメータに CONCAT を指定している。</li> </ul> 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 <ul style="list-style-type: none"> <li>• DD 要素でファイル連結として指定している。</li> <li>• DD_EXISTCHECK パラメータに CONCAT を指定している。</li> </ul>	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、「[3.10 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の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<sup>※</sup>、または EXEC 要素の LANG 属性の値が COBOL の場合に追加書きする。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 である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 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。また、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、「3.14.7(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る。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 = "1.2" os="windows">
  <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、使用しないこと。
- 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 = "1.2" os="windows">
  <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 = "1.2" os="windows">
  <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たり、割り当てたファイルの割り当てを解除したりする場合、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で、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 3 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。*	

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 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 ジョブ終了時に処理する。 <u>処理内容は、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> (2)DISP 属性の第 2 パラメータが DELETE の場合 恒久ファイルの処理と同様（世代管理情報を更新しない）。	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。

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）。 GDG_REGISTER_JOBRC_THRESHOLD 適用 (1)DISP 属性の第 2 パラメータが KEEP の場合 <u>(a)DISP 属性の第 1 パラメータが NEW の場合</u> ジョブ終了時に削除する。 <u>(b)DISP 属性の第 1 パラメータが NEW 以外の場合</u> <ul style="list-style-type: none"> <li>• <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> ジョブ終了時に削除し、世代管理情報は更新しない。</li> <li>• <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。</li> </ul> (2)DISP 属性の第 2 パラメータが DELETE の場合 <u>(a)DISP 属性の第 1 パラメータが NEW の場合</u>	



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	<p>ジョブステップ終了時に削除する。</p> <p>(b)DISP 属性の第 1 パラメータが NEW 以外の場合</p> <ul style="list-style-type: none"> <li>ジョブ実行中に作成された世代ファイルである場合 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。</li> <li>上記以外の場合 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。</li> </ul>	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 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

NAME 属性が同一である DD 要素を連続して記述した場合、ファイルの連結の指定と解釈する。先頭の DD 要素に指定された名称の環境変数に、DD 要素の記述の順序で、割り当てた各ファイルのパス名を区切り文字セミコロン";"で連結した文字列を設定する。最大 255 の DD 要素を連続して記述できる。

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い場合は、先頭に"DDN\_"または"CBL\_"を付けた 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。

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、セミコロン";"をパス名の区切りと判断できるプログラム（バッチジョブ実行システムが連結したファイルの入出力に対応した COBOL2002, または SORT EE), JOBLIB DD 要素, または STEPLIB DD 要素で利用できる。

JOBLIB DD 要素, 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、ファイル連結できるかどうかが決めている。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	○
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	×

(凡例)

- ：ファイル連結できる。
- ×：ファイル連結できない。

#### 注意事項

- ダミーファイルをファイルの連結処理の途中で指定した場合、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 ファイルにこのような指定があり、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。
- ファイルの連結処理によって区切り文字で連結したパス文字列の長さが 16,383 を超える場合、KAKC1632-E を出力し、ジョブを終了する。区切り文字で連結したパス文字列の長さが 16,383 を超えないよう、ファイル連結処理の指定内容を修正すること。

## 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。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。

<例> DD 名が SYSUT2 のファイルを、一時ファイルとして使用する場合

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

### (2) 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。次に示すファイル名が作成される。

```
BJX※ユニークな名称.tmp
```

注※ "BJX"は固定である。

- 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。
- 一時ファイルは設定ファイルの TEMP\_FILE\_DIR パラメータで指定したディレクトリに格納する。
- 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 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るが、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 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 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。

### 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。

<例> DD 名が SYSUT2 のファイルを、一時 ISAM ファイルとして使用する場合

```
<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".KDF"）
- 主キーファイル（拡張子".K01"）
- データファイル（拡張子".DRF"）

## (3) 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。次に示すファイル名が生成される。

```
BJX※ユニークな名称.TMP
```

注※ "BJX"は固定である。

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けられる。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.KDF
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.DRF

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ず、拡張子を除いたファイル名が1つだけ設定される。

## (4) 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	認めない。
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	認めない。

## (5) 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。
- 一時 ISAM ファイルは、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。
- 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ファイルを割り当てるが、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。
- 一時 ISAM ファイルを使用する場合、実行ユーザに Windows のグローバルオブジェクトの作成権限を与えること。
- 一時 ISAM ファイルの名称はすべて大文字となる。
- バッチジョブ実行システムは、一時 ISAM ファイルを構成するファイル名から拡張子 (".KDF",".K01",".DRF") を除いたファイル (BJX ユニークな名称.TMP) を作成する。このファイルは、バッチジョブ実行システムが一時 ISAM ファイルの名称を管理するために作成するものであり、一時 ISAM ファイルと同じ契機で作成・削除される。一時 ISAM ファイルを使用中にこのファイルを削除した場合、バッチジョブ実行システムは KAKC0165-I メッセージを出力して処理を継続する。
- 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。ダミーファイルを指定した場合、一時ファイルが割り当てられる。

#### (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。

- 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
ドライブレター:¥[パス名¥]ファイル名, または先頭が¥¥	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。
¥ または ..¥	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。
\$(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。
上記以外	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。

## (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 C:¥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=C:¥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 C:¥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=C:¥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 値に文字列を付けた名称とする。付ける文字列は、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_"を付けた DD 要素名 ("CBL_" + DD 要素名) 例 <DD NAME="ABC" /> 環境変数名: CBL_ABC	先頭に"DDN_"を付けた 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="C:¥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、C:¥home¥file1 のファイルを使用できる。

```
DDN_DD1=C:¥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	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
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"> <li>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";"(セミコロン)で区切った文字列である。</li> <li>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ける。</li> </ul>

#### (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\_"を付け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="C:¥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、C:¥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=C:¥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。
  - 「[3.17 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

- システム環境変数など、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スクリプト（Windows のバッチファイル）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。

## (5) DSN 属性への環境変数指定機能

DSN 属性への環境変数指定機能は、設定ファイルの DSN\_USE\_ENV パラメータのオペランドに"YES"を指定して使用可能となる。DSN 属性への環境変数指定機能を使用した場合、DD 要素の DSN 属性値に指定した環境変数を、その環境変数の値に置換する。

### (a) 環境変数の指定方法

DSN 属性値に"\${"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、DSN 属性値の任意の位置に指定できる。

```
DSN="…${環境変数名}…"
```

(凡例)

…：0 文字以上の任意文字列を示す。

### (b) 指定可能な環境変数名

DSN 属性値に指定可能な環境変数名の一覧を次の表に示す。DSN 属性値には、次の表に示す環境変数名以外を指定しないこと。

表 3-15 DSN 属性に指定可能な環境変数

環境変数名	内容
BJEX_DIR_で始まる名称	バッチジョブ実行システムのパス名置換識別子。設定パラメータ内に指定されているパス名置換用識別子を必ず使用すること。設定パラメータ内に指定しないで、ジョブ実行時に指定しないこと。
JP1 が設定する環境変数	内容については、JP1 のマニュアルを参照のこと。

### (c) 注意事項

- DSN 属性値に指定した環境変数が bjexec コマンド起動時に設定されていないとき、ジョブ定義 XML ファイルの解析時にエラーとなる。
- 環境変数は、bjexec コマンド起動時に設定されている値に置換される。ジョブ実行開始後にジョブステップで環境変数を設定したり、SETENV 要素を用いて環境変数を設定したりしても、そこで設定した値には置換されない。

- "\${"と"}"の間には"\${"を含めないこと。"\${"と"}"の間に含まれる"\${"は、環境変数指定の開始とは解釈されず、通常の文字列として処理する。
- "\${"が存在するが、それに対応する"}"が存在しない場合、その"\${"は環境変数指定の開始とは解釈しないで、通常の文字列として処理する。
- DSN 属性値の長さは、環境変数指定を含めて規定の長さの範囲内とすること。範囲外の指定をした場合、ジョブ定義 XML ファイルの解析時にエラーとなる。DSN 属性値の長さの範囲については、「9. ジョブ定義 XML ファイル」を参照のこと。
- 環境変数の指定を置換した結果の DSN 属性値が規定の長さの範囲を超えた場合、ジョブ定義 XML ファイルの解析時にエラーとなる。DSN 属性値の長さの範囲については、「9. ジョブ定義 XML ファイル」を参照のこと。

### 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。

逆方向参照は参照元 DD 要素、および先行 DD 要素の TYPE 属性がともに FILE であることが必要である。

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。プロシジャ内で指定された場合、そのプロシジャ内で 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="1.2" os="windows">
  <JOB NAME="SAMPLE" >
    <STEP NAME="STEP1">
      <EXEC PGM="UAP1" />
      <DD NAME="DD1" TYPE="FILE" DSN="C:\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。

## 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、プログラムまたはコマンドを実行する方法と、スクリプト（Windows のバッチファイル）内でコマンドを実行する方法がある。

また、EXEC 要素の LANG 属性値に"Java"を指定した場合は、EXEC 要素に記述した Java アプリケーションを実行できる。Java アプリケーションを実行する方法については、「[3.15 Cosminexus との連携によるバッチアプリケーション実行](#)」を参照のこと。

### 3.5.1 プログラムまたはコマンドを実行

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、コマンドをスクリプトからではなく直接実行する。

- 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 スクリプトを実行

EXEC 要素内の PGM 属性値に"\*"の指定と EXEC 要素の内容にスクリプトを記述した場合、そのスクリプトを実行する。内容の形式は「[3.8.2 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"!"にコマンドを記述する。

- 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
```



- 指定例

```
<EXEC PGM="*">  
<![CDATA[  
!  
type account.info  
!  
]]>  
</EXEC>
```

(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

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 3-16 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。

注※ PGM\_RCABEND パラメータを指定しなかった場合、1, 2, 3 を仮定する。

### 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、スクリプトとして実行すること。

## 3.6 ジョブステップの条件付実行

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実行という。条件付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 0~4095 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 3-17 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0~4095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ある。演算符号については、「表3-17 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)"
```

- ジョブステップの終了コードが0~4095の範囲でない場合、ジョブステップの終了コードの下位12ビットだけを判定に使用する。したがって、EXEC要素に記述するプログラムの終了コードは0~4095の値を使用すること。

## 3.6.2 条件付実行

### (1) JOB要素でCOND属性を指定した場合

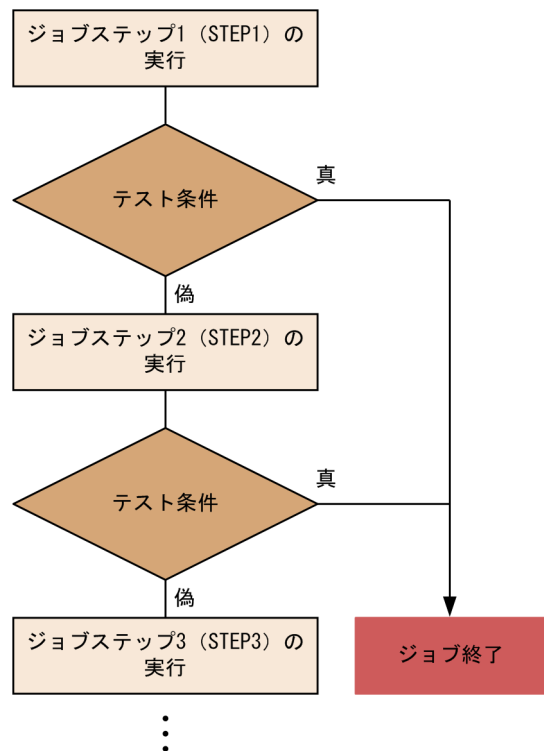
ジョブ内の各ジョブステップの終了時点で、各ジョブステップが生成する終了コードとJOB要素のCOND属性で指定した演算符号を基に論理演算をする。

- 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



## (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 を指定した場合は、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。

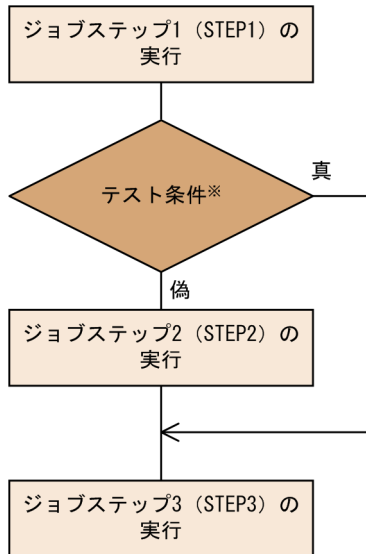
### (b) 先行するジョブステップ名を指定した場合

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



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

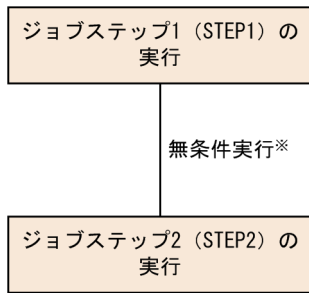
### (3) EXEC要素のCOND属性でEVENまたはONLYを指定した場合

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。

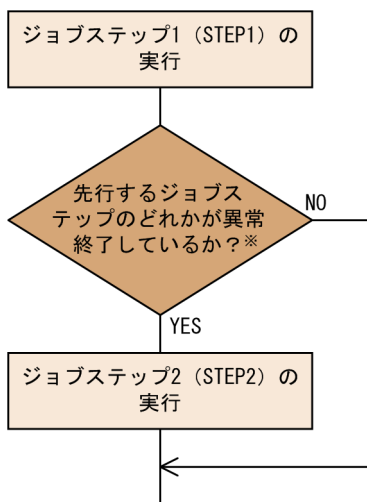
図 3-9 EXEC 要素の COND 属性で EVEN を指定した場合の流れ



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

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

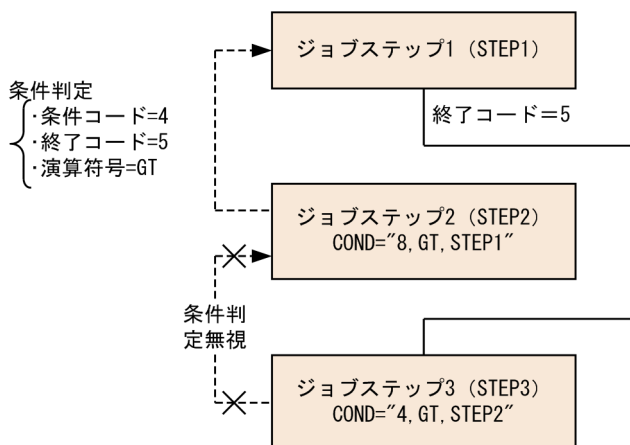
図 3-10 EXEC 要素の COND 属性で ONLY を指定した場合の流れ



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

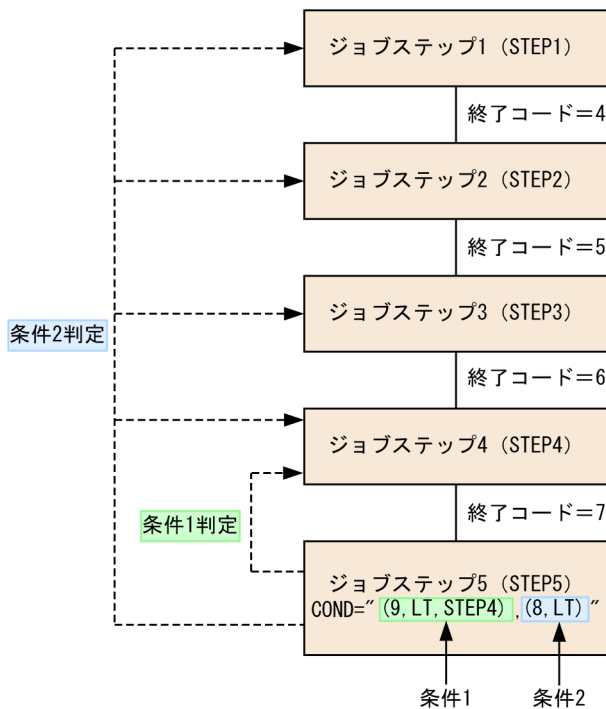
ジョブステップの条件付きの実行を以下の図に示す。

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 (9,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 (8,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。

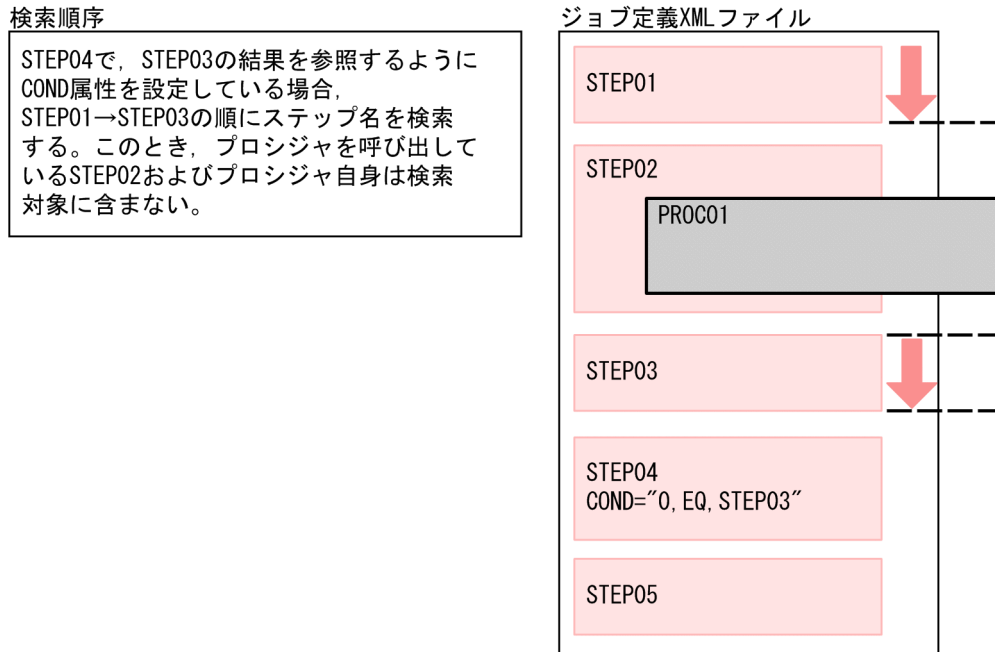
#### (1) COND属性がプロシジャ外に指定されている場合

##### (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)



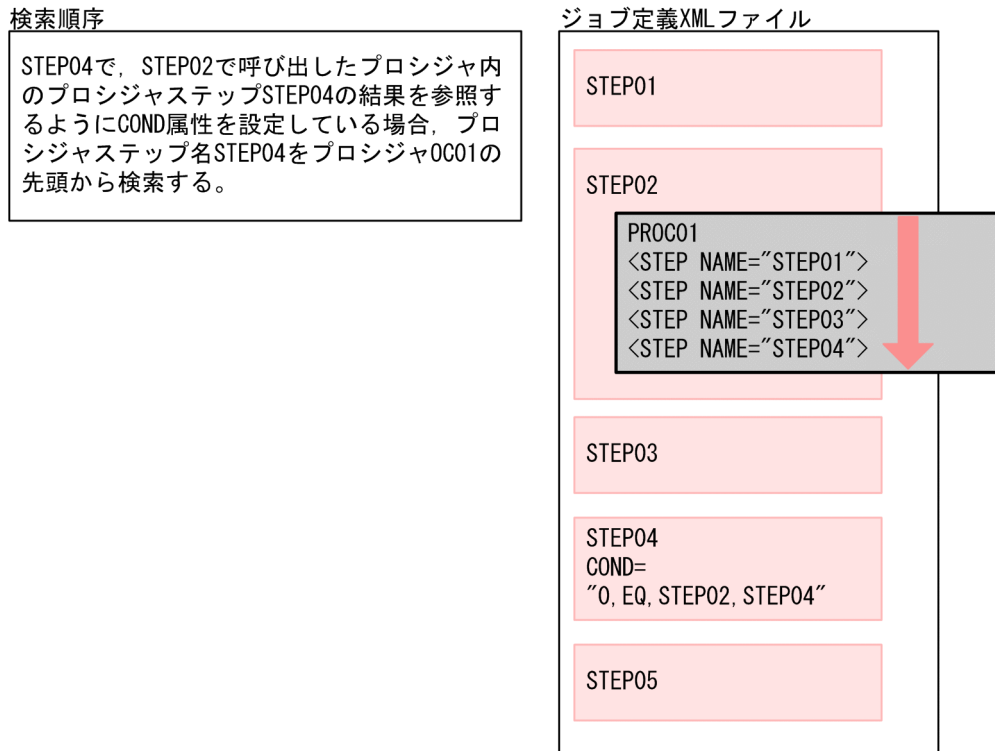
(凡例)

-  : 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)



(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、その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、その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、このとき検索途中で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。

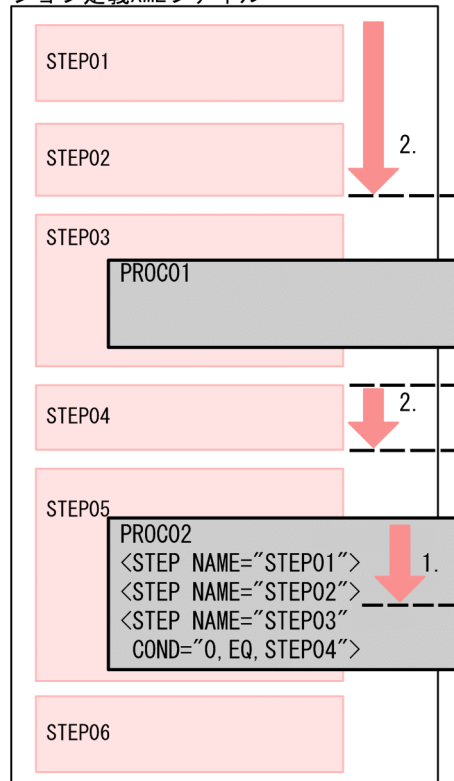
図 3-15 ステップ名だけで検索する例 (COND 属性がプロシジャ内に指定されている場合)

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。

(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

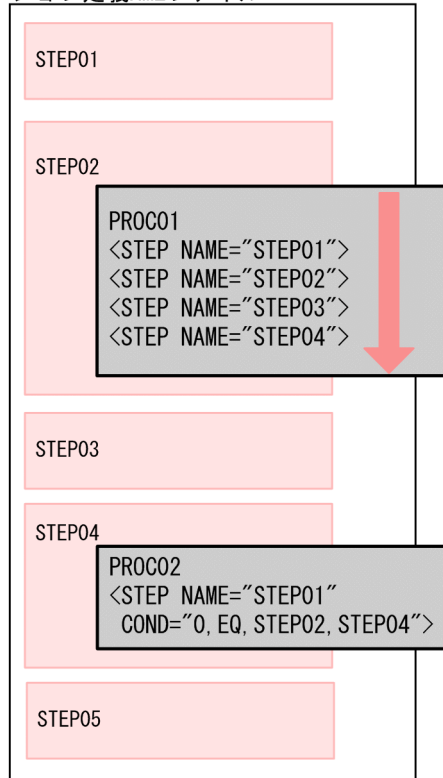
バッチジョブ実行システムは、COND 属性で指定したステップから呼び出されているプロシジャに限定して、その先頭からプロシジャステップ名を検索する。検索例を次の図に示す。

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 世代ファイル

### 3.7.1 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。

世代番号の指定は、相対世代番号で指定する。

#### (1) 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) で付け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
```

#### (2) 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囲んで付け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。

### (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、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) となり、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。

また、世代ファイルは隠しファイルであり、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。

- 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。

## 3.7.2 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。

```
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 C:¥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 C:¥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。

### 3.7.3 世代ファイルの作成と参照

#### (1) 世代ファイルの作成

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="1.2" os="windows">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM" />
      <DD NAME="DD1" DSN="C:¥home¥gdg¥ACCOUNT.X(+1)" TYPE="GDG" DISP="NEW,KEEP" /> • ACCOUNT.X.G0001の作成
      <DD NAME="DD2" DSN="C:¥home¥gdg¥ACCOUNT.X(+2)" TYPE="GDG" DISP="NEW,KEEP" /> • ACCOUNT.X.G0002の作成
      <DD NAME="DD3" DSN="C:¥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

## (2) 世代ファイルの参照

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="1.2" os="windows">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM" />
      <DD NAME="DD1" DSN="C:\home\gdg\ACCOUNT.X(-2)" TYPE="GDG" DISP="SHR" /> • ACCOUNT.X.G0001の参照
      <DD NAME="DD2" DSN="C:\home\gdg\ACCOUNT.X(-1)" TYPE="GDG" DISP="SHR" /> • ACCOUNT.X.G0002の参照
      <DD NAME="DD3" DSN="C:\home\gdg\ACCOUNT.X(0)" TYPE="GDG" DISP="SHR" /> • ACCOUNT.X.G0003の参照
    </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。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.2" os="windows">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM" />
      <DD NAME="DD1" DSN="C:\home\gdg\ACCOUNT.X" TYPE="GDG" DISP="SHR" /> • ACCOUNT.Xに登録されている
                                                                    世代ファイルの参照
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

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="1.2" os="windows">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM" />
      <DD NAME="DD1" DSN="C:\home\gdg\ACCOUNT.X(0)" TYPE="GDG" DISP="SHR" />
      <DD NAME="DD1" DSN="C:\home\gdg\ACCOUNT.X(-1)" TYPE="GDG" DISP="SHR" />
      <DD NAME="DD1" DSN="C:\home\gdg\ACCOUNT.X(-2)" TYPE="GDG" DISP="SHR" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

### 注意事項

世代データ群名指定によって区切り文字で連結したパス文字列の長さが 16,383 を超える場合、KAKC1632-E を出力し、ジョブを終了する。区切り文字で連結したパス文字列の長さが 16,383 を超えないよう、パスの長さに注意して世代ファイルを作成すること。



## 3.7.4 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

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="1.2" os="windows">
  <JOB NAME="SAMPLE">
    <STEP NAME="STEP1">
      <EXEC PGM="PROGRAM" />
      <DD NAME="DD1" DSN="C:\home\gdg\ACCOUNT.X(-2)" TYPE="GDG" DISP="OLD,DELETE" />・ACCOUNT.X.G0001の削除
      <DD NAME="DD2" DSN="C:\home\gdg\ACCOUNT.X(-1)" TYPE="GDG" DISP="OLD,DELETE" />・ACCOUNT.X.G0002の削除
      <DD NAME="DD3" DSN="C:\home\gdg\ACCOUNT.X(0)" TYPE="GDG" DISP="OLD,DELETE" />・ACCOUNT.X.G0003の削除
    </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。

世代ディレクトリ名 C:\home\gdg

```
bjexgdg -q C:\home\gdg\ACCOUNT.X
```

## 3.7.5 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（ユーザ指定に従って処理する）。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、世代管理情報を更新するかどうかを判定し、登録の可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

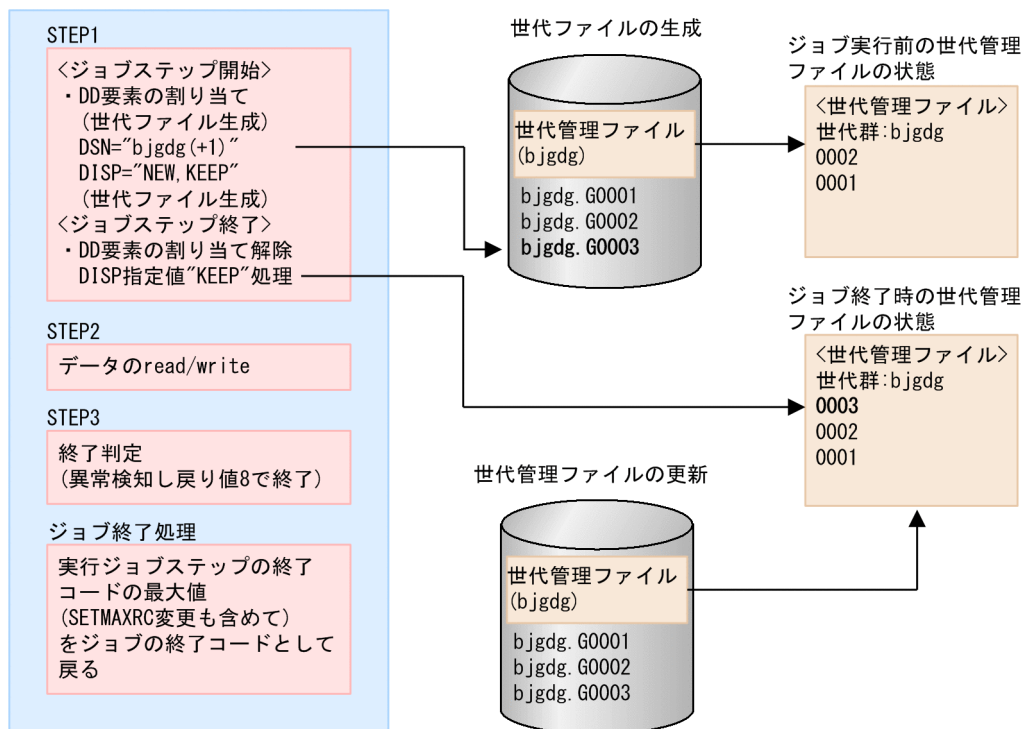
世代ファイルを生成するジョブステップが正常に実行された場合、生成した世代ファイルは、DD 要素の DISP 属性の指定値に従って登録される。そのため、後続のジョブステップでの異常有無に関係なく、更新された世代管理ファイルの状態はジョブ実行前の状態には戻らない。

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 3-17 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

ジョブA



## (b) ジョブ終了コードが異常終了と認識された場合

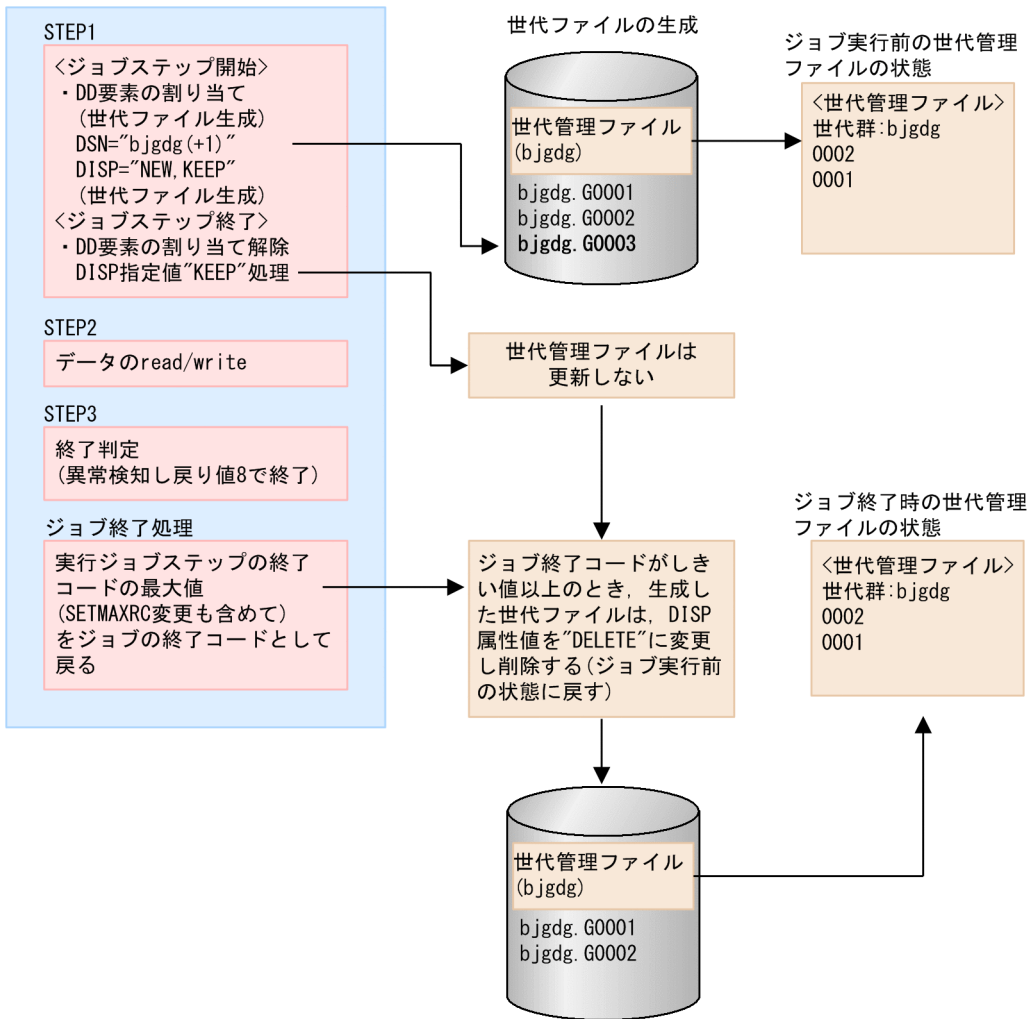
世代ファイルを生成するジョブステップでは、ジョブステップ終了時に世代管理ファイルを更新しない。ジョブ終了時に更新の可否を判定し、更新できない（ジョブは異常終了と認識された場合、世代管理ファイルの状態はジョブ開始時点の状態を保持する）と判定した場合、世代管理ファイルを更新しないで、生成した世代ファイルを削除（DELETE）する。

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 3-18 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

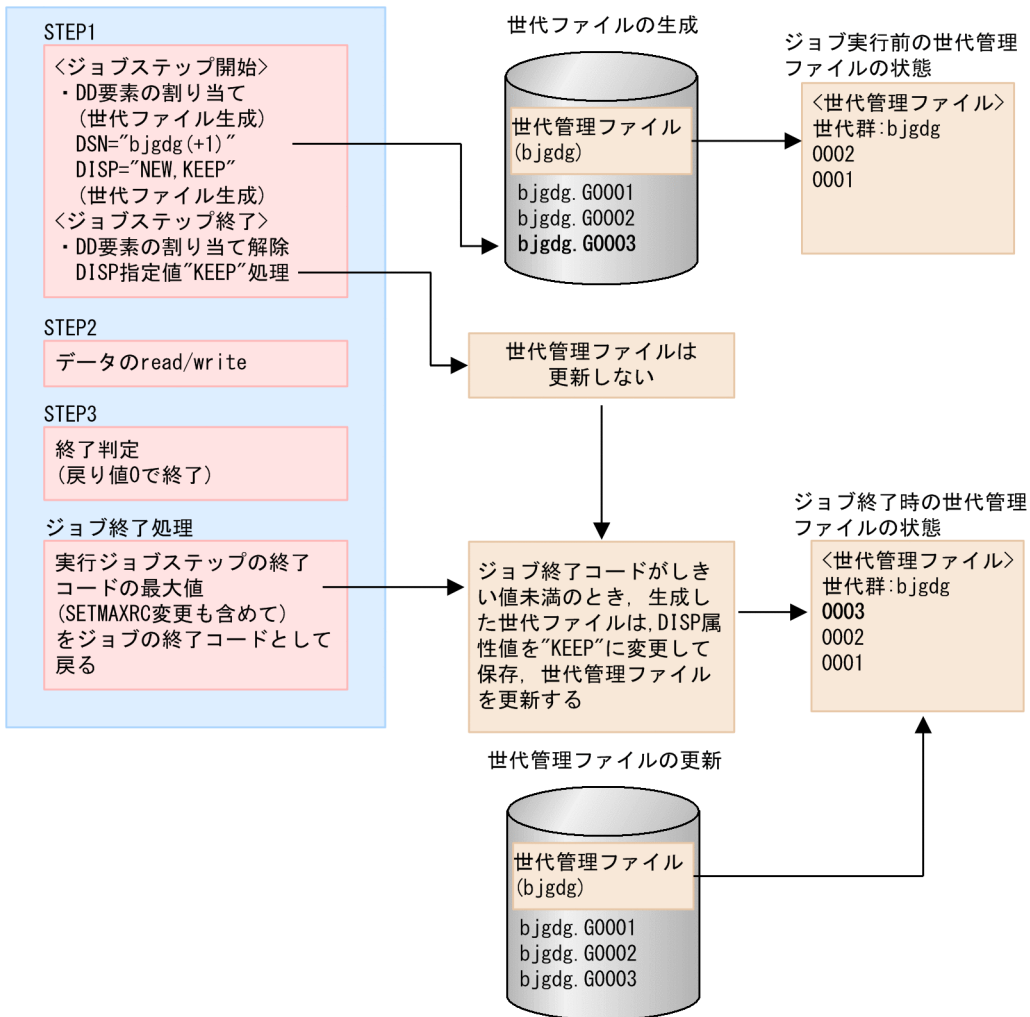
世代ファイルを生成するジョブステップでは、ジョブステップ終了時に世代管理ファイルは更新されない。ジョブ終了時に更新の可否を判定し、更新できる（ジョブは正常終了と認識された場合、世代管理ファイルの状態は最新世代ファイルの情報を更新した状態となる）と判定した場合、世代管理ファイルを更新して、生成した世代ファイルは保持（KEEP）される。

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 3-19 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" のように削除指定しているときは、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、「3.7.3(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。

## 3.7.6 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、1~255 個である。登録可能数は、bjexgdg コマンドの -I 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

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。

#### (b) データ保存機能 (NOSCRATCH) が設定された世代データ群の場合

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 桁の通し番号 (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 del コマンドで削除すること。

### (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。
- 同一の世代ファイル、世代データ群を別ジョブから同時に作成する場合は、同時にジョブが実行しないようにジョブをスケジューリングする。

### (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。

<例>

```
<DD NAME="DD1" DSN="ACCOUNT.X(0)" TYPE="GDG" DISP="SHR" />
```



- 同一の世代ファイル、世代データ群を作成するジョブと同時に実行しないようにジョブをスケジューリングする。
- 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

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

通常の大小関係では、"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、大小関係は、"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 3-18 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。
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 3-19 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 3-20 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一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、del 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)
- "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)
- "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。

## 3.8 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。

### 3.8.1 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。

### 3.8.2 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。

### 3.8.3 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...AAAAAAAAAAAAAAAAAAAAAA¥r¥n ←80バイト+改行
BBBBBBBBBB...BBBBBBBBBB¥r¥n ←70バイト+改行
!
]]>
</DD>
```

(凡例)

¥r¥n：改行を示す。

レコード

```
レコード 1 AAAAAAAAAA...AAAAAAAAAAAAAAAAAAAAAA¥r¥n ←80バイト+改行
レコード 2 BBBBBBBBBB...BBBBBBBBBB¥r¥n ←70バイト+改行
```

(凡例)

¥r¥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¥r¥n  
BBBBBBBBBB...BBBBBBBBBB¥r¥n  
[EOF]
```

(凡例)

¥r¥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¥r¥n ←80バイト+改行  
BBBBBBBBBB...BBBBBBBBBB¥r¥n ←70バイト+改行  
!  
]]>  
</DD>
```

(凡例)

¥r¥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。

## 3.9 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)PDE ファイル形式の指定（FTYPE 属性の指定）
- (r)レベル 1 外字名の指定（GAIJ11 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。

## 3.9.1 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。

## 3.9.2 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。

## 3.9.3 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。

## 3.9.4 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、ジョブ終了時にプリントサーバに転送される。印刷が不要なものは、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。出力保留した SYSOUT データファイルは、あとから印刷指示する機能がないため、印刷しない SYSOUT データに適用する。

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。

## 3.9.5 出力後保存

バッチジョブ実行システム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、スプールディレクトリに"ジョブ識別子-ジョブ名"の形式名称のディレクトリに格納されている。これらのファイルは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、印刷サーバに転送後にスプールから削除する。転送処理で失敗した場合、スプールジョブはスプールに残るが、再度転送する手段はないため、必要ならジョブを再実行する。

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。

## 3.9.6 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 3-21 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。

### 3.9.7 PDE ファイル形式

PDE で扱うファイルの形式として、次に示す「PDE ファイル形式 1」と「PDE ファイル形式 2」がある。

ファイル形式	説明
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。
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。

PDE ファイル形式 2 は、PDE と COBOL2002 とが連携することで VOS3 の帳票印刷からの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、次に示すどちらかの方法で指定する。

- 設定ファイルで SysoutClass パラメータの FTYPE オペランドおよび RECFM オペランドを指定する。
- SYSOUT データファイルを定義するジョブ定義 XML ファイルの DD 要素の FTYPE 属性および RECFM 属性で指定する。

ジョブ定義 XML ファイルに属性が指定されていない場合、SysoutClass パラメータで指定した値がデフォルト値として使用される。ジョブ定義 XML ファイルに属性が指定されている場合、その属性が優先的に使用される。

また、SysoutClass パラメータの TRUNC オペランドを使用することで、PDE ファイル形式 2 の SYSOUT データファイル作成時に行末の空白を削除する指定ができる。これによって、SYSOUT データファイルのファイルサイズを小さくできる。

SysoutClass パラメータの各オペランドの詳細については「[8. 設定ファイル](#)」の「[SysoutClass \(出力クラスの属性値定義\)](#)」を、各 XML 属性の詳細については「[9. ジョブ定義 XML ファイル](#)」の「[DD 要素 \(ファイルの定義\)](#)」を参照のこと。

## 3.9.8 レベル 1 外字名指定

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、設定ファイルで SysoutClass パラメータの GAIJ11 オペランド、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。

ジョブ定義 XML ファイルに属性が指定されていない場合、SysoutClass パラメータで指定した値がデフォルト値として使用される。ジョブ定義 XML ファイルに属性が指定されている場合、その属性が優先的に使用される。

SysoutClass パラメータの GAIJ11 オペランドの詳細については「8. 設定ファイル」の「SysoutClass (出力クラスの属性値定義)」を、ジョブ定義 XML ファイルの GAIJ11 属性の詳細については「9. ジョブ定義 XML ファイル」の「DD 要素 (ファイルの定義)」を参照のこと。

## 3.9.9 SYSOUT 機能での注意事項

スプール内のデータは定期的に削除すること。次の場合、スプールファイルは自動的に削除されず、スプール内に累積される。

- スプールファイルを出力保留した場合
- スプールファイルを出力後保存した場合
- 設定ファイルの SPOOL\_JOB\_TRANSFER パラメータで NO を指定した場合
- パラメータ指定の誤りやネットワーク障害などによって、スプールファイルの転送または削除に失敗した場合

システム上のファイル数の上限到達時にバッチジョブを実行した場合、各種ファイルを新規作成できず、バッチジョブの実行に失敗することがある。

## 3.10 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。

また、排他/共用制御機能を適用してバッチジョブを実行する場合、BjexSvc サービスが開始されている必要がある。BjexSvc サービスの開始については、「[2.1.1 インストール](#)」を参照のこと。

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解除される。

- 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される。
- 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/共用制御が解除される。

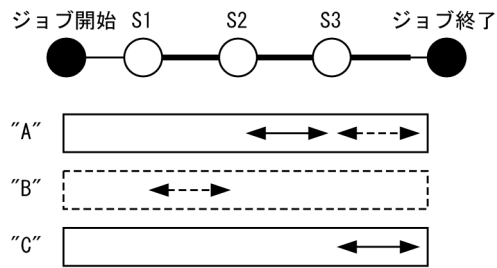
ジョブ中の DD 要素で指定したファイルおよびディレクトリに、一つでも排他/共用制御によるリソースに対するロック確保待ちが発生した場合、すべてのリソースに対するロック確保を行わないで、すべてのリソースに対するロック確保が可能となった時点でロック確保をする。

- **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。

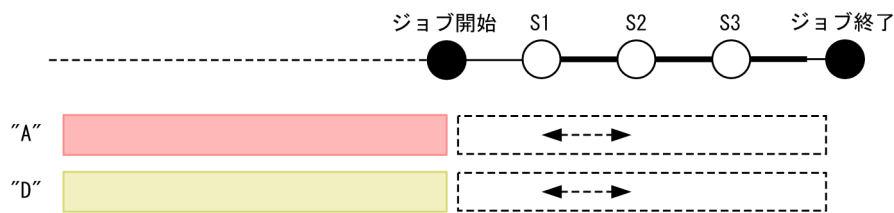
ジョブ単位のファイルおよびディレクトリに対する排他/共用制御を次の図に示す。

図 3-20 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)

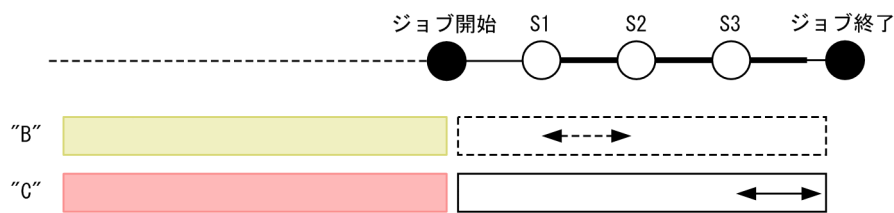
<JOB:JOB0001>



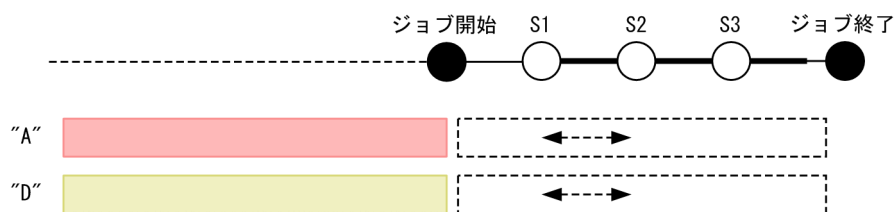
<JOB:JOB0002>



<JOB:JOB0003>



<JOB:JOB0004>



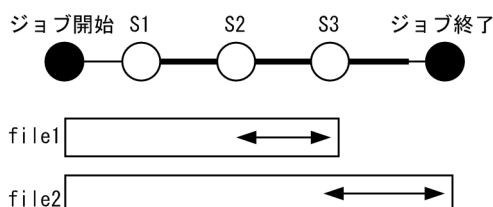
(凡例)

- : 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。

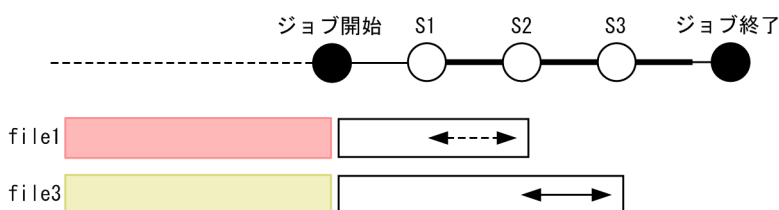
ジョブステップ終了時にファイルおよびディレクトリに対する排他/共用制御を解除する場合を次の図に示す。

図 3-21 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)

<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。
- [ ] : 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 3-22 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。

### 3.10.1 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 3-23 DISP 属性値ごとの排他/共用要求

DISP 属性値	排他/共用※
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※ 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。

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。

- DISP 属性値に"SHR"を指定し、第 2 パラメータに"DELETE"を指定

```
<DD NAME="DD0005" 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="windows">
  <JOB NAME="BJEX_JOB001">
    <STEP NAME="STEP00001">
      <EXEC PGM="PGM001"/>
      <DD NAME="DD00001" TYPE="FILE" DSN="FILE1" DISP="OLD" />      ... (1)
    </STEP>
    <STEP NAME="STEP00002">
      <EXEC PGM="PGM002"/>
      <DD NAME="DD00002" 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。

次に示す DD 要素の割り当てをする場合、指定したファイルを含むディレクトリに対して排他要求が行われていると、該当する DD 要素はロック確保待ちとなる。

- 共用要求でファイルを割り当てる DD 要素の指定

```
<DD NAME="DD0010" TYPE="FILE" DSN="C:\opt\hitachi\bjex\bin\PUBLIC\FILE10" DISP="SHR"/>
```

- 排他要求でディレクトリを割り当てる DD 要素の指定※

```
<DD NAME="DD0010" TYPE="DIR" DSN="C:\opt\hitachi\bjex\bin\PUBLIC" DISP="OLD"/>
```

注※ この指定のジョブを実行している場合、このディレクトリおよびディレクトリ配下のファイルパスの割り当てをするジョブは、ロック確保待ちとなる。

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 3-24 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。
FILE	○	
GDG	○*	
LIB	○	
DUMMY	×	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。
DATA	×	
SYSOUT	×	
TEMP	×	
TEMPISAM	×	

(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※ 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。

## 3.10.2 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 3-22 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 JOBLLOG 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          PPID          JP1JobID  JOBNAME
-----  -----  -----  -
000001 LOCK    00:10:16  21614        21605 0000001001 JOB0001
000002 WAIT    00:05:09  42802        42788 0000001002 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          PPID          JP1JobID  JOBNAME
-----  -----  -----  -
000002 WAIT    00:05:09  42802        42788 0000001002 JOB0002
MODE STATUS LOCKJOB  TYPE  FILEPATH
-----  -----  -----  -
EXC  WAIT    000001  FILE  C:¥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 mutex.
```

- 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 mutex.
```

- 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あと、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 PPID JP1JobID JOBNAME
-----
000084 LOCK 00:00:40 17570 17568 0000001055 JOB0003
000085 WAIT 00:00:32 17578 17574 0000001056 JOB0004
```

上の例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 PPID JP1JobID JOBNAME
-----
000085 WAIT 00:00:32 17578 17574 0000001056 JOB0004
MODE STATUS LOCKJOB TYPE FILEPATH
-----
EXC UNLK NONE FILE C:\opt\hitachi\bjex\bin\public%B
EXC UNLK NONE FILE C:\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の 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。

### 3.10.3 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/共用制御では共有メモリおよび Mutex を使用する。バッチジョブ実行システムが使用する共有メモリおよび Mutex は、BjexSvc サービスの開始時に生成され、BjexSvc サービス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 BjexSvc サービス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。
- JP1 の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する。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。

## 3.11 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。

### 3.11.1 表示されるジョブ情報

表示されるジョブ情報には、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 3-25 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優先度	—	○	○
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

### 3.11.2 セットアップ時の設定

実行中のジョブ情報表示機能を使用する場合、BjexSvc サービスが開始されている必要がある。BjexSvc サービスの開始については、「[2.1.1 インストール](#)」を参照のこと。

### 3.11.3 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。

### 3.11.4 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、排他/共有制御と同様にジョブの多重度制御が実行される。

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 mutex.
```

- 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 a mutex.
```

- 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
```



## 3.12 メッセージ制御機能

---

メッセージ制御機能は、JP1/IM - View の画面にメッセージを出力し、応答を入力する機能である。

### 3.12.1 メッセージ制御機能の概要

メッセージの出力と応答の入力をするには、COBOL2002 で作成した COBOL アプリケーションを使用する。

メッセージ制御機能によって、従来 COBOL2002 で作成した COBOL アプリケーションから stdout/stdin でメッセージを、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。

メッセージ制御機能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。

### 3.12.2 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。

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。

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 アプリケーションが応答を受け取るのを待っている状態である。

### 3.12.3 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 3-26 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。	×
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。
- ×：応答できない。

### 3.12.4 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。

## 3.12.5 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 3-27 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="1.2" os="windows">
  <JOB NAME="SAMPLE">
    <SETENV NAME="BJEX_DISPLAY_REPLY" VALUE="YES"/> . . . (1)
    <STEP NAME="STEP1">
      <EXEC PGM="MYPROG1" LANG="COBOL"/>
    </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)
    </STEP>
  </JOB>
</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。

### 3.12.6 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「8. 設定ファイル」の「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。

### 3.12.7 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 3-28 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="1.2" os="windows">
  <JOB NAME="SAMPLE">
    <SETENV NAME="BJEX_SUPPRESS_CBLDISPLAY" VALUE="YES"/> . . . (1)
    <STEP NAME="STEP1">
      <EXEC PGM="MYPROG1" LANG="COBOL"/>
    </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)
```

(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。

## 3.12.8 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可能性がある。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のリトライを 3 回まで行い、それ以降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。

### 3.12.9 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。

- 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。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。

メッセージを出力する件数の推奨値を次に示す。

表 3-29 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
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		

#### (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。KCCC2060R-S 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 3-30 KCCC2060R-S 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。
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"> <li>• BjexSvc サービスが起動していない。</li> <li>• bjexchmsg コマンドで応答状態を強制的に解除した。</li> </ul>	BjexSvc サービス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 bjexlsmmsg 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おそれがある。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 : 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

- 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、それ以外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。

## 3.13 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。

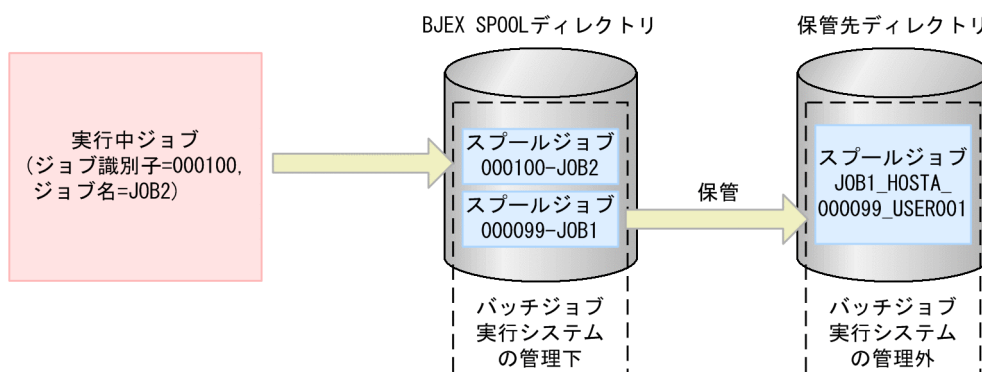
スプールジョブ保管機能を使用して、次のことができる。

- 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。
- 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毎または実行開始日時毎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。

### 3.13.1 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 3-22 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通り残り、転送処理される。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。

ホスト名は設定ファイルの JP1\_HOSTNAME パラメータで指定したホスト名を使用する。また、JP1 の論理ホスト名を指定した場合はその論理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、JP1 の論理ホスト名が有効になる。詳細については、「7. コマンド」の「`bjexec (バッチジョブの実行 (一般ユーザ用))`」を参照のこと。

ユーザ名は環境変数 JP1\_USERNAME から求めたユーザ名を使用する。

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。

保管先ディレクトリパス名に加え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毎または実行開始日時毎に作成され、該当するサブディレクトリ下に保管される。

図 3-23 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

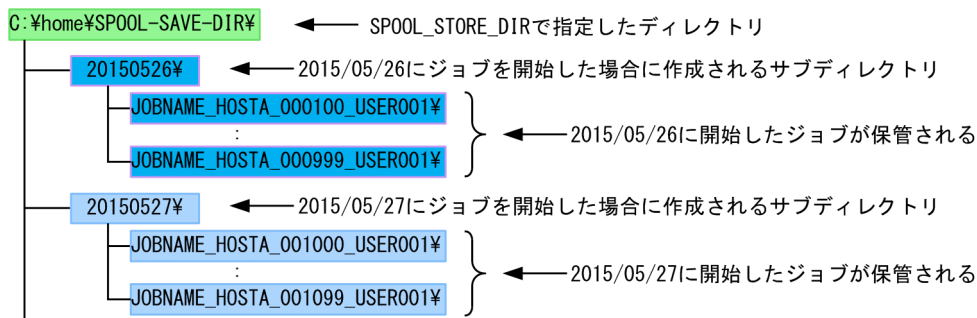
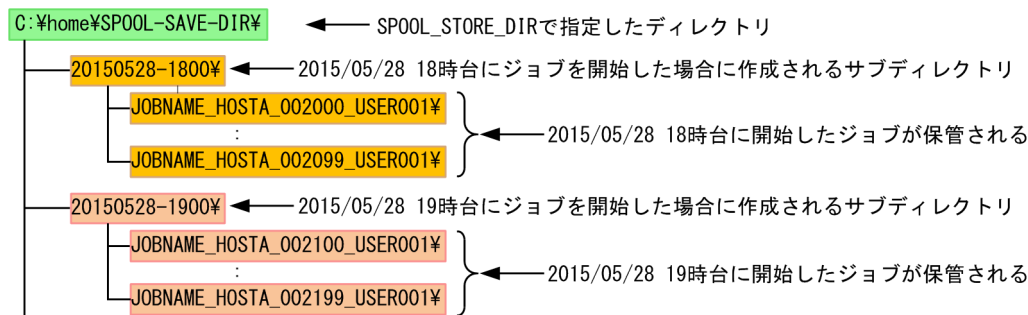


図 3-24 サブディレクトリ定義（ジョブの実行開始日+開始時刻をサブディレクトリ名とする）を指定した場合のスプールジョブ保管



### 3.13.2 スプールジョブ保管機能の設定方法

スプールジョブは、ジョブ終了時にジョブコントローラによって自動保管される。保管先はバッチジョブ実行システムの設定ファイルで定義する。設定ファイルに保管先のディレクトリを定義することで、スプールジョブ保管機能は有効になる。

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を次の表に示す。

表 3-31 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	必須/任意
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。	必須
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。	任意
SPOOL_JOB_STORE	ジョブ終了時にスプールジョブを保管する。	必須

ジョブコントローラによるジョブ実行の終了時に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、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。

### 3.13.3 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。
- ジョブ終了を契機とする保管時に「ジョブ名\_ホスト名\_ジョブ識別子\_ユーザ名」のディレクトリがすでに存在する場合、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関係なく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。
- 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せない。このため、印刷す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。

## 3.14 ファイル操作ユーティリティ

---

### 3.14.1 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

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\_ismut および bjexu\_ismut64

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と複写を行う。索引順編成ファイルを作成および削除でき、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) bjexu\_pcopy

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。

#### (4) bjexu\_scopy

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。

### 3.14.2 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 3-32 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"> <li>• COBOL2002 で作成したファイル</li> <li>• ISAM で作成したファイル</li> </ul>	○※1	○※1
可変長索引順編成ファイル	<ul style="list-style-type: none"> <li>• COBOL2002 で作成したファイル</li> <li>• ISAM で作成したファイル</li> </ul>	○※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'0D0A') で区切られている次のファイル <ul style="list-style-type: none"> <li>• COBOL2002 で作成したテキスト編成ファイル</li> <li>• テキストエディタなどで作成したテキストファイル</li> </ul>	○※2	○※2
CSV 編成ファイル	<ul style="list-style-type: none"> <li>• COBOL2002 で作成した CSV 編成ファイル</li> <li>• 表計算プログラムで作成した CSV ファイル</li> </ul>	—	—

(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。

ファイル操作ユーティリティでは、ハードリンクとシンボリックリンクを意識した機能はサポートしていない。ファイル名やディレクトリ名の重複チェックを行う場合、ジョブ定義 XML ファイルや制御文に指定された内容でチェックする。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。

## (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（¥：円記号）および（/：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。

### 3.14.3 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。JOBLIB 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 3-33 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	<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>¥bin
bjexu_ismut	索引順編成ファイルの作成および削除、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 (ISAM ライブラリの 32 ビット版使用時)	
bjexu_ismut64	索引順編成ファイルの作成および削除、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 (ISAM ライブラリの 64 ビット版使用時)	
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)	

### 3.14.4 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]△<sub>1</sub> コマンド名 △<sub>1</sub>オペランド [△<sub>1</sub> 注釈]**

(凡例)

△<sub>1</sub> : 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注釈を記述できる。

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。

### 3.14.5 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="windows" >
  <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。

## 3.14.6 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。

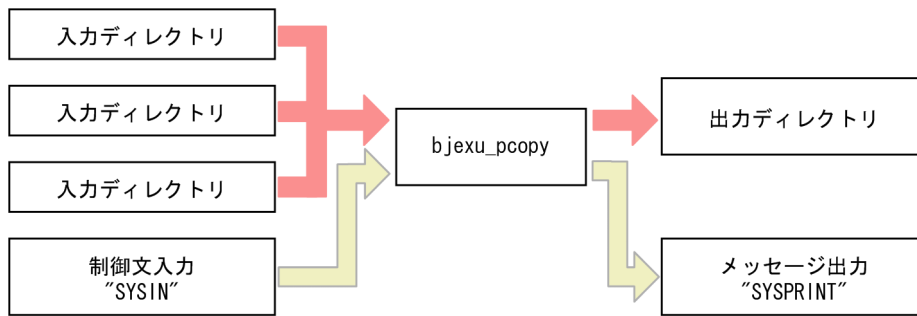
入力ディレクトリが 2 個以上指定されている場合、指定した順にすべての入力ディレクトリを順次処理する。ただし、bjexu\_pcopy 制御文が SELECT 文の MEMBER 指定の場合だけは、先に指定された入力ディレクトリに対象のファイルが見つかったら、後続の入力ディレクトリに同じ名称のファイルがあっても無視する。詳細については、「(4)(b) SELECT 文」の注意事項を参照のこと。

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 3-25 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 3-34 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。

終了コード	意味
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"> <li>制御文の指定誤り</li> <li>制御文や入出力ファイルのオープンエラー、I/O エラー</li> </ul>

注※1 入力ディレクトリと出力ディレクトリが同じ場合は、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ため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 3-35 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実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 3-26 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。

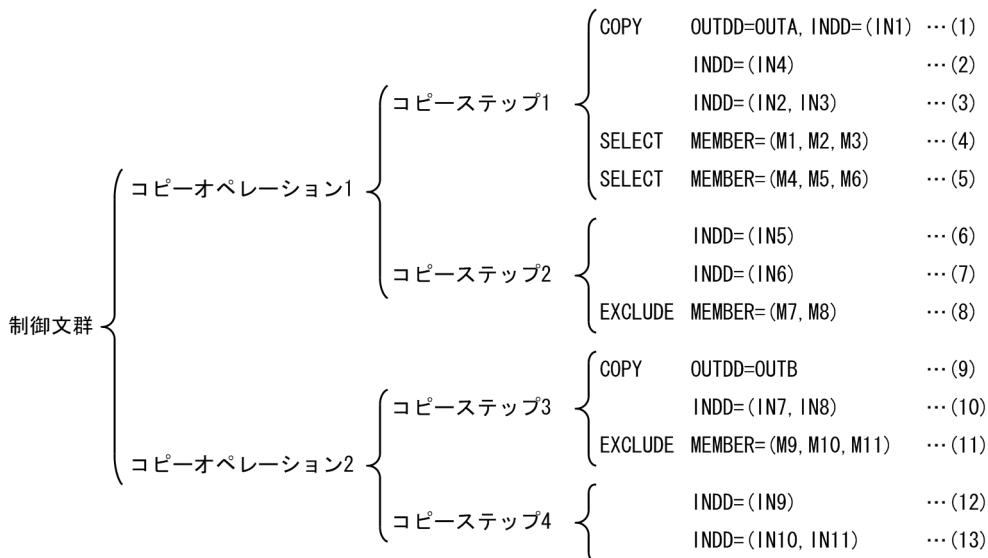
複写処理は、コピーオペレーションの指定順に実行し、複数のコピーステップに分割される場合も、コピーステップ単位に順次実行する。

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 3-26 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

<p>【記号】 Δ<sub>1</sub> COPY Δ<sub>1</sub></p>	<p>OUTDD=DD名            [, INDD=({DD名 (DD名, R)}[, {DD名 (DD名, R)}]...)]            [, LIST=NO]</p>
--	---

## ■ 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ないときに指定する。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 <sub>1</sub> SELECT Δ <sub>1</sub>	{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（¥：円記号）および（/：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。

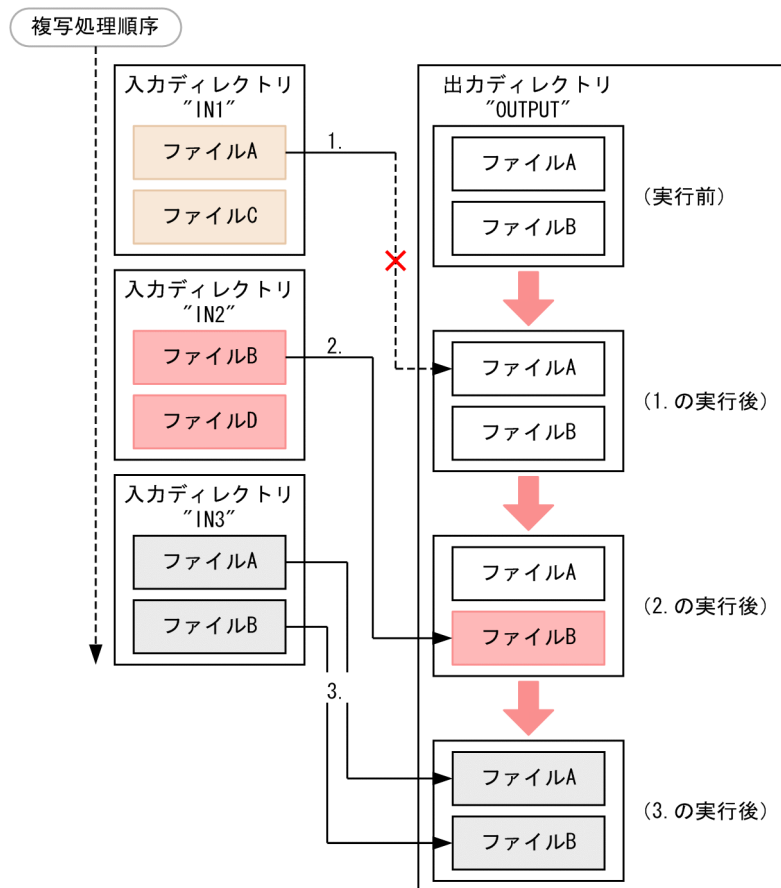
- 制御文

```
COPY  OUTDD=OUTPUT, INDD=(IN1, (IN2, R), (IN3, R))
      SELECT GROUP=(A, B)
```

- 処理の流れ

1. 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。
2. 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。
3. 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 3-27 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、複製しなかった場合も、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製する。ただし、ファイル A は置き換えを行わない。

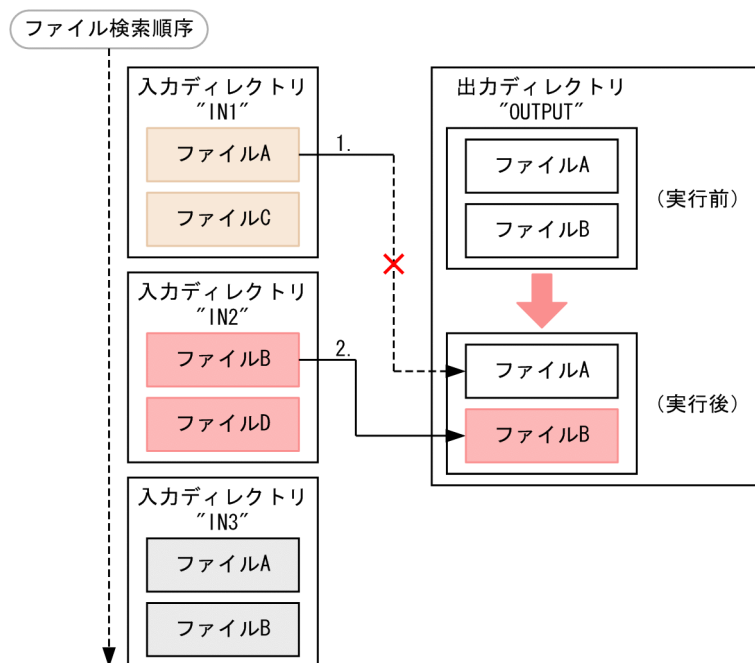
- 制御文

```
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製しない。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、これを複製する。この時点でファイル B の処理は終了する。

図 3-28 複製処理の流れ(SELECT 文の MEMBER オペランド指定の場合)



### (c) EXCLUDE 文

#### ■ 形式

[記号] Δ <sub>1</sub> EXCLUDE Δ <sub>1</sub>	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 <sub>1</sub>	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。

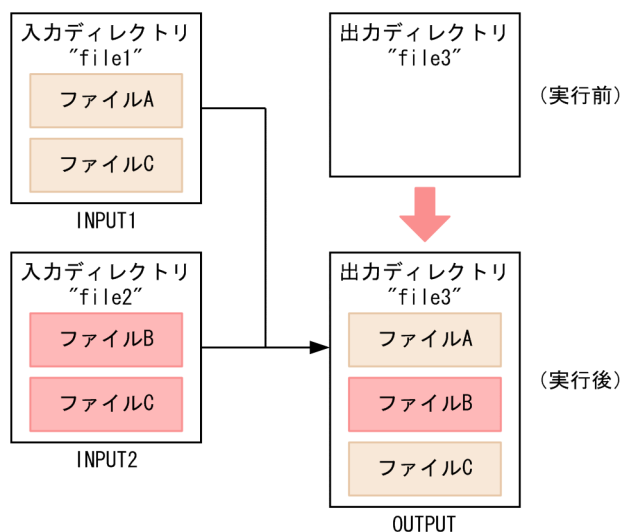
## (5) 使用例

<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。
2. 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。
3. 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 3-29 入力ディレクトリからすべてのファイルを複製（置き換え指定なし）



```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="windows" >
  <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>

```

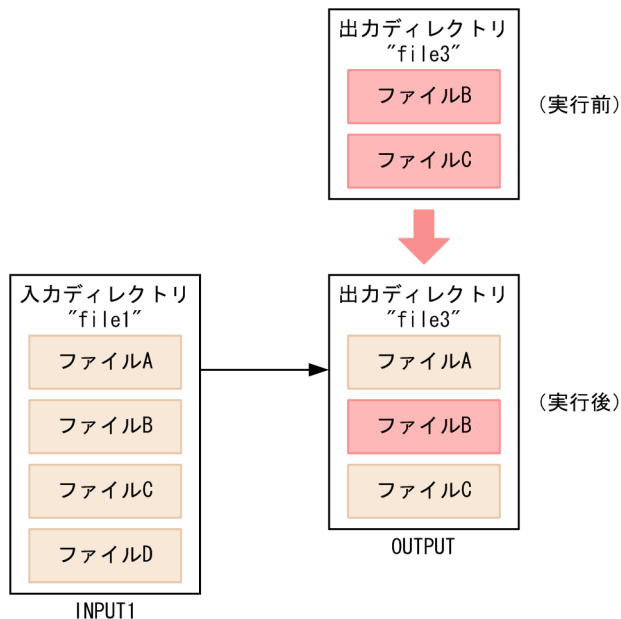
- (1)bjexu\_pcopy を実行する。
- (2)メッセージ出力先に SYSOUT データファイルの A クラスを指定する。
- (3)入力ディレクトリを定義する。ディレクトリは, "file1"を指定する。
- (4)入力ディレクトリを定義する。ディレクトリは, "file2"を指定する。
- (5)出力ディレクトリを定義する。ディレクトリは, "file3"を指定する。
- (6)制御文の入力元を定義する。DD 要素の中に入力データがあることを指定する。
- (7)コピーオペレーションの開始を示す。出力ディレクトリとして OUTPUT(ディレクトリ file3)を指定し, 入力ディレクトリとして 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 3-30 入力ディレクトリからファイル名が A と C のファイルを複写（置き換え指定あり）



```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="windows" >
  <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。

### 3.14.7 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。

ファイルを複写するときに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 3-36 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\_scopsy 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に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。

- 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することができる。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に示す。

表 3-37 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くと、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。

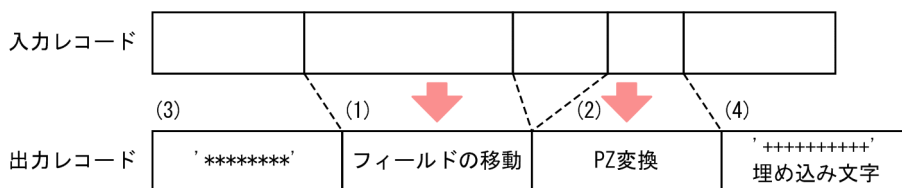
## (c) レコードの編集

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くと、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 3-31 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。

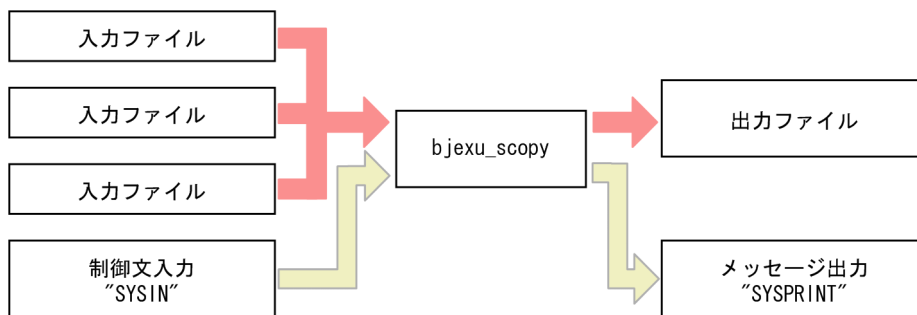
(3)出力レコードの指定位置から指定の文字列を埋め込む。

(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\_scopcy の入出力構成を次の図に示す。

図 3-32 bjexu\_scopcy 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。
- 制御文 (SYSIN)  
bjexu\_scopcy 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。
- メッセージ (SYSPRINT)  
bjexu\_scopcy が出力するメッセージの出力先。

### (c) 終了コード

bjexu\_scopcy の終了コードを次の表に示す。



表 3-38 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"> <li>• 実行できない制御文を指定している。</li> <li>• 制御文や入出力ファイルのオープンエラー、I/O エラー</li> </ul>

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。

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\_scopcy 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\_scopcy 制御文

bjexu\_scopcy 制御文と機能を次の表に示す。

表 3-39 bjexu\_scopcy 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\_scopcy 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 3-40 bjexu\_scopcy 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	OUT FILE	GENERATE		ME MBE R	REC ORD	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 <sub>1</sub> INPFILE Δ <sub>1</sub>	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

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、改行文字で区切られている。

#### RECLEN=レコード長

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 RECLLEN 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ときは、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、INPFILE 文で FILEORG=V を指定する必要がある。

## (b) OUTFILE 文

### ■ 形式

[記号] Δ <sub>1</sub> OUTFILE Δ <sub>1</sub>	FILEORG=ファイル形式[, RECLLEN=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

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、改行文字で区切られている。

## RECLLEN=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 バイト))

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

- RECLLEN 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、RECLLEN 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\_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。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 3-41 出力レコード長の決定方法

環境変数 BJEX_SCOPY_FIXRECLLEN 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 RECLLEN 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とき、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 バイト))**

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 ≥ 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 <sub>1</sub> GENERATE Δ <sub>1</sub>	[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～<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。

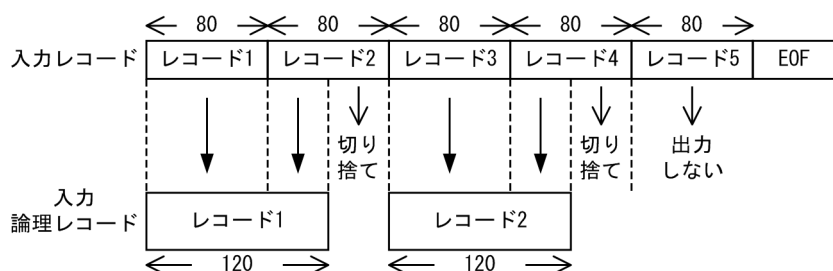
このオペランドは、入力ファイル形式が固定長順編成ファイルのときだけ指定できる。省略した場合はレコードを連結しない。

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。

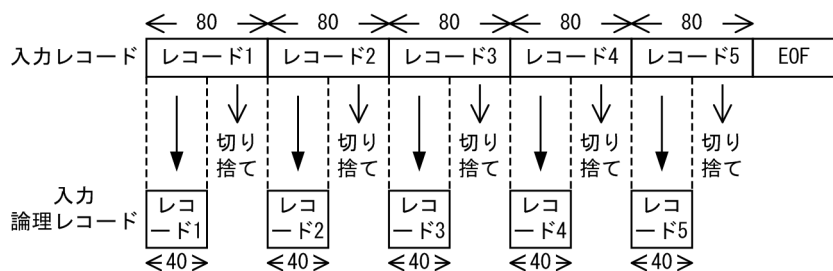
このオペランドを使用した例を次に示す。

### 図 3-33 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 <sub>1</sub> MEMBER Δ <sub>1</sub>	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 (¥: 円記号) および (/: 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 文を指定するときは、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 <sub>1</sub> RECORD Δ <sub>1</sub>	[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 の定義により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 3-42 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。

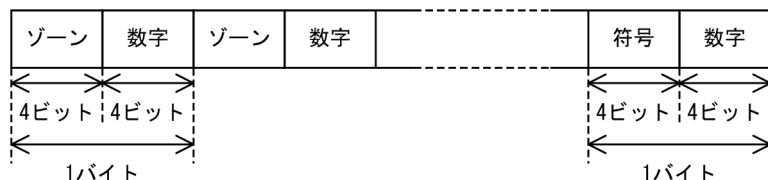
ZP を指定した場合、外部 10 進形式から内部 10 進形式に変換する。

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。



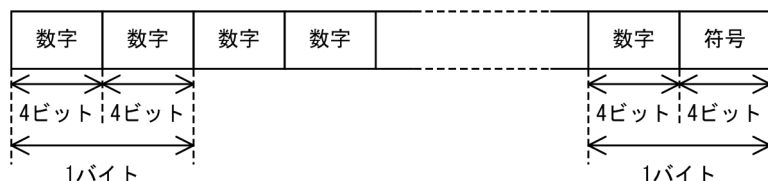
ゾーン部には 3 の値を、数字部には 0~9 の値を指定する。

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)<sub>16</sub> が、ZP 変換のときは (00)<sub>16</sub> 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		出力フィールド
12   34   5C	出力長指定なし	31   32   33   34   35
	出力長=3	33   34   35
	出力長=7	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		出力フィールド
31   32   33   34	出力長指定なし	01   23   4C
	出力長=2	23   4C
	出力長=5	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。

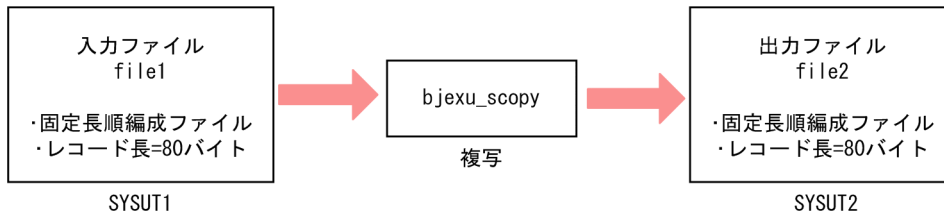
■ 注意事項

- 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\_SCOPIY\_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、「3.14.7(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 3-34 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

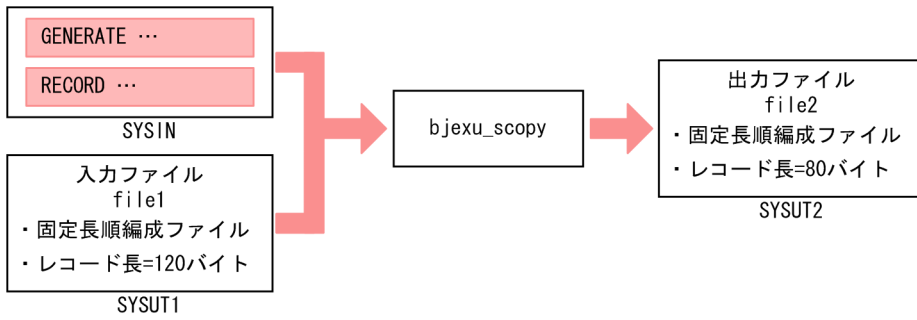


```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="windows" >
  <JOB NAME="SAMPLE1" >
    <STEP NAME="STEP0001" >
      <EXEC PGM="bjexu_scopy" /> ... (1)
      <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" /> ... (2)
      <DD NAME="SYSUT1" TYPE="FILE" DSN="file1" 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)
!
        ]]>
      </DD>
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

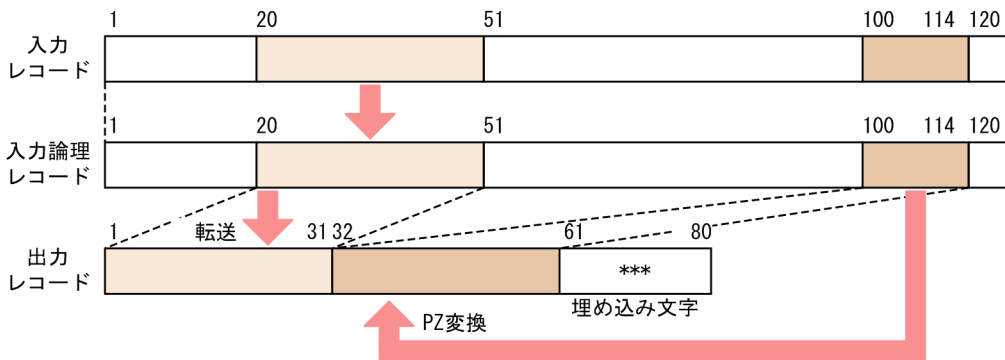
- (1)bjexu\_scopy を実行する。
- (2)メッセージ出力先に SYSOUT データファイルの A クラスを指定する。
- (3)入力ファイルを定義する。ファイルは, "file1"を指定する。
- (4)出力ファイルを定義する。ファイルは, "file2"を指定する。
- (5)制御文の入力元を定義する。DD 要素の中に入力データがあることを指定する。
- (6)入力ファイルの形式を指定する。入力ファイルは, 固定長順編成ファイルでレコード長が 80 バイトであることを指定する。OUTFILE 文の指定を省略した場合, INPFILE 文で指定した形式を仮定する。

<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。

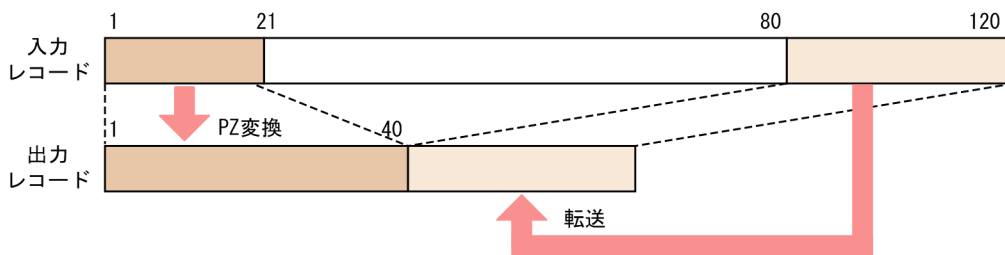
図 3-35 入力の固定長順編成ファイルをレコード編集して出力ファイルに複写



・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="windows" >
  <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>

```

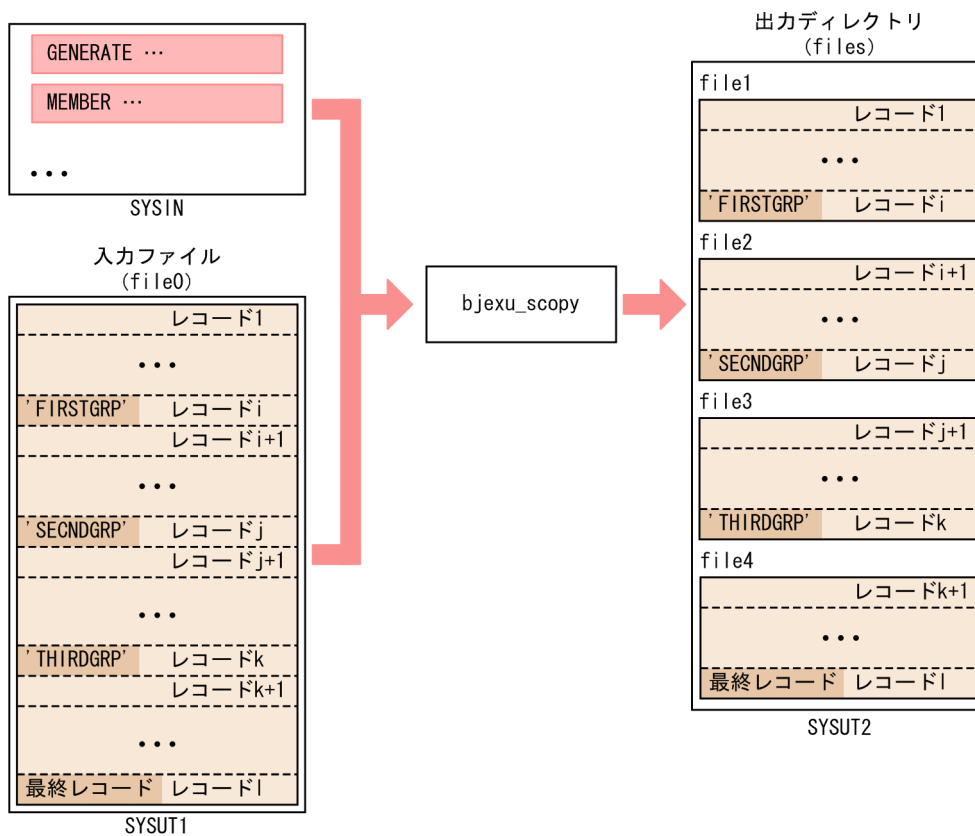
- (1)bjexu\_scopy を実行する。
- (2)メッセージ出力先に SYSOUT データファイルの A クラスを指定する。
- (3)入力ファイルを定義する。ファイルは, "file1"を指定する。
- (4)出力ファイルを定義する。ファイルは, "file2"を指定する。
- (5)制御文の入力元を定義する。DD 要素の中に入力データがあることを指定する。
- (6)入力ファイルの形式を指定する。入力ファイルのレコード長は, 120 バイトであることを指定する。
- (7)出力ファイルの形式を指定する。出力ファイルのレコード長を 80 バイトにすることを指定する。
- (8)ファ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 3-36 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="windows" >
  <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。

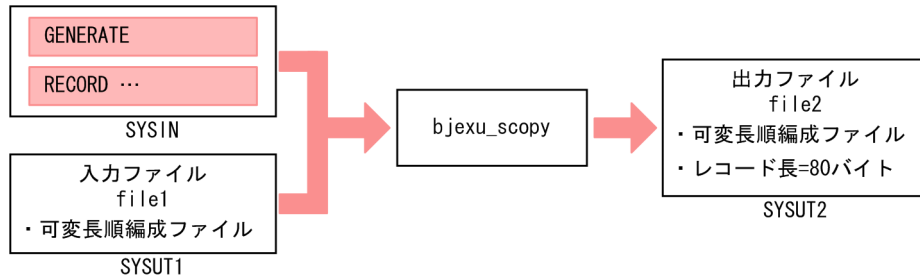


(14)第3レコードグループの最後のレコードを指定する。レコードの1バイト目から8バイトの内容が'THIRDGRP'のレコードを示す。

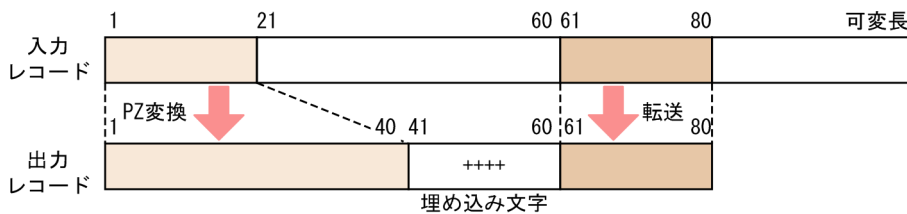
(15)第4レコードグループに付けるファイル名 file4 を指定する。RECORD 文を省略し、入力ファイルの最後までを第4レコードグループとして出力することを指定する。

<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 3-37 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="windows" >
  <JOB NAME="SAMPLE2" >
    <STEP NAME="STEP001" >
      <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>

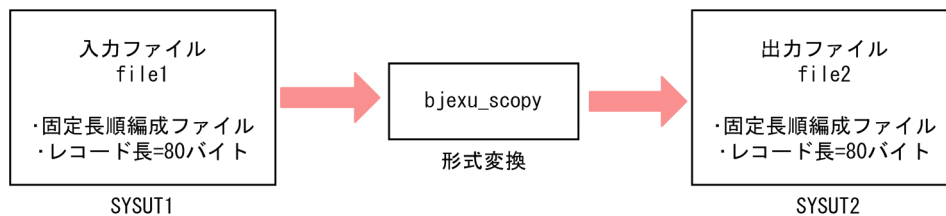
```

- (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 3-38 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="windows" >
  <JOB NAME="SAMPLE1" >
    <STEP NAME="STEPO001" >
      <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。

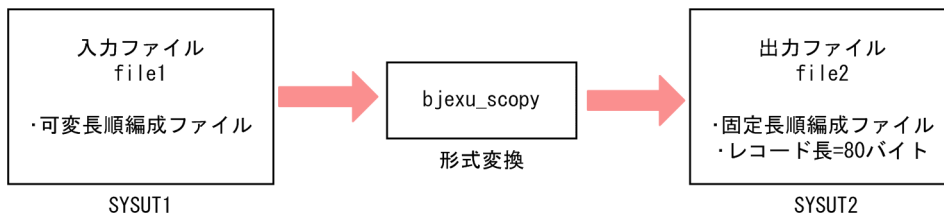
(6)入力ファイルの形式を指定する。入力ファイルは、固定長順編成ファイルでレコード長が 80 バイトであることを指定する。

(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 3-39 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="windows" >
  <JOB NAME="SAMPLE1" >
    <STEP NAME="STEPO001" >
      <EXEC PGM="bjexu_scopy" /> ... (1)
      <DD NAME="SYSOOUT" TYPE="SYSOOUT" SYSOOUT="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\_scopy を実行する。

(2)メッセージ出力先に SYSOOUT 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。

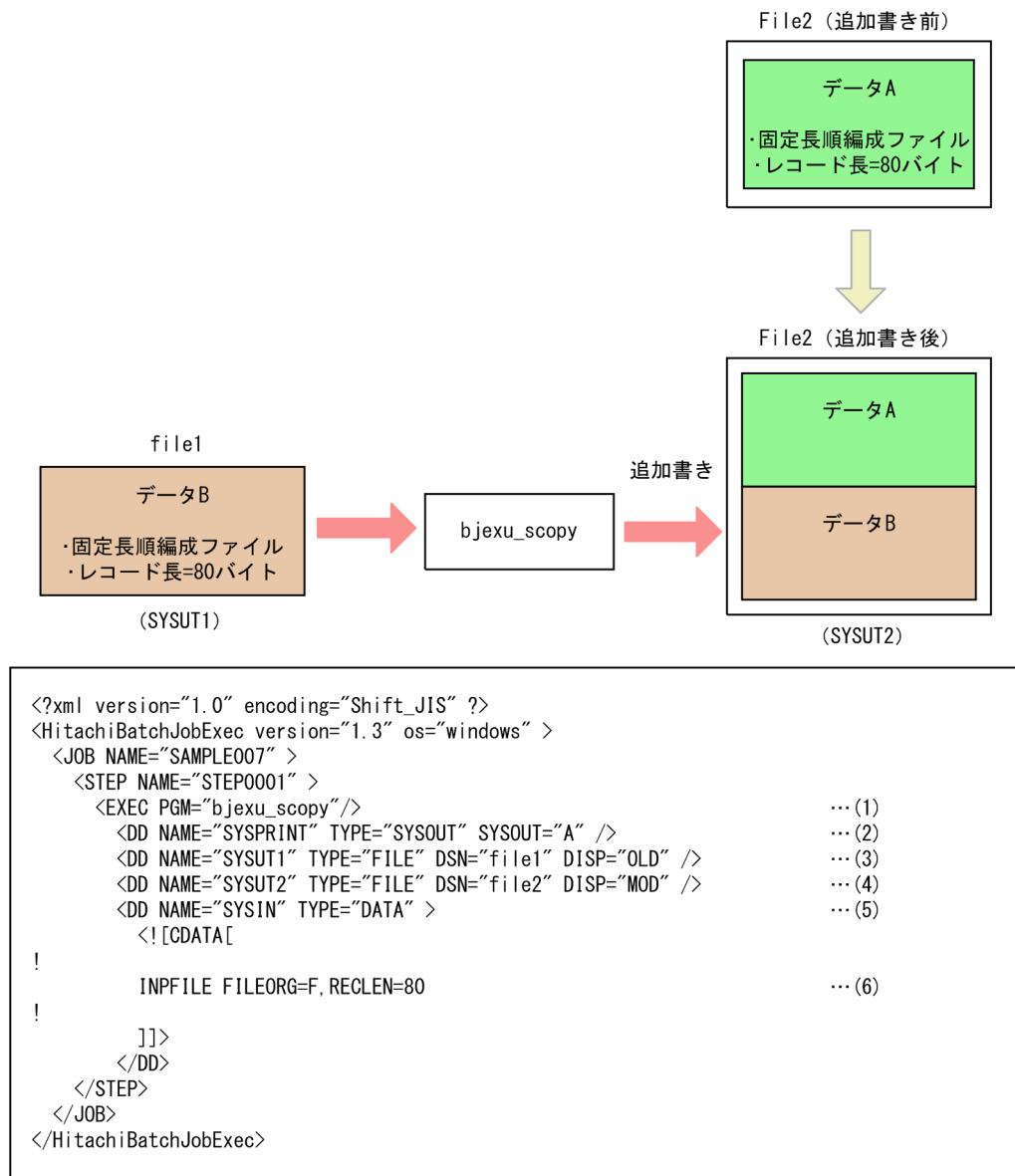
(7)出力ファイルの形式を指定する。出力ファイルは、固定長順編成ファイルで、レコード長を 80 バイトにすることを指定する。

(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 3-40 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)bjemu\_scoppy 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、INPFILE 文で指定した形式を仮定する。

### 3.14.8 bjexu\_ismut および bjexu\_ismut64 (索引順編成ファイルの作成および削除, 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および削除, 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 3-43 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 3-44 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
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 3-45 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 3-46 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トには、次のものがある。

表 3-47 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。

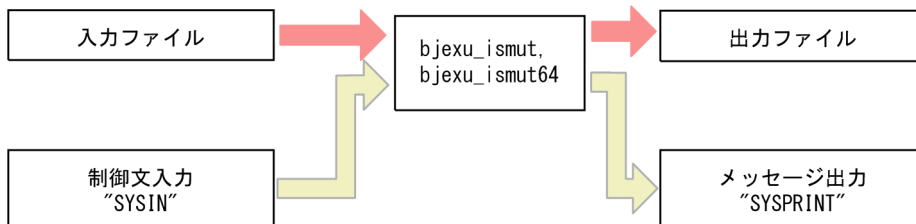
- 可変長索引順編成ファイルを削除する場合  
索引順編成ファイルの dd 要素名を指定する。

```
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 3-41 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。
- ・ 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。

## (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元となるファイル。  
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。
- ・ 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, 変換または併合の出力先となるファイル。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。
- ・ 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。

## (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。

## (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 3-48 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いるため、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こと。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"><li>• 制御文の指定誤り</li><li>• 制御文 (SYSIN) や入出力ファイルのオープンエラー</li></ul>

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

- 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" />
```

- 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" />
```

### (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。

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 3-49 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 3-50 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 3-51 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、変換または併合の実行開始を指定する。
DELETE 文	削除するファイルを指定する。
SET 文	変更する終了コードを指定する。

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組とし、DEFINE 文、DELETE 文、および SET 文と組み合わせ指定することもできる。



INPFILE 文と OUTFILE 文で指定できる組み合わせと処理内容の対応の詳細については、「(1) 機能」の「表 3-43」を参照のこと。

## (a) DEFINE 文

### ■ 形式

[記号] Δ <sub>1</sub> DEFINE Δ <sub>1</sub>	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 3-52 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	認めない
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	認めない

#### 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 バイト))

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 バイト))

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 <sub>1</sub> INPFILE Δ <sub>1</sub>	FILEORG=ファイル形式～((I X F V))[, 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。

### RECLLEN=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 バイト))

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 <sub>1</sub> OUTFILE Δ <sub>1</sub>	FILEORG=ファイル形式～((I X F V))[, RECLLEN=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。

### RECLLEN=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バイト))

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。

- RECLLEN 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。
- RECLLEN オペランドを省略すると，INPFIL 文の RECLLEN 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

出力ファイルのレコード長は，入力ファイルの形式によって次のとおりになる。

- 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，RECLLEN 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。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。

- 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バイト))

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。

### ■ 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 ≤ 入力レコード長 ≤ 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。
- RECLEN 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 <sub>1</sub> REPRO Δ <sub>1</sub>	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、2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 <sub>1</sub> DELETE Δ <sub>1</sub>	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 <sub>1</sub> SET Δ <sub>1</sub>	[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

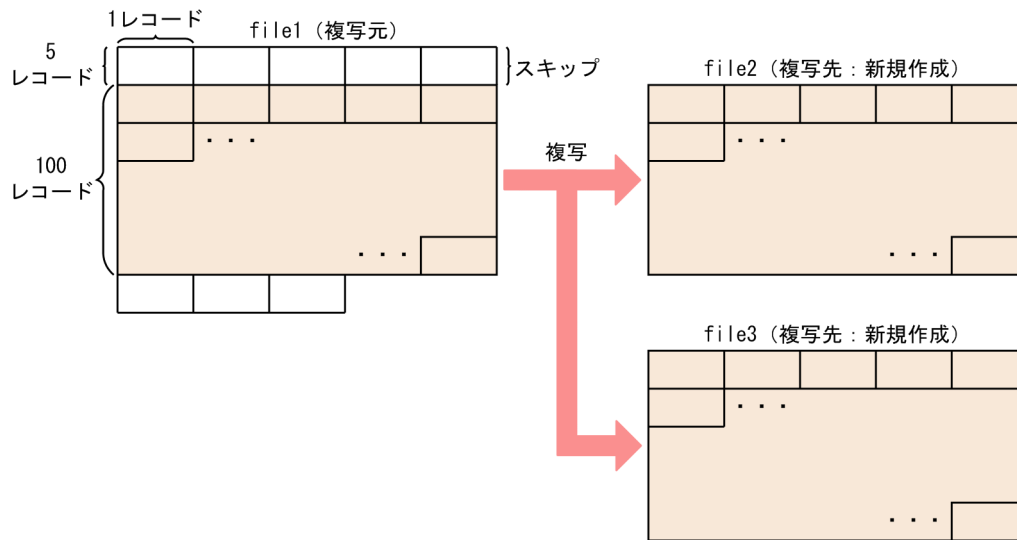
### (a) ファイルの複写

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 3-42 索引順編成ファイルの一部を複製



```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="windows" >
  <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)
      INPFILE FILEORG=I : (15)
      OUTFILE FILEORG=I : (16)
      REPRO KEY=1, : (17)
        INDD=SYSUT1, : (18)
        OUTDD=SYSUT2, : (19)
        COPYDD=SYSUT3, : (20)
        SKIP=5, : (21)
        COUNT=100 : (22)
!
      ]]>
    </DD>
  </JOB>
</HitachiBatchJobExec>

```

```
</STEP>  
</JOB>  
</HitachiBatchJobExec>
```

- (1)bjexu\_ismut プログラムを実行する。
- (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、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。
- (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。
- (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。

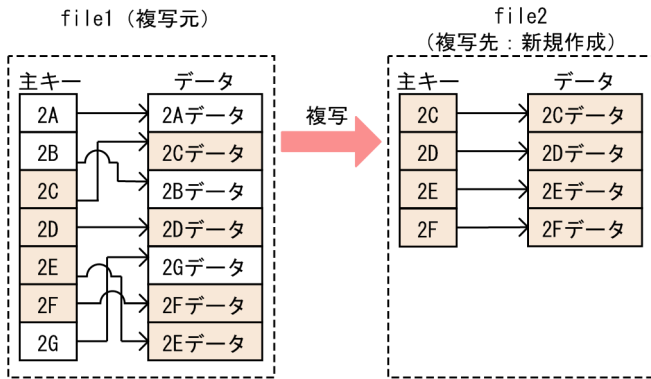
(20)COPYDD オペランドは、もう 1 つの出力ファイルを定義した DD 名を指定する。

(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 3-43 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="windows" >
  <JOB NAME="SAMPLE2" >
    <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" />       :(2)
      <DD NAME="SYSIN" TYPE="DATA" >
        <![CDATA[
!
      DEFINE ISAM,                                           :(3)
          DD=SYSUT2,                                         :(4)
          ISRECL=(100),                                       :(5)
          ISRECFM=F,                                         :(6)
          ISKEY=(2,0,C)                                       :(7)
      INPFILE FILEORG=I                                       :(8)
      OUTFILE 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。

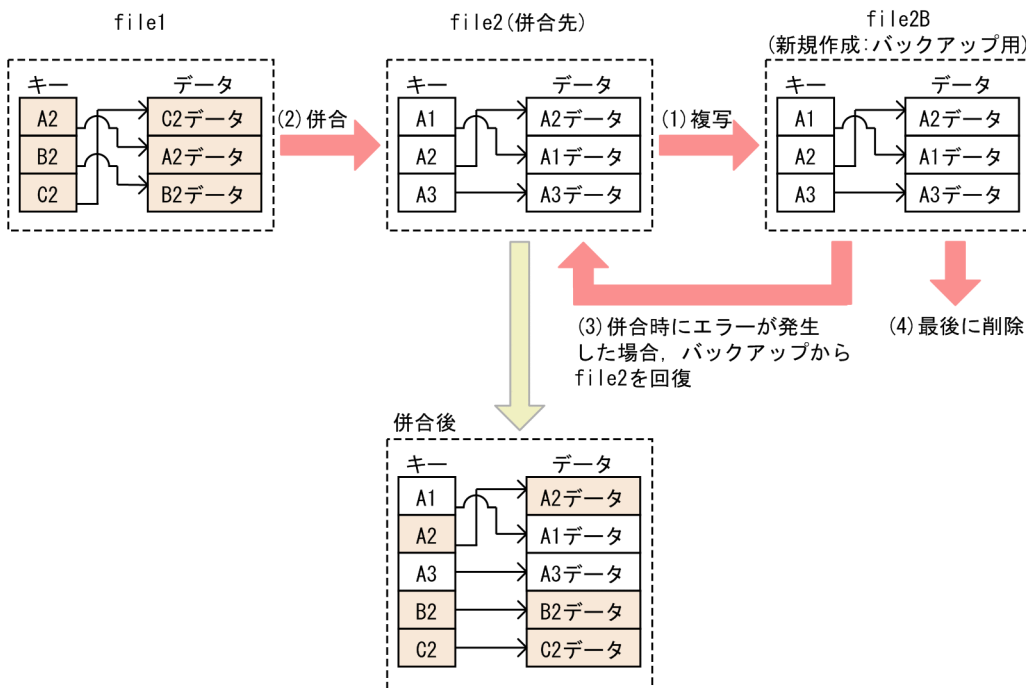
ファイルの併合は、レコードを複写する点ではファイルの複写と同じだが、併合先の出力ファイルにすでにレコードがある点が異なる。ファイルの併合の例では、次の処理を実施している。

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 3-44 索引順編成ファイルで特定の主キーのレコードを複製



```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="windows" >
  <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)
!
      ]]]>
    </STEP>
    <STEP NAME="STEP0002" >
      <EXEC COND="8,NE,STEP0001" PGM="*" > : (4)
    </STEP>
  </JOB>
</HitachiBatchJobExec>

```

```

    <![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記録があるため、ファイルの併合を実施する。

(3)REPLACE 指定がある場合、同じ主キーの記録が出力先にあると、その記録を置き換える。

(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。

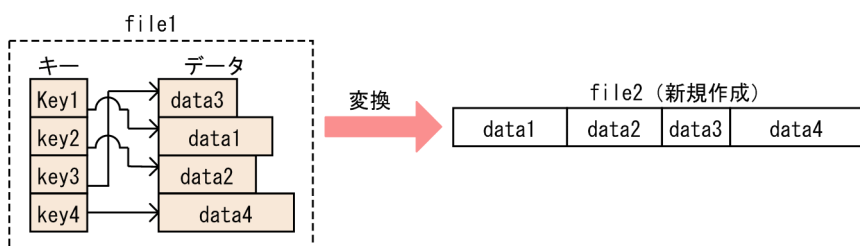
注 この例では、複写する記録を指定するオペランドが省略されているため、主キーの順にすべての記録が複写される。

### (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 3-45 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



```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="windows" >
  <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。

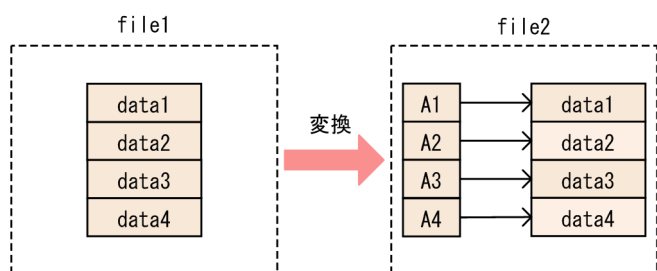
(4)OUTDD オペランドに出力ファイルを定義した DD 名を指定する。出力するレコードは、入力ファイルのレコード長で出力される。

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 > ISAM ライブラリの 64 ビット版の bjexu\_ismut64 プログラムを指定して、入力元の固定長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 3-46 固定長順編成ファイルを固定長索引順編成ファイルに変換





```

<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="windows" >
  <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)ISAM ライブラリの 64 ビット版の 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 は固定長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 は固定長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あと、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="windows" >
  <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>
  </STEP>
</JOB>
</HitachiBatchJobExec>

```

```

<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あと、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="windows" >
  <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>

```

```

!
  ]]>
  </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なる。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="windows" >
  <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="windows" >
  <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

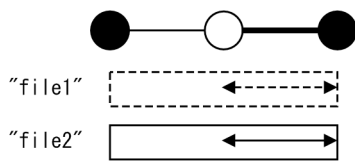
(4)OUTFILE 文で、出力が索引順編成ファイルであることを指定

(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>

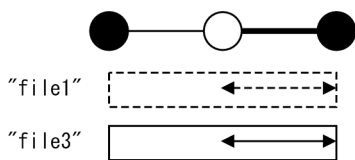
ジョブ開始 S1 ジョブ終了



JOB0001とJOB0002でfile1を入力ファイルとして指定して、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>

ジョブ開始 S1 ジョブ終了



(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要素のスコップを示す。
- ←——> : 排他要求のDD要素のスコップを示す。
- [- - - -] : 共用ロック確保区間を示す。
- [———] : 排他ロック確保区間を示す。

### 3.14.9 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。

## 3.15 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「3.15.2 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。
- 強制終了時に、Java バッチアプリケーションに対して内部的に cjkilljob 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 3-53 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
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「3.15.1(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。

### 3.15.1 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 3-54 ジョブ定義 XML ファイルに指定する情報

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)	247 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ルするとき割り当てるバッチサーバのグループを指定する。 例: Group="MyScheduleGroup"

要素	属性	指定する値	最大長	説明
EXEC	Option	Java オプション	1,023 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 C:¥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 C:¥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="C:¥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\_" を与えた文字列と 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="C:¥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"C:¥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="Dオプションのname部分" DSN="Dオプションのvalue部分"/>
```

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。

```
cjexecjob△スケジュールグループ名△-jobID△ジョブID△  
<Javaオプション△"-DDD_<Dオプションのname部分">  
<-Dオプションのvalue部分">"△"Javaアプリケーションクラス名">  
<mainメソッドに渡す引数>
```

構築例を次に示す。

```
<EXEC PGM="jp.co.Hitachi.MyBank.MyJob" LANG="Java"  
Group="MyGroup" Option="-cp C:¥mydir¥hello.jar -DHello=World"  
PARM="myjob check_transaction" /><DD NAME="WRITEDATA"  
DSN="C:¥myinfo¥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。

```
cjexecjob△MyGroup△-jobID△BJEXが自動割り当てした29桁の英数字>  
△-cp△C:¥mydir¥hello.jar△-DHello=World△  
-DDD_WRITEDATA=C:¥myinfo¥mydata.txt△"myjob check_transaction"
```

## (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 3-47 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 3-55 Java アプリケーションに対するジョブ ID の割り当て形式の項目と内容

項目	長さ	内容
BJEX	4 桁	バッチジョブ実行システムが割り当てるジョブ ID に自動的に与えられる。"BJEX"固定である。
ジョブ識別子	6 桁	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 桁	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 桁	内部的な時刻識別用情報である。
ID	8 桁	内部的な識別用 ID である。

## (5) 注意事項

- 空白を含む文字列を属性値として指定する場合は、シングルクォートとダブルクォートの順に囲むこと。次に例を示す。
  - main メソッドへの引数として空白入りの文字列「KAIKEI HOUKOKU」を渡したい場合は、`PARM="KAIKEI HOUKOKU"`と指定する。
  - D オプション「PROFIT」に対し、value 部分に空白入りの文字列「KESSAN VALUE」を指定したい場合は、`Option='-DPROFIT="KESSAN VALUE"'`と指定する。
- 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。
- 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。

## 3.15.2 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。

## 3.15.3 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。

## 3.16 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。

### 3.16.1 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}}
```

#### (1) 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。

## 3.16.2 使用例

### (1) ジョブ定義 XML ファイル

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<HitachiBatchJobExec version="1.3" os="windows" >
  <JOB NAME="ENV_REPLACE">
    <STEP NAME="STEP01" >
      <EXEC PGM="{BJEX_DIR_SHELLDIR}%{EXEC_PGM}" />
      <DD TYPE="{TYPEVAL}" NAME="DDNAME" %{DSNATTR} />
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

### (2) 環境変数の設定

環境変数名	指定値	指定個所
BJEX_DIR_SHELLDIR	abc123	JP1/AJS の [詳細定義 - [PC Job]] 画面の [環境変数] 欄
	C:¥shell¥	バッチジョブ実行システムの設定ファイル(bjex.conf)
EXEC_PGM	mkDD.bat	JP1/AJS の [詳細定義 - [PC Job]] 画面の [環境変数] 欄
TYPEVAL	TEMP	JP1/AJS の [詳細定義 - [PC Job]] 画面の [環境変数] 欄
DSNATTR	指定なし	該当しない

### (3) 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="1.3" os="windows" >
000003 <JOB NAME="ENV_REPLACE">
000004 <STEP NAME="STEP01" >
000005 <EXEC PGM="C:¥shell¥mkDD.bat" />
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.bjx" started.
16:29:17 002404 KAKC1175-I Environment variable %{BJEX_DIR_SHELLDIR} was replaced to "C:¥shell¥".
16:29:17 002404 KAKC1175-I Environment variable %{EXEC_PGM} was replaced to "mkDD.bat".
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
```

## 3.17 環境変数

### 3.17.1 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 3-56 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
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。
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。

### 3.17.2 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ジョブステップを実行した場合に受け取ることのできる環境変数を次の表に示す。環境変数名は、大文字・小文字を区別しない。

表 3-57 ジョブコントローラからジョブステップを実行した場合に受け取ることのできる環境変数

環境変数名	内容
BJEX_DIR_で始まる名称	設定ファイルに記述されたパス名。
BJEX_JOB_NAME	JOB 要素の NAME 属性に指定したジョブ名。
BJEX_JOBID	ジョブ識別子 (6 桁固定 (000001~999999) の 10 進数の値)。
BJEX_STEP_NAME	STEP 要素の NAME 属性に指定したジョブステップ名。 NAME 属性の指定を省略した場合は環境変数自体が設定されない。
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設定される。
CBLDISP_DD 名※2※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。

環境変数名	内容
CBL_で始まる名称※2	LANG="COBOL"の指定があるステップ内の DD 要素のパス名。 LANG="COBOL"の指定があるステップ内の 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。
DDN_で始まる名称※2	LANG="COBOL"の指定がないステップ内の DD 要素のパス名。 LANG="COBOL"の指定がないステップ内の 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。
JP1 が設定する環境変数	内容については、JP1 のマニュアルを参照のこと。
PATH	DD 要素で TYPE="LIB"を指定したときの DSN 属性に指定したディレクトリのパス名を追加。
RENAME 属性で指定した名称※2	RENAME 属性を指定した DD 要素のパス名。
任意	SETENV 要素で定義した環境変数。

注※1 詳細は、マニュアル「PRINT DATA EXCHANGE for Open」を参照のこと。

注※2 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※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。

#### 注意事項

ジョブコントローラは、ジョブステップと連携した機能を実現するために、環境変数名が BJEX から始まる環境変数をジョブステップに設定している。そのため、SETENV 要素や OS の実行環境などで、環境変数名が BJEX から始まる環境変数は設定しないこと。

### 3.17.3 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に示す。

表 3-58 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_SCOPY_FIXRECLLEN	bjexu_scopy で OUTFILE 文に RECLLEN 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。
BJEX_SCOPY_DISP_MOD	bjexu_scopy で 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指定する。

# 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、管理者特権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一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

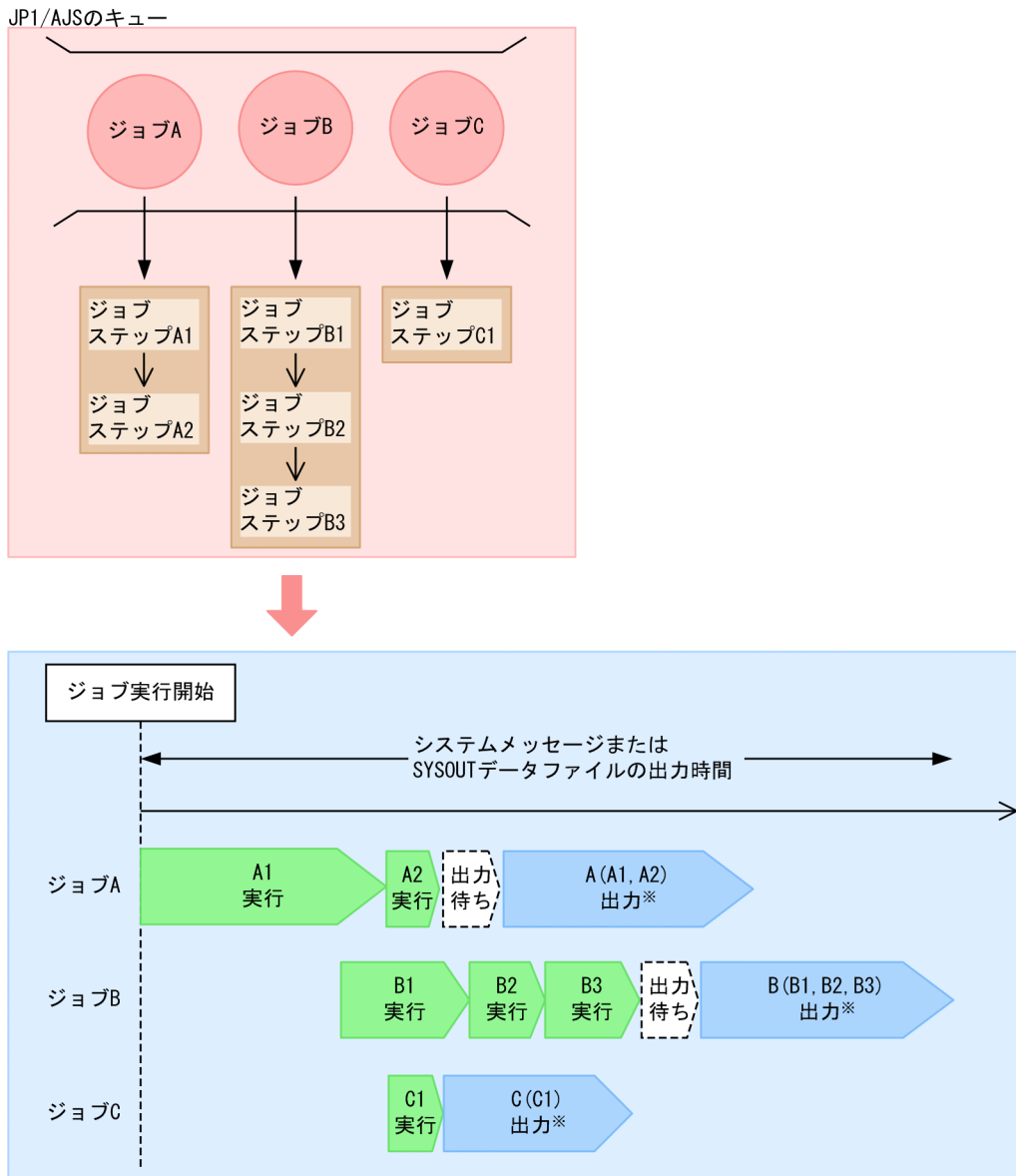
ジョブとジョブとは互いに独立である。したがって、同時に処理するジョブ同士、または先に実行したジョブがあとに実行するジョブに影響を及ぼすことはない。さらに、ファイル上の情報を除けば、ジョブからジョブに情報を引き継がれることもない。ただし、JP1/AJS によるスケジュールによっては、ジョブ相互に実行順序関係ができる。

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示している。

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、JOB LIB 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。

JOB LIB 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<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥temp 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。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

恒久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<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥perm である。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けられ、このパラメータ配下に限定できる。

## (5) スプール

スプール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<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥spool である。バッチジョブが作成するスプールファイルの総量を見積もり、必要であれば専用のディスクを割り当てるなど、十分な空き領域を準備すること。また、スプールファイル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

システムを構築する場合、ディスク容量や仮想メモリを調整し、バッチジョブ実行システムの実行処理に必要なリソースを最適化すること。

メモリ所要量、ディスク占有量およびプロセス数については、リリースノートを参照のこと。

## (7) ファイルサイズ制限値設定時の注意事項

ジョブ実行ユーザごとのファイルサイズ設定時の JP1/AJS の注意事項については、JP1 のマニュアルを参照のこと。

## (8) 大文字と小文字の区別に関する注意事項

- Windows では、ファイル名やディレクトリ名を含むパス情報は、大文字と小文字を区別しない。例えば、C:¥Temp ディレクトリと c:¥temp ディレクトリは同じディレクトリとして扱われる。
- 次に示す項目は大文字と小文字が区別されるため注意すること。
  1. 設定ファイルのパラメータ名
  2. ジョブ定義 XML ファイルの要素名、属性名、およびパス情報を除く各属性の指定値
  3. コマンドのオプション

### 4.3.2 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。

### 4.3.3 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（ログオン）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、ログオン後は、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。

## (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、OS のアクセス制御リスト（ACL（Access Control List））によって設定される。ジョブコントローラによって作成されるファイルは、親ディレクトリのアクセス権が継承される。または、作成ユーザだけにアクセス権が与えられる。

ユーザ間で共用するファイルはあらかじめ共通のディレクトリを用意しておき、共有するユーザ間で共用できるように ACL を設定しておく必要がある。

### 4.3.4 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。

### 4.3.5 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](#)、[LOCK\\_MAXFILE](#)」を参照のこと。

## 4.3.6 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 COBOL プログラムが終了しなくなる。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、COBOL プログラム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 に同時期に大量の応答要求メッセージを送付しないようにするため、サーバ内の応答要求メッセージ送付数を指定する。これによって、JP1/IM - Manager に送付する応答要求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

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。

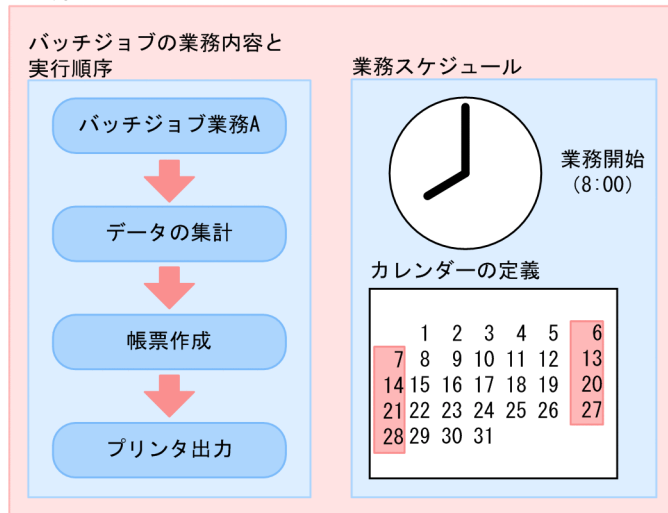
JP1/AJS を使用して、バッチジョブ業務を自動的に実行する場合、次に示す内容を定義する必要がある。

- 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

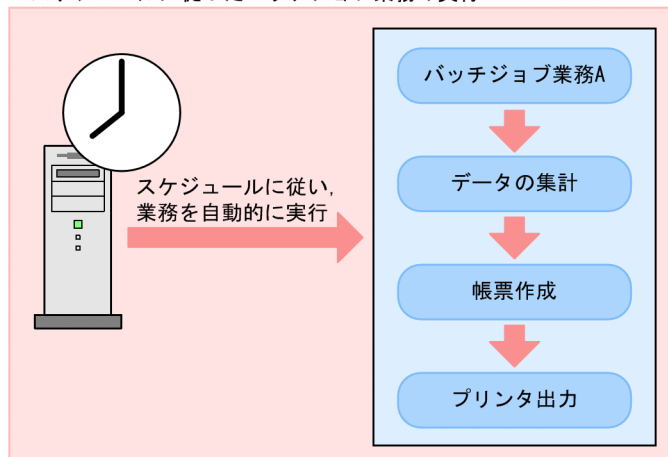
JP1/AJS を使用したバッチジョブ業務の自動化の概要を次の図に示す（図中の番号は、次に示す説明の項番と対応している）。

## 図 4-2 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

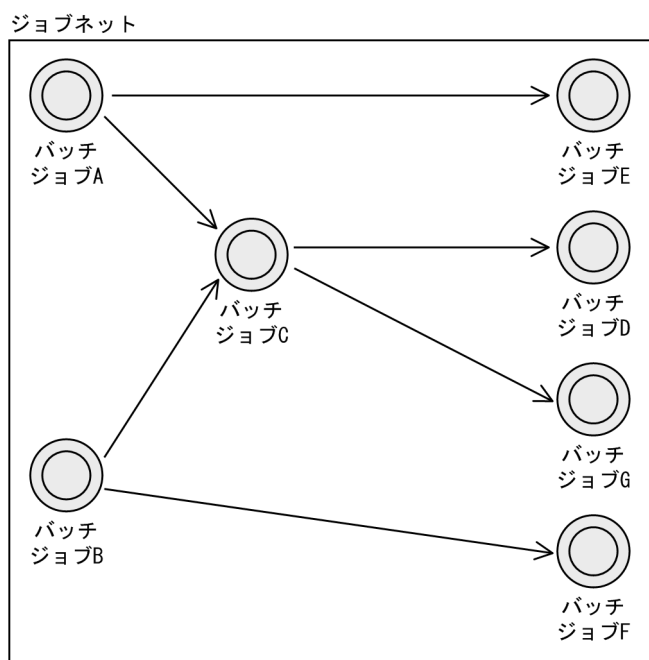
これらの流れをジョブコントローラのジョブステップとして、1~4をジョブ定義XMLファイルに定義することで自動化を実現できるが、12:00にデータベースからデータを抽出するという作業の自動化は行えない。バッチジョブ実行システムとJP1/AJSでバッチジョブ業務と実行順序の定義を実行するには、業務を構成する一連の流れをジョブコントローラで定義し、それぞれのバッチジョブ業務と実行順序の定義の関係をJP1/AJSの実行順序、または実行時間として定義する。

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 4-3 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) 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。

- PC ジョブ (ただし、キュージョブ)
- サブミットジョブ

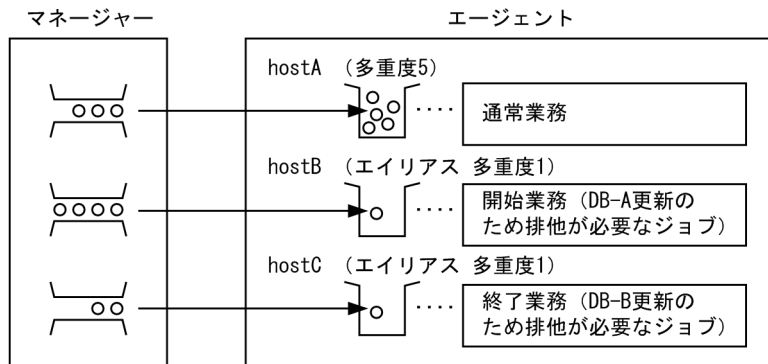
上記以外の種類のジョブで、バッチジョブ実行システムのジョブを実行した場合、動作は保証されない。

#### (2) 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 4-4 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、PC 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バッチジョブ実行システムのジョブ定義 XML ファイル名を指定する。ジョブ定義 XML ファイル (.bjx ファイル) は bjexec コマンドと関連づけられているため、このパラメータに指定したジョブ定義 XML ファイルを入力として bjexec コマンドが実行される。

#### 指定例

```
sc="C:\Program Files\Hitachi\BJEX\sample\sample.bjx"
```

### • 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 NUL 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場合。  
"NUL"を記述すること。



"NUL"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指定しているときは、Windows のコマンドプロンプトに記述できるコマンド行やスクリプトファイル内で、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 を介さないで直接起動した場合、対話処理と同等の優先度は「3」となる（Windows で規定される NORMAL\_PRIORITY\_CLASS を設定する）。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 バッチジョブおよびジョブネットの強制終了

#### (1) 強制終了の方法

##### (a) JP1/AJS から直接バッチジョブを実行している場合

JP1/AJS の操作によって実行中のバッチジョブ実行システムのバッチジョブおよびジョブネットを強制終了できる。強制終了の指示を受けたジョブコントローラは、実行中のジョブステップのプロセスを強制終了する。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または ajskill 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、「強制終了」となる。また、ジョブログを含むジョブコントローラの標準エラー出力の一部または全体が、JP1/AJS から参照できないおそれがある。このときジョブの実行結果を確認するには、スプール内のジョブログを参照すること。

## (b) JP1/AJS から直接バッチジョブを実行していない場合、およびコマンドラインから起動している場合

ユーザが直接 taskkill コマンドなどでバッチジョブを終了させる必要がある。このとき、bjexec.exe プロセスだけ終了させなければならない。bjexecsub.exe プロセスを終了させた場合、一時ファイルの削除や世代ファイルの更新などの後処理がされない。

bjexec.exe プロセスのプロセス ID は bjexlslock コマンドの PPID の列に出力される。bjexlslock コマンドで bjexec.exe プロセスのプロセス ID を参照し、taskkill コマンドなどで終了させること。bjexlslock コマンドでプロセス ID を調べ、バッチジョブを強制終了する例を次に示す。

```
> bjexlslock
JOBID STATUS ELAPSED PID PPID JP1JobID JOBNAME
-----
000907 LOCK 00:01:23 3188 3128 0000001270 SAMPLE
> taskkill /F /PID 3128
```

ただし、bjexlslock コマンドで bjexec.exe プロセスのプロセス ID が出力されるのは、排他/共用制御が適用される DD 要素が指定されているジョブだけである。詳細は「[3.10 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。

それ以外のジョブについては、OS の機能などを利用してプロセス ID を調べること。

バッチジョブのジョブログをファイルにリダイレクトしている場合、bjexec.exe プロセスを taskkill コマンドなどで終了させると、バッチジョブのジョブログがファイルに最後まで出力されないことがある。Advanced Shell はデフォルトでジョブログをファイルにリダイレクトしているため、必要に応じて Advanced Shell の出力モードの設定を変更すること。

また、JP1/AJS から強制終了した場合、バッチジョブの親プロセスによってはバッチジョブが終了され一時ファイルの削除や世代ファイルの更新などの後処理がされないことがある。例えば、bat ファイルは終了されたとき子プロセスに何もしないため、バッチジョブが終了されることはない。Advanced Shell は終了されたとき子プロセスを終了させるため、ジョブステッププロセスを含めバッチジョブを終了させられる。

JP1/AJS から直接バッチジョブを実行しない場合は注意が必要である。

## (2) バッチジョブ実行システムがジョブステップのプロセスを終了させる方法

バッチジョブ実行システムは、次の方法でジョブステップのプロセスを強制終了する。

## (a) EXEC 要素の LANG 属性が Java の場合

cjkilljob コマンドを呼び出し、バッチアプリケーションを強制終了する。cjkilljob コマンドについては、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。

## (b) EXEC 要素の LANG 属性が Java 以外の場合

ジョブステップのジョブオブジェクトに対して Win32(R) API の TerminateJobObject 関数を実行し、ジョブステップのプロセスを強制終了する。TerminateJobObject 関数の仕様については、Windows の資料を参照のこと。

## 4.4.6 注意事項

- 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、標準エラー出力ファイル名に"NUL"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。
- UNIT 定義ファイルの sc パラメータにジョブ定義ファイルを指定する場合は、次に示す点に注意すること。
  - ジョブ定義ファイルに定義するジョブステッププログラムは、ウィンドウ画面を持たない。
  - bjexec コマンドのオプションは使用できない。オプションを使用する場合は、Windows のコマンドプロンプトから直接コマンドを実行すること。
  - 標準出力ファイル名、標準エラー出力ファイル名（追加書きを含める）は指定しない。
  - ジョブの終了判定は、最終ステップではなく各ジョブステップの最大値で判定する。
  - インタラクティブな環境を必要とするジョブは JP1/AJS から起動して実行できない。
  - 多数のステップが存在して長時間実行するジョブがある場合、JP1/AJS - Agent 側の多重度に対する設計を十分に行い、長時間の実行待ちが問題にならないように設計すること。
  - ジョブステッププログラムは、ジョブオブジェクトに登録できない (AssignProcessToJobObject が失敗する)。
  - ジョブコントローラプロセスがすでにジョブオブジェクトに関連づけられていた場合、ジョブステッププロセスの子孫プロセスのどれかが存在する状態でジョブを強制終了したとき次の制限事項がある。
    - ジョブステッププロセスの子プロセスは終了しない
    - スプールジョブディレクトリ名をジョブ識別子-ジョブ名に変更できない
  - EXEC 要素内の PGM 属性値に "\*" を指定した場合、スクリプトがジョブステッププロセスになるため、CDATA に記述したコマンドはジョブステッププロセスの子プロセスになる。

- Advanced Shell からジョブコントローラを起動した場合、ジョブコントローラプロセスはジョブオブジェクトに関連づけられる。
- ジョブステッププログラムは、標準入力を使用できない。
- ジョブステッププログラムが終了した場合、DD 要素に記述したファイルは後処理される。ただし、ジョブステッププログラムからさらに起動したプロセスがこのファイルを使用したままの状態だと、DELETE 指定などの後処理やスプールに出力した情報の印刷などが失敗する。この場合、エラーメッセージを見てファイルの削除などを実施すること。
- JP1/AJS の操作によって実行中のバッチジョブ実行システムのバッチジョブおよびジョブネットを強制終了した場合、JP1/AJS - View の [実行結果詳細] 画面から参照できるジョブログが途中で打ち切られることがある。強制終了したジョブの詳細を確認する場合は、スプールに出力されたジョブログも参照すること。

## 4.5 スプールジョブ転送の運用

スプールジョブ（帳票などの SYSOUT データファイル）は、ジョブ実行完了後にプリントサーバのディレクトリに転送（コピー）することで印刷される。プリントサーバが、ジョブコントローラ実行サーバと別のサーバの場合は Windows の共有ディレクトリを利用する。

### 4.5.1 スプールジョブ転送先の設定

スプールジョブは、出力クラスごとにバッチジョブ実行システムの設定ファイルで転送先を設定する。バッチジョブ実行システムの設定ファイルの変更は、bjex\_setup コマンド実行後にシステム管理者が、動作環境に作成された設定ファイルに対して、任意のテキストエディタを使用して行う。

```
<SysoutClass クラス名>  
[PRTCNTL スプールジョブ管理ファイル転送先ディレクトリ名]  
[PRTDATA SYSOUTデータファイル転送先ディレクトリ名]  
</SysoutClass>
```

### 4.5.2 帳票処理の概要

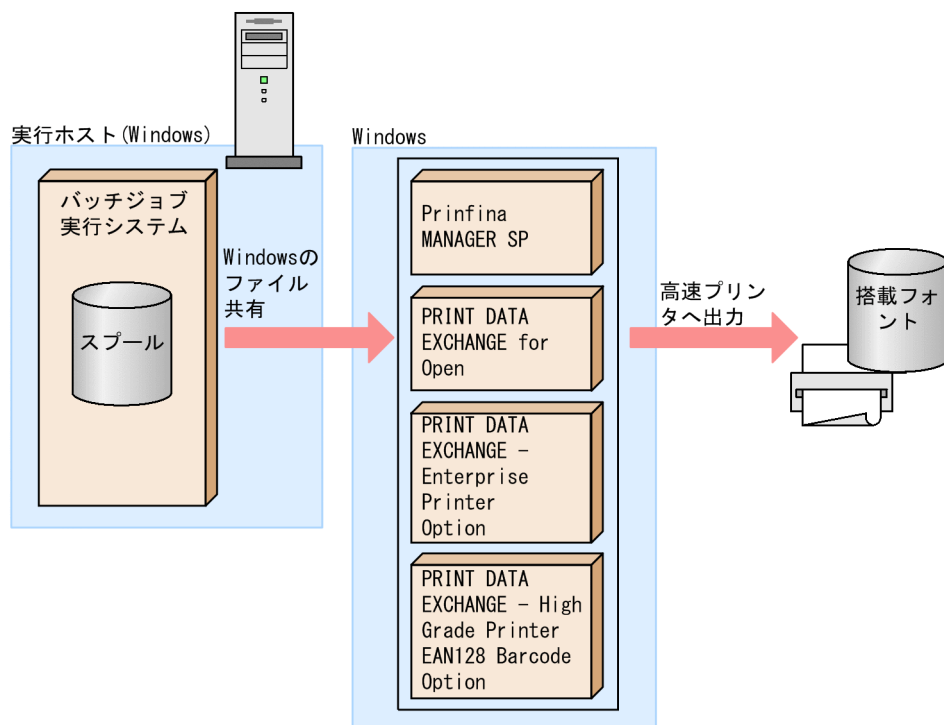
バッチジョブ実行システムは、PDE や Prinfinia MANAGER SP と連携して帳票データの印刷や PDF 変換を実行する。バッチジョブの実行終了後、スプールに出力された SYSOUT データファイルを出力クラスに従い、PDE が動作する印刷ホストに転送することで実行する。

#### (1) 高速プリンタへの印刷

Prinfinia MANAGER SP と連携して高速プリンタへ印刷を実行する。

Prinfinia 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 4-5 Prinfina MANAGER SP との連携による高速プリンタへの印刷の流れ

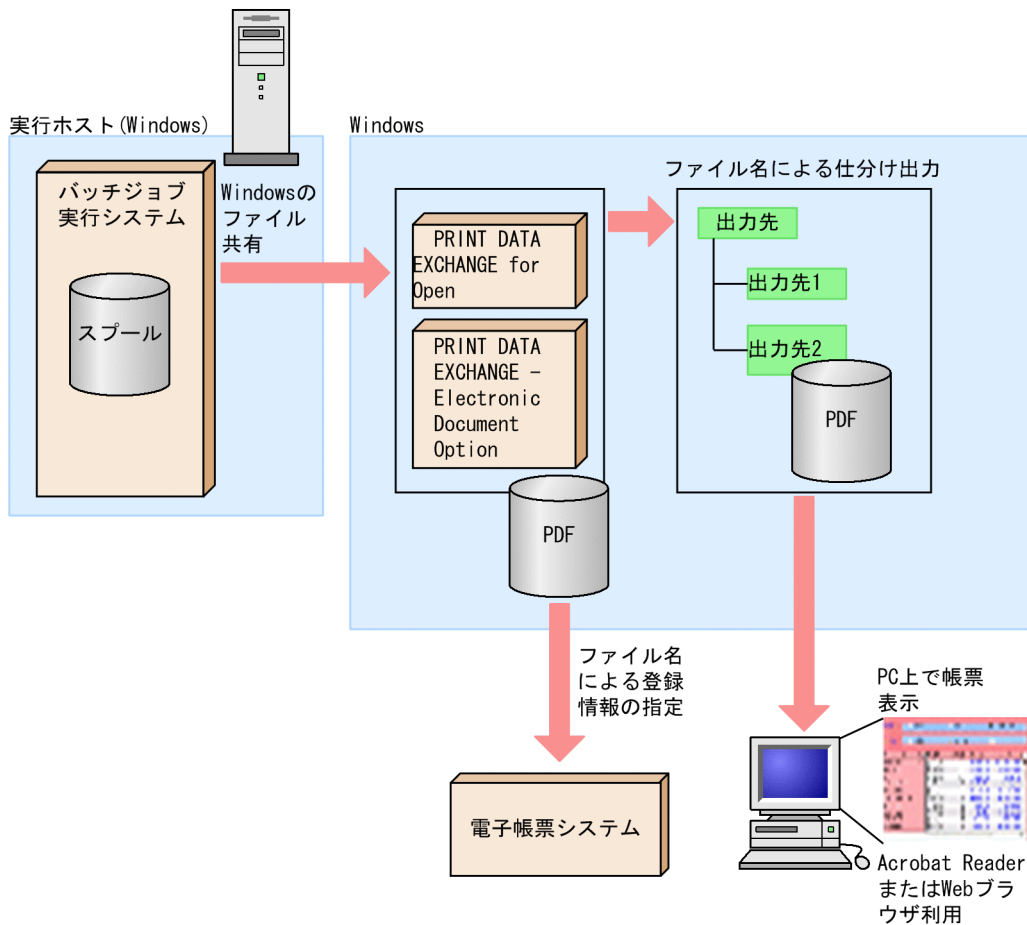


## (2) 電子帳票として出力

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 4-6 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

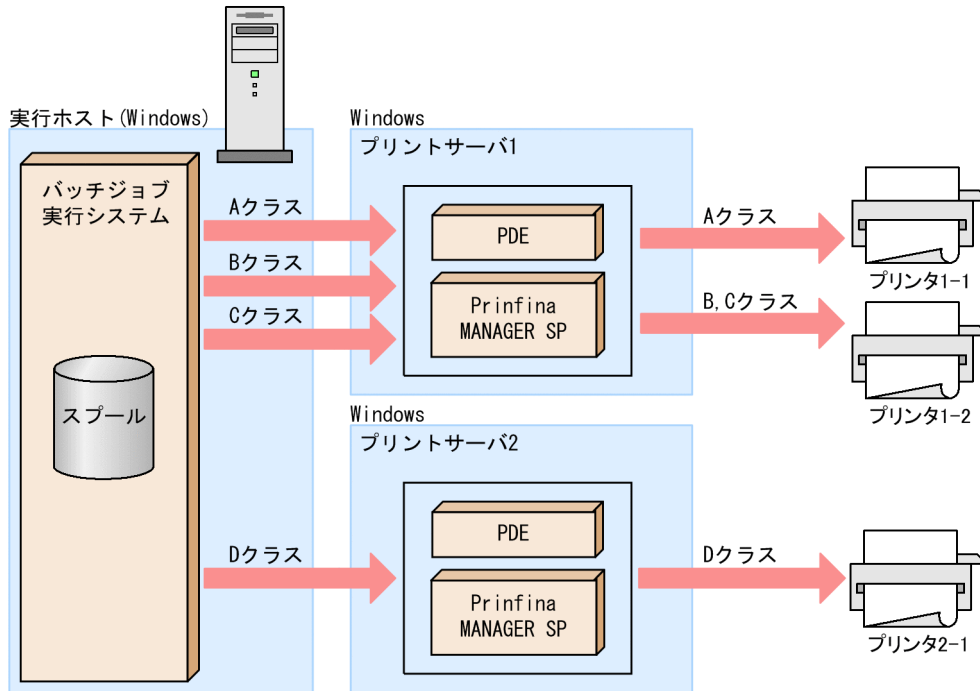


### 4.5.3 出カクラスによる制御

出カ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、この出カ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 4-7 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



- 設定ファイルに従い、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Prinfina 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。このため、出力保留は印刷しない SYSOUT データファイルに適用する。

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 4-1 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

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

2GB 以上の SYSOUT データファイル

## 4.5.5 スプールジョブの削除

出力保留属性の SYSOUT データファイルがあるスプールジョブは、プリントサーバへ転送後も削除されず、スプールに"ジョブ識別子-ジョブ名"のディレクトリとして残る。このため、bjexrmjob コマンドなど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 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。

## 4.5.7 スプールジョブ転送失敗時の注意事項

出力後保存属性がない SYSOUT データファイルは、印刷サーバに転送後スプールから削除する。転送処理で失敗した場合、スプールジョブはスプールに残るが、再度転送する手段はないため、必要ならジョブを再実行する。

スプールジョブ転送が失敗した場合でも、ジョブコントローラの終了コードは変更されない。転送処理の失敗を判断するためには、ジョブコントローラの終了コードではなく、実行ログまたは JP1/IM に出力される KAKC2104-E メッセージまたはイベントを監視すること。



## 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。

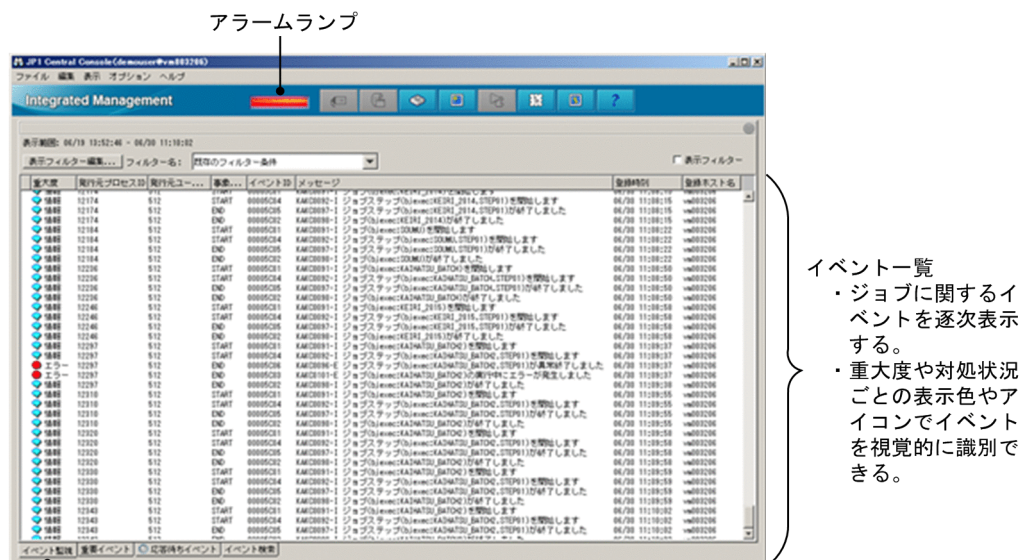
統合コンソールは、バッチジョブで発生した事象を JP1 イベントとして集め、時系列に表示することによって、システムを集中的に監視する。

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、「10.5 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 4-8 [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。

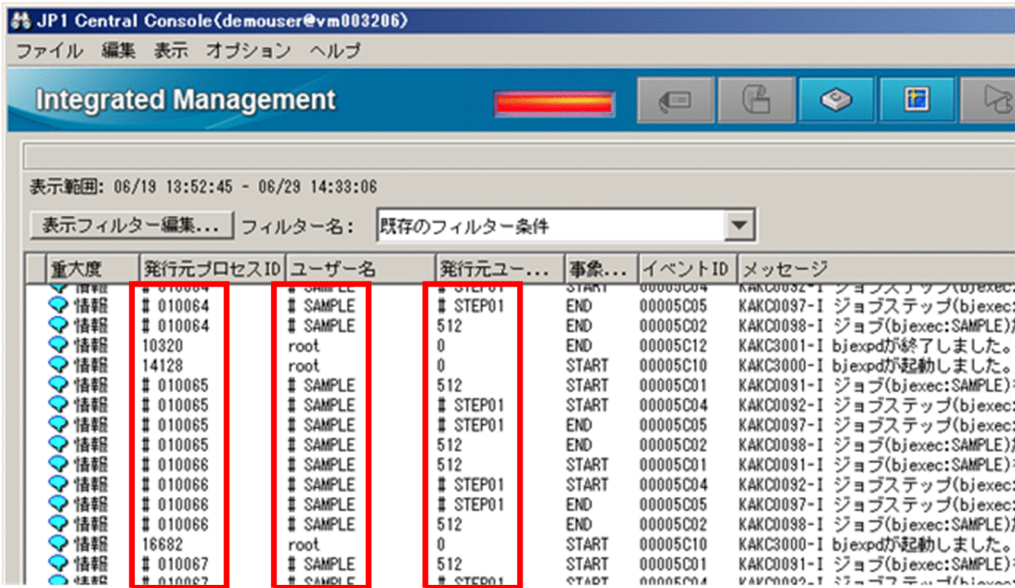
## 4.6.2 ジョブ名などの情報をイベント一覧に表示

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、「10.5 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 4-9 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



The screenshot shows the JP1 Central Console interface. The main window displays a table of events. Three columns are highlighted with red boxes: '発行元プロセスID' (containing job IDs like 010064), 'ユーザー名' (containing job names like SAMPLE), and '発行元ユーザー名' (containing step names like STEP01). Red arrows point from these boxes to labels below: 'ジョブ通し番号', 'ジョブ名', and 'ステップ名'.

重大度	発行元プロセスID	ユーザー名	発行元ユーザー名	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	# 010064	# SAMPLE	# STEP01	END	00005C02	KAKC0098-I ジョブ(bjexec:SAMPLE)?
情報	10320	root	0	END	00005C12	KAKC3001-I bjexpdが終了しました。
情報	14128	root	0	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	# 010065	# SAMPLE	# STEP01	END	00005C02	KAKC0098-I ジョブ(bjexec:SAMPLE)?
情報	# 010066	# SAMPLE	# STEP01	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	# STEP01	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	# STEP01	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

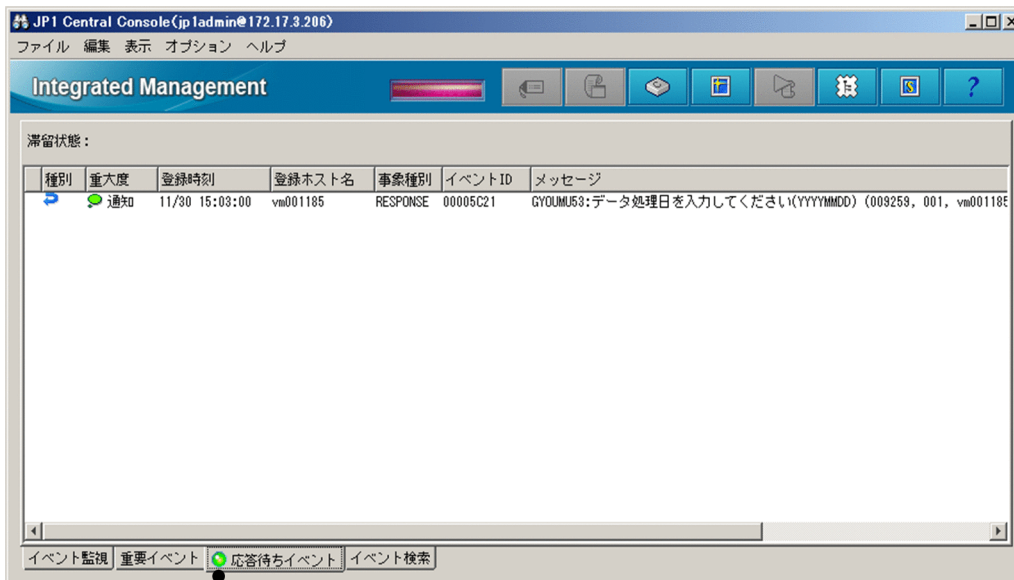
## 4.6.3 応答要求メッセージの監視

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、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 4-10 「応答待ちイベント」画面



タブランプ

## 4.7 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。

### 4.7.1 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

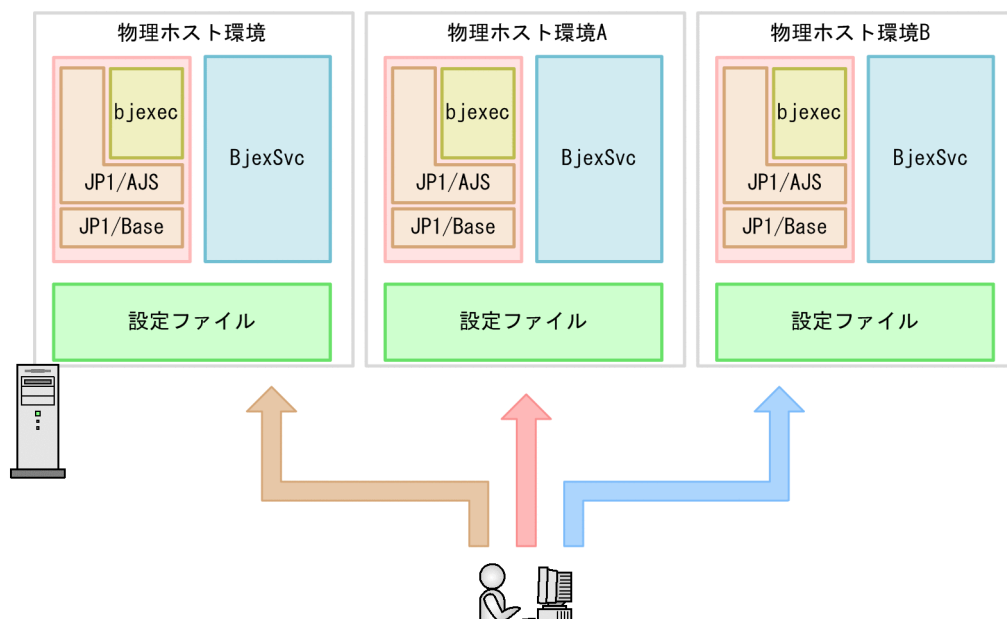
バッチジョブ実行システムには物理HOST環境と論理HOST環境の2種類が存在する。

- 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、`BjexSvc` サービスと連携する。  
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、`BjexSvc_論理HOST名` サービスと連携する。  
ここで、`bjex_論理HOST名.conf` と `BjexSvc_論理HOST名` サービスの論理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。



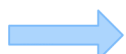
バッチジョブ実行システムの物理HOST環境は1つであるが、バッチジョブ実行システムの論理HOST環境は複数構築できる。複数のバッチジョブ実行システムの環境の構成例を次の図に示す。

図 4-11 複数のバッチジョブ実行システム環境

バッチジョブ実行システムが稼働するサーバ



(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。

## 4.7.2 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、同じ物理ホスト上で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

<All Users のプロファイルのドキュメントディレクトリの場所>¥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) サービスの追加

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バッチジョブ実行システムの環境ごとにサービスが必要となる。そのため、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、BjexSvc\_論理ホスト名サービスを追加登録する必要がある。登録方法の詳細については、「7. コマンド」の「bjexsvc (バッチジョブ実行システムのサービスの登録およびサービス起動環境の設定 (管理者用))」を参照のこと。

### 4.7.3 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<All Users のプロファイルのドキュメントディレクトリの場所>¥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の場合、連携する JP1 が論理ホスト環境か物理ホスト環境かによって、JP1\_HOSTNAME パラメータに指定すべきホスト名が異なる。詳細は、「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する。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「4.7.2 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。

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、「4.9 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。

## 4.8 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。

### 4.8.1 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アドレスで通信をする。

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。

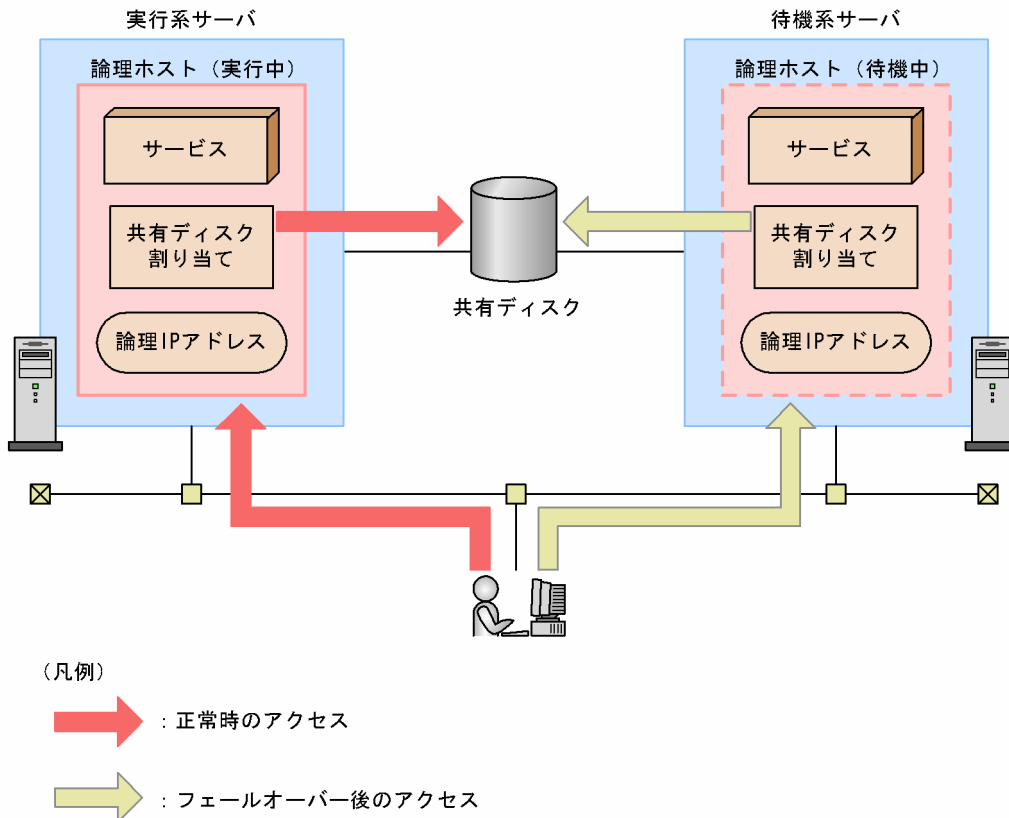
表 4-2 クラスタシステム構成のJP1の論理ホストの構成要素

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。実行系のサーバで障害が発生すると、待機系の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、実行系のサーバで障害が発生した場合、待機系の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。実行系の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のサーバが引き継ぐ。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 4-12 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。

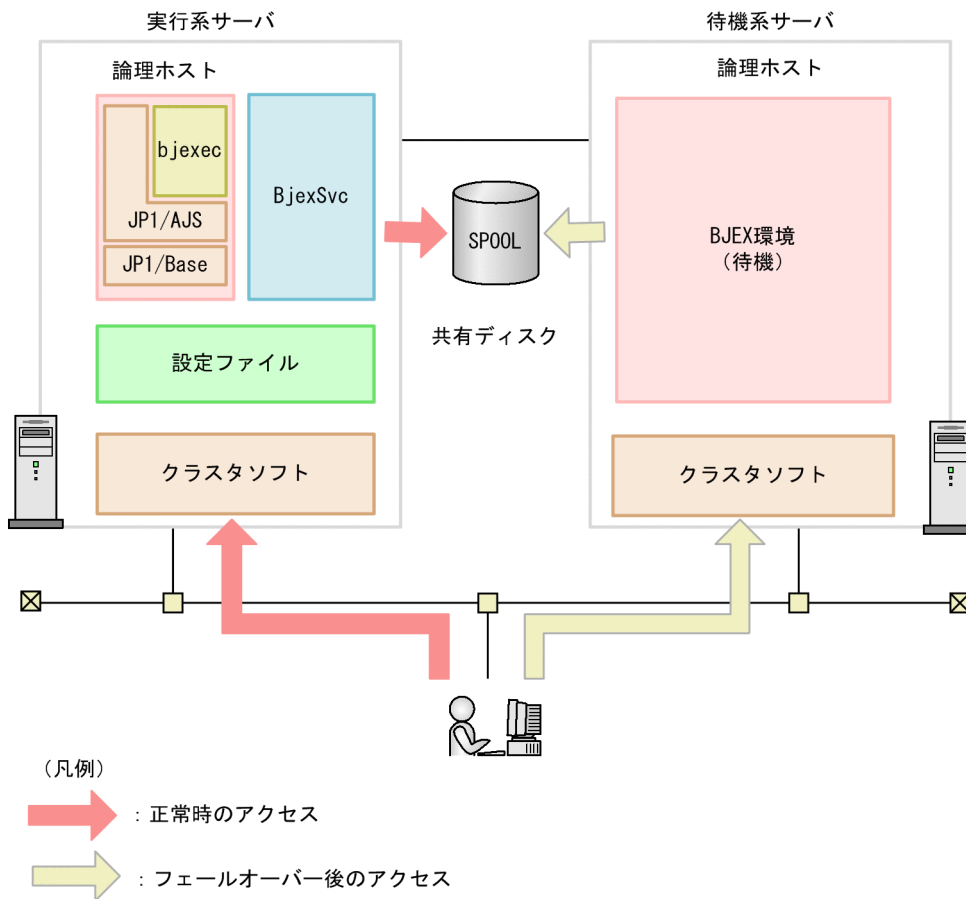
## (2) バッチジョブ実行システムのクラスタ運用の概要

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と論理ホスト環境のどちら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 4-13 バッチジョブ実行システムでのクラスタ運用の概要



## 4.8.2 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

バッチジョブ実行システムの論理ホスト名の長さの上限は JP1 と同様 196 文字であるが、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。詳細については 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。論理ホスト環境（共有ディスクおよび論理 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的环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。

## 4.8.3 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的环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 4-3 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

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、クラスタソフトにバッチジョブ実行システムのサービスを登録すること。クラスタソフトに登録するサービスを次の表に示す。

表 4-4 クラスタソフトに登録するサービス

クラスタシステムで運用するバッチジョブ実行システム	クラスタソフトに登録するサービス
物理ホスト	BjexSvc サービス
論理ホスト	BjexSvc_論理ホスト名サービス

## (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を次のファイルと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

## 4.8.4 クラスタ運用に関する注意事項

- 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、「[4.9 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。

## 4.9 バッチジョブ実行システムの運用時の注意事項

---

### 4.9.1 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、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、「3.7.4 世代ファイルおよび世代データ群の削除」に示された方法に従って、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。

### 4.9.2 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。

- インストール先フォルダ以下のすべて
- <All Users のプロファイルのアプリケーションデータディレクトリの場所>¥Hitachi¥BJEX 以下のすべて
- <All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX 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

# 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 Prinfinia 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。

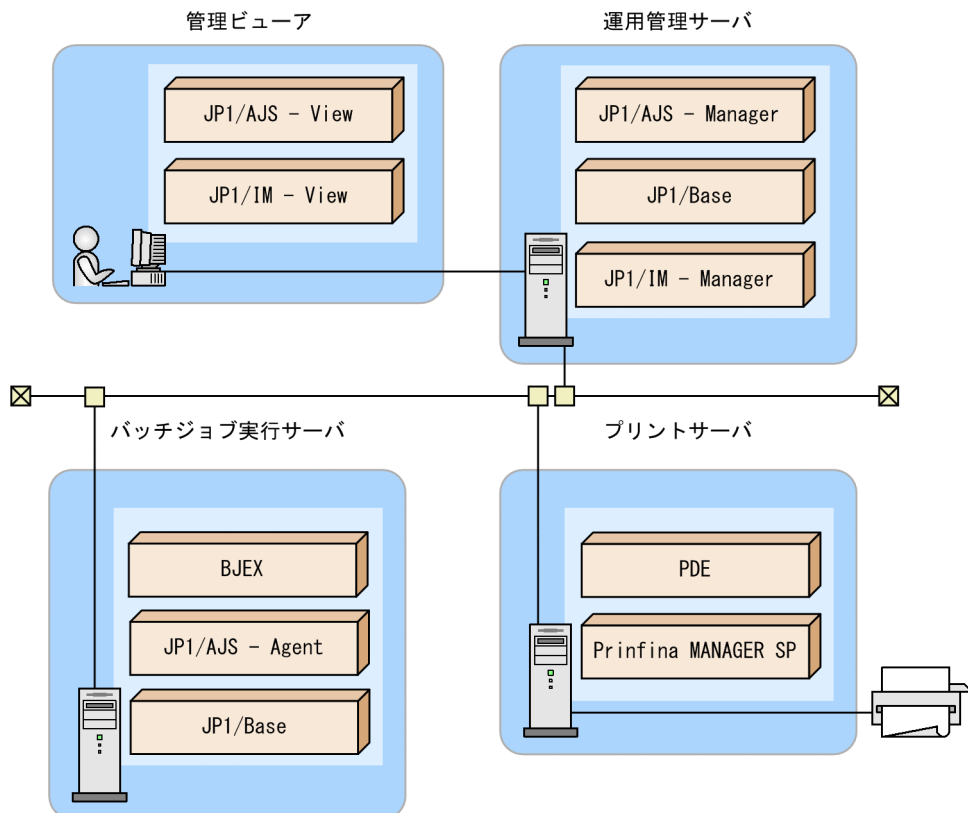
## 5.1 バッチジョブ実行システムの使用概要

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 JP1/AJS によるジョブスケジュール

---

ここでは、バッチジョブ実行の JP1./AJS によるジョブスケジュールについて説明する。ジョブネットワークの定義方法などの JP1/AJS の操作については、JP1 のマニュアルを参照のこと。

### 5.2.1 概要

バッチジョブ実行システムのバッチジョブは、JP1/AJS から bat ファイルや Advanced Shell 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。

#### (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。

- PC ジョブ  
実行するジョブを PC 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、PC 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 (jpqjobsub コマンド) を使ってジョブをキューに直接登録できる。

#### (2) コマンドラインから起動する場合

コマンドラインからバッチジョブ実行システムを起動する場合、次の注意事項がある。

- 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。
- 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。

### 5.2.2 バッチジョブ実行システムのバッチジョブの実行

バッチジョブ実行システムのバッチジョブを実行するには、PC ジョブまたはサブミットジョブを使用する。

#### (1) PC ジョブの実行

[詳細定義 - [PC 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、PC 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

- [実行ファイル名]

バッチジョブ起動コマンド名として、ジョブ定義 XML ファイルまたはジョブコントローラである bjexec コマンドを指定する。

指定例を次に示す。

[ジョブ定義 XML ファイルを直接指定する場合]

```
"C:¥Program Files¥HITACHI¥BJEX¥sample¥sample.bjx"
```

[bjexec コマンドを指定する場合]

```
"C:¥Program Files¥HITACHI¥BJEX¥bin¥bjexec.exe"
```

- [パラメーター]

[実行ファイル名] に bjexec コマンドを指定した場合は、[パラメーター] にジョブ定義 XML ファイルを指定すること。[実行ファイル名] にジョブ定義 XML ファイルを直接指定した場合は、[パラメーター] に再度ジョブ定義 XML ファイルを指定する必要はない。

指定例を次に示す。

```
"C:¥Program Files¥HITACHI¥BJEX¥sample¥sample.bjx"
```

- [実行優先順位]

ジョブコントローラおよびユーザプログラムの優先順位を指定する。ユーザプログラムにこの指定を有効にするには、設定ファイルの SET\_PGM PRIORITY パラメータに YES を指定すること。実行優先順位は 3 以上を指定すること。詳細については JP1/AJS のマニュアルを参照のこと。

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 - [PC Job]] 画面の [定義] タブの指定例

ユニット名	sample
コメント	
実行ホスト	
<b>定義</b>   転送ファイル   属性	
実行ファイル名	Program Files\HITACHI\BJEX\sample\sample.bjx
パラメーター	
環境変数	
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 <input type="text" value="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。

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－ [PC Job]] 画面の [属性] タブの指定例

詳細定義-[PC 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、jppqjobsub 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。

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。

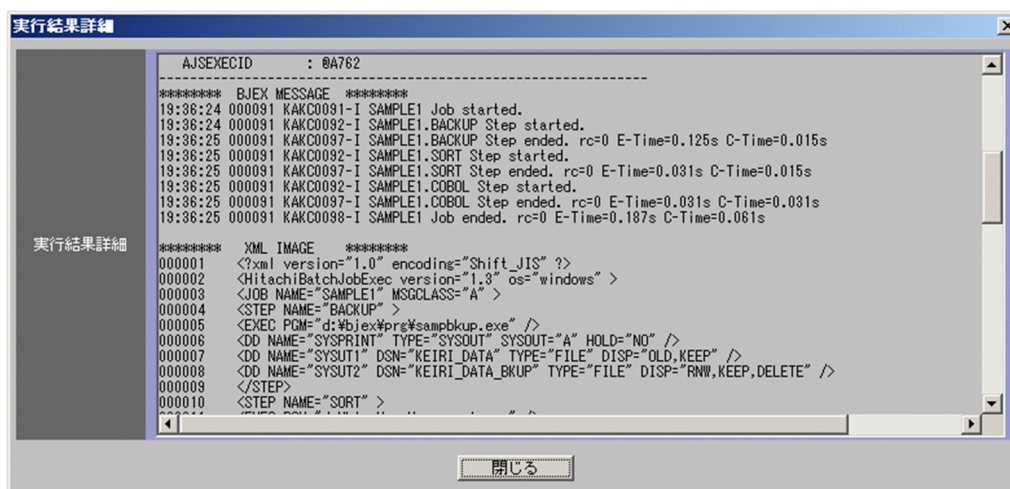
```
-sc "%C:\Program Files\HITACHI\BJEX\sample\sample.bjx"
```

## 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



The screenshot shows a window titled "実行結果詳細" (Execution Result Details) with a close button in the top right corner. The main content area displays the following text:

```
AJSEXCID      : @A782
-----
*****  BJEX MESSAGE  *****
18:36:24 000091 KAKC0091-I SAMPLE1 Job started.
18:36:24 000091 KAKC0092-I SAMPLE1.BACKUP Step started.
18:36:25 000091 KAKC0097-I SAMPLE1.BACKUP Step ended. rc=0 E-Time=0.125s C-Time=0.015s
18:36:25 000091 KAKC0092-I SAMPLE1.SORT Step started.
18:36:25 000091 KAKC0097-I SAMPLE1.SORT Step ended. rc=0 E-Time=0.031s C-Time=0.015s
18:36:25 000091 KAKC0092-I SAMPLE1.COBOL Step started.
18:36:25 000091 KAKC0097-I SAMPLE1.COBOL Step ended. rc=0 E-Time=0.031s C-Time=0.031s
18:36:25 000091 KAKC0098-I SAMPLE1 Job ended. rc=0 E-Time=0.187s C-Time=0.061s

*****  XML IMAGE  *****
000001 <?xml version="1.0" encoding="Shift_JIS" ?>
000002 <HitachiBatchJobExec version="1.3" os="windows" >
000003 <JOB NAME="SAMPLE1" MSGCLASS="A" >
000004 <STEP NAME="BACKUP" >
000005 <EXEC PGM="d:\bjex\org\sample\bkup.exe" />
000006 <DD NAME="SYSPRINT" TYPE="SYSOUT" SYSOUT="A" HOLD="NO" />
000007 <DD NAME="SYSUT1" DSN="KEIRI_DATA" TYPE="FILE" DISP="OLD,KEEP" />
000008 <DD NAME="SYSUT2" DSN="KEIRI_DATA_BKUP" TYPE="FILE" DISP="RNW,KEEP,DELETE" />
000009 </STEP>
000010 <STEP NAME="SORT" >
```

## 5.3 Prinfina MANAGER SP および PDE との連携による帳票印刷

ここでは、Prinfina 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 概要

帳票印刷は、Prinfina 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、出力クラスに従い、Prinfina 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 印刷データ転送

バッチジョブ実行システムは、ジョブ実行時に共用ファイル経由でプリントサーバに印刷データを転送する。

#### (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>
  :
  PRTCNTL   スプールジョブ管理ファイル転送先ディレクトリ名
  PRTPDATA  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>
  :
  PRTCNTL   ¥¥PRTSERVER1¥PDE¥prtcntl
  PRTPDATA  ¥¥PRTSERVER1¥PDE¥prtpdata
  :
</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) 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。

Prinfinia 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、Prinfinia 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 PRTCNTL 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、Prinfinia 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 Prinfinia 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。これらのディレクトリは共有ディレクトリにする。

- **ディレクトリアクセス権の設定**

バッチジョブ実行システムから書き込みができるようにバッチジョブを実行する全ユーザに対して書き込み可能な権限を設定する。

- **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

### 5.3.3 SYSOUT 属性の PDE および Prinfinia 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,Prinfinia 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 Prinfinia 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る必要がある属性について次の表に示す。

表 5-1 プログラマが留意する必要がある SYSOUT 属性の PDE, Prinfinia 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	Prinfinia 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 5-2 の項番 1 へ	
	FCB 名称	FCB	○	○	表 5-2 の項番 2 へ	
	書式番号	FORMS	○	○	表 5-2 の項番 3 へ	
	漢字モード	MODE	○	○	表 5-2 の項番 4 へ	
	ANSI 制御 コード	RECFM	○	×	表 5-2 の項番 5 へ	
	文字配列 テーブル順 序番号属性	OPTCDJ	○	×	表 5-2 の項番 6 へ	
	複写回数	COPIES	○	○	表 5-2 の項番 7 へ	
	グループ複 写回数	COPYG	○	○	表 5-2 の項番 8 へ	
	オーバレ イ名	OVERLAY	○	○	表 5-2 の項番 9 へ	
	適用枚数	OVERC	○	○	表 5-2 の項番 10 へ	
	デフォルト 機能キャラ クタイメー ジ名	FCDFLT	○	○	表 5-2 の項番 11 へ	
	出力先識 別子	DEST	○	○	表 5-2 の項番 12 へ	



(凡例)

○：指定できる。

×：指定できない。

表 5-2 プログラマが留意する必要がある SYSOUT 属性の PDE, Prinfinia MANAGER SP での扱い (扱いについての詳細)

項番	PDE での扱い	Prinfinia 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イメージ名 (デフォルト機能キャラクタイメージの適用)	—
12	出力先コード (対象ジョブ条件に使用)	—

(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る必要がある属性について次の表に示す。

表 5-3 運用管理者が留意する必要がある SYSOUT 属性の PDE, Prinfinia 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	Prinfinia 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	表 5-4 の項番 1 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			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)	×	×	×	当てるジョブ 識別子	表 5-4 の項番 1 へ	
	ジョブ名	NAME	○	×	×	表 5-4 の項番 2 へ	
	ユーザ名	×	×	×	環境変数 JP1_USERN AME または OS ユーザ名	表 5-4 の項番 3 へ	
	仕分け番号	ROOM	○	×	×	表 5-4 の項番 4 へ	
	ホスト名	×	×	×	設定ファイル の JP1_HOSTN AME パラ メータで設定 したホスト名	表 5-4 の項番 5 へ	
D D	DD 名称	NAME	○	×	×	表 5-4 の項番 6 へ	
	出力クラ ス名	SYSOUT	○	○	×	表 5-4 の項番 7 へ	
	注釈情報	CMT	○	×	×	表 5-4 の項番 8 へ	
ジョ ブ ス テッ プ	ステップ名	NAME	○	×	×	表 5-4 の項番 9 へ	
	ステップ 番号	×	×	×	ステップ通し 番号*	表 5-4 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 5-4 運用管理者が留意する必要がある 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

項番	PDE での扱い	Prinfina MANAGER SP での扱い
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	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。

印刷を一時保留したい場合は、Prinfina MANAGER SP の機能で印刷保留することを推奨する。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ため、印刷しない SYSOUT データファイルに適用する。

## 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。COBOL プログラムには、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。

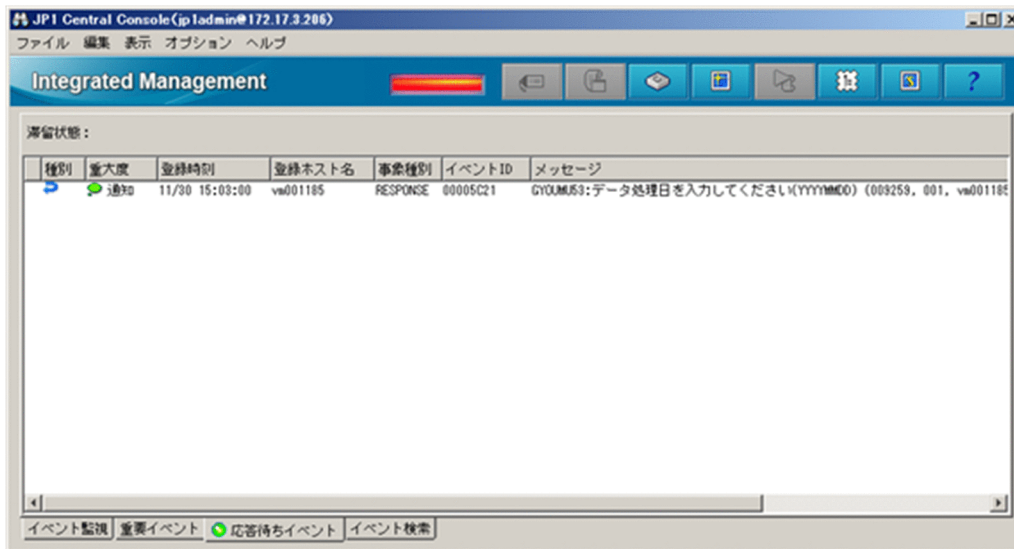
## 5.4.4 応答入力

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。

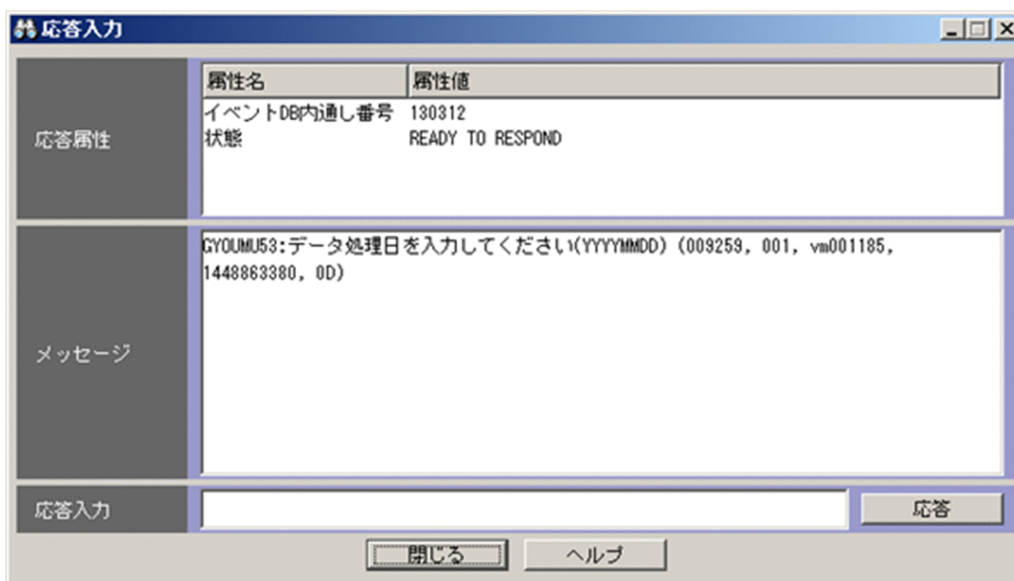
図 5-5 [応答待ちイベント] 画面の例



[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 (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

#### (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に出力される。

```
seqnum=1, date=2008-12-01T10:37:07.669+09:00, pgmid=Exec_c, jobid=12, pid=5832,
msgid=KAKC0004-1, msg="Jobid=000012, JP1NBQSQueueName=¥¥io10212¥10.209.40.12,
JP1JobID=0000000073"
seqnum=2, date=2008-12-01T10:37:07.685+09:00, pgmid=Exec_c, jobid=12, pid=5832,
msgid=KAKC0091-1, msg="SAMPLE1 Job started."
seqnum=3, date=2008-12-01T10:37:07.685+09:00, pgmid=Exec_c, jobid=12, pid=5832,
msgid=KAKC0092-1, msg="SAMPLE1.BAKCUP Step started."
seqnum=4, date=2008-12-01T10:37:19.700+09:00, pgmid=Exec_c, jobid=12, pid=5832,
msgid=KAKC0097-1, msg="SAMPLE1.BAKCUP Step ended. rc=0 E-Time=12.015s C-Time=0.031s"
seqnum=5, date=2008-12-01T10:37:19.700+09:00, pgmid=Exec_c, jobid=12, pid=5832,
msgid=KAKC0092-1, msg="SAMPLE1.SORT Step started."
seqnum=6, date=2008-12-01T10:37:31.716+09:00, pgmid=Exec_c, jobid=12, pid=5832,
msgid=KAKC0097-1, msg="SAMPLE1.SORT Step ended. rc=0 E-Time=12.016s C-Time=0.015s"
seqnum=7, date=2008-12-01T10:37:31.716+09:00, pgmid=Exec_c, jobid=12, pid=5832,
msgid=KAKC0092-1, msg="SAMPLE1.COBOL Step started."
seqnum=8, date=2008-12-01T10:37:43.747+09:00, pgmid=Exec_c, jobid=12, pid=5832,
msgid=KAKC0097-1, msg="SAMPLE1.COBOL Step ended. rc=0 E-Time=12.031s C-Time=0.015s"
seqnum=9, date=2008-12-01T10:37:43.747+09:00, pgmid=Exec_c, jobid=12, pid=5832,
msgid=KAKC0098-1, msg="SAMPLE1 Job ended. rc=0 E-Time=36.062s C-Time=0.061s "
```

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が出力される。

Exec\_c：ジョブコントローラ

Msg\_lib：メッセージ制御機能を使用した COBOL

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

トレース<sup>※</sup>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。また、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) に出力される。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。

### 注※

- トレースファイルは、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。
- トレースファイルは、最初にバッチジョブ実行システムを実行したユーザを所有者として作成される。
- 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。

### (1) システム情報

採取が必要なシステム情報の採取方法を次に示す。

## (a) Windows Server 2008 R2 の場合

1. [スタート] メニューから [ファイル名を指定して実行] を選択する。
2. [名前] に [msinfo32] を指定し, [OK] ボタンをクリックする。  
[システム情報] 画面が表示される。
3. 左ペインから [システムの概要] を選択する。
4. [ファイル] メニューから [エクスポート] を選択し, システム情報をテキストファイルで保存する。
5. Windows の [コントロールパネル] から [システム] を選択する。  
[システム] 画面が表示される。
6. 左ペインから [システムの詳細設定] を選択する。  
[システムのプロパティ] 画面が表示される。
7. [詳細設定] タブをクリックする。
8. [起動と回復] の [設定] ボタンをクリックする。  
[起動と回復] 画面が表示される。
9. [デバッグ情報の書き込み] で設定されているダンプファイルの出力先を確認し, ファイルを採取する。

## (b) Windows Server 2012/Windows Server 2012 R2 の場合

1. デスクトップで, Windows ロゴキーを押しながら R キーを押して, [ファイル名を指定して実行] ダイアログボックスを開く。
2. [名前] に [msinfo32] を指定し, [OK] ボタンをクリックする。  
[システム情報] 画面が表示される。
3. 左ペインから [システムの概要] を選択する。
4. [ファイル] メニューから [エクスポート] を選択し, システム情報をテキストファイルで保存する。
5. Windows の [コントロールパネル] から [システム] を選択する。  
[システム] 画面が表示される。
6. 左ペインから [システムの詳細設定] を選択する。  
[システムのプロパティ] 画面が表示される。
7. [詳細設定] タブをクリックする。
8. [起動と回復] の [設定] ボタンをクリックする。  
[起動と回復] 画面が表示される。

9. [デバッグ情報の書き込み] で設定されているダンプファイルの出力先を確認し、ファイルを採取する。

## (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。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 6-2 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	「バッチジョブ実行システムインストールディレクトリ」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設定ファイル	バッチジョブ実行システムで使用している設定ファイル
実行ログ	設定ファイルの LOG_DIR パラメータで指定されたディレクトリ下のすべてのファイル
トレース	設定ファイルの TRACE_DIR パラメータで指定されたディレクトリ下のすべてのファイル
スプールの情報	設定ファイルの SPOOL_DIR パラメータで指定されたディレクトリ下の、次の資料 • トラブルが発生したジョブのスプールジョブ管理ファイル (sysout.ini) • ディレクトリとファイル一覧 (隠しファイル属性のものも含む)

## (3) バッチジョブ実行システムのプロセス一覧

タスクマネージャからバッチジョブ実行システムの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 bjexec.exe および bjexecsub.exe 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。

- ジョブ定義 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, および PDE 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。

## 6.3 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おそれがあるため、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。

## 6.4 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た環境でバッチジョブが異常終了する場合の確認手順

---

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た環境でバッチジョブが異常終了する場合、次に示す手順で要因を確認する。

《バッチジョブや bjexlslock コマンドおよび bjexlsexjob コマンドがエラーメッセージを出力して異常終了した場合》

出力されたエラーメッセージに従って対処する。

《BjexSvc サービスが停止している場合》

1. BjexSvc サービスが停止する要因となったイベントを、アプリケーションのイベントログから検出する（イベントの種類が"エラー"で、かつイベントソースが"BjexSvc"であるイベントを検出する）。
2. 検出したイベントのプロパティの説明に表示されているメッセージに従って対処する。



## 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、リトライ回数を 3 回とし、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 BjexSvc サービスを再起動し、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 BjexSvc サービスを再起動し、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。



```
>bjexlsmgs
MESSAGE-NO STATUS JOBID STEP DATE/TIME MESSAGE or RESPONSE
[0000010001] [Wait] 000001 001 15/12/01 11:46:40 msg=KCCC2000R-I FROM CONSOLE指定のAC
CEPT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 bjexlsmgs 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

```
>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 bjexlsmgs コマンドで表示される MESSAGE-NO を指定
- -d オプションを指定

```
>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。
- BjexSvc サービスを停止した。

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。

# 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。

$\Delta_0$ コマンド名 [ $\Delta_1$ -オプション名 [ $\Delta_1$ 値] ] … [ $\Delta_1$ -オプション名 [ $\Delta_1$ 値] ] [任意名]

(凡例)

$\Delta_0$  : 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。

$\Delta_1$  : 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。オプションの前に任意名を指定した場合は、指定内容をすべて任意名として処理する。
- オプションを複数指定する場合、指定順序は任意である。
- 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

コマンド名	機能概要	対象ユーザ	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。	バッチジョブ実行システム のユーザ	<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>%bin
bjexec	バッチジョブを実行する。		
bjexgdg	世代データ群を生成および更新する。		
bjexlsexjob	実行中のジョブ情報を表示する。		
bjexlsjob	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。		
bjexlsmmsg	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。		
bjexrmjob	スプールジョブを削除する。	バッチジョブ実行システム の運用管理者	<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 >%sbin
bjexsvc	バッチジョブ実行システムのサービスの登録およびサービス起動環境の設定をする。		
bjex_setup	バッチジョブ実行システムの動作環境を作成する。		

# bjexchmsg (応答待ちメッセージへの応答入力 (一般ユーザ用))

## 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) >((1~196))

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 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 { [-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

### -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。

## 注意事項

- 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、BjexSvc\_論理ホスト名サービスではなく、BjexSvc サービスが作成した排他/共用制御の情報および実行中のジョブ情報を参照する。
- 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 C:¥xml¥sample.bjx
```



# 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 [-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)]] |  
         [-v]
```

## 機能

世代データ群の生成および更新を行う。

## 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。

**-l エントリ数~<数字>((1~255)) 《255》**

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、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する (自動消去機能: SCRATCH)。

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 (データ保存機能: NOSCRATCH)。世代ファイルに与える 4 桁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>((1~247 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一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を表示する。相対世代番号を指定した場合、特定の世代ファイル名だけを表示する。

## 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。

-v

バージョン情報を表示する。

## 注意事項

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。del 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

## 使用例

- 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
```

- 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)"
```

## 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
```

```

GDG BASE ----- bjpgdg                ... (1)
ATTRIBUTES
LIMIT ----- 10 SCRATCH                ... (2)
INFILE ----- 4                        ... (3)
ASSOCIATIONS
( 0) -- bjpgdg.G0004                    ... (4)
(-1) -- bjpgdg.G0003                    ... (5)
(-2) -- bjpgdg.G0002                    ... (6)
(-3) -- bjpgdg.G0001                    ... (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可能な世代ファイル数を"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だけ表示する。

- 世代データ群 bjpgdg の管理情報と相対世代番号 (-1) の世代ファイル名を表示する。

```
bjexgdg -i "bjgdg(-1)"
```

```

GDG BASE ----- bjpgdg                ... (1)
ATTRIBUTES
LIMIT ----- 10 SCRATCH                ... (2)
INFILE ----- 4                        ... (3)
ASSOCIATIONS
(-1) -- bjpgdg.G0003                    ... (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可能な世代ファイル数を"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、世代ファイル名を" (相対世代番号) - -"に続いて表示する。

# 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) >((1～196))

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 JP1 の論理ホスト名を指定する。論理ホスト名を指定することで、設定ファイル (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=2015/03/09 TIME=11:04:51									
JOBID	JP1JOBID	JOBSTART	STEPSTRT	JOBNAME	STEPNAME	PROCSTEP	PPID	PID	STEP PID
007953	0000004916	11:02:00	11:04:50	SAMPLE_JOB01	STEP03	PROC01_STEP1	6880	3944	2840
007954	0000004917	11:03:22	11:04:46	SAMPLE_JOB02		PROC01_STEP1	1152	5220	1836
007955	0000004918	11:03:33	11:04:06	BATCH001	STEP2		756	5368	3680
007956		11:04:07	11:04:51	JOB0091a	SORT3		1348	6984	6392
007957		11:04:11	11:04:46	PRINT_JOB	print_a2	PROC01_PRINT	6140	6516	5980

### 出力項目の説明

- JOBID  
ジョブ識別子 (6 桁)
- JP1JOBID  
JP1 ジョブ識別子 (10 桁。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無い場合は空白)
- 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無い場合は空白)
- 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 桁)
- PID  
ジョブコントローラプロセス (bjexecsub) のプロセス番号 (10 進数, 10 桁)
- STEP PID  
ジョブステッププロセスのプロセス番号 (10 進数, 10 桁)
- 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=2015/03/09 TIME=11:04:59								
JOBID	JP1JOBID	JOBSTART	STEPSTRT	JOBNAME	STEPNAME	C-TIME	PRI	JP1PRI
007953	0000004916	11:02:00	11:04:50	SAMPLE_JOB01	STEP03	104.954	2	32
007954	0000004917	11:03:22	11:04:46	SAMPLE_JOB02		16.987	0	64
007955	0000004918	11:03:33	11:04:06	BATCH001	STEP2	21.293	0	64
007956		11:04:07	11:04:51	JOB0091a	SORT3	27.252	2	
007957		11:04:11	11:04:46	PRINT_JOB	print_a2	17.689	2	

### 出力項目の説明

- JOBID  
ジョブ識別子 (6 桁)
- JP1JOBID  
JP1 ジョブ識別子 (10 桁。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無い場合は空白)
- C-TIME  
実行中のジョブが使用した CPU 時間。6 桁の秒 (約 277 時間) と 3 桁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。
  - GetProcessTimes 関数で求めたカーネルモードでのプロセス動作時間とユーザーモードでのプロセス動作時間の和で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。
  - GetPriorityClass 関数で求めた優先順位クラスに対応した値。

GetPriorityClass 関数で求めた優先順位クラス	表示する値
IDLE_PRIORITY_CLASS	0
BELOW_NORMAL_PRIORITY_CLASS	1

## 7. コマンド

GetPriorityClass 関数で求めた優先順位クラス	表示する値
NORMAL_PRIORITY_CLASS	2
ABOVE_NORMAL_PRIORITY_CLASS	3
HIGH_PRIORITY_CLASS	4
REALTIME_PRIORITY_CLASS	5
上記以外	?

- 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=2015/03/09 TIME=11:05:10
JOBID          : 007953
JOB NAME       : SAMPLE_JOB01
JOB START TIME : 2015/03/09 11:02:00
JOB C-TIME     : 110.476
STEP NAME      : STEP03
PROC-STEP NAME : PROC01_STEP2
STEP START TIME : 2015/03/09 11:05:07
STEP C-TIME    : 1.263
PGM            : c:\users\bjexuser\bin\sample.exe
XML FILE       : c:\users\bjexuser\bjex\sample\xml\sample1.bjx
PPID          : 6880
PID           : 3944
STEP PID      : 6064
JP1JOBID      : 0000004916
JP1PRI        : 32
PRI           : 2
```

### 出力項目の説明

- JOBID  
ジョブ識別子 (6 桁)
- JOB NAME  
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 桁の秒 (約 277 時間) と 3 桁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。

- GetProcessTimes 関数で求めたカーネルモードでのプロセス動作時間とユーザーモードでのプロセス動作時間の和で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無い場合は空白)
- 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無い場合は空白)
- 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 桁の秒 (約 277 時間) と 3 桁のミリ秒で表示する。最大値を超えた場合 999999.999 となる。
  - GetProcessTimes 関数で求めたカーネルモードでのプロセス動作時間とユーザーモードでのプロセス動作時間の和で求めた参考値である。
- PGM  
実行プログラム名称 (最大 1023 桁)
- 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 桁)
- PID  
ジョブコントローラプロセス (bjexecsub) のプロセス番号 (10 進数, 10 桁)
- STEP PID  
ジョブステッププロセスのプロセス番号 (10 進数, 10 桁)
- JP1JOBID  
JP1 ジョブ識別子 (10 桁。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。
  - GetPriorityClass 関数で求めた優先順位クラスに対応した値



GetPriorityClass 関数で求めた優先順位クラス	表示する値
IDLE_PRIORITY_CLASS	0
BELOW_NORMAL_PRIORITY_CLASS	1
NORMAL_PRIORITY_CLASS	2
ABOVE_NORMAL_PRIORITY_CLASS	3
HIGH_PRIORITY_CLASS	4
REALTIME_PRIORITY_CLASS	5
上記以外	?

## 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ずにジョブの情報だけが表示される。
- 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。
- 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) >((1~196))

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 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。

詳細表示では実行済みのジョブの情報だけを表示する。

実行済みであっても、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実行済み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

実行済み以外のジョブについても詳細を表示する。

実行済み以外のジョブについては、通常は詳細を表示しない。

これらのジョブを、スプールディレクトリ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。

実行済みのジョブについてだけ表示する。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

スプールジョブ一覧をローカルタイムで簡易表示する。

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。

ジョブ名を指定した場合、実行済み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

- 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 1446291250 1446291320 JOB0030
000004 X -----
000005 H 1446295250 1446295320 JOB0005
000008 H 1446290250 1446300320 JOB0005
000009 H 1446490250 1446490320 JOB0005
000029 H 1447437793 1447437794 JOB0001
000123 H 1447442872 1447442872 JOB001A
000125 H 1447442899 1447442899 JOB001A001
000227 H 1447443000 1447443000 JOB001A001
000231 X -----
000300 X -----
```

## 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	実行済みのジョブであることを示す。
X	実行中であることを示す。

- **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 2015/10/31 20:34:10 2015/10/31 20:35:20 JOB0030
000004 X ----/--/-- --:--:-- ----/--/-- --:--:-- -----
000005 H 2015/10/31 21:40:50 2015/10/31 21:42:00 JOB0005
000008 H 2015/10/31 20:17:30 2015/10/31 23:05:20 JOB0005
000009 H 2015/11/03 03:50:50 2015/11/03 03:52:00 JOB0005
000029 H 2015/11/14 03:03:13 2015/11/14 03:03:14 JOB0001
000123 H 2015/11/14 04:27:52 2015/11/14 04:27:52 JOB001A
000125 H 2015/11/14 04:28:19 2015/11/14 04:28:19 JOB001A001
000227 H 2015/11/14 04:30:00 2015/11/14 04:30:00 JOB001A001
000231 X ----/--/-- --:--:-- ----/--/-- --:--:-- -----
000300 X ----/--/-- --:--:-- ----/--/-- --:--:-- -----
```

- 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 X -----
000005 H JOB0005
000008 H JOB0005
000009 H JOB0005
000029 H JOB0001
000123 H JOB001A
```

```
000125 H JOB001A001
000227 H JOB001A001
000231 X -----
000300 X -----
```

- 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=2015/10/31 20:34:10 2015/10/31 20:35:20
JOBID  ssshk DD NAME              SYSOUT CLASS
000003 ---HK JOBLOG              A
000003 ---HK XML                 A
000003 ---HK SYSMSG             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=2015/10/31 21:40:50 2015/10/31 21:42:00
JOBID  ssshk DD NAME              SYSOUT CLASS
000005 ---HK JOBLOG              A
000005 ---HK XML                 A
000005 ---HK SYSMSG             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 JOBLOG          A
000003 ---HK XML             A
000003 ---HK SYSMSG          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
000005 ---HK JOBLOG          A
000005 ---HK XML             A
000005 ---HK SYSMSG          A
000005 001HK STDOUT          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 X  -----
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) >((1~196))

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 JP1 の論理ホスト名を指定する。論理ホスト名を指定することで設定ファイル (bjex\_論理ホスト.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

**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 10-29 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有メモリ使用量定義で見積もった共有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。
- JP1/AJS から直接実行していないバッチジョブを強制終了する場合、PPID に表示されるプロセス ID のプロセスだけを終了させること。

## 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      PPID      JP1JobID  JOBNAME
-----
000001 LOCK   00:01:00   1002     1001     0000000401 JOB1
000002 WAIT   00:02:00   1004     1003     0000000402 JOB2
000003 LOCK   00:03:00   1006     1005     0000000403 JOB3
```

出力項目の説明

- JOBID**  
ジョブ識別子 (6桁 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。
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 (bjexecsub.exe) のプロセス番号 (10 進数)。
- **PPID**  
 ジョブコントローラプロセス (bjexec.exe) 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

```

> bjexlslock -j 1
(1) JOBID STATUS ELAPSED PID PPID JP1JobID JOBNAME
-----
000001 LOCK 00:01:00 1002 1001 0000000401 JOB1

(2) MODE STATUS LOCKJOB TYPE FILEPATH
-----
EXC LOCK SELF FILE C:\home\userid\exclusiveLockdFile
SHR LOCK SELF FILE C:\home\userid\sharedLockFile

> bjexlslock -j 2
(1) JOBID STATUS ELAPSED PID PPID JP1JobID JOBNAME
-----
000002 WAIT 00:02:00 1004 1003 0000000402 JOB2

(2) MODE STATUS LOCKJOB TYPE FILEPATH
-----
EXC WAIT 000001 FILE C:\home\userid\exclusiveLockdFile
EXC WAIT 000001 FILE C:\home\userid\sharedLockFile
SHR UNLK NONE FILE C:\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。
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桁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。
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="C:¥home¥user1¥SEDAI(-1)" TYPE="GDG" DISP="SHR"/>

この場合、パス名は"C:¥home¥user1¥SEDAI"と表示される。

- パス名を指定した場合

```
>cd C:¥home¥userid
>bjexlslock exclusiveLockedFile sharedLockedFile allocateWaitingFile
WAITCT  EXCJOB  SHRCT  TYPE  FILEPATH
  1  000001    0  FILE  C:¥home¥userid¥exclusiveLockedFile
      JOBID  STATUS  MODE
      000001  LOCK   EXC
      000002  WAIT   SHR
WAITCT  EXCJOB  SHRCT  TYPE  FILEPATH
  1  -----    2  FILE  C:¥home¥userid¥sharedLockedFile
      JOBID  STATUS  MODE
      000001  LOCK   SHR
      000002  WAIT   EXC
      000003  LOCK   SHR
WAITCT  EXCJOB  SHRCT  TYPE  FILEPATH
  0  -----    0  FILE  C:¥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。
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="C:¥home¥user1¥SEDAI(-1)" TYPE="GDG" DISP="SHR"/>

この場合、パス名は"C:¥home¥user1¥SEDAI"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

```
>bjexlslslock -t FILE
WAITCT  EXCJOB  SHRCT  TYPE  FILEPATH
  1 000001    0 FILE  C:¥home¥userid¥exclusiveLockedFile
      JOBID  STATUS  MODE
      000001  LOCK   EXC
      000002  WAIT   SHR
WAITCT  EXCJOB  SHRCT  TYPE  FILEPATH
  1 -----  2 FILE  C:¥home¥userid¥sharedLockedFile
      JOBID  STATUS  MODE
      000001  LOCK   SHR
      000002  WAIT   EXC
      000003  LOCK   SHR
WAITCT  EXCJOB  SHRCT  TYPE  FILEPATH
  0 -----  0 FILE  C:¥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。
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="C:¥home¥user1¥SEDAI(-1)" TYPE="GDG" DISP="SHR"/>

この場合、パス名は"C:¥home¥user1¥SEDAI"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 (一般ユーザ用))

## 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) >((1~196))

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 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
[0000005680] [Wait] 024368 001 15/11/30 18:30:40 msg=KCCC2003R-I STOP (001673, 001, host1,
1198043673, 0)
[0000001928] [Set ] 024369 001 15/11/30 18:30:40 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 (024369, 001, host1, 1198044152, 0)  
[0000001928] [Set ] 024369 001 15/11/30 18:30:40 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。bjexlsmmsg 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）

# bjexrmjob (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、実行済み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) >((1~196))

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 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ス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、ス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。

#### -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ス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。

#### -f

このオプションを指定した場合、スプールジョブの状態に関係なくスプールジョブディレクトリがあればス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る。一般的にはスプールジョブディレクトリ内に SYSOUT データファイル以外のファイルが存在するときなど、スプールジョブのディレクトリが削除できない場合に使用する。

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ス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、実行済みのス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ス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。

#### -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、スプールディレクトリ、ス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。

ス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スプールジョブの操作を行うと更新される。

#### -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。

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、コマンド実行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、バッチジョブの実行ユーザにかかわらず削除可能であるため、指定するジョブ識別子やジョブ名など、削除対象に注意して実行すること。
- 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。
- 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 のスプールジョブを強制削除する。

```
bjexrmjob -f 231
```

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、スプールジョブ管理ファイルまたはディレクトリの最終更新時刻がコマンド投入時刻より 30 秒以前であれば削除の対象となる。

2月2日 10:15:40 にこの指定をした場合、スプールジョブ管理ファイルまたはディレクトリの最終更新時刻が 2月2日 10:15:09 以前の場合にスプールジョブを削除する。

- ジョブ実行開始時刻が 10 日以前の実行済みのスプールジョブを削除する。

```
bjexrmjob -T 10
```

2月12日にこの指定をした場合、2月2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コマンド投入時刻より 40 秒以上前の場合に削除する。

```
bjexrmjob -f -a 40 123
```

2月2日 11:15:30 にこの指定をした場合、2月2日 11:14:49 以前に開始されたスプールジョブが対象となる。

- 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 時間以前の実行済みのスプールジョブを削除する。

```
bjexrmjob -H 10
```

2月2日 13:15:30 にこの指定をした場合、2月2日 3:15:29 以前に開始されたスプールジョブが削除の対象となる。

- ジョブ実行開始時刻が 36 時間以前の実行済み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*
```

2月2日 16:15:30 にこの指定をした場合、2月1日 4:15:29 以前に開始されたスプールジョブで、かつ、ジョブ名の先頭が SOUMU で始まるものが削除の対象となる。

# bjexsvc (バッチジョブ実行システムのサービスの登録およびサービス起動環境の設定 (管理者用))

## 形式

```
bjexsvc { [-h 論理ホスト名] -install | -setup }
```

## 機能

バッチジョブ実行システムのサービス (BjexSvc) の登録およびサービス起動環境の設定を行う。

## オプション

**-h 論理ホスト名** ~<英字と数字と- (ハイフン) >((1~196))

-install オプションを指定する場合に指定できる。バッチジョブ実行システムの論理ホスト環境を構築するために、BjexSvc\_論理ホスト名サービスを登録する。

論理ホスト名の長さは63バイト以下を推奨する。63バイトを超える名称を指定すると、動作しないことがある。

**-install**

バッチジョブ実行システムのサービスを登録する。

バッチジョブ実行システムのサービスは、バッチジョブ実行システムをセットアップしたときに自動的に登録される。しかし、登録したレジストリ情報を消去してしまった場合は、このオプションを使って手動で再登録する必要がある。

また、バッチジョブ実行システムの論理ホスト環境を構築する場合は、-h オプションと合わせて指定することによって BjexSvc\_論理ホスト名サービスを登録する。

**-setup**

バッチジョブ実行システムのサービス (BjexSvc) を起動するための環境設定を行う。

このオプションを指定すると、Windows のローカルグループ"BJEX File Exclusive Control Group"が生成される。

## 戻り値

戻り値	意味
0	正常終了
0以外	異常終了

## 注意事項

- オプションを指定しなかったり、指定できるオプション以外を指定して実行したりした場合でも、コマンドはエラーにはならないで終了する。この場合、レジストリ情報の更新は行われぬ。
- h オプションを指定する場合は、最初に指定すること。

## 使用例

BjexSvc\_論理ホスト名サービスを登録する。

```
bjexsvc -h 論理ホスト名 -install
```

# bjex\_setup (バッチジョブ実行システムを実行する環境の設定 (管理者用))

## 形式

```
bjex_setup [-k [-h 論理ホスト名]]
```

## 機能

バッチジョブ実行システムのインストールディレクトリにインストールされた設定ファイルのテンプレートを入力し、バッチジョブ実行システムの動作に必要な標準的なディレクトリと設定ファイルのひな形を生成する。

### • 生成するディレクトリ

<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥spool

<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥conf

<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥log

<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥perm

<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥proc

<All Users のプロファイルのアプリケーションデータディレクトリの場所>¥Hitachi¥BJEX¥trace

<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥temp

### • 生成するファイル

<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥conf¥bjex.conf

### • 設定ファイルの変更

bjex\_setup コマンドでは設定ファイルのテンプレートの次の内容について、標準的なディレクトリを使用するように変更して設定ファイルのひな形を生成する。ただし、テンプレートファイルで明示指定されている場合には正しく作成されないため、テンプレートの設定ファイルは変更しないこと。

- JP1\_HOSTNAME ホスト名  
バッチジョブ実行システムが稼働するホスト名
- [LOG\_DIR△パス名]  
実行ログ出力ディレクトリのパス
- [PERMANENT\_FILE\_DIR△パス名]  
恒久ファイル用ディレクトリのパス
- [PROCEDURE\_DIR△パス名]  
プロシジャライブラリのパス
- [SPOOL\_DIR△パス名]  
スプールディレクトリのパス
- [TEMP\_FILE\_DIR△パス名]  
一時ファイル作成ディレクトリのパス



- [TRACE\_DIR△パス名]  
トレース出力ディレクトリのパス

## オプション

-k

<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥conf¥bjex.conf の設定ファイルをカスタマイズしたあとに、このファイルのシンタクスチェックだけ行うときに指定する。

-h 論理ホスト名

<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥conf¥bjex\_論理ホスト名.conf の設定ファイルを作成したあとに、このファイルのシンタクスチェックをするときに指定する。  
このオプションは-k オプションと組み合わせて指定すること。

## 注意事項

- ディレクトリがすでに生成されている場合は、ディレクトリをそのまま利用する。
- 作成する環境に設定ファイルがすでに存在していた場合は、エラーで終了する。

## 使用例

```
bjex_setup
```

# 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

$\Delta_0$ パラメータ名 $\Delta_1$ パラメータのオペランド名

- 出力クラスのパラメータ

$\Delta_0$ <SysoutClass $\Delta_1$ クラス名>  
 $\Delta_0$ パラメータの記述内容  
 $\Delta_0$ </SysoutClass>

- 注釈

$\Delta_0$ # $\Delta_0$ 注釈

(凡例)

$\Delta_0$  : 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。

$\Delta_1$  : 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。

- 設定ファイルは、bjex\_setup コマンドで生成される内容に基づき、システム管理者が運用に合わせて変更する。
- 設定ファイルは、バッチジョブ実行システムのプログラム起動時に  
<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥conf¥bjex.conf から読み込まれる。
- 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、行全体が注釈であると見なし、読み飛ばされる。
- 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 $\Delta$ 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なる。
- 次のどちらかの条件に該当するパスを絶対パスと判断する。
  - 先頭文字が "ドライブレター:¥"
  - 先頭文字が "¥¥"
- 1 行の長さが 4,092 文字を超える場合はエラーとなる。
- 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なる。

- 複数指定可能な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なる。
- パラメータ指定値に空白を含む場合は、"で囲んで指定できる。
- 値に"を含む値を指定するとパラメータエラーになり、bjex\_setup コマンドで正しく処理できないことがあるため、"を含む値を指定しないこと。

## 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
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。		1
DD_EXISTCHECK	DD 要素に指定したファイルパスの存在確認をするかどうかを定義する (EXEC 要素の LANG 属性に"COBOL"以外を指定した場合)。		
DD_EXISTCHECK_COBOL	DD 要素に指定したファイルパスの存在確認をするかどうかを定義する (EXEC 要素の LANG 属性に"COBOL"を指定した場合)。		
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。		
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。		
DSN_USE_ENV	DD 要素の DSN 属性への環境変数指定機能を使用するかどうかを定義する。		
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が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_PARMSPLIT	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。		
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れる条件を定義する。		
ISAMLIB_MODE	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。		

パラメータ名	定義内容	パラメータ指定	指定できる個数
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。	任意	1
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。		
JP1_HOSTNAME	JP1 が稼働する論理ホスト名を定義する。	必須	1
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。	任意	無制限
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。		1
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_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_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	排他制御機能を使用するかどうかを定義する。		
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/共有制御の解除契機を定義する。		
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	JP1/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
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。	必須	
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	

パラメータ名	定義内容	パラメータ指定	指定できる個数
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。	任意	無制限
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。		
SET_PGMRIORITY	ユーザプログラムに JP1/AJS で指定した実行優先順位を引き継ぐかどうかを定義する。		
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。		
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。		
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	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。		
STDOUTCLASS_DEFAULT	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。		
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	各出力クラスの属性値を定義する。	1 個以上 必須	256
TEMP_FILE_DIR	一時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	1
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、識別子を付けて記述することで、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>((1～128 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「¥」が付けられていない場合、そのパス名に対して自動的に末尾に「¥」が付けられる。



## 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する際、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場合に指定する (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する。

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するのではなく、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する。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する。

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する。

**NO** : 存在確認をしない。

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	DATA	DUMMY	TEMP	TEMPISAM	SYSOUT
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する。
- : 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する必要がない。

## 注意事項

- ファイルパスの存在確認をする 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する。
- 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する際、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場合に指定する (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、JOBLIB/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する。

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するのではなく、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する。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する。

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する。

NO : 存在確認をしない。

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	DIR	GDG	DATA	DUMMY	TEMP	TEMPISAM	SYSOUT
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する。

○ : 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する必要がない。

## 注意事項

- ファイルパスの存在確認をする 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する。
- 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。

## 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	GDG	DATA	DUMMY	TEMP	TEMPISAM	SYSOUT
NEW	-	○	-	×	-	-	○	○	-
ERROR	-	×	-	×	-	-	×	×	-

(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。

## 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。



## DSN\_USE\_ENV (DD 要素の DSN 属性への環境変数指定機能)

---

### 形式

```
[DSN_USE_ENV {YES|NO}]
```

### 機能

ジョブ定義 XML ファイルの DD 要素の DSN 属性値にある環境変数を環境変数の値に置換する。

### オペランド

DSN\_USE\_ENV~{YES | NO}

ジョブ定義 XML ファイルの DSN 属性値に環境変数を記述した場合、環境変数の部分を環境変数の値に置換する機能を使用するかどうかを指定する。

**YES**：ジョブ定義 XML ファイルの DD 要素の DSN 属性値に環境変数を記述した場合、環境変数の部分を環境変数の値に置換する。

**NO**：ジョブ定義 XML ファイルの DD 要素の DSN 属性値に環境変数を記述した場合、環境変数の部分を環境変数の値に置換しないで、通常の文字列として扱う。

## 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。

## 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**：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。

### 注意事項

このパラメータはジョブ全体に影響する。プログラムに渡す引数の形式を個々のジョブステップで区別したい場合は、ジョブ定義 XML ファイルの EXEC 要素の PARMSPLIT 属性を使用すること。

# 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 パラメータに"KEEP"を指定している場合、"PASS"を指定した場合の動作をする。

ジョブが正常終了時、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 パラメータ値を"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実行する。

終了コードがしきい値以上の場合、およびジョブの異常終了時は、"PASS"を"DELETE"に変更しファイルを削除する。このとき、世代管理ファイルにも登録されない。

ジョブの異常終了時、ジョブ終了コードによらず、DD 要素の DISP 属性の第 3 パラメータ値が"KEEP"のときは"DELETE"に変更してファイルを削除する。このとき、世代管理ファイルにも登録されない。

このファイルの登録・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、パラメータ値が"KEEP"に変更され、保存される。

## オペランド

### しきい値～<符号なし整数>((1～4097))

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正常終了した場合は、世代ファイルの保存と世代管理ファイルへの登録がされ、「しきい値」以上の値のジョブ終了コードでジョブが正常終了した場合は、世代ファイルの保存と世代管理ファイルへの登録はされない。

なお、ジョブ終了コードが 4096 のときに、世代ファイルの保存と世代管理ファイルへの登録がされないために、4097 まで指定できる。

ジョブが異常終了した場合は、ジョブ終了コードの値に関係なく、世代ファイルの保存と世代管理ファイルへの登録はされない。

# ISAMLIB\_MODE (ISAM ライブラリ (32 ビット版または 64 ビット版) の定義)

## 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。

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。bjexec から ISAM ライブラリの 64 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。

## 注意事項

- このパラメータで M64 を指定する場合、実行環境が 64 ビット OS で 64 ビットの ISAM が組み込まれている必要がある。一時 ISAM ファイル使用時の動作方法を次の表に示す。

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。

# 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、最終ジョブ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 でジョブを強制終了した場合、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 10090 となる。
- ジョブステップの終了コードが 4095 より大きい場合、または負の値の場合、ジョブの終了コードは 4096 とする。
- このパラメータに 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="windows">
  <JOB NAME="TESTJOB">
    <STEP NAME="STEP1_RC4">
      <EXEC PGM="CBLRTN" PARM="004" LANG="COBOL"/>
    </STEP>
    <STEP NAME="STEP2_RC0">
      <EXEC PGM="CBLRTN" PARM="000" LANG="COBOL"/>
    </STEP>
    <SETMAXRC VALUE="16" />
    <STEP NAME="STEP3_RC0">
      <EXEC PGM="CBLRTN" PARM="000" LANG="COBOL"/>
    </STEP>
    <SETMAXRC VALUE="4"/>
  </JOB>
</HitachiBatchJobExec>
```

## JOBLOG

```
***** BJEX MESSAGE *****
16:14:50 030968 KAKC0091-I TESTJOB Job started.
16:14:50 030968 KAKC0092-I TESTJOB.STEP1_RC4 Step started.
16:14:50 030968 KAKC0097-I TESTJOB.STEP1_RC4 Step ended. rc=4 E-Time=0.045s C-Time=0.000s
16:14:50 030968 KAKC0092-I TESTJOB.STEP2_RC0 Step started.
16:14:50 030968 KAKC0097-I TESTJOB.STEP2_RC0 Step ended. rc=0 E-Time=0.047s C-Time=0.000s
16:14:50 030968 KAKC1909-I TESTJOB The maximum value of return code was changed by SETMAXRC element. rc=16
16:14:50 030968 KAKC0092-I TESTJOB.STEP3_RC0 Step started.
16:14:50 030968 KAKC0097-I TESTJOB.STEP3_RC0 Step ended. rc=0 E-Time=0.048s C-Time=0.000s
16:14:50 030968 KAKC1909-I TESTJOB The maximum value of return code was changed by SETMAXRC element. rc=4
16:14:50 030968 KAKC0098-I TESTJOB Job ended. rc=4 E-Time=0.140s C-Time=0.000s
```

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 STEP3\_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\_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

- この指定をした場合、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。
- 表示できるジョブ数として、LOCK\_JOBMLPX パラメータで指定したジョブ多重度を使用する。LOCK\_JOBMLPX 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。



# JP1\_HOSTNAME (バッチジョブ実行システムが稼働するホスト名定義)

---

## 形式

JP1_HOSTNAME ホスト名
-------------------

## 機能

連携する JP1 が稼働しているホスト名を定義する。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。

バッチジョブ実行システムの論理ホストを運用する場合、JP1\_HOSTNAME パラメータの指定は任意であり、指定しても無視される。

## オペランド

ホスト名～<英字, 数字, @, #, \_ (アンダースコア), - (ハイフン) >～((1～196 文字))

- JP1 の物理ホスト環境でバッチジョブ実行システムの物理ホストを運用する場合  
hostname コマンドを実行したときに表示されるホスト名を指定する。
- JP1 の論理ホスト環境でバッチジョブ実行システムの物理ホストを運用する場合  
JP1\_HOSTNAME パラメータに、連携する JP1 の論理ホスト名を指定する。

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。

この値は接続先の JP1 イベントサーバ名, SYSOUT ファイル名, スプール管理ファイルの情報の一部として利用される。

# 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「[10.5 JP1 イベント一覧](#)」および「[10.6 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。

# 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、「[3.12.8 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

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着確認を実施する。通常メッセージ出力については、リトライ回数を3回とし、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イベント到着確認のリトライ時間を指定できる。

# 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

メッセージ制御機能を使用することで発行される 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))

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、メッセージ制御機能を使用することで発行される 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、次に示すホスト名を指定したものとみなす。

- 物理ホストの場合  
JP1\_HOSTNAME パラメータに指定したホスト名
- 論理ホストの場合  
JP1 の論理ホスト名

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、メッセージ制御機能を使用することで発行される JP1 イベントが JP1/IM - View に表示されなくなる。
- バッチジョブ実行システムが発行する JP1 イベントは、メッセージ制御機能が発行する JP1 イベントだけ発行先ホストが異なる。詳細は、「[2.3.3\(1\) JP1 イベントの発行先ホスト](#)」を参照のこと。

# 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

## オペランド

ジョブ多重度数 ～<数字> ((1～64) < 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 Mutex の確保ができないジョブは、KAKC1652-W メッセージを出力し、Mutex 確保待ちとなる。Mutex 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。

## 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。LOCK\_JOBMPLEX および LOCK\_MAXFILE の値が有効になる。

**DISABLE** : 排他制御機能を使用しない。LOCK\_JOBMPLEX および LOCK\_MAXFILE の値は有効とならない。

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\_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/共有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/共有制御をジョブ終了時に解除するか、ジョブステップ終了時に解除するかを定義する。

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属性でスキップされたジョブステップの場合も同様である。

## 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。
- STEP オペランドを指定した場合でも、JOBLIB 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>((1~128文字)) 《<All Users のプロファイルのドキュメントディレクトリの場所>¥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

このパラメータは bjex\_setup コマンドを実行することで設定される。このパラメータを設定ファイルのテンプレートに指定すると、bjex\_setup コマンドが設定ファイルのひな形を作成できなくなるため、設定ファイルのテンプレートにこのパラメータを指定しないこと。

## 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 $\leq 2,000$

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。
- 通常メッセージのイベント送信および到達確認については、パラメータの指定に関係なく転送失敗のエラーが発生した場合にリトライを 3 回実施し、それ以降はリトライをやめて RC=0 で終了し、この機能を実行しているプログラムを継続させる。
- 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。

## 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おそれがあるため、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。

恒久ファイルは、次のどの形式にも該当しない DSN 属性値のファイルを恒久ファイルと判断する。

- 絶対パス  
ドライブレター:¥[パス名¥]ファイル名, または先頭が¥¥
- 相対パス  
先頭が.¥または..¥
- パス名置換識別子を付けるパス  
\$[BJEX\_DIR\_任意名]¥[パス名¥]ファイル名

## オペランド

パス名~<パス名>((1~128 文字)) 《<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥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

このパラメータは bjex\_setup コマンドを実行することで設定される。このパラメータを設定ファイルのテンプレートに指定すると、bjex\_setup コマンドが設定ファイルの雛形を作成できなくなるため、設定ファイルのテンプレートにこのパラメータを指定しないこと。

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。



# 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～<整数>((-10000～10000))

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。このパラメータの指定が省略されている場合、1, 2, 3を仮定する。

## 指定例

ユーザプログラムの終了コードが-1以下、1～3、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 -10000_-1  
PGM_RCABEND 1_3  
PGM_RCABEND 12_10000
```

## 注意事項

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 パス名]

### 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) のパス名を定義する。

### オペランド

パス名~<パス名>((1~128 文字)) 《<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥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。

### 注意事項

このパラメータは bjex\_setup コマンドを実行することで設定される。このパラメータを設定ファイルのテンプレートに指定すると、bjex\_setup コマンドが設定ファイルのひな形を作成できなくなるため、設定ファイルのテンプレートにこのパラメータを指定しないこと。

## SET\_PGMPRIORITY(ユーザプログラムへの実行優先順位引き継ぎ)

---

### 形式

```
[SET_PGMPRIORITY {YES|NO}]
```

### 機能

ユーザプログラムに JP1/AJS で指定した実行優先順位を引き継ぐかどうかを定義する。

### オペランド

SET\_PGMPRIORITY～((YES | NO))

YES：ユーザプログラムに JP1/AJS で指定した実行優先順位を引き継ぐ。

NO：ユーザプログラムに JP1/AJS で指定した実行優先順位を引き継がない。

## 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)) 《<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥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

このパラメータは bjex\_setup コマンドを実行することで設定される。このパラメータを設定ファイルのテンプレートに指定すると、bjex\_setup コマンドが設定ファイルのひな形を作成できなくなるため、設定ファイルのテンプレートにこのパラメータを指定しないこと。

## 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。

YES を指定した場合、PRTCNTL と PRTPDATA で指定したディレクトリに印刷データを転送する。NO を指定した場合、転送しない。

### 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。保管先ディレクトリのアクセス権限は、全ジョブ実行ユーザに対して読み込み許可、書き込み許可、作成権限、削除権限を設定する必要がある。

### 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。

# 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アクセス権限は、SPOOL\_STORE\_DIR で指定したディレクトリの権限を引き継いで設定し作成する。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せず、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。

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するときの書き込みモードを定義する。

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 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する。

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する。

# 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}]
[FCB FCB名称]
[FCDFLT デフォルト機能キャラクタ名]
[FORMS 書式番号]
[FTYPE {1|2}]
[GAIJI1 レベル1外字名]
[HOLD {YES|NO}]
[KEEP {YES|NO}]
[MODE {AN|K}]
[OVERC 適用枚数]
[OVERLAY 書式オーバーレイ名]
[PRTCNL スプールジョブ管理ファイル転送先ディレクトリ名]
[PRTDATA スプールジョブデータファイル転送先ディレクトリ名]
[RECFM {A|M|N}]
[TRUNC {YES|NO}]
</SysoutClass>
```

## 機能

各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。

#### 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。

#### 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、行制御印刷をするファイル形式

#### GAIJ1 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、あとから印刷するかどうかを指定する。

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。

#### 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 データファイルをプリントサーバに転送後、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 データファイルをプリントサーバに転送後、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。

#### PRTCNTL 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。

DUMMY オペランドに YES を指定している場合、このオペランドは省略できる。

#### PRTDATA スプールジョブ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。

DUMMY オペランドに YES を指定している場合、このオペランドは省略できる。

#### RECFM~{A | M | N}

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 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。

**A** : ANSI 制御文字。

**M** : 機械制御文字。

**N** : 制御文字なし。

#### 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。プールジョブはジョブ終了後に削除される。

# 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

一時ファイルを格納するディレクトリのパス名を定義する。定義するディレクトリは、バッチジョブ実行システムの一時ファイル以外では使用しないこと。

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。

TYPE 属性値に TEMP または TEMPISAM 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](#)」

## オペランド

パス名~<パス名>((1~128 文字)) 《<All Users のプロファイルのドキュメントディレクトリの場所>¥Hitachi¥BJEX¥temp》

一時ファイルを格納するディレクトリのパス名を指定する。

## 注意事項

このパラメータは bjex\_setup コマンドを実行することで設定される。このパラメータを設定ファイルのテンプレートに指定すると、bjex\_setup コマンドが設定ファイルのひな形を作成できなくなるため、設定ファイルのテンプレートにこのパラメータを指定しないこと。



## 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>((1~128文字)) 《<All Users のプロファイルのアプリケーションデータディレクトリの場所>¥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。

### 注意事項

このパラメータは bjex\_setup コマンドを実行することで設定される。このパラメータを設定ファイルのテンプレートに指定すると、bjex\_setup コマンドが設定ファイルのひな形を作成できなくなるため、設定ファイルのテンプレートにこのパラメータを指定しないこと。

## 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
JP1_HOSTNAME      BATCHHOST
SPOOL_DIR         C:¥BJEX¥spool
LOG_DIR           C:¥BJEX¥log
PROCEDURE_DIR     C:¥BJEX¥proc
PERMANENT_FILE_DIR C:¥BJEX¥perm
TEMP_FILE_DIR     C:¥temp
TRACE_DIR         C:¥BJEX¥trace
BJEX_DIR_GYOUUMU C:¥BJEX¥gyoumu
##Select the SysoutClass name.###
MSGCLASS_DEFAULT A
##sysout class
<SysoutClass A>
  HOLD           NO
  FCB            A4L6
  MODE          K
  PRTCNTL       ¥¥PRTSERVO1¥PDE¥PRTCNTL
  PRTPDATA     ¥¥PRTSERVO1¥PDE¥PRTPDATA
</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。

## 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
&	&amp;
<	&lt;
>	&gt;
' (シングルクォート)	&apos;
" (ダブルクォート)	&quot;

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/>]
  [<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/>][...]
    </STEP>
    [<SETMAXRC要素/>]
    [<STEP要素/>|<SETMAXRC要素/>][...]
  </PROC>][...]
  <STEP要素>
    <EXEC要素/>|
    <CALL要素>
      [<SESYMBOL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/>][...]
  </STEP>
  [<SETMAXRC要素/>]
  [<STEP要素/>|<SETMAXRC要素/>][...]
</JOB>|
[<PROC要素>
  [<SYMBOL要素/>][...]
  <STEP要素>
    <EXEC要素/>
    [<DD要素/>|<SETENV要素/>][...]
  </STEP>
  [<SETMAXRC要素/>]
  [<STEP要素/>|<SETMAXRC要素/>][...]
</PROC>]
</HitachiBatchJobExec>

```

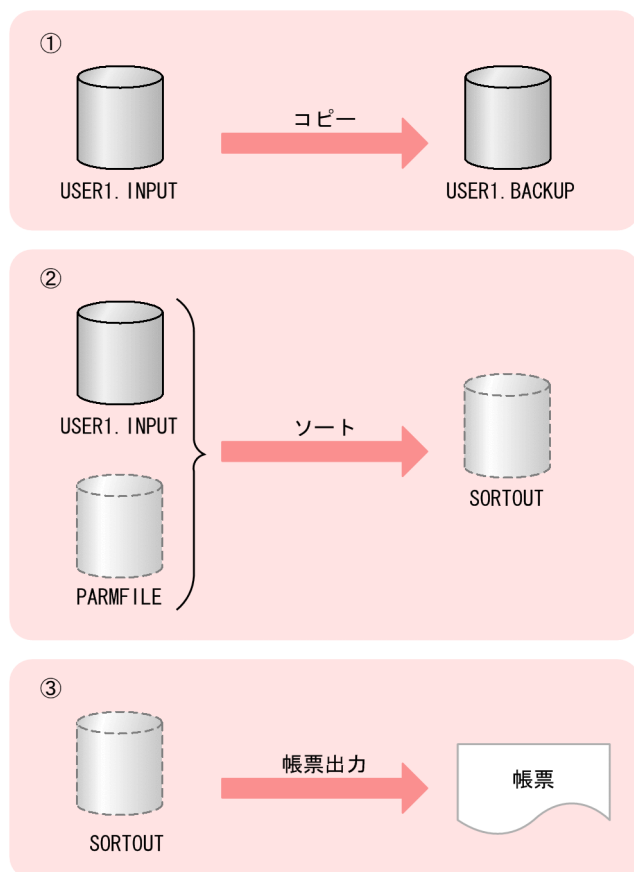
## 使用例

### 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 に転送したあと、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 XML 定義ファイルの例（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="1.2" os="windows" > . . . 2
  <JOB NAME="SAMPLE1" COND="0,NE" > . . . 3
    <DD NAME="JOBLIB" TYPE="LIB" DSN="C:¥BJEX¥batch¥job1¥bin" /> . . . 4
    <DD NAME="JOBLIB" TYPE="LIB" DSN="D:¥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[!copy %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
!]]> . . . 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[! . . . 19
-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 C:¥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 に転送する）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。転送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。転送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、ユーザアプリケーションプログラムの実行を指定する。ユーザアプリケーションプログラムは 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ユーザアプリケーション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。ユーザアプリケーション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 = "1.2" os="windows" >	. . . 2
<PROC NAME="SAMPLE2_PROC" >	. . . 3
<STEP NAME="SAMPLE2_PROC_STEP1">	. . . 4
<EXEC PGM="cmd" PARM="/C ECHO %%%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 = "1.2" os="windows" >	. . . 2
<JOB NAME="SAMPLE2_JCL" >	. . . 3
<STEP NAME="STEP1" >	. . . 4
<EXEC PGM="cmd" PARM="/C ECHO %%%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="/C ECHO %%%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 = "1.2" os="windows" >   . . . 2
<JOB NAME="SAMPLE3_JCL" >                             . . . 3
  <PROC NAME="SAMPLE3_PROC" >                          . . . 4
    <STEP NAME="SAMPLE3_PROC_STEP1">                   . . . 5
      <EXEC PGM="cmd" PARM="/C ECHO %%%%%%%%%SAMPLE3_PROC_STEP1%%%%%%%%" /> . . . 6
    </STEP>                                             . . . 7
  </PROC>                                              . . . 8
  <STEP NAME="STEP1" >                                  . . . 9
    <EXEC PGM="cmd" PARM="/C ECHO %%%%%%%%%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="/C ECHO %%%%%%%%%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="VersionNum"]  
[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="VersionNum"～<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)) 《Shift\_JIS》

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

文字符号化方式名は"Shift\_JIS"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
- アンダスコア(\_)とハイフン(-)

ジョブ定義 XML ファイルの実際の符号化方式に関係なく、必ず Shift\_JIS を指定すること。

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、または存在することがあることを示す。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。

## 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。
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。
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。
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 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、その対象となるステップを定義する。
SETENV	環境変数を定義する。
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ないこと
- DTD 上で定義されている要素・属性のうち、「表 9-2 XML ジョブ制御要素一覧」に記載されていないものはバッチジョブ実行システムではサポートしない。それらの要素・属性を指定した場合、バッチジョブ実行システムはジョブ定義 XML ファイルの解析エラーとなる。

# 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="1.2" os="windows">
  <JOB NAME="ASM">
    <STEP NAME="STEP">
      <CALL NAME="ASMCG" >
        <SETSYMBOL NAME="PARM1" VALUE="C:¥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
  |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)
  |ファイルパス名
  |ディレクトリパス名
  |* [. ステップ名 [. プロシジャステップ名] ]. DD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}"]
[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}"]
[RENAME="環境変数名"]
[SAMF="{YES|NO}"]
[SYSOUT="{クラス名|*}"]
[TYPE="{DATA|DUMMY|FILE|TEMP|TEMPISAM|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、「[3.10 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ある場合は、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ある場合、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あるときは、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ときは、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。

- OLD

ファイルが該当するジョブステップ以前にすでにある場合、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ある場合、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ある場合、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するときに指定する。

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, もしくは 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 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。

- 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  
|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)  
|ファイルパス名  
|ディレクトリパス名  
|*.[ステップ名[.プロシジャステップ名]].DD名  
}"
```

ファイルの名称を指定する。

設定ファイルのDSN\_USE\_ENVパラメータにYESを指定している場合、この属性に環境変数を指定できる。環境変数の指定方法と指定できる環境変数については、「[3.4.7\(5\) DSN属性への環境変数指定機能](#)」を参照のこと。

- 一時ファイル識別名～<パス名>((1～247文字))

TYPE属性値にTEMP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"¥"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"¥"より後ろの文字列を一時ファイル識別名と判断する。

- 一時ISAMファイル識別名～<パス名>((1～247文字))

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"¥"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"¥"より後ろの文字列を一時ISAMファイル識別名と判断する。

- 世代データ群パス名(相対世代番号)～<パス名>((1～247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。

#### 世代データ群パス名.Gmmmm※

注※ mmmmm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。

- ファイルパス名～<パス名>((1～247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>((1～247 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。
- \*.ステップ名[.プロシジャステップ名].DD 名  
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))  
\*.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。

**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

**GAIJ1 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、あとから印刷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  
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[{C | 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}"

スプールジョブ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  
スプールジョブ転送後、スプールジョブを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  
スプールジョブ転送後、スプールジョブを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け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。

付ける文字列は 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、ステップ内でだけ有効となる（次のステップでは、PATH 環境変数から除く）。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}"**

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。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。

- 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)

**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\_" を付けた名称になる。

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 | 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

TYPE="TEMP" と同等であり、一時ファイルを割り当てる。

- 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。
- 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（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"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、「[3.8 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 要素外に指定可能な 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 で印刷するデータ形式の場合に、VOS3 の改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

印刷データに付け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

印刷データに付け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ジョブ定義 XML ファイルの解析エラーとなる。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	DUMMY	SYSOUT	LIB	TEMPISAM
NAME	○	○	○	○	○	○	○	○	○
DSN	○	○	○	△	×	×	×	○	△
DISP	△	△	△	△	×	×	×	△	△
SYSOUT	×	×	×	×	×	×	○	×	×
CHARS	×	×	×	×	×	×	△	×	×
CMT	×	×	×	×	×	×	△	×	×
COPIES	×	×	×	×	×	×	△	×	×
COPYG	×	×	×	×	×	×	△	×	×
DEST	×	×	×	×	×	×	△	×	×
FCB	×	×	×	×	×	×	△	×	×
FCDFLT	×	×	×	×	×	×	△	×	×
FORMS	×	×	×	×	×	×	△	×	×
FTYPE	×	×	×	×	×	×	△	×	×
GAJJI1	×	×	×	×	×	×	△	×	×
HOLD	×	×	×	×	×	×	△	×	×
KEEP	×	×	×	×	×	×	△	×	×
MODE	×	×	×	×	×	×	△	×	×
OPTCDJ	×	×	×	×	×	×	△	×	×
OVERC	×	×	×	×	×	×	△	×	×
OVERLAY	×	×	×	×	×	×	△	×	×
RECFM	×	×	×	×	×	×	△	×	×



DD 要素の属性	DD 要素の TYPE 属性								
	FILE	DIR	GDG	TEMP	DATA	DUMMY	SYSOUT	LIB	TEMPISAM
ISKEY	×	×	×	×	×	×	×	×	○*
ISRECFM	×	×	×	×	×	×	×	×	○*
ISRECL	×	×	×	×	×	×	×	×	○*
RENAME	△	△	△	△	△	△	△	×	△
SAMF	×	×	×	×	△	×	×	×	×

(凡例)

- ：必ず指定する。
- △：指定を省略できる。
- ×

注※

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。

- DTD 上は TYPE 属性の値として PREST および TEMPPREST が含まれるが、これらの属性値はバッチジョブ実行システムではサポートしない。これらの値を TYPE 属性として指定した場合、バッチジョブ実行システムはジョブ定義 XML ファイルの解析エラーとする。
- DTD 上は DD 要素の属性に FPERM 属性が含まれるが、これらの属性値はバッチジョブ実行システムではサポートしない。FPERM 属性を指定した場合、バッチジョブ実行システムはジョブ定義 XML ファイルの解析エラーとする。
- 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ことがある。そのため、これらの値を NAME 属性に指定しないこと。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 属性にドライブレーターの指定以外の目的で":"を指定しないこと。指定した場合、ジョブ定義 XML ファイルの解析エラーとなる。
- 別ユーザまたはバッチジョブ実行システム以外のプロセスが使用中のため、読み取り専用となっているファイルに対して、DELETE や RNW などそのファイルを操作、変更する DISP 属性値を指定した場合、そのファイルの変更は失敗する。

- 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。

## 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"/>
```



# EXEC 要素 (プログラムの実行)

## 形式

```
[COND="
  [(しきい値, 演算符号 [, ステップ名 [, プロシジャステップ名]] [D])
  [, (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}] " ]
[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"]
[LANG="{COBOL|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 | Javaアプリケーションクラス名}"]
[PARM="{引数 | <Javaバッチアプリケーションの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 [, ステップ名 [, プロシジャステップ名]] [D])
[, (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>((0~4095))

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。

- **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、ステップ名としてプロシジャを呼び出した CALL 要素の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。

プロシジャ内でプロシジャステップ名を指定しないでステップ名だけ指定した場合、はじめにその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。

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。

**LANG="{COBOL | 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\_"を付けて、"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\_"を付けて、"DDN\_"+"DD 要素名"という名称に変換する。

- **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"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 属性にスケジューラグループ名, 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>((1~1023))

- **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。

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。

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 | 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, 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>((1~247 文字))**

実行するプログラム, コマンド, スクリプトの名称を指定する。

プログラムの名称の一部として拡張子を指定しなかった場合, ".exe"を追加する。最後が1つのピリオド"."で終了し拡張子が含まれていない場合や, パス名が含まれている場合, ".exe"は追加しない。

- **Java アプリケーション~<クラス名>((1~247 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。

- Java バッチアプリケーションの main 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

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、Windows のコマンドプロンプトに記述できるコマンド行を指定する。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く、", 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。

## 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 "\*" を指定してスクリプトを記述し、OS 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>
```

- DTD 上は DD 要素の属性に JSPEC 属性が含まれるが、この属性値はバッチジョブ実行システムではサポートしない。JSPEC 属性を指定した場合、バッチジョブ実行システムはジョブ定義 XML ファイルの解析エラーとする。
- TYPE 属性値に"TEMP"および"TEMPISAM"を指定した場合、ジョブコントローラは DSN 属性が指定されていても無視する。

## 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="1.2" os="windows">
  <JOB NAME="JOB1">
    <STEP NAME="STEP">
      <EXEC PGM="PROG1" />
    </STEP>
  </JOB>
</HitachiBatchJobExec>
```

また、Windows のバッチファイ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="1.2" os="windows">
  <JOB NAME="JOB1">
    <STEP NAME="STEP">
      <EXEC PGM='*'>
        <![CDATA[
          !
          PROG1
          !
        ]]>
      </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、指定した内容をチェックしない。

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、指定した内容をチェックしない。

例

- 1.3  
ジョブ定義 XML ファイルの DTD のバージョン番号が 1.3 であることを指定する。

version 属性値と対応するバッチジョブ実行システムのバージョンより、実行するバッチジョブ実行システムのバージョンが新しい場合は、そのジョブ定義 XML ファイルが実行できる。一方、version 属性値と対応するバッチジョブ実行システムのバージョンより、実行するバッチジョブ実行システムのバージョンが古い場合、そのジョブ定義 XML ファイルは実行できない。

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
08-01 以降	1.3
08-00	1.2



## 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="windows">
```



# 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。

JOB 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。
- 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。

## 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]) [(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>((0~4095))  
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そこで終了し、後続の STEP 要素や SETMAXRC 要素などの XML 要素の処理は行わない。
- 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 = "1.2" os="windows" >
  <JOB NAME="JOB1" COND="0,LT" MSGCLASS="A">
    <JOBPARM FCB="FCB1"/>
    <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"]  
[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。

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"/>  
:
```

# 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。

## 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atalog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="1.2" os="windows">
  <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ステップ以外の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 D:¥BJEX¥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="D:¥BJEX¥sysout"/>  
</REPLACE>  
</CALL>
```



# REPLACESTEP 要素 (DD 要素または EXEC 要素の属性を変更対象のジョブステップの定義)

---

## 形式

```
NAME="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="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, 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ステップ名部分を NEWSTEP に置き換えてからプロシジャを解析する。

## 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ステップ内だけで有効となる。
- NAME 属性値に指定した環境変数名が、システム環境変数で設定する環境変数名と一致する場合、この環境変数の指定が有効となる。
- NAME 属性値と同じ環境変数がシステム環境変数中にあった場合、次に示す優先順位に従う（優先順位は 1 が最も高い）。
  1. ジョブ定義 XML ファイル内での定義
  2. システム環境変数の定義
- CALL 要素を定義したステップ内に、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 MYNAME に「HITACHI TARO」を設定する場合の使用例を次に示す。

```
:  
<SETENV NAME="MYNAME" VALUE="HITACHI TARO"/>  
:
```

# 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終ジョブ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>((0~4095))

先行ジョブステップの終了コードの最大値を指定する。

## 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。
- SETMAXRC 要素で指定した終了コードは、COND 属性の判定では使用しない。

## 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 属性および DD 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="ASMC" >  
<SETSYMBOL NAME="PARM1" VALUE="C:¥home¥user1"/>  
</CALL>
```

## STEP 要素 (ジョブステップの定義)

---

### 形式

```
[NAME="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, または SETENV 要素  
複数回記述, または省略できる。ただし, このステップ内に CALL 要素を指定した場合は, DD 要素, または SETENV 要素を指定しても無視する。

### 属性

NAME="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

ステップ名に STEP01 という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 要素と DD 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 属性および DD 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、C:¥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="C:¥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"><li>• 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。</li><li>• 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。</li></ul>
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**太字**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
4000~4199	世代コマンド関連メッセージ
5500~5699	bjex_setup (バッチジョブ実行システム動作環境の設定) コマンドのメッセージ
7000~7199	バッチジョブ実行システムサービス処理関連メッセージ
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（イベントログ）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。

## 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	イベントログ
KAKC0001-E	-	○	○	○	○	-	-
KAKC0004-I	-	-	-	-	○	-	-
KAKC0005-E	-	○	○	○	○	-	-
KAKC0020-E	-	○	-	-	○	-	-
KAKC0030-E~KAKC0090-I	-	○	-	-	-	-	-
KAKC0091-I, KAKC0092-I	-	-	○	-	○	○	-
KAKC0093-E	-	○	○	-	○	-	-
KAKC0096-E~KAKC0098-I	-	○	○	-	○	○	-
KAKC0099-I	-	○	-	-	-	-	-
KAKC0100-E	-	-	-	-	-	○	-
KAKC0101-E	-	○	○	-	○	○	-
KAKC0105-E	-	○	-	-	-	-	-
KAKC0161-E~KAKC0165-I	-	-	-	○	-	-	-
KAKC0200-E	-	○	-	-	○	-	-
KAKC0202-E~KAKC0238-E	-	○	-	-	○	-	-
KAKC0299-E	-	○	○	○	○	-	-
KAKC0300-I~KAKC0702-E	-	○	-	-	-	-	-
KAKC0703-E	-	○	-	-	○	-	-
KAKC0704-E~KAKC0706-E	-	○	-	-	-	-	-
KAKC0707-I	-	-	-	○	-	-	-
KAKC0709-E~KAKC0712-E	-	-	-	○	○	-	-
KAKC0719-I~KAKC0802-E	-	○	-	-	-	-	-

メッセージ ID の範囲	stdout	stderr	JOBLOG	SYSMSG	実行ログ	JP1	イベントログ
KAKC0803-E	-	○	-	○	○	-	-
KAKC0804-E	-	○	-	-	-	-	-
KAKC0805-E	-	○	-	○	○	-	-
KAKC0900-I~KAKC1648-E	-	-	-	○	-	-	-
KAKC1603-I	-	-	-	○	-	-	-
KAKC1609-E	-	-	-	○	-	-	-
KAKC1650-W~KAKC1655-I	-	○	○	-	○	○	-
KAKC1660-E, KAKC1680-I	-	-	-	○	-	-	-
KAKC1717-E~KAKC1719-E	-	○	-	-	-	-	-
KAKC1720-E	-	○	-	○※2	○※2	-	-
KAKC1724-E~KAKC1725-E	-	○	-	○	○	-	-
KAKC1726-E	-	○	○	○	○	-	-
KAKC1728-E	-	○	-	○	○	-	-
KAKC1729-E	-	○	○	-	○	-	-
KAKC1734-E	-	○	-	○	○	-	-
KAKC1900-E, KAKC1901-E	-	○※1	-	-	○	-	-
KAKC1905-E	-	○	○	-	○	-	-
KAKC1906-I~KAKC1922-E	-	-	○	-	○	-	-
KAKC1923-W	-	○	○	-	○	-	-
KAKC1924-E	-	-	○	-	○	-	-
KAKC1932-I	-	-	○	-	-	-	-
KAKC1950-E, KAKC1951-E	-	○※1	-	-	○	-	-
KAKC2104-E	-	○	-	-	○	○	-
KAKC2110-I, KAKC2112-E	-	-	-	-	○	-	-
KAKC2120-E~KAKC2205-E	-	○	-	-	○	-	-
KAKC2206-E	-	○	-	-	-	-	-
KAKC2207-E	-	○	-	-	○	-	-
KAKC2208-E~KAKC2209-E	-	○	-	-	-	-	-
KAKC2211-E	-	○	-	-	-	-	-
KAKC2213-E	-	○	-	-	-	-	-

メッセージ ID の範囲	stdout	stderr	JOBLOG	SYSMSG	実行 ログ	JP1	イベン トログ
KAKC2214-E	-	○	-	-	○	-	-
KAKC2215-E	-	○	-	○	-	-	-
KAKC2300-E	-	○	-	-	-	-	-
KAKC2301-E~KAKC2302-E	-	○	-	-	○	-	-
KAKC2303-E	-	○	-	-	-	-	-
KAKC2310-E~KAKC2499-E	-	○	-	-	○	-	-
KAKC4000-E~KAKC4005-E	-	-	-	○	○	-	-
KAKC4010-I, KAKC4011-I	-	○	-	-	-	-	-
KAKC4020-E, KAKC4021-E	-	○	-	○	○	-	-
KAKC4022-E, KAKC4023-E	-	○	-	-	-	-	-
KAKC4024-E~KAKC4026-E	-	-	-	○	○	-	-
KAKC4050-E~KAKC4065-E	-	○	-	○	○	-	-
KAKC4210-E~KAKC4567-E	-	○	-	-	-	-	-
KAKC4600-I	○	-	-	-	-	-	-
KAKC4624-E	-	○	-	-	-	-	-
KAKC4625-E	-	○	-	○	○	-	-
KAKC5000-E	-	-	○	-	○	-	-
KAKC5001-I	-	-	○	-	○	○	-
KAKC5002-I	-	-	-	-	○	-	-
KAKC5003-I	-	-	-	-	○	-	-
KAKC5005-E	-	-	○	-	○	-	-
KAKC5009-E	-	-	-	-	○	-	-
KAKC5010-E	-	-	-	-	○	-	-
KAKC5011-E	-	-	○	-	○	-	-
KAKC5012-I	-	-	○	○	○	-	-
KAKC5013-I	-	-	○	○	○	-	-
KAKC5014-I	-	-	-	-	○	-	-
KAKC5015-E	-	-	○	-	○	-	-
KAKC5016-I	-	-	-	-	○	-	-
KAKC5017-E	-	-	○	-	○	-	-

メッセージ ID の範囲	stdout	stderr	JOBLOG	SYSMSG	実行ログ	JP1	イベントログ
KAKC5018-I	-	-	-	-	○	-	-
KAKC5019-E	-	-	○	-	○	-	-
KAKC5020-E	-	-	○	-	○	-	-
KAKC5021-E	-	-	○	-	-	-	-
KAKC5407-E	-	○	-	-	-	-	-
KAKC5408-I	-	-	-	-	-	○	-
KAKC5409-E~KAKC5429-E	-	○	-	-	-	-	-
KAKC5500-I~KAKC5530-E	-	○	-	-	-	-	-
KAKC7000-I~KAKC7011-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。

SYSMSG：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。

イベントログ：イベントログを示す。

○：出力する。

-：出力しない。

注※1 ジョブステップごとの標準エラーに出力する。

注※2 コマンド実行時は出力しない。

## 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	SYSPRINT	stdout	stderr	JOBLOG	SYSMSG	実行ログ	JP1	イベントログ
KAKD1001-E~KAKD1006-E	○	-	○	-	-	-	-	-
KAKD1009-I	○	-	-	-	-	-	-	-
KAKD1010-E~KAKD1091-E	○	-	○	-	-	-	-	-

メッセージ ID の範囲	SYSPRINT	stdout	stderr	JOBLOG	SYSMSG	実行ログ	JP1	イベントログ
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	○	-	-	-	-	-	-	-
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	○	-	-	-	-	-	-	-



メッセージ ID の範囲	SYSPRINT	stdout	stderr	JOBLOG	SYSMSG	実行ログ	JP1	イベントログ
KAKD3139-E~ KAKD3144-E	○	—	○	—	—	—	—	—

(凡例)

SYSPRINT：DD 名が"SYSPRINT"の DD 要素で指定した出力先を示す。

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。

SYSMSG：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。

イベントログ：イベントログを示す。

○：出力する。

—：出力しない。

## 10.3 メッセージに含まれるエラー情報

バッチジョブ実行システムが出力するメッセージは、C ランタイムの関数および Win32 API のエラー情報を含む場合がある。バッチジョブ実行システムの環境で発生しやすい、代表的な C ランタイム関数のエラー情報に対する原因と対策を次の表に示す。表にないエラーおよび Win32 API のエラー情報については、使用している OS のマニュアルを参照すること。

表 10-6 代表的な C ランタイム関数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	Input/Output 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 要素や SETENV 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"> <li>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。</li> <li>メッセージキューが満杯である。</li> </ul>	<ul style="list-style-type: none"> <li>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。</li> </ul>
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"> <li>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。</li> <li>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。</li> </ul>	<ul style="list-style-type: none"> <li>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。</li> <li>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。</li> </ul>
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	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

エラー番号	エラー詳細	二モニック	考えられる主な原因	対策
14	Bad address	EFAULT	うとしたディスクが切り離されたおそれがある。	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	次のどちらかの原因が考えられる。 1. DD 要素の TYPE 属性値に"FILE"を指定し、DSN 属性値にディレクトリパス名を指定した。 2. メモリ管理情報の不正を検知した。	1. DD 要素を見直し、TYPE 属性値や DSN 属性値を修正すること。 2. システム管理者に連絡すること。
23	Too many open files in system	ENFILE	ファイルのオープン数がシステムの上限を超えた。	システム全体で使用中のファイルの数を確認し、不要なファイルを閉じること。
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。	システム管理者に連絡すること。
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。

## 10.4 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桁の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可能な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。

### KAKC0004-I

Jobid=バッチジョブ実行システムのジョブ識別子, JP1NBQQueueName=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JP1NBQQueueName 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。

## KAKC0020-E

Initialization of bjexecsub process failed. (function=**関数名**, detail=**詳細情報**)

bjexecsub プロセスの初期化に失敗した。要因として、次のことが考えられる。

- bjexecsub.exe を直接実行している。

(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**関数名**と**詳細情報**からエラーの原因を取り除いてジョブを再実行する。

## KAKC0030-E

An error occurred while starting bjexec. function="**関数名**",error code=**エラーコード**,reason="**エラー詳細**"

ジョブコントローラの開始処理中にエラーが発生した。

(S)処理を終了する。

(O)エラー情報を調べ、原因を取り除き、再起動する。

## KAKC0031-E

An error occurred while completing bjexec process. function="**関数名**",error code=**エラーコード**,reason="**エラー詳細**"

ジョブコントローラの終了処理中にエラーが発生した。

(S)処理を終了する。

(O)エラー情報を調べ、原因を取り除き、再起動する。

## KAKC0039-E

An error occurred bjexec. function="**機能名**",error code=**エラー番号**,reason="**エラー詳細**"

ジョブコントローラの実行中にエラーが発生した。

(S)処理を終了する。

(O)エラー情報を調べ、原因を取り除き、ジョブを再実行する。

## 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。

#### 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.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.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が強制終了の要求を受け取った。

(S)ジョブコントローラはジョブステップを強制終了し、ファイルの削除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削除する。

## KAKC0096-E

```
ジョブ名.ステップ名[.プロシジャステップ名] Step ended abnormally. 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.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、「[10.7 終了コード](#)」を参照のこと。

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.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.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  
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: 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。詳細については、「10.7 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)。\_ftime 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)。GetProcessTimes 関数のリターン情報である構造体 FILETIME 内の dwLowDateTime と dwHighDateTime 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。

#### 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。

#### KAKC0161-E

Failed to create work file. (file name=ファイル名, function=関数名, detail=詳細情報)

ジョブコントローラが使用するワークファイルの作成に失敗した。

ワークファイルの名称が取得できないでエラーとなった場合、ファイル名には「"」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。システム管理者はジョブ終了後に、ファイル名が「"」以外の場合はファイル名で示すファイルを削除する。関数名と詳細情報からエラーの原因を取り除いてジョブを再実行する。

#### KAKC0162-I

Work file was created. (file name=ファイル名)

ファイル名で示す、ジョブコントローラが使用するワークファイルを作成した。

(S)処理を続行する。

(O)なし。

#### KAKC0163-I

Work file was deleted. (file name=ファイル名)

ファイル名で示す、ジョブコントローラが使用するワークファイルを削除した。

(S)処理を続行する。

(O)なし。

## KAKC0165-I

Failed to delete work file. (file name=ファイル名, function=関数名, detail=詳細情報)

ジョブコントローラが使用するワークファイルの削除に失敗した。

(S)処理を続行する。

(O)システム管理者に連絡する。システム管理者はジョブ終了後にファイル名で示すファイルを削除し、関数名と詳細情報からエラーの原因を取り除く。

## 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:エラー詳細	スプールジョブ管理ファイルで"エラー詳細"で示すエラーが発生した。
open spool job dir:エラー詳細	スプールジョブディレクトリのオープン中に"エラー詳細"で示すエラーが発生した。
failed to copy:エラー詳細	スプールジョブの保管中に"エラー詳細"で示すエラーが発生した。

(S)スプールジョブの保管処理を中断する。ジョブの実行は続ける。

(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た後、必要に応じてジョブを再実行する。

## KAKC0202-E

Failed to create spool job store directory. (jobid=ジョブ識別子, error=エラー詳細)

保管先のスプールジョブディレクトリ作成処理中に"エラー詳細"で示すエラーが発生した。

ジョブ識別子：スプールジョブのジョブ識別子

(S)スプールジョブの保管処理を中断する。ジョブの実行は続ける。

(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を再実行する。それ以外の場合はシステム管理者に連絡する。

## 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可能な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)ジョブ実行環境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。

## KAKC0408-E

```
Config file does not exist.
```

設定ファイルが存在しない。

(S)処理を終了する。

(O)設定ファイルが存在するか確認する。

## 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のオペランドに誤りがある。

(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。

## 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おそれがある。

(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。

(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。

## 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。

#### KAKC0451-E

An error occurred while getting a default directory: "パラメータ名".

デフォルトのディレクトリ名を求める処理でエラーが発生した。

(S)処理を終了する。

(O)動作環境にディレクトリ作成場所が存在するか確認する。

#### 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。

#### 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,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。

#### KAKC0800-E

Failed to create the sysout.ini. 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。

#### KAKC0802-E

The sysout.ini 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.ini 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可能な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.ini.

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.ini.

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。

#### KAKC0903-E

```
Only Shift-JIS can be specified as encoding attribute.
```

encoding 宣言には Shift\_JIS 以外の値を指定できない。

(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)LANG 環境変数の値が"Ja\_JP"であることを確認し、ジョブ定義 XML ファイルの XML 宣言に encoding="Shift\_JIS"を指定してジョブを再実行する。

#### 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。

#### 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。

## KAKC0910-E

This job cannot be executed on specified OS. elementNo=番号

番号で示すタグの解析中に HitachiBatchJobExec 要素を見つけたが、os 属性に指定された OS ではこのジョブを実行できない。

(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)ジョブ定義 XML ファイルを見直し、os 属性の値を正しく指定してジョブを再実行する。

## 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一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可能な文字でない場合に発生する。要素名として使用可能な文字は、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可能な文字でない場合に発生する。要素名として使用可能な文字は、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、以下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, 以下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.TXT" /> : 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.TXT" /> : 属性間に不当な文字  
<DD % NAME="ERRDD" DSN="INPUT.TXT" /> : 要素名のあとに不当な文字  
<DD NAME="ERRDD" DSN="INPUT.TXT" % /> : 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可能な文字でない場合に発生する。属性名として使用可能な文字は、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 l?>
- 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 "ステップ名" not found. elementNo=番号

番号で示すタグの解析中にステップ名で示すステップを置換しようとしたが、該当する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（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（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 要素を定義しないこと。

## 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 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 "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、属性値に指定したステップ名が見つからない。

ステップ名：ステップ名が表示される場合と、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 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ステップ数が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ステップを含むステップ数が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、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 名には同一ステップ、または指定した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。

### KAKC1129-E

Environment variable "環境変数名" is not set. elementNo=番号

番号で示すタグ内で、DSN 属性に環境変数名で示す環境変数を指定しているが、環境変数が設定されていない。

(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、**環境変数名**が正しいかどうか確認する。正しい場合、**環境変数名**で示す環境変数がジョブコントローラ起動時に設定されるかどうか、システムの設定を確認する。

#### 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可能な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" stat 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 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)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 "ステップ名" is already replaced. elementNo=番号

番号で示すタグの解析中にステップ名で示すステップが見つかったが、この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, SYSNDUMP, 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、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"以外を指定しないこと。

(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、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。

## 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。

## KAKC1163-I

The attribute "属性名" value was converted into "変換後属性値". elementNo=番号

番号で示すタグ内で、属性名に示す属性の値が変換後属性値に変換された。

(S)処理を続行する。

(O)なし。

## 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。

## KAKC1170-E

Unsupported feature "機能名" is specified. elementNo=番号

番号で示すタグ内に機能名で示す機能が指定されているが、サポートしていない機能である。

(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)機能名で示す機能を指定しないようにジョブ定義 XML ファイルを修正し、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確認する。

#### 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.ステップ名 Step allocated.

ジョブ名と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。

なお、ファイル名部分がスプールジョブディレクトリを含む場合、ジョブ実行後にスプールジョブディレクトリの名称がジョブ名を含む内容に変更される点に留意すること。詳細は「[3.1.4\(1\)\(a\)\(2\) スプールジョブディレクトリ](#)」を参照のこと。

(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"DUMMY", "FILE", "TEMP", "GDG", "SYSOUT"を指定した場合に表示される。ショートカットを

指定した場合のファイルサイズは、ショートカットファイルそのものの長さを表示する。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。

なお、ファイル名部分がプールジョブディレクトリを含む場合、ジョブ実行後にプールジョブディレクトリの名称がジョブ名を含む内容に変更される点に留意すること。詳細は「[3.1.4\(1\)\(a\)\(2\) プールジョブディレクトリ](#)」を参照のこと。

(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。

このメッセージは、ファイルを割り当てたあと、ジョブステップが完了するまでの間にエラーが発生した場合にも、出力されることがある。

(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。

このメッセージの前に、エラー要因を示す別のメッセージが出力されている場合は、そのメッセージを参照してエラー要因を取り除く。

割り当てられたファイルがジョブ終了時に削除されないで残った場合、DD 要素の DISP パラメータの指定値を見直す。

#### 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。

(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おそれがある。

(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。

(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）を指定したおそれがある。

(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".KDF"が付けられた名称である。
- キーファイル  
一時 ISAM ファイル名で示す名称に拡張子".K01"が付けられた名称である。
- データファイル  
一時 ISAM ファイル名で示す名称に拡張子".DRF"が付けられた名称である。

## 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".KDF"が付けられた名称である。
- キーファイル  
一時 ISAM ファイル名で示す名称に拡張子".K01"が付けられた名称である。
- データファイル  
一時 ISAM ファイル名で示す名称に拡張子".DRF"が付けられた名称である。

## 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には次のどれかが表示される。

FTYPE：SYSOUT データファイルのファイル形式

RECFM：制御文字の種別

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。

## KAKC1632-E

Length of environment variable value exceeded the limit. (environment variable name=環境変数名)

バッチジョブ実行システムが環境変数名で示す名称の環境変数を設定しようとしたが、値の長さがバッチジョブ実行システムで定められた上限値を超えた。次の要因が考えられる。

- バッチジョブ実行システムに JOBLIB DD 要素、または STEPLIB DD 要素を指定しており、PATH 環境変数の値の長さが、バッチジョブ実行システムで定められた制限を超えた。詳細は「[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。
- ジョブ定義 XML ファイルにファイルの連結処理を指定しており、区切り文字で連結したパス文字列の長さが、バッチジョブ実行システムで定められた制限を超えた。詳細は「[3.4.3 ファイルの連結処理](#)」を参照のこと。
- ジョブ定義 XML ファイルに世代データ群名指定をしており、区切り文字で連結したパス文字列の長さが、バッチジョブ実行システムで定められた制限を超えた。詳細は「[3.7.3 世代ファイルの作成と参照](#)」を参照のこと。

(S)処理を終了する。

(O)ジョブ定義 XML ファイルやシステムで定義した PATH 環境変数の値を見直し、環境変数の値の長さが上限値を超えないよう対処してジョブを再実行する。

## 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 mutex.

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 mutex.

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。

## 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 mutex 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 Mutex 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。

- BjexSvc サービスが開始されていない。
- BjexSvc サービスの開始時に参照した設定ファイルの LOCK\_OPTION パラメータで ENABLE が指定されていない。
- BjexSvc サービスの開始時に参照した設定ファイルの JOB\_STATUS\_DISPLAY パラメータで ENABLE が指定されていない。
- 設定ファイルの LOCK\_OPTION パラメータに ENABLE を設定したあと、設定ファイルを有効にするための BjexSvc サービスの再起動をしていない。
- 設定ファイルの JOB\_STATUS\_DISPLAY パラメータに ENABLE を設定したあと、設定ファイルを有効にするための BjexSvc サービスの再起動をしていない。
- bjexec コマンド、bjexlslock コマンド、および bjexlsexjob コマンドを実行するローカルユーザが、バッチジョブ実行システムをインストールしたときに生成されるローカルグループに登録されていない。

(S)処理を中止する。

(O)要因に応じて次の対処をする。

- コマンド実行に先立ち、BjexSvc サービスを起動しておく。
- 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。
- 設定ファイルの指定 (LOCK\_OPTION パラメータに ENABLE を指定) をした場合、設定ファイルを有効にするために BjexSvc サービスを再起動する。
- 設定ファイルの指定 (JOB\_STATUS\_DISPLAY パラメータに ENABLE を指定) をした場合、設定ファイルを有効にするために BjexSvc サービスを再起動する。
- bjexec コマンド、bjexlslock コマンド、および bjexlsexjob コマンドを実行するローカルユーザを、バッチジョブ実行システムをインストールしたときに生成されるローカルグループに登録する。

上記以外の要因の場合、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、共有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。

## 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 the maximum.

指定したファイルパスが長過ぎる。

(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。

## KAKC1734-E

An error occurred during lock release processing. (function=**関数名**)

ジョブステップ終了時の排他/共有制御の解除処理で、エラーが発生した。

(S)ジョブステップ終了時の排他/共有制御を解除処理しないで、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、マニュアルの6章「**障害対策**」に従って資料を採取し、BjexSvc サービスを再起動する。

## 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。

## 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.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.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。

ジョブステップが終了コード 1, 2, 3 で異常終了すると、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係なくこのメッセージが出力される。

(S)処理を続行する。

(O)なし。

## 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。

## KAKC1913-W

ジョブ名 The maximum value of return code is adjusted.

ジョブステップの終了コードの最大値が、次のどれかである。

- すべてのジョブステップが正常終了し、ジョブステップの最大の終了コードが 4096 以上、または-1 以下である。
- PGM\_RCABEND パラメータに指定した終了コードである。  
PGM\_RCABEND パラメータの指定を省略した場合は、1, 2, 3 が仮定される。

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)ジョブのリターンコードを「[10.7 終了コード](#)」で示すジョブコントローラの終了コードとし、処理を続行する。

(O)このコードで問題ないか確認する。

## KAKC1914-W

ジョブ名 The last value of return code is adjusted.

最後に実行完了したジョブステップの終了コードが、次のどれかである。

- 4096 以上、または-1 以下である。
- PGM\_RCABEND パラメータに指定した終了コードである。  
PGM\_RCABEND パラメータの指定を省略した場合は、1, 2, 3 が仮定される。

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)ジョブのリターンコードを「[10.7 終了コード](#)」で示すジョブコントローラの終了コードとし、処理を続行する。

(O)このコードで問題ないか確認する。

## 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。

## KAKC1922-E

Failed to terminate step process. (step=ステップ名, function=関数名, detail=詳細情報)

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。

(O)システム管理者に連絡する。システム管理者は関数名と詳細情報からエラーの原因を取り除き、ステップ名で示すジョブステップのプロセスが終了していなければ、そのプロセスを強制終了する。

## KAKC1923-W

Step process was terminated. (step=ステップ名)

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。

## KAKC1924-E

Failed to create step process. (step=ステップ名, function=関数名, detail=詳細情報)

ステップ名で示すジョブステップのプログラムを実行しようとしたが、エラーが発生したため実行できなかった。関数名が「AssignProcessToJobObject」であり、詳細情報が「Access is denied.」の場合は、Advanced Shell からジョブコントローラを起動している可能性がある。この場合、ジョブオブジェクトに関する処理は Advanced Shell ですでに実行されているため、このメッセージが出力されても(O)に示す対処は必要ない。ただし、バッチジョブを強制終了したとき、ジョブステップのプログラムが作成したプロセスについては終了しない。

(S)処理を終了する。ただし、ジョブオブジェクトへのプロセスの関連づけでエラーが発生した場合は、処理を継続する。

(O)関数名と詳細情報からエラーの原因を取り除き、ジョブを再実行する。

LANG="Java"指定のときは cjexecjob のパスが通っていない場合があるので確認すること。

#### 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。

#### 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。

#### KAKC2104-E

```
Failed to transfer spool job file.  
スプールジョブの転送に失敗しました
```

スプールジョブの転送時、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。

## KAKC2110-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,出力先識別子,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。

## KAKC2112-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,出力先識別子,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おそれがある。エラー情報を調べ, 原因を取り除く。

## KAKC2120-E

An error occurred during file processing. (file または directory=パス名, function=関数名, detail=詳細情報)

スプールジョブ転送処理でファイルエラーが発生した。

file または directory が file の場合, パス名にはファイル名, directory の場合, パス名にはディレクトリ名が表示される。関数名および詳細情報はエラー情報を示す。

(S)処理を続行する。

(O)関数名と詳細情報からエラーの原因を取り除き, ジョブを再実行する。

## KAKC2199-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処理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おそれがあるため、システム管理者に連絡する。

#### 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 JOBLLOG 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、設定ファイルのスプ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、設定ファイルのスプ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、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)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。

(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, os\_code=保守情報 2

トレースの初期化に失敗した。

(S)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。

(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 32, 33 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)処理を続行する。起動したジョブではトレースヘ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)処理を続行する。起動したジョブではトレースヘ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、処理を続行する。起動したジョブでの実行ログヘ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32, 33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可能な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可能な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可能な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可能な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可能な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可能な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可能な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可能な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可能な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可能な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可能な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可能な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可能な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可能な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可能な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可能な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。

#### 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 相対世代番号 is not exist.
```

相対世代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NOSCRATCH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(-I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)] gdg file of the specified generation data group is not found.

世代群名[(相対世代番号)]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一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。

(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。

#### 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。

(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 0001～0255 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。

#### 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
EXEC	実行中

(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。

#### 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。

#### 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、コマンドを再入力する。

#### 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。

### 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)該当ジョブのスプール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スプールジョブ管理ファイルは空のファイルである。

(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)該当ジョブのスプール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、スプールジョブ管理ファイルが破壊されている場合は、該当スプールジョブを強制削除して、ジョブを再実行する。

スプールジョブ管理ファイルが破壊されている場合、該当するスプール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スプールジョブの 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スプールジョブのファイルではなく、スプールジョブ管理ファイルが破壊されているか、スプールディレクトリ名を変更したなどのおそれがある。

#### KAKC4314-E

The spool job control file is not accessible. (jobid = ジョブ識別子, error message = エラー詳細)

スプール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。

#### 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ないで残っている場合は、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、ほかで使用中的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入力する。

#### KAKC4364-E

An error occurred while searching for a spool job. (process = 処理など, error message = エラー詳細)

スプールジョブの検索処理でエラーが発生した。

(S)スプールジョブの処理を中断する。

(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。

#### 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。

#### 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。



## 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。

## 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。

## 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 process info. pid=プロセス番号 (code=保守情報,errormessage=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。

## KAKC4625-E

An error occurred during the security information setting of the process. 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使用時、起動したプロセスに対するセキュリティ情報の設定中にエラーが発生した。

bjexlsexjob コマンドでは、エラーが発生したプロセスの C-TIME およびプロセスの優先度が表示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。

#### 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)BjexSvc サービスを起動して、ジョブを再実行する。

## 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.

SYSMSG 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。

#### 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。

## 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。

## KAKC5500-I

Start verification of a template of configuration file.

設定ファイルのテンプレートの解析を開始する。

(S)処理を続行する。

(O)なし。

## KAKC5501-I

Start verification of a modified configuration file.

作成した設定ファイルの解析を開始する。

(S)処理を続行する。

(O)なし。

## KAKC5502-I

The specified parameters in the configuration file were correct.

設定ファイルのパラメータは正しい。

(S)処理を続行する。

(O)なし。

## KAKC5503-E

The specified parameters in the configuration file were incorrect.

設定ファイルのパラメータが正しくない。

(S)処理を終了する。

(O)設定ファイルのパラメータを正しく指定したあとに、コマンドを再実行する。



## KAKC5511-E

An error occurred while getting a path. (パスの種別名, エラー内容)

設定環境のディレクトリを求める処理でエラーが発生した。

### パスの種別名

MyDocuments：ユーザごとのドキュメント格納先のパス  
APPDATA：ユーザごとのアプリケーションデータのパス  
ALLUSERAPP：全ユーザ共通のアプリケーションデータの格納先のパス  
ALLUSERDOC：全ユーザ共通のドキュメント格納先のパス  
USERPROFILE：ユーザプロファイルのディレクトリパス  
spool：スプールディレクトリ  
conf：設定ファイル格納ディレクトリ  
log：ログ用ディレクトリ  
perm：恒久ファイル用ディレクトリ  
proc：カタログドプロシジャ用ディレクトリ  
trace：トレース用ディレクトリ  
temp：一時ディレクトリ  
BJEX：バッチジョブ実行システムのインストールディレクトリ

(S)処理を終了する。

(O)該当ディレクトリが存在するか、パス名が長過ぎないか確認する。パス名が長過ぎる場合、パス名が短くなるように設定環境を変更する。

## KAKC5520-E

A template of configuration file is missing.

設定ファイルのテンプレートファイルの格納先を見つけることができない。

バッチジョブ実行システムがインストールされていないか、レジストリが破壊されているおそれがある。

(S)処理を終了する。

(O)バッチジョブ実行システムを正しくインストールする。

## KAKC5521-E

A size of template configuration file name is too long.

設定ファイルのテンプレートファイルのファイル名が長過ぎる。

レジストリが破壊されているか、インストール先ディレクトリのパス名が長過ぎるおそれがある。

(S)処理を終了する。

(O)インストールの設定を見直してから、バッチジョブ実行システムを再度インストールする。

#### KAKC5522-E

A size of configuration file name is 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)設定ファイルのファイル名を見直したあとに、コマンドを再実行する。

#### KAKC5528-E

An error occurred while opening a configuration file. (エラー内容)

bjex\_setup コマンドで設定ファイルを作成するときに、設定ファイルのオープンでエラーが発生した。

(S)処理を終了する。

(O)エラー内容に従って原因を調査し、障害を取り除いてコマンドを再実行する。

#### KAKC5529-E

An error occurred while writing a configuration file. (エラー内容)

bjex\_setup コマンドで設定ファイルを作成するときに、設定ファイルへの書き込みでエラーが発生した。

(S)処理を終了する。

(O)エラー内容に従って原因を調査し、障害を取り除いてコマンドを再実行する。

#### KAKC5530-E

An error occurred while creating a directory. (ディレクトリの種別, エラー内容)

ディレクトリ作成時にエラーが発生した。

ディレクトリの種別とエラー内容を基に障害を取り除いて、コマンドを再実行する。

##### ディレクトリの種別

spool : スプールディレクトリ

conf : 設定ファイル格納ディレクトリ

log : ログ用ディレクトリ

perm : 恒久ファイル用ディレクトリ

proc：カタログドプロシジャ用ディレクトリ  
trace：トレース用ディレクトリ  
temp：一時ディレクトリ

(S)処理を終了する。

(O)設定ファイルを削除したあとにコマンドを再実行する。

#### KAKC7000-I

Service installed successfully.

BjexSvc サービスの登録が成功した。

(S)処理を続行する。

(O)なし。

#### KAKC7001-E

Service installed unsuccessfully. (API=保守情報,Error code=エラーコード)

BjexSvc サービスの登録に失敗した。

(S)処理を中止する。

(O)BjexSvc サービスがすでに登録されている場合、このメッセージは無視する。それ以外の場合は、システム管理者に連絡する。

#### KAKC7002-E

BjexSvc initialize error. (API=保守情報,Error code=エラーコード)

BjexSvc サービスのサービス初期化処理でエラーが発生した。次の要因が考えられる。

- BjexSvc サービスを起動するユーザアカウントが管理者権限を持っていない。
- バッチジョブ実行システムのセットアップで、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ための設定をしていない。

(S)処理を中止する。

(O)要因に応じて次の対処をする。それ以外の場合は、システム管理者に連絡する。

- BjexSvc サービスのプロパティでログオンするアカウントに指定したユーザに、管理者権限を持つユーザを指定する。

- bjexsvc -setup コマンドを実行し、バッチジョブ実行システムで排他/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ための設定を行う。詳細については、「2.3.1(5) 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ための設定」を参照のこと。

## KAKC7003-E

Failed to API 名. (Error code=エラーコード)

BjexSvc サービスの処理中に Mutex の生成またはオープンに失敗した。

BjexSvc サービスを起動するアカウントにサーバの管理者権限を持つユーザを指定していない場合にこのエラーとなることがある。

(S)処理を中止する。

(O)指定に誤りがある場合、誤りを修正して再実行する。それ以外の場合は、システム管理者に連絡する。

## KAKC7004-E

Config file does not exist.

設定ファイルが存在しない。

(S)処理を中止する。

(O)設定ファイルが存在するか確認する。

## KAKC7005-E

Parsing config file ended. (Error code=エラーコード)

設定ファイルのパラメータの指定に誤りがあるため、解析を終了した。

(S)処理を中止する。

(O)bjex\_setup コマンドで-k オプションを指定して設定ファイルのシンタクスチェックを行い、誤りがあれば修正する。

## KAKC7010-I

Service setup successfully.

BjexSvc サービスの設定処理が成功した。

(S)処理を続行する。

(O)なし。

## KAKC7011-E

Service setup unsuccessfully. (API=**保守情報**, Error code =**エラーコード**)

BjexSvc サービスの設定処理が失敗した。

(S)処理を中止する。

(O)システム管理者に連絡する。

## 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あと再実行する。

#### 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 文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超えた。

(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。

INPFILE 文を制御文の最初に指定する必要がある。

OUTFILE 文は、INPFILE 文の次に指定する必要がある。

GENERATE 文は、INPFILE 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 文でないにも関わらず、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 RECLEN 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。

## 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。

(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す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。

INPFILE 文を制御文の最初に指定する必要がある。

OUTFILE 文は、INPFILE 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製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。

## 10.5 JP1 イベント一覧

JP1 イベント一覧を次の表に示す。

表 10-7 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
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
00005C0C	スプールジョブ転送失敗時	KAKC2104-E
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
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 が出力するメッセージ
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 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	BjexSvc サービス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

## 10.6 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。

PC ジョブの場合

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

サブミットジョブの場合

bjexec となる。

表 10-8 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"> <li>• PC ジョブの場合 JP1/AJS のジョブ情報:bjexec</li> <li>• サブミットジョブの場合 bjexec</li> </ul>
		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 10-9 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 (1970/1/1 00:00:00 (UTC (Coordinated Universal Time))) からの通算秒)
	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 10-10 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 (1970/1/1 00:00:00 (UTC) からの通算秒)
		終了時刻	END_TIME	ジョブ終了時刻 (1970/1/1 00:00:00 (UTC) からの通算秒)
		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 10-11 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 (1970/1/1 00:00:00 (UTC) からの通算秒)
		終了時刻	END_TIME	ジョブ終了時刻 (1970/1/1 00:00:00 (UTC) からの通算秒)
		終了コード	RESULT_CODE	終了コード

属性種別		項目	属性名	内容
拡張属性	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
		エラー詳細	ERROR_INFO	エラー詳細

(凡例)

－：なし。

表 10-12 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.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.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 (1970/1/1 00:00:00 (UTC) からの通算秒)
	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 10-13 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
		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,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,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 (1970/1/1 00:00:00 (UTC) からの通算秒)
		終了時刻	END_TIME	ジョブステップ終了時刻 (1970/1/1 00:00:00 (UTC) からの通算秒)
		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 10-14 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,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
		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,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 (1970/1/1 00:00:00 (UTC) からの通算秒)
		終了時刻	END_TIME	ジョブステップ終了時刻 (1970/1/1 00:00:00 (UTC) からの通算秒)
	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 10-15 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 10-16 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
		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

属性種別		項目	属性名	内容
拡張属性	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 10-17 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
		ジョブ名	JOB_NAME	ジョブ名

(凡例)

－：なし。

表 10-18 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

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	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 (00005C0C の詳細)

属性種別		項目	属性名	内容
基本属性		イベント ID	－	00005C0C
		メッセージ	－	KAKC2104-E スプールジョブの転送に失敗しました
拡張属性	共通情報	重大度	SEVERITY	Error
		ユーザ名	USER_NAME	JP1 ユーザ名
		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
		オブジェクト ID	OBJECT_ID	－
		事象種別	OCCURRENCE	NOTICE
		開始時刻	START_TIME	－
		終了時刻	END_TIME	－
		終了コード	RESULT_CODE	－
	固有情報	ジョブ定義 XML ファイル名	XML_JCL	－

属性種別		項目	属性名	内容
拡張属性	固有情報	ジョブ通し番号	JOB_ID	—
		ジョブ名	JOB_NAME	—
		ジョブステップ番号	STEP_ID	—
		ジョブステップ名	STEP_NAME	—
		エラー詳細	ERROR_INFO	エラー詳細

(凡例)

—：なし。

表 10-20 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
		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 (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

属性種別		項目	属性名	内容
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 10-22 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

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
		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 10-23 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
		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 10-24 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

属性種別		項目	属性名	内容
基本属性		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 10-25 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	BjexSvc Service
		登録名タイプ	ROOT_OBJECT_TYPE	SERVICE
		登録名	ROOT_OBJECT_NAME	BjexSvc Service
		事象種別	OCCURRENCE	CANCEL



属性種別		項目	属性名	内容
拡張属性	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 10-26 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
		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 10-27 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)

属性種別		項目	属性名	内容
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。

## 10.7 終了コード

### 10.7.1 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（プログラムコマンド、スクリプト、Java アプリケーション）の終了コード、またはジョブコントローラがエラーを検出した場合に設定する値である。

EXEC 要素に記述したプログラムの終了コードが PGM\_RCABEND パラメータに指定した値と異なる場合、ジョブステップの終了コードを 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の終了コードが PGM\_RCABEND パラメータに指定した値と同じである場合、またはジョブコントローラがエラーを検出して処理を終了した場合は、ジョブステップの終了コードを KAKC0096-E メッセージに出力する。なお、PGM\_RCABEND パラメータに何も指定しなかった場合、1, 2, 3 が仮定される。

ジョブステップの終了コード一覧を次の表に示す。

表 10-28 ジョブステップの終了コード一覧

終了コード	意味
0	EXEC 要素に記述したプログラムの終了コードが 0 である。
1	次のどちらかである。 <ul style="list-style-type: none"><li>EXEC 要素に記述したプログラムまたは cjexecjob コマンドが異常終了した。</li><li>EXEC 要素に記述したプログラムの終了コードが 1 である。</li></ul>
2	次のどちらかである。 <ul style="list-style-type: none"><li>EXEC 要素に記述したプログラムまたは cjexecjob コマンドが異常終了した。</li><li>EXEC 要素に記述したプログラムの終了コードが 2 である。</li></ul>
3	次のどちらかである。 <ul style="list-style-type: none"><li>EXEC 要素に記述したプログラムまたは cjexecjob コマンドが異常終了した。</li><li>EXEC 要素に記述したプログラムの終了コードが 3 である。</li></ul>
10000 以上	次のどちらかである。 <ul style="list-style-type: none"><li>KAKC0096-E メッセージを出力した場合、ジョブコントローラが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または、ジョブステップを実行したが終了コードを取得できなかった。</li><li>KAKC0097-I メッセージを出力した場合、EXEC 要素に記述したプログラムの終了コードが 10000 以上である。</li></ul>
上記以外	EXEC 要素に記述したプログラムの終了コード。

## 10.7.2 ジョブコントローラの終了コード

ジョブコントローラの終了コードは次の値であり、KAKC0099-I メッセージに出力する。

### (1) 異常終了したジョブステップが存在しない場合

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、最終ジョブステップの終了コードである。

ただし、-1 以下の終了コード、または 4096 以上の終了コードのジョブステップが存在する場合は、ジョブコントローラの終了コードを 4096 とする。

SETMAXRC 要素で変更した場合は、その値に従う。

### (2) 異常終了したジョブステップが存在する場合

JOB\_RCDEFINE パラメータで STEPMAX を指定した場合

異常終了したジョブステップ内での最大の終了コードに 5000 を加算した値である。

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、最終ジョブステップの終了コードである。

最終ジョブステップが異常終了した場合、終了コードに 5000 を加算した値である。

SETMAXRC 要素で変更した場合は、その値に従う。

### (3) ジョブコントローラがエラーを検出した場合

ジョブコントローラがエラーを検出して処理を終了した場合は、10000 以上の値である。

SETMAXRC 要素で変更はできない。

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 10-29 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"><li>• -v オプションを指定した場合、正常終了した。</li><li>• -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。</li><li>• 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。</li><li>• JOB_RCDEFINE パラメータで STEPLAST を指定した場合</li></ul>

終了コード	意味
0	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。
1~4095	<ul style="list-style-type: none"> <li>• JOB_RCDEFINE パラメータで STEPMAX を指定した場合次の両方の条件を満たす。 <ul style="list-style-type: none"> <li>・ジョブステップの最大の終了コードが1~4095である。</li> <li>・異常終了したジョブステップが存在しない。</li> </ul> </li> <li>• 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1~4095である。</li> </ul>
4096	<ul style="list-style-type: none"> <li>• JOB_RCDEFINE パラメータで STEPMAX を指定した場合次の両方の条件を満たす。 <ul style="list-style-type: none"> <li>・ジョブステップの最大の終了コードが-1以下、または4096以上である。</li> <li>・異常終了したジョブステップが存在しない。</li> </ul> </li> <li>• JOB_RCDEFINE パラメータで STEPLAST を指定した場合次の両方の条件を満たす。 <ul style="list-style-type: none"> <li>・最終ジョブステップの終了コードが-1以下、または4096以上である。</li> <li>・最終ジョブステップが正常終了した。</li> </ul> </li> </ul>
5000~9096	<ul style="list-style-type: none"> <li>• JOB_RCDEFINE パラメータで STEPMAX を指定した場合異常終了したジョブステップが存在する。</li> <li>• 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が異常終了した。</li> </ul>
10010	<p>ジョブコントローラが次に示すエラーを検出したため、処理を終了した。</p> <ul style="list-style-type: none"> <li>• ファイルの割り当てに失敗した。</li> <li>• ディレクトリの割り当てに失敗した。</li> <li>• 世代ファイルの割り当てに失敗した。</li> </ul>
10020	<p>ジョブコントローラが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。</p> <ul style="list-style-type: none"> <li>• 入力パラメータの文法エラー。</li> <li>• 設定ファイルの文法エラー。</li> <li>• ジョブ定義 XML ファイルの文法エラー。</li> </ul>
10030	<p>ジョブコントローラが次に示すエラーを検出したため、処理を終了した。</p> <ul style="list-style-type: none"> <li>• DD 要素で指定した一時ファイルが存在しない。</li> <li>• DD 要素で指定した一時 ISAM ファイルが存在しない。</li> <li>• 世代管理ファイルの割り当てエラー。</li> </ul>
10040	<p>ジョブコントローラが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。</p> <ul style="list-style-type: none"> <li>• 設定ファイルのオープンエラー、入出力エラー。</li> <li>• ジョブ定義 XML ファイルのオープンエラー、入出力エラー。</li> <li>• ジョブログファイルのオープンエラー、入出力エラー。</li> <li>• ISAM ライブラリのロードエラー。</li> <li>• 世代管理ファイルのオープンエラー、入出力エラー。</li> </ul>
10050	<p>ジョブコントローラが次に示すエラーを検出したため、処理を終了した。</p>

終了コード	意味
10050	<ul style="list-style-type: none"> <li>ジョブステッププロセス生成エラー。</li> <li>ジョブステッププロセス終了待ちエラー。</li> </ul>
10060	メモリ割り当てエラーなど、ジョブコントローラが続行できないエラーを検出したため、処理を終了した。
10070	ジョブコントローラがメッセージを出力できなかったため、処理を終了した。
10080	プログラムの内部矛盾検出など、ジョブコントローラが内部エラーを検出したため、処理を中断した。
10090	ジョブの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。

## (4) 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="windows">
  <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>
```

ジョブコントローラの終了コード

表 10-30 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	
		STEPSMAX	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ジョブ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ジョブ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。

### 10.7.3 注意事項

- EXEC 要素に記述したコマンドやプログラムの終了コードが PGM\_RCABEND パラメータの指定値と一致した場合、ジョブコントローラはプログラムが異常終了したと判断する。PGM\_RCABEND パラメータに何も指定していない場合、1, 2, 3 が仮定される。

testcommand プログラムを呼び出したあとに、Windows バッチファイルの ERRORLEVEL 変数を参照して終了コードを変更する例を次に示す。実行完了時に設定される終了コードはコマンドやプログラムによって異なるため、使用するコマンドやプログラムごとに終了コードに合ったエラー処理を記述すること。

```
<EXEC PGM="*">
<![CDATA[
!
testcommand
IF ERRORLEVEL 1 GOTO ERROREND
exit 0

: ERROREND
IF ERRORLEVEL 4 EXIT %ERRORLEVEL%
IF ERRORLEVEL 3 EXIT 4003
IF ERRORLEVEL 2 EXIT 4002
IF ERRORLEVEL 1 EXIT 4001
!
]]>
</EXEC>
```

- JOB 要素または EXEC 要素にジョブステップの終了コードを評価する COND 属性を指定した場合、評価にはジョブステップの終了コードの下位 12 ビット (0~4095) を使用する。したがって、EXEC 要素に記述するプログラムの終了コードは (0~4095) の値を使用すること。
- LANG="Java"を指定した場合のジョブステップの終了コードは cjexecjob の戻り値に従う。cjexecjob の戻り値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。
- 転送処理で失敗した場合でも、ジョブコントローラの終了コードは KAKC0099-I メッセージに出力した値である。

# 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-#xFFFF] |
           [#x10000-#x10FFFF]
[ 3] S ::= (#x20 | #x9 | #xD | #xA)+
[ 4] NameChar ::= Letter | Digit | '.' | '-' | '_' | ':'
[ 5] Name ::= (Letter | '_' | ':') (NameChar)*
[10] AttValue ::= '"' ([^&"] | Reference)* '"' |
             "'" ([^&' ] | Reference)* "'"
[14] CharData ::= [^&]* - ([^&]* '&#x[0-9A-Fa-f]+')* [^&]*
[15] Comment ::= <!-- ((Char - '-') | ('-' (Char - '-')))* -->
[18] CDsect ::= CDStart CData CDEnd
[19] CDStart ::= <![CDATA[
[20] CData ::= (Char* - (Char* '&#x[0-9A-Fa-f]+'))* 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 ::= [#x0041-#x005A] | [#x0061-#x007A]
[88] Digit ::= [#x0030-#x0039]
```

### 付録 A.1 ジョブ定義 XML ファイルの DTD

ジョブ定義 XML ファイルの DTD を次に示す。

```
<!ELEMENT HitachiBatchJobExec (JOB|PROC)>
<!ELEMENT JOB (JOBPARM?, (DD|SETENV)*, PROC*, (STEP, SETMAXRC?)+)>
<!ELEMENT JOBPARM EMPTY>
<!ELEMENT DD (#PCDATA)>
<!ELEMENT SETENV EMPTY>
<!ELEMENT PROC (SYMBOL*, (STEP, SETMAXRC?)+)>
<!ELEMENT STEP ((EXEC|CALL), (DD|SETENV)*)>
<!ELEMENT SETMAXRC EMPTY>
<!ELEMENT SYMBOL EMPTY>
<!ELEMENT EXEC (#PCDATA)>
```

```

<!ELEMENT CALL (SETSYMBOL*, REPLACE?)>
<!ELEMENT SETSYMBOL EMPTY>
<!ELEMENT REPLACE (EXEC?, DD*, REPLACESTEP*)>
<!ELEMENT REPLACESTEP (EXEC?, DD*)>

<!ATTLIST HitachiBatchJobExec
  version CDATA #IMPLIED
  os      CDATA #IMPLIED
>
<!ATTLIST JOB
  NAME      CDATA #REQUIRED
  COND      CDATA #IMPLIED
  MSGCLASS CDATA #IMPLIED
>
<!ATTLIST JOBPARM
  FORMS      CDATA #IMPLIED
  FCB        CDATA #IMPLIED
  ROOM       CDATA #IMPLIED
>
<!ATTLIST DD
  NAME      CDATA #REQUIRED
  TYPE      (DATA|DUMMY|TEMP|TEMPISAM|GDG|FILE|LIB|SYSOUT|DIR)#IMPLIED
  DSN       CDATA #IMPLIED
  DISP      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) #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
>
<!ATTLIST SETENV
  NAME      CDATA #REQUIRED
  VALUE     CDATA #REQUIRED
>
<!ATTLIST PROC
  NAME      CDATA #REQUIRED
>
<!ATTLIST STEP

```

```

NAME      CDATA      #IMPLIED
>
<!ATTLIST SETMAXRC
VALUE     CDATA      #REQUIRED
>
<!ATTLIST SYMBOL
NAME      CDATA      #REQUIRED
VALUE     CDATA      #IMPLIED
>
<!ATTLIST EXEC
PGM       CDATA      #IMPLIED
PARM      CDATA      #IMPLIED
COND      CDATA      #IMPLIED
LANG      (COBOL|Java) #IMPLIED
PARMSPLIT (YES|NO) #IMPLIED
Group     CDATA      #IMPLIED
Option    CDATA      #IMPLIED
>
<!ATTLIST CALL
NAME      CDATA      #REQUIRED
>
<!ATTLIST SETSYMBOL
NAME      CDATA      #REQUIRED
VALUE     CDATA      #REQUIRED
>
<!ATTLIST REPLACESTEP
NAME      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。

タスクマネージャを使って、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。

表 B-1 タスクマネージャに表示されるプロセス一覧

名前	機能	ユーザ名	機能の詳細
bjexec.exe	ジョブコントローラ	JP1 ユーザ名※	ジョブの監視を行う。
bjexecsub.exe	ジョブコントローラ	JP1 ユーザ名※	ジョブの制御を行う。
bjexsvc.exe	BjexSvc サービス	bjexsvc サービスを起動したユーザ名	共有メモリおよび Mutex の管理を行う。

注※ バッチジョブ実行システムを起動した JP1/AJS のユーザ名。

## 付録 C 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 C.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。

#### COBOL 関連

- COBOL2002 ユーザーズガイド (3021-3-600)
- COBOL2002 操作ガイド (3021-3-601)
- COBOL2002 言語 標準仕様編 (3021-3-604)
- COBOL2002 言語 拡張仕様編 (3021-3-605)

#### Cosminexus 関連

- Cosminexus アプリケーションサーバ V8 概説 (3020-3-U01)
- Cosminexus アプリケーションサーバ V8 機能解説 拡張編 (3020-3-U08)
- 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)
- Cosminexus アプリケーションサーバ V8 リファレンス 定義編 (サーバ定義) (3020-3-U15)

#### 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)

## PRINT DATA EXCHANGE 関連

- PRINT DATA EXCHANGE for Open (3020-3-P02)
- PRINT DATA EXCHANGE - Form Designer (3020-3-P03)

## その他

- ソートマージ (3020-3-N73)
- 索引順編成ファイル管理 ISAM (3020-3-D88)

## 付録 C.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	
COBOL2002 Net Server Runtime	COBOL2002	
COBOL2002 Net Server Suite		
ISAM Version7	ISAM	
JP1/Advanced Shell	Advanced Shell	
Advanced Shell		
JP1/Automatic Job Management System 3 - Agent	JP1/AJS3	JP1/AJS
JP1/Automatic Job Management System 3 - Manager		
JP1/Automatic Job Management System 3 - View		
JP1/Integrated Management 2 - Manager	JP1/IM - Manager	JP1/IM
JP1/Integrated Management 3 - Manager		
JP1/Integrated Management 2 - View	JP1/IM - View	
JP1/Integrated Management 3 - View		
PRINT DATA EXCHANGE - Electronic Document Option	PDE	

正式名称	このマニュアルでの表記
PRINT DATA EXCHANGE - Enterprise Printer Option	PDE
PRINT DATA EXCHANGE - Form Designer	
PRINT DATA EXCHANGE for Open	
SORT Version9	SORT EE
SORT Version9 - Extended Edition	
uCosminexus Application Server Enterprise	Cosminexus
uCosminexus Batch Job Execution Server	バッチジョブ実行システムまたはBJEX
Virtual-storage Operating System 3	VOS3

### 付録 C.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。



## 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。

こ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	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	2	共有メモリの参照に失敗した。 BjexSvc サービスが起動していない可能性がある。	BjexSvc サービス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。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。
	258	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。
	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。	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。
	409	同一 PID から応答要求メッセージが出力された。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。	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。
	Win32 API のエラー情報	「10.3 メッセージに含まれるエラー情報」を参照のこと。	

戻り値	エラー番号	意味	意味対処
3	C ランタイム関数のエラー情報	「10.3 メッセージに含まれるエラー情報」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。	

戻り値	エラー番号	意味	意味対処
4 および 5	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。
	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。

### (英字)

#### 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。

#### 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。

#### XML (eXtensible Markup Language)

拡張可能な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一つである。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 を指定することによって定義できる。

#### 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 を指定することによって定義できる。

#### 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。

## (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によって内容が展開され、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い、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。

プロシジャの部分的修正を可能にし、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。

### 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 文書に含まれる一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。

## 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。

## 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、世代管理情報を更新するかどうかを判定し、登録の要否を決定する機能である。

## 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。

## 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 を指定することによって定義される。バッチジョブ実行システムでは、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。

### ファイル操作ユーティリティ

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ー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。

## (マ行)

### メタキャラクタ

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：'
- バックスラッシュまたは円記号
- 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

## (ヤ行)

### 要素 (Element)

XML 文書での情報の単位である。

# 索引

## 記号

- a [bjexrmjob コマンド] 343
- c [bjexec コマンド] 315
- c [bjexlsexjob コマンド] 320
- D LT [bjexlsjob コマンド] 327
- D UTC [bjexlsjob コマンド] 327
- f [bjexlsjob コマンド] 327
- f [bjexrmjob コマンド] 343
- h [bjexec コマンド] 315
- h [bjexlsexjob コマンド] 320
- h [bjexlsjob コマンド] 326
- h [bjexlslock コマンド] 333
- H [bjexrmjob コマンド] 343
- h [bjexrmjob コマンド] 342
- i [bjexgdg コマンド] 317
- j [bjexlsexjob コマンド] 320
- J [bjexlsjob コマンド] 327
- j [bjexlslock コマンド] 333
- J [bjexrmjob コマンド] 343
- l [bjexgdg コマンド] 317
- l [bjexlsjob コマンド] 326
- l [bjexlslock コマンド] 333
- m [bjexgdg コマンド] 317
- n [bjexec コマンド] 315
- n [bjexlsjob コマンド] 326
- q [bjexgdg コマンド] 317
- s [bjexlsexjob コマンド] 320
- s [bjexlsjob コマンド] 326
- t [bjexlsjob コマンド] 327
- T [bjexlsjob コマンド] 327
- t [bjexlslock コマンド] 333
- T [bjexrmjob コマンド] 342
- u [bjexgdg コマンド] 317
- v [bjexec コマンド] 315
- v [bjexgdg コマンド] 318
- w [bjexlslock コマンド] 333

## B

- BJEX\_DIR\_任意名 356
- BJEX\_DIR\_任意名の指定方法 93
- bjex\_setup 348
- bjexchmsg 313
- bjexec 315
- bjexgdg 317
- bjexlsexjob 320
- bjexlsjob 326
- bjexlslock 333
- bjexlsmsg 340
- bjexrmjob 342
- bjexsvc 346
- 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) 164
- bjexu\_ismut および bjexu\_ismut64 (索引順編成  
ファイルの作成および削除, 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) 201
- bjexu\_pcopy (ファイルの複写) 165
- bjexu\_scopy (ファイルのレコード単位の複写) 177

## C

- CALL 要素 (プロシジャの呼び出し) 422
- CHARS [DD 要素] 423
- CHARS [SysoutClass パラメータ] 400
- CMT [DD 要素] 424
- COND [EXEC 要素] 437
- COND [JOB 要素] 445
- COPIES [DD 要素] 424
- COPIES [SysoutClass パラメータ] 400
- COPYG [DD 要素] 424
- COPYG [SysoutClass パラメータ] 400
- Cosminexus との連携によるバッチアプリケーション  
実行 234

## D

- DD\_DEALLOC\_ON\_ALLOCERR (パラメータ) 357

DD\_EXISTCHECK\_COBOL [パラメータ] 360  
DD\_EXISTCHECK [パラメータ] 358  
DD\_SAMF 364  
DD 要素 (ファイルの定義) 423  
DD 要素の書き方 92  
DD 要素を修正するための指定方法 70  
DD 要素をプログラムから使用する方法 94  
DEST [DD 要素] 424  
DEST [SysoutClass パラメータ] 401  
DISP [DD 要素] 424  
DISPLAY\_REPLY\_MESSAGE [パラメータ] 362  
DISPMOD\_NOFILE [パラメータ] 363  
DSN\_USE\_ENV 365  
DSN\_USE\_ENV (DD 要素の DSN 属性への環境変数  
指定機能) 365  
DSN [DD 要素] 426  
DSN 属性の指定方法 93  
DSN 属性への環境変数指定機能 96  
DTD [用語解説] 632  
DUMMY [SysoutClass パラメータ] 401

## E

encoding [XML 宣言] 419  
EXEC\_COND\_NOSTEPNAME [パラメータ] 366  
EXEC\_PARMSPLIT 367  
EXEC\_PARMSPLIT [EXEC\_PARMSPLIT パラメータ]  
367  
EXEC 要素 (プログラムの実行) 437

## F

FCB [DD 要素] 427  
FCB [JOBPARM 要素] 448  
FCB [SysoutClass パラメータ] 401  
FCDFLT [DD 要素] 427  
FCDFLT [SysoutClass パラメータ] 401  
FORMS [DD 要素] 427  
FORMS [JOBPARM 要素] 448  
FORMS [SysoutClass パラメータ] 401  
FTYPE [DD 要素] 427

FTYPE [SysoutClass パラメータ] 401

## G

GAIJI1 [DD 要素] 427  
GAIJI1 [SysoutClass パラメータ] 401  
GDG\_REGISTER\_JOBRC\_THRESHOLD [パラメータ]  
368  
Group 438

## H

HitachiBatchJobExec 要素 (ジョブ定義 XML ファ  
イルの宣言) 443  
HOLD [DD 要素] 427  
HOLD [SysoutClass パラメータ] 401

## I

ISAMLIB\_MODE [パラメータ] 369  
ISKEY [DD 要素] 428  
ISRECFM [DD 要素] 428  
ISRECL [DD 要素] 429

## J

Java バッチアプリケーションの強制終了 239  
Java バッチアプリケーション [用語解説] 632  
JOB\_RCDEFINE [パラメータ] 370  
JOB\_STATUS\_DISPLAY [パラメータ] 372  
JOBLIB, STEPLIB, および PATH 環境変数の使用  
方法 75  
JOBLIB DD 要素 74  
JOBPARM 要素 (ジョブ単位の制御情報の定義) 448  
JOB 要素 (ジョブの属性と終了条件の定義) 445  
JP1\_HOSTNAME 373  
JP1/AJS 環境設定 42  
JP1/AJS によるジョブスケジュール 254, 285  
JP1/AJS を使用したバッチジョブ業務の自動化の概要  
255  
JP1/IM による実行状態の監視と実行結果の確認 269  
JP1/IM の設定 42  
JP1/IM - Manager が稼働するホスト名  
[JP1IM\_MANAGER\_HOST パラメータ] 376

JP1/IM - Manager との連携 296  
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 296  
JP1EVENT\_SUPPRESS\_ID 374  
JP1IM\_MANAGER\_CHKDELAY [パラメータ] 375  
JP1IM\_MANAGER\_HOST [パラメータ] 376  
JP1 イベント一覧 599  
JP1 イベントの詳細 600

## K

KEEP [DD 要素] 429  
KEEP [SysoutClass パラメータ] 401

## L

LANG 438  
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 235  
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 239  
LOCK\_JOBMPLEX [パラメータ] 377  
LOCK\_MAXFILE [パラメータ] 378  
LOCK\_OPTION [パラメータ] 379  
LOCK\_RELEASE [パラメータ] 380  
LOG\_DIR 381  
LOG\_FILE\_CNT 382  
LOG\_FILE\_SIZE 383

## M

MAX\_RESPONSE\_REQUEST [パラメータ] 384  
MESSAGE\_REPLACE\_SPACE [パラメータ] 385  
MESSAGE\_RETRY\_TIMEOUT [パラメータ] 386  
MODE [DD 要素] 429  
MODE [SysoutClass パラメータ] 402  
MSGCLASS\_DEFAULT 387  
MSGCLASS [JOB 要素] 446

## N

NAME [CALL 要素] 422  
NAME [DD 要素] 429  
NAME [JOB 要素] 445

NAME [PROC 要素] 449  
NAME [REPLACESTEP 要素] 453  
NAME [SETENV 要素] 454  
NAME [SETSYMBOL 要素] 457  
NAME [STEP 要素] 458  
NAME [SYMBOL 要素] 459

## O

OPTCDJ [DD 要素] 430  
Option 439  
os [HitachiBatchJobExec 要素] 443  
OVERC [DD 要素] 430  
OVERC [SysoutClass パラメータ] 402  
OVERLAY [DD 要素] 430  
OVERLAY [SysoutClass パラメータ] 402

## P

PARM [EXEC 要素] 440  
PARMSPLIT [EXEC 要素] 440  
PDE との連携による帳票出力 27  
PDE ファイル形式 135  
PERMANENT\_FILE\_DIR 388  
PERMANENT\_FILE\_DIR の指定方法 93  
PGM\_RCABEND [パラメータ] 389  
PGM [EXEC 要素] 439  
Prinfina MANAGER SP および PDE との連携による帳票印刷 290  
Prinfina MANAGER SP との連携による高速プリンタへの印刷の流れ 265  
PROCEDURE\_DIR 390  
PROC 要素 (プロシジャの定義) 449  
prolog (XML の宣言) 419  
PRTCNTL スプールジョブ管理ファイル転送先ディレクトリ名 402  
PRTDATA スプールジョブデータファイル転送先ディレクトリ名 402

## R

RECFM [DD 要素] 430  
RECFM [SysoutClass パラメータ] 402

RENAME [DD 要素] 430  
REPLACESTEP 要素 (DD 要素または EXEC 要素の属性を変更対象のジョブステップの定義) 453  
REPLACE 要素 (DD 要素または EXEC 要素の属性の変更内容の定義) 451  
ROOM [JOBPARM 要素] 448

## S

SAMF [DD 要素] 431  
SET\_PGM PRIORITY 391  
SET\_PGM PRIORITY [Set\_Pgmpriority パラメータ] 391  
SETENV 要素 (環境変数の定義) 454  
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) 455  
SETSYMBOL 要素 (記号パラメータの値の定義) 457  
SPOOL\_DIR 392  
SPOOL\_JOB\_STORE [パラメータ] 393  
SPOOL\_JOB\_TRANSFER 394  
SPOOL\_JOB\_TRANSFER [SysoutClass パラメータ] 394  
SPOOL\_STORE\_DIR [パラメータ] 395  
standalone [XML 宣言] 419  
STDERRCLASS\_DEFAULT [パラメータ] 397  
STDOUTCLASS\_DEFAULT [パラメータ] 398  
STEPLIB DD 要素 74  
STEP 要素 (ジョブステップの定義) 458  
SYMBOL 要素 (記号パラメータの定義) 459  
SYSIN データとなる文字列 126  
SYSIN データの書き方 126  
SYSIN データの読み込み 126  
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 128  
SYSIN データ [用語解説] 632  
SYSOUT\_APPEND\_WRITE 399  
SYSOUT\_APPEND\_WRITE [Sysout\_Append\_Write パラメータ] 399  
SysoutClass 400  
SYSOUT [DD 要素] 431

SYSOUT 機能での注意事項 136  
SYSOUT 属性の PDE および Prinfina MANAGER SP での扱い 292  
SYSOUT の機能 131  
SYSOUT [用語解説] 632

## T

TEMP\_FILE\_DIR 404  
TRACE\_DIR 405  
TRACE\_FILE\_CNT 406  
TRACE\_FILE\_SIZE 407  
TRACE\_LEVEL 408  
TRUNC [SysoutClass パラメータ] 402  
TYPE [DD 要素] 431

## V

VALUE [SETENV 要素] 454  
VALUE [SETMAXRC 要素] 455  
VALUE [SETSYMBOL 要素] 457  
VALUE [SYMBOL 要素] 459  
version [HitachiBatchJobExec 要素] 443  
VersionNum [XML 宣言] 419

## X

XML ジョブ制御要素一覧 421  
XML ファイルへの環境変数指定機能 240  
XML [用語解説] 632

## あ

アンインストール 35

## い

一時 ISAM ファイル 88  
一時 ISAM ファイルの指定方法 88  
一時 ISAM ファイルのファイル名 89  
一時 ISAM ファイルを構成するファイル 89  
一時 ISAM ファイルを構成するファイル名 89  
一時 ISAM ファイルを利用する場合の注意事項 90  
一時 ISAM ファイル [用語解説] 632  
一時ファイル 87, 249

一時ファイルの指定方法 87  
一時ファイルを利用する場合の注意事項 87  
一時ファイル〔用語解説〕 632  
一般ユーザ 246  
イベント ID〔JP1EVENT\_SUPPRESS\_ID パラメータ〕 374  
イベント到達確認までの待ち時間〔JP1IM\_MANAGER\_CHKDELAY パラメータ〕 375  
印刷データ転送 290  
印刷データの転送契機 295  
インストール 33, 34  
インストールおよびアンインストール時の注意事項 35  
引用符〔用語解説〕 632

## う

運用 244

## え

演算符号 103

## お

応答入力 297  
応答の入力 296  
応答要求メッセージの監視 270  
応答要求メッセージの同時出力数 296

## か

概要 245  
カタログドプロシジャ 64  
カタログドプロシジャと入力ストリームプロシジャ 64  
カタログドプロシジャの使用方法 65  
カタログドプロシジャ〔用語解説〕 633  
環境変数 242  
環境変数に設定される内容 94  
環境変数名を任意の名称に設定する方法 95  
環境変数〔用語解説〕 633

## き

記号パラメータに対する値の割り当て 73  
記号パラメータの定義 73

記号パラメータ〔用語解説〕 633  
機能 47  
逆方向参照 97  
逆方向参照〔用語解説〕 633  
共用制御 137

## く

クラスタシステムで運用する場合の設定 276  
クラス名〔SysoutClass パラメータ〕 400

## こ

恒久ファイル 249, 633  
コード〔MESSAGE\_REPLACE\_SPACE パラメータ〕 385  
コマンド 310  
コマンド一覧 312  
コマンドの記述形式 311

## さ

最大応答要求メッセージ数〔MAX\_RESPONSE\_REQUEST パラメータ〕 384  
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) 201  
作成する一時 ISAM ファイルの定義内容 90

## し

システム管理者 246  
システム管理者への指示 463  
システムのプログラム使用方法 74  
実行結果の参照 289  
実行ログ 302  
実行ログ〔用語解説〕 633  
実体〔用語解説〕 633  
自動消去機能 (SCRATCH) が設定された世代データ群の注意事項 123  
終了コード 615  
出力クラス 132  
出力クラスによる制御 266  
出力クラスの設定 250



- 出力クラス名 [MSGCLASS\_DEFAULT パラメータ] 387
- 出力クラス [用語解説] 633
- 出力形式 132
- 出力後保存 132
- 出力後保存 [用語解説] 633
- 出力情報の複写 132
- 出力保留 132
- 出力保留属性および出力非保留属性 267
- 出力保留 [用語解説] 634
- 障害対策 299
- 条件付実行 104
- 条件付実行の指定方法 103
- ジョブ 246
- ジョブが長時間待ち状態となった場合の対処方法 306
- ジョブコントローラ 50
- ジョブコントローラが使用する環境変数 242
- ジョブコントローラの終了コード一覧 616
- ジョブコントローラ [用語解説] 634
- ジョブ識別子管理ファイル 61
- ジョブ識別子 [bjexlsjob コマンド] 327
- ジョブ識別子 [bjexrmjob コマンド] 343
- ジョブ識別子 [用語解説] 634
- ジョブ実行の処理手順 48
- ジョブ実行のスケジュール 248
- ジョブステップ 99, 246
- ジョブステップの終了コード一覧 615
- ジョブステップの条件付実行 103
- 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 108
- ジョブステップ [用語解説] 634
- ジョブ多重度数 [LOCK\_JOBMPLEX パラメータ] 377
- 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 137
- ジョブ定義 XML ファイル 48, 410
- ジョブ定義 XML ファイル仕様 621
- ジョブ定義 XML ファイルのパス名 [bjexec コマンド] 315
- ジョブ定義 XML ファイルの文法 411
- ジョブ定義 XML ファイル [用語解説] 634
- ジョブとジョブステップの概念 246
- ジョブとジョブステップの実行順序の関係 247
- ジョブの構成 49
- ジョブのシンタクスチェック機能 50
- ジョブ名などの情報をイベント一覧に表示 270
- ジョブ名ワイルドカード [bjexlsjob コマンド] 327
- ジョブ名ワイルドカード [bjexrmjob コマンド] 344
- ジョブ名 [bjexlsjob コマンド] 327
- ジョブ名 [bjexrmjob コマンド] 344
- ジョブライブラリ [用語解説] 634
- ジョブログ 301
- ジョブログ出力機能 51
- ジョブログ出力機能 [用語解説] 634
- ジョブログ [用語解説] 634
- ジョブを構成するためのジョブ定義 XML ファイル 49

## す

- ステップライブラリ [用語解説] 634
- ストレージ設計 46
- スプール 59
- スプールジョブ管理ファイル 60
- スプールジョブディレクトリ 60
- スプールジョブ転送先の設定 264
- スプールジョブ転送失敗時の注意事項 268
- スプールジョブ転送の運用 264
- スプールジョブの構造 61
- スプールジョブの削除 268
- スプールジョブのディレクトリの構造 61
- スプールジョブのデータの流れ 62
- スプールジョブ保管機能 [用語解説] 635
- スプールジョブを転送しない運用 268
- スプールジョブ [用語解説] 635
- スプールディレクトリ 60
- スプールに関する注意事項 63
- スプールの構造 60
- スプールの制御 60
- スプール [用語解説] 635

## せ

- セキュリティ情報の設定 250
- 世代ディレクトリ名 [bjexgdg コマンド] 317
- 世代データ群 113
- 世代データ群の削除 117
- 世代データ群の作成 114
- 世代データ群の配置 115
- 世代データ群名 [bjexgdg コマンド] 317
- 世代データ群を構成するファイル 114
- 世代データ群 [用語解説] 635
- 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 122
- 世代ファイル 113
- 世代ファイルおよび世代データ群の削除 117
- 世代ファイルの概要 113
- 世代ファイルの削除 117
- 世代ファイルの作成と参照 115
- 世代ファイルの参照 116
- 世代ファイルの参照 (世代データ群名指定) 116
- 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] 635
- 世代ファイルの注意事項 122
- 世代ファイルを削除する場合の注意事項 124
- 世代ファイルを作成する場合の DD 要素の注意事項 122
- 世代ファイルを参照する場合の DD 要素の注意事項 122
- 世代ファイル [用語解説] 635
- 絶対世代番号 113
- 設定ファイル 350
- 設定ファイルの使用例 409
- 設定ファイルのパラメーター一覧 353
- 設定ファイル [用語解説] 635
- セットアップ 39
- 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 283

- 相対世代番号 113
- 相対世代番号 [bjexgdg コマンド] 318
- 属性 [用語解説] 635

## た

- タイプコード 461
- ダミーファイル 91
- ダミーファイルの主な用途 91
- ダミーファイルの指定方法 92
- ダミーファイル [用語解説] 635

## ち

- 帳票処理の概要 264

## て

- 定義ファイル 249
- 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 266

## と

- 統合コンソールによる集中監視 269
- 統合コンソール [用語解説] 636
- 動作環境の確認 249
- トラブルシューティング 301
- トラブル発生時に採取が必要な資料 303
- トレース 303
- トレースレベル [TRACE\_LEVEL パラメータ] 408
- トレース [用語解説] 636

## に

- 入力ストリームプロシジャ 64
- 入力ストリームプロシジャの作成方法 64
- 入力ストリームプロシジャの使用方法 66
- 入力ストリームプロシジャ [用語解説] 636

## ね

- ネットワーク上のファイルシステムのサポート内容 30

## は

- バージョン間の移行時の注意事項 36



- 排他/共用制御 137
- 排他/共用制御機能の設定 251
- 排他/共用制御機能〔用語解説〕 636
- 排他/共用制御に関する注意事項 145
- 排他/共用制御による実行待ち状態の確認方法 142
- 排他/共用制御の指定方法と適用範囲 140
- 排他制御 137
- パス名〔BJEX\_DIR\_任意名パラメータ〕 356
- パス名〔bjexlsock コマンド〕 333
- パス名〔LOG\_DIR パラメータ〕 381
- パス名〔PERMANENT\_FILE\_DIR パラメータ〕 388
- パス名〔PROCEDURE\_DIR パラメータ〕 390
- パス名〔SPOOL\_DIR パラメータ〕 392
- パス名〔TEMP\_FILE\_DIR パラメータ〕 404
- パス名〔TRACE\_DIR パラメータ〕 405
- バックアップ 300
- バックアップとリカバリー 300
- バッチジョブおよびジョブネットの強制終了 260
- 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 257
- バッチジョブ業務と実行順序の定義 256
- 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 257
- バッチジョブ業務の自動化 254
- バッチジョブ業務の自動化の検討 251
- バッチジョブ実行システム実行ユーザ管理 260
- 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 256
- バッチジョブ実行システムと Prinfinia MANAGER SP および PDE の連携による印刷の流れ 267
- バッチジョブ実行システムとは 21
- バッチジョブ実行システムの運用時の注意事項 282
- 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 272
- バッチジョブ実行システムの関連プログラム 28
- バッチジョブ実行システムの機能 48
- バッチジョブ実行システムの機能の特長 48
- バッチジョブ実行システムのコマンド一覧 312
- バッチジョブ実行システムのシステム設計 249
- バッチジョブ実行システムの使用概要 284
- バッチジョブ実行システムの処理の流れ 22
- バッチジョブ実行システムのセットアップ手順の概要 37
- バッチジョブ実行システムの全体構成 32
- バッチジョブ実行システム的前提プログラム 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ッチジョブの実行 285
- 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 284
- バッチジョブ実行システム〔用語解説〕 636
- バッチジョブ定義の記述方法 258
- バッチジョブの実行環境 257
- バッチジョブの実行結果の一元管理 26

## ふ

- ファイルサイズ〔LOG\_FILE\_SIZE パラメータ〕 383
- ファイルサイズ〔TRACE\_FILE\_SIZE パラメータ〕 407
- ファイルシステム 30
- ファイル数〔LOCK\_MAXFILE パラメータ〕 378
- ファイル操作ユティリティ 161
- ファイル操作ユティリティ〔用語解説〕 636
- ファイルの後処理の指定 80
- ファイルの作成 (DEFINE 文) 201
- ファイルの準備 77
- ファイルの状態と後処理 78
- ファイルの状態と後処理の指定省略 85
- ファイルの状態の指定 78
- ファイルの複写 203
- ファイルの併合 205
- ファイルの変換 204
- ファイルの連結処理 86
- ファイルの連結処理〔用語解説〕 636
- ファイルの割り当て 77
- 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 133
- 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 74

プログラムへのファイル割り当て機能 92  
プログラムへのファイル割り当て機能〔用語解説〕  
636  
プロシジャ内の DD 要素の修正方法 69  
プロシジャ内の EXEC 要素の修正方法 67  
プロシジャ内への DD 要素の追加方法 72  
プロシジャの概要 64  
プロシジャの記号パラメータによる修正方法 72  
プロシジャの修正方法 67  
プロシジャライブラリ〔用語解説〕 637  
プロシジャ〔用語解説〕 636  
プロセス一覧 624

## ほ

ホスト名 373

## め

メタキャラクタ〔用語解説〕 637  
メッセージ一覧 472  
メッセージ制御機能 149  
メッセージ制御機能での障害対応 308  
メッセージ制御機能の設定 252  
メッセージ制御機能を使用する場合の環境設定 44  
メッセージと終了コード 460  
メッセージに含まれるエラー情報 470  
メッセージの記載形式 461  
メッセージの形式 461  
メッセージの出力 463  
メッセージの出力形式 461  
メッセージの出力先一覧 464  
メッセージ番号の割り当て 462  
面数〔LOG\_FILE\_CNT パラメータ〕 382  
面数〔TRACE\_FILE\_CNT パラメータ〕 406

## ゆ

ユーザプログラム 249  
ユーザプログラムの使用方法 74

## よ

要素〔用語解説〕 637

## り

リカバリー 300  
リトライ打ち切り時間  
〔MESSAGE\_RETRY\_TIMEOUT パラメータ〕 386  
利用者の役割 246

## れ

レベル 1 外字名指定 136

## ろ

ログ情報の種類 301