

# ソートマージ

手引書

3020-3-N73-70

## ■対象製品

●適用 OS : HP-UX 11i V2(IPF), HP-UX 11i V3(IPF)

P-1J21-1181 SORT Version8 08-10

P-F1J21-11811 SORT Version8 - Extended Edition 08-40

P-1J21-1281 SORT Version8(64) 08-11

P-F1J21-12811 SORT Version8 - Extended Edition(64) 08-40

●適用 OS : AIX 5L V5.2, AIX 5L V5.3, AIX V6.1, AIX V7.1

P-1M21-1181 SORT Version8 08-10

P-F1M21-11811 SORT Version8 - Extended Edition 08-40

P-1M21-1281 SORT Version8(64) 08-11

P-F1M21-12811 SORT Version8 - Extended Edition(64) 08-50<sup>※1</sup>

●適用 OS : Red Hat Enterprise Linux 5 Advanced Platform (x86), Red Hat Enterprise Linux 5 (x86), Red Hat Enterprise Linux 5 (AMD/Intel 64)<sup>※2</sup>, Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64)<sup>※2</sup>, Red Hat Enterprise Linux Server 6 (32-bit x86), Red Hat Enterprise Linux Server 6 (64-bit x86\_64), Red Hat Enterprise Linux Server 7 (64-bit x86\_64)

P-9S21-1181 SORT Version8 08-10

P-F9S21-11811 SORT Version8 - Extended Edition 08-40

●適用 OS : Red Hat Enterprise Linux 5 (AMD/Intel 64)<sup>※2</sup>, Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64)<sup>※2</sup>, Red Hat Enterprise Linux Server 6 (64-bit x86\_64), Red Hat Enterprise Linux Server 7 (64-bit x86\_64)

P-9W21-1281 SORT Version8(64) 08-11

P-F9W21-12811 SORT Version8 - Extended Edition(64) 08-50

●適用 OS : Windows 2000, Windows Server 2003, Windows Server 2008, Windows Server 2012, Windows Vista, Windows XP, Windows 7, Windows 8, Windows 8.1, Windows 10

P-2621-1184 SORT Version8 08-11

P-F2621-11841 SORT Version8 - Extended Edition 08-50<sup>※1</sup>

●適用 OS : Windows Server 2003 x64 Edition, Windows Server 2008 x64 Edition, Windows Server 2012, Windows Vista x64 Edition, Windows XP x64 Edition, Windows 7 x64 Edition, Windows 8 x64 Edition, Windows 8.1 x64 Edition, Windows 10 x64

P-2921-1284 SORT Version8(64) 08-11

P-F2921-12841 SORT Version8 - Extended Edition (64) 08-50<sup>※1</sup>

注※1 このプログラムプロダクトについては、出荷時期をご確認ください。

注※2 この製品の動作環境は、Intel EM64T だけに対応しています。

## ■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 は、株式会社 日立製作所の商標または登録商標です。

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

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

IBM, AIX 5L は、世界の多くの国で登録された International Business Machines Corporation の商標です。

Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

Microsoft は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Red Hat は、米国およびその他の国で Red Hat, Inc. の登録商標もしくは商標です。

UNIX は、The Open Group の米国ならびに他の国における登録商標です。

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

Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Vista は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

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	このマニュアルでの表記
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> 2000 Advanced Server Operating System	Windows 2000
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> 2000 Datacenter Server Operating System	
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> 2000 Professional Operating System	
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> 2000 Server Operating System	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2003, Enterprise Edition	Windows Server 2003
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2003, Enterprise x64 Edition Operating System	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2003, Standard Edition	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2003, Standard x64 Edition Operating System	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2003 R2, Enterprise Edition Operating System	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2003 R2, Enterprise x64 Edition Operating System	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2003 R2, Standard Edition Operating System	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2003 R2, Standard x64 Edition Operating System	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2003 R2, Standard x64 Edition Operating System	

製品名	このマニュアルでの表記
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2003, Enterprise x64 Edition Operating System	Windows Server 2003 x64 Edition
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2003, Standard x64 Edition Operating System	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2003 R2, Enterprise x64 Edition Operating System	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2003 R2, Standard x64 Edition Operating System	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2008 Enterprise	Windows Server 2008
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2008 Standard	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2008 Enterprise x64 Edition	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2008 Standard x64 Edition	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2008 R2 Datacenter	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2008 R2 Enterprise	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2008 R2 Standard	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2008 Enterprise x64 Edition	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2008 Standard x64 Edition	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2008 Enterprise x64 Edition	Windows Server 2008 x64 Edition
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2008 Standard x64 Edition	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2012 Standard	Windows Server 2012
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2012 Datacenter	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2012 R2 Standard	
Microsoft <sup>(R)</sup> Windows Server <sup>(R)</sup> 2012 R2 Datacenter	
Microsoft <sup>(R)</sup> Windows Vista <sup>(R)</sup> Business	Windows Vista
Microsoft <sup>(R)</sup> Windows Vista <sup>(R)</sup> Enterprise	
Microsoft <sup>(R)</sup> Windows Vista <sup>(R)</sup> Ultimate	
Microsoft <sup>(R)</sup> Windows Vista <sup>(R)</sup> Business x64	
Microsoft <sup>(R)</sup> Windows Vista <sup>(R)</sup> Enterprise x64	
Microsoft <sup>(R)</sup> Windows Vista <sup>(R)</sup> Ultimate x64	
Microsoft <sup>(R)</sup> Windows Vista <sup>(R)</sup> Business x64	Windows Vista x64 Edition
Microsoft <sup>(R)</sup> Windows Vista <sup>(R)</sup> Enterprise x64	
Microsoft <sup>(R)</sup> Windows Vista <sup>(R)</sup> Ultimate x64	
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> XP Home Edition Operating System	Windows XP
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> XP Professional Operating System	
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> XP Professional x64 Edition Operating System	

製品名	このマニュアルでの表記
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> XP Professional x64 Edition Operating System	Windows XP x64 Edition
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> 7 Enterprise	Windows 7
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> 7 Professional	
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> 7 Ultimate	
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> 7 Enterprise x64	
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> 7 Professional x64	
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> 7 Ultimate x64	
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> 7 Enterprise x64	Windows 7 x64 Edition
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> 7 Professional x64	
Microsoft <sup>(R)</sup> Windows <sup>(R)</sup> 7 Ultimate x64	
Windows <sup>(R)</sup> 8 Pro	Windows 8
Windows <sup>(R)</sup> 8 Enterprise	
Windows <sup>(R)</sup> 8 Pro x64	
Windows <sup>(R)</sup> 8 Enterprise x64	
Windows <sup>(R)</sup> 8 Pro x64	Windows 8 x64 Edition
Windows <sup>(R)</sup> 8 Enterprise x64	
Windows <sup>(R)</sup> 8.1 Pro	Windows 8.1
Windows <sup>(R)</sup> 8.1 Enterprise	
Windows <sup>(R)</sup> 8.1 Pro x64	
Windows <sup>(R)</sup> 8.1 Enterprise x64	
Windows <sup>(R)</sup> 8.1 Pro x64	Windows 8.1 x64 Edition
Windows <sup>(R)</sup> 8.1 Enterprise x64	
Windows <sup>(R)</sup> 10 Pro(x86)	Windows 10
Windows <sup>(R)</sup> 10 Enterprise(x86)	
Windows <sup>(R)</sup> 10 Pro(x64)	
Windows <sup>(R)</sup> 10 Enterprise(x64)	
Windows <sup>(R)</sup> 10 Pro(x64)	Windows 10 x64
Windows <sup>(R)</sup> 10 Enterprise(x64)	

なお、Windows 2000、Windows Server 2003、Windows Server 2008、Windows Server 2012、Windows Vista、Windows XP、Windows 7、Windows 8、Windows 8.1、および Windows 10 を総称する場合は、Windows と表記します。

■発行

2016年6月 3020-3-N73-70

■著作権

All Rights Reserved. Copyright (C) 2008, 2016, Hitachi, Ltd.

## 変更内容

変更内容 (3020-3-N73-70) SORT Version8(64) 08-11, SORT Version8 - Extended Edition(64) 08-50, SORT Version8 - Extended Edition 08-50

追加・変更内容	変更箇所
SORT と SORT EE の入出力構成の図を追加した。	1.1
エンコードに EUC-HJ を追加した。	2.26, 2.36, 3.2, 3.3.1, 5.6, 5.7, 5.8.7, 5.8.10, 5.8.13 ~ 5.8.17, 5.10.3, 5.16, 5.21.3, 5.21.4, 15.1.1(11), 15.1.1(14), 15.1.1(16), 15.1.1(17), 15.1.1(19), 15.1.1(21), 17.1.1, 表 C-2, 付録 E, 付録 I, 付録 J
SORT EE の機能として、次を追加した。 <ul style="list-style-type: none"> <li>• -outfile パラメタ数の制限解除オプション</li> <li>• exsort パラメタファイルサイズの制限解除オプション</li> <li>• CSV 数値データ (固定小数点数) の形式チェックオプション</li> <li>• 選択条件式のブレイクオプション</li> <li>• CSV 日付データの部分指定</li> <li>• 出力ファイル名重複チェック抑止オプション</li> <li>• 最適化したソートライブラリの使用</li> </ul>	2.41, 2.42, 2.43, 2.44, 2.45, 3.3.1, 4.4.1, 表 5-5, 表 5-6, 5.5.24, 5.5.25, 17.1.9, 17.2.4, 付録 B.1, 付録 B.5
SORT EE と SORT で利用できるファイルのレコード長の最小値, 最大値および注釈を追加した。	表 3-1
円記号 (¥) の扱いについての説明を追加した。	4.3.6, 4.4.3, 9.1.3(6), 9.2.3(5), 18.2, 18.2.4, 付録 N
コマンド実行時のロケール (UNIX の場合) に SORT EE 08-50 以降のバージョンの説明を追加した。	4.3.8
コマンド実行時のコードページ (Windows の場合) に SORT EE 08-50 以降のバージョンの説明を追加した。	4.3.9
選択条件式の書き方の説明を追加した。	4.4.3
環境変数の説明を変更した。	4.4.5
パラメタファイルを記述する文字コードに SORT EE 08-50 以降のバージョンの説明を追加した。	4.4.7
SORT EE の環境変数を追加した。 <ul style="list-style-type: none"> <li>• EXSORTOUTFILENUMUNLIMIT</li> <li>• EXSORTPRMSIZEUNLIMIT</li> <li>• EXSORTOUTFILENODUPCHK</li> <li>• EXSORTLOGSUPPRESS</li> <li>• EXSORT64OUTFILENUMUNLIMIT</li> <li>• EXSORT64PRMSIZEUNLIMIT</li> <li>• EXSORT64OUTFILENODUPCHK</li> <li>• EXSORT64LOGSUPPRESS</li> <li>• EXSORT64SORTLIBOPTION</li> </ul>	表 5-5, 5.5.2, 表 14-1, 表 14-2, 14.1.1(21), 14.1.1(33) ~ 14.1.1(36), 14.1.2(21), 14.1.2(33) ~ 14.1.2(37)

追加・変更内容	変更箇所
10 進キーのチェックオプション (KEYNOCHK オプション) の説明を変更した。	5.5.9
-inpfile パラメタの, reclen オペランドの指定値の説明を変更した。	5.8.3
-inpfile パラメタの, {include   omit} オペランドの cond-value の指定値に, 'S' 文字列' を追加した。	5.8.16
-inpfile パラメタの, {csvinclude   csvomit} オペランドの cond-value の指定値を追加した。 <ul style="list-style-type: none"> <li>• 'S' 文字列'</li> <li>• 'D' マスク日付'</li> <li>• 'H' マスク日付 T マスク時刻'</li> </ul>	5.8.17
-copyinp パラメタの, fileorg オペランドの指定値の説明および注意事項を変更した。	5.28.1
入力ファイル編成ごとのレコード長の指定方法を変更した。	5.8.3, 表 7-4, 表 7-14, 表 8-4, 表 8-11, 表 9-4, 表 9-13, 表 10-5, 表 10-12, 表 11-6
ワークサイズ設定の優先順位の, SORT の説明を変更した。	14.1.3
SORT EE の環境ファイルに設定できる項目を追加した。 <ul style="list-style-type: none"> <li>• EXS_OUTFILENUMUNLIMIT</li> <li>• EXS_PRMSIZEUNLIMIT</li> <li>• EXS_DUMPLOG_FILESIZE</li> </ul>	15.1.1, 15.1.1(24) ~ 15.1.1(26)
SORT EE の環境ファイルに指定できる項目の, EXS_CODETYPE の説明を変更した。	15.1.1(21)
ワークバッファサイズの自動見積もり機能の, ワークバッファサイズの指定値の説明を変更した。	18.1.2(4)
SORT の機能拡張オプションのメモリ所要量の説明を追加した。	18.2.1
デバッグ情報ファイルおよびログファイルの, ファイルの説明を変更した。	19.1.2, 19.1.3
CSV ファイルのデータの扱いの説明を追加した。	付録 B.1, 付録 B.5
SORT EE メッセージ出力オプションの説明を追加した。	付録 K.1(5)
次のメッセージを変更した。 KBLS1104-E, KBLS1120-E, KBLS1211-E, KBLS1314-E, KBLS1320-E, KBLS1607-E, KBLS1670-E, KBLS2209-E	付録 K.2
次のメッセージを追加した。 KBLS1122-E	付録 K.2

単なる誤字・脱字などはお断りなく訂正しました。



# はじめに

---

このマニュアルは、次に示すプログラムプロダクトの機能と使い方について説明したものです。

- P-2621-1184 SORT Version8
- P-2921-1284 SORT Version8 (64)
- P-1M21-1181 SORT Version8
- P-1M21-1281 SORT Version8 (64)
- P-9S21-1181 SORT Version8
- P-9W21-1281 SORT Version8 (64)
- P-1J21-1181 SORT Version8
- P-1J21-1281 SORT Version8 (64)
- P-F2621-11841 SORT Version8 - Extended Edition
- P-F2921-12841 SORT Version8 - Extended Edition (64)
- P-F1M21-11811 SORT Version8 - Extended Edition
- P-F1M21-12811 SORT Version8 - Extended Edition (64)
- P-F9S21-11811 SORT Version8 - Extended Edition
- P-F9W21-12811 SORT Version8 - Extended Edition (64)
- P-F1J21-11811 SORT Version8 - Extended Edition
- P-F1J21-12811 SORT Version8 - Extended Edition (64)

## ■対象読者

SORT Version8 - Extended Edition (以降、SORT EE と呼びます) では、ソート機能、マージ機能、コピー機能、選択機能、集約機能、編集機能、ファイル分割機能、およびファイル連結機能を利用される方、またこれらの機能を複数使用される方を対象としています。

SORT Version8 (以降、SORT と呼びます) では、ソート機能、マージ機能、選択機能および集約機能を利用される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、次に示す編、章と付録から構成されています。

### 第1編 概要

#### 第1章 SORT EE と SORT

SORT EE と SORT の機能の概要について説明しています。

#### 第2章 便利な機能

SORT EE と SORT の便利な機能について説明しています。

#### 第3章 利用できるファイルの種類とデータ形式

SORT EE と SORT で利用できるファイルの種類、キー属性と形式、指定できる範囲について説明しています。

はじめに

## 第 2 編 SORT EE

### 第 4 章 SORT EE の使い方

SORT EE で使用するコマンドについて説明しています。

### 第 5 章 exsort コマンドのパラメタおよびオペランドの詳細

exsort コマンドのパラメタおよびオペランドの詳細について説明しています。

### 第 6 章 SORT EE の使用例

SORT EE の使用例について説明しています。

## 第 3 編 SORT

### 第 7 章 ソートコマンドの使い方

ソートコマンドの使い方について説明しています。

### 第 8 章 マージコマンドの使い方

マージコマンドの使い方について説明しています。

### 第 9 章 選択コマンドの使い方

選択コマンドの使い方について説明しています。

### 第 10 章 集約コマンドの使い方

集約コマンドの使い方について説明しています。

### 第 11 章 拡張 SORT コマンドの使い方 (Windows の場合)

拡張 SORT コマンドの使い方について説明しています。

### 第 12 章 シェル・スクリプトコマンドの使い方 (UNIX の場合)

シェル・スクリプトコマンドの使い方について説明しています。

### 第 13 章 C 言語プログラムでの使い方 (Windows の場合)

C 言語プログラムでの使い方について説明しています。

## 第 4 編 運用

### 第 14 章 環境変数の設定

環境変数の設定について説明しています。

### 第 15 章 環境ファイルの設定

環境ファイルの設定について説明しています。

### 第 16 章 一時ファイルの作成条件とファイル容量の算出方法

一時ファイルの作成条件とファイル容量の算出方法について説明しています。

### 第 17 章 性能チューニング方法

性能チューニング方法について説明しています。

## 第 18 章 SORT の機能拡張オプションの使い方

SORT EE 固有の機能を SORT で使用できるようになるオプションについて説明しています。

## 第 19 章 障害時の対処方法

障害時の対処方法について説明しています。

## 第 20 章 SORT から SORT EE への移行ガイド

SORT から SORT EE に移行する場合のコマンド、およびワークバッファサイズの移行内容について説明しています。

## 付録 A JIS8 単位コード体系と EBCDIK コード体系の変換表

コード変換で使用できるコード体系のコード対応表を一覧にして説明しています。

## 付録 B CSV ファイルのデータの扱い (SORT EE の機能)

入力データが数値の場合の端数処理、文字列の場合の端数処理、日付の場合の端数処理について説明しています。

## 付録 C CSV ファイル形式と CSV データ形式 (SORT EE の機能)

CSV ファイル形式および CSV データ形式について説明しています。

## 付録 D CSV ファイル内のナル文字列の扱い (SORT EE の機能)

SORT EE で扱える CSV ファイル内のナル文字列について説明しています。

## 付録 E 左分離符号付き数字のデータ形式 (SORT EE の機能)

SORT EE で扱える左分離符号付き数字のデータ形式について説明しています。

## 付録 F レコード編集処理で基になるレコード (SORT EE の機能)

レコード編集処理で基になるレコードについて説明しています。

## 付録 G altseq 変換ファイル (SORT EE の機能)

altseq 変換ファイルについて説明しています。

## 付録 H Unicode 操作文字コード一覧 (SORT EE)

Unicode 操作文字コードについて説明しています。

## 付録 I マルチバイト文字データの扱い

マルチバイト文字データの扱いについて説明しています。

## 付録 J 日付時刻データの形式 (SORT EE の機能)

日付時刻データの形式について説明しています。

## 付録 K メッセージ

SORT EE および SORT から出力されるメッセージについて説明しています。

## 付録 L 各バージョンの変更内容

各バージョンの変更内容について説明しています。

はじめに

## 付録 M 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 N 用語解説

SORT EE および SORT で使われる用語について説明しています。

## ■ 読書手順

このマニュアルは、利用目的に合わせて次の個所をお読みいただくことをお勧めします。

マニュアルを読む目的	記述箇所
SORT EE および SORT の概要および機能について知りたい	1 章
SORT EE および SORT の便利な機能について知りたい	2 章
利用できるファイルの種類とデータ形式について知りたい	3 章
SORT EE の使い方について知りたい	4～6 章
SORT の使い方について知りたい	7～13 章
環境変数の設定について知りたい	14 章
環境ファイルの設定について知りたい	15 章
ソート機能の一時ファイルの作成条件とファイル容量の算出方法について知りたい	16 章
性能チューニング方法について知りたい	17 章
SORT EE 固有の機能を SORT で使用できるようになるオプションについて知りたい	18 章
障害時の対処方法について知りたい	19 章
SORT から SORT EE に移行する方法について知りたい	20 章
JIS8 単位コード体系と EBCDIK コード体系の変換について知りたい	付録 A
CSV ファイルのデータの扱いについて知りたい	付録 B
CSV ファイル形式および CSV データ形式について知りたい	付録 C
SORT EE で扱える CSV ファイル内のナリ文字列について知りたい	付録 D
SORT EE で扱える左分離符号付き数字のデータ形式について知りたい	付録 E
SORT EE のレコード編集処理で基になるレコードについて知りたい	付録 F
altseq 変換ファイルについて知りたい	付録 G
Unicode 操作文字コードについて知りたい	付録 H
マルチバイト文字データの扱いについて知りたい	付録 I
日付時刻データの形式について知りたい	付録 J
SORT EE および SORT のメッセージについて知りたい	付録 K
過去の版での変更内容について知りたい	付録 L
このマニュアルでの表記などについて知りたい	付録 M
SORT EE および SORT の用語について知りたい	付録 N

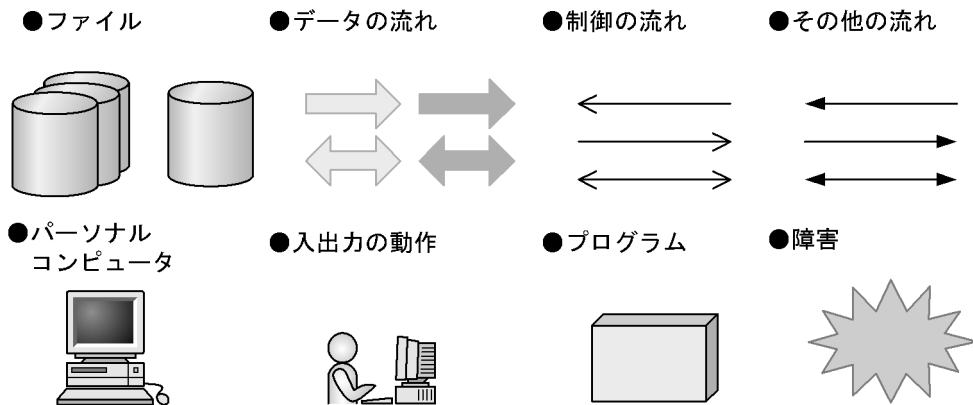
## ■このマニュアルで使用する記号

このマニュアルで使用している記号とその意味は、次のとおりです。

記号	説明
[ ]	メニューの項目、それぞれのメニューが持つ項目、およびダイアログボックスのボタンを表します。 この記号で示された項目またはボタンを選択するときの操作を、次に示します。 マウスを使用する場合 項目またはボタンをクリックする。 キーボードを使う場合 <ul style="list-style-type: none"> <li>・ &lt; Alt &gt;を押したまま、または押しながら、項目の名称のあとにある ( ) 内に示された英字のキーを押す。</li> <li>・ 移動キーまたは &lt; Tab &gt; を使用して、指定する項目またはボタンにカーソルを移動し、 &lt; Enter &gt; を押す。</li> </ul>
< >	キーボードのキーを表します。
…	…は、直前の { } で囲まれた部分を複数繰り返して指定できることを示します。
△	半角のスペース（空白）を示します。
文字列	文字列が斜体になっている項目は、可変の文字列を表します。

## ■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。





# 目次

## 第 1 編 概要

<b>1</b>	<b>SORT EE と SORT</b>	<b>1</b>
1.1	SORT EE と SORT の概要	3
1.2	ソート機能とは	5
1.3	マージ機能とは	6
1.4	コピー機能とは (SORT EE の機能)	7
1.5	選択機能とは	8
1.6	集約機能とは	9
1.7	編集機能とは (SORT EE の機能)	10
1.8	ファイル分割機能とは (SORT EE の機能)	12
1.9	ファイル連結機能とは (SORT EE の機能)	14
1.10	ファイル編成変更機能とは (SORT EE の機能)	15
1.11	処理の流れの概要	16
1.11.1	ソート機能の処理の流れ	16
1.11.2	マージ機能の処理の流れ	19
1.11.3	コピー機能の処理の流れ	19
<b>2</b>	<b>便利な機能</b>	<b>21</b>
2.1	重複レコードの入力順序保存機能	24
2.2	EBCDIK コードキー比較機能	26
2.3	強制マージ機能	27
2.4	無条件レコード抽出機能	28
2.5	10 進数オーバーフロー判定機能	29
2.6	エンディアンオプション機能	30
2.7	ファイル排他機能 (UNIX の場合)	31
2.7.1	SORT のファイル排他機能	31
2.7.2	SORT EE のファイル排他機能	32
2.8	キャッシュファイルの同期出力機能	34
2.9	内部 10 進数符号保障機能	35
2.10	10 進キーのキーチェック緩和機能 (SORT EE の機能)	36
2.11	10 進集約項目のチェック緩和機能 (SORT EE の機能)	38

2.12	キー位置のチェック緩和機能 (SORT EE の機能)	39
2.13	日付データのチェック緩和機能 (SORT EE の機能)	41
2.14	複数ファイル出力機能 (SORT EE の機能)	42
2.15	入力レコードの読み飛ばし・打ち切り機能 (SORT EE の機能)	43
2.16	文字順序変更機能 (SORT EE の機能)	44
2.17	既存ファイルの書き換え防止機能 (SORT の機能)	45
2.18	I/O サイズ変更機能 (SORT EE の機能)	46
2.19	ワークバッファサイズ自動見積もり機能 (SORT EE の機能)	47
2.20	2GB 超ワークバッファサイズの指定 (64 ビット版 SORT EE の機能)	48
2.21	SORT の機能拡張オプション (SORT の機能)	49
2.22	強制ファイル分割機能 (SORT EE の機能)	51
2.23	ファイル逐次 OPEN/CLOSE 機能 (SORT EE の機能)	52
2.24	メッセージ抑止機能 (SORT EE の機能)	53
2.25	0 件ファイル出力抑止機能 (SORT EE の機能)	54
2.26	キー・項目の上限値拡張機能 (SORT EE の機能)	55
2.27	メッセージ出力オプション (SORT EE の機能)	56
2.28	CSV ファイル関連の便利な機能 (SORT EE の機能)	57
2.28.1	入力開始レコード番号の指定	57
2.28.2	連続区切り文字オプション	57
2.28.3	ナル文字列の集約オプション (SUMNULKEY, SUMNONULL オプション)	57
2.28.4	出力ファイルのデリミタ変更	58
2.28.5	出力ファイルの引用符変更	58
2.28.6	出力ファイルのセル数調整	58
2.29	集約結果オーバーフロー検知オプション (SORT EE の機能)	59
2.30	改行文字の自動判定機能 (SORT EE の機能)	60
2.31	シフト JIS コード比較オプション (SORT EE の機能)	61
2.32	ゼロデータ比較オプション (SORT EE の機能)	62
2.33	空行削除オプション (SORT EE の機能)	63
2.34	ファイルの上書きオプション (SORT EE の機能)	64
2.34.1	既存ファイルの上書きオプション	64
2.34.2	入力ファイルの上書きオプション	64
2.35	Unicode シグニチャ (BOM) 関連オプション (SORT EE の機能)	65
2.35.1	Unicode シグニチャ読み飛ばし・付加機能	65
2.35.2	Unicode シグニチャ自動判別機能	65
2.36	マルチバイト文字データの比較 (SORT EE の機能)	66
2.37	内部 10 進数符号オプション (SORT EE の機能)	67



2.38	外部 10 進数データ形式オプション (SORT EE の機能)	68
2.39	符号なし 2 進数のマスクビット指定 (SORT EE の機能)	69
2.40	簡易コピー機能 (SORT EE の機能)	70
2.41	-outfile パラメタ数の制限解除オプション (SORT EE の機能) 【08-50 以降】	71
2.42	exsort パラメタファイルサイズの制限解除オプション (SORT EE の機能) 【08-50 以降】	72
2.43	CSV 数値データ (固定小数点数) の形式チェックオプション (SORT EE の機能) 【08-50 以降】	73
2.44	選択条件式のブレイクオプション (SORT EE の機能) 【08-50 以降】	75
2.45	CSV 日付データの部分指定 (SORT EE の機能) 【08-50 以降】	78

### 3

	利用できるファイルの種類とデータ形式	79
3.1	利用できるファイルの種類	80
3.2	キー属性と形式	82
3.3	指定できる値の範囲	98
3.3.1	パラメタ	98
3.3.2	メモリ量	106
3.3.3	入力レコード件数とファイルサイズ	107

## 第 2 編 SORT EE

### 4

	SORT EE の使い方	109
4.1	機能の実行順序と実行形態	110
4.1.1	機能の実行順序	110
4.1.2	機能の実行形態	115
4.2	exsort コマンドの形式	117
4.2.1	ソート機能	118
4.2.2	マージ機能	122
4.2.3	コピー機能	126
4.2.4	ファイル分割機能	130
4.2.5	ファイル連結機能	134
4.2.6	簡易コピー機能 (コマンド形式だけ)	137
4.3	コマンドの表記規則	138
4.3.1	パラメタ形式	138

4.3.2	オペランド形式	138
4.3.3	パラメタおよびオペランドの指定順序	138
4.3.4	コマンド予約語の表記	139
4.3.5	ファイル名およびディレクトリ名の指定形式	139
4.3.6	選択条件式の書き方	140
4.3.7	レコード編集の挿入データの書き方	142
4.3.8	コマンド実行時のロケール (UNIX の場合)	142
4.3.9	コマンド実行時のコードページ (Windows の場合)	144
4.4	exsort パラメタ ファイルの表記規則	146
4.4.1	ファイル形式	146
4.4.2	コーディング規則	146
4.4.3	選択条件式の書き方	148
4.4.4	レコード編集の挿入データの書き方	148
4.4.5	環境変数	148
4.4.6	ファイル名およびディレクトリ名	151
4.4.7	パラメタファイルを記述する文字コード	152
4.4.8	Unicode シグニチャ (BOM) 付きパラメタファイルの扱い	154

## 5

exsort コマンドのパラメタおよびオペランドの詳細		157
5.1	パラメタおよびオペランドの組み合わせ	159
5.2	-p パラメタ	166
5.3	-function パラメタ	171
5.4	-locatajst パラメタ	172
5.5	-cmdopt パラメタ	174
5.5.1	重複レコードの入力順序保存機能 (DUP   NODUP オプション)	178
5.5.2	ファイルの上書きオプション (OVWRT   INPOVWRT オプション)	179
5.5.3	EBCDIK コード比較オプション (EBCJ, EBCE オプション)	180
5.5.4	入力ファイルのエンディアンオプション (BIG   INPBIG   INPLTL オプション)	181
5.5.5	出力ファイルのエンディアン変更オプション (OUTBIG   OUTLTL オプション)	182
5.5.6	強制マージオプション (FMRG オプション)	183
5.5.7	10 進数オーバーフロー判定オプション (OVFLW オプション)	184
5.5.8	内部 10 進数符号保障オプション (ABS オプション)	185
5.5.9	10 進キーのチェックオプション (KEYNOCHK オプション)	185
5.5.10	10 進集約項目のチェックオプション (SUMNOCHK オプション)	192
5.5.11	キー位置のチェックオプション (POSNOCHK オプション)	195

5.5.12	日付データのチェックオプション (DATENOCHK オプション)	202
5.5.13	日付時刻データのチェックオプション (DATETIMENOCHK オプション)	207
5.5.14	入出力ファイル共用・排他オプション (NOLOCK   EXC オプション)	209
5.5.15	キャッシュファイルの同期出力オプション (SYNC オプション)	210
5.5.16	強制ファイル分割機能 (FSPLIT オプション)	210
5.5.17	ファイル逐次 OPEN/CLOSE 機能 (SEQOPNCLS オプション)	211
5.5.18	連続区切り文字オプション (INPCONSEPA, OUTCONSEPA オプション)	211
5.5.19	ナル文字列の集約オプション (SUMNULKEY, SUMNONULL オプション)	215
5.5.20	集約結果オーバーフロー検知オプション (OVFERR オプション)	215
5.5.21	シフト JIS コード比較オプション (SJIS オプション)	216
5.5.22	ゼロデータ比較オプション (DIFFZERO オプション)	222
5.5.23	空行削除オプション (DELBLANKLINE オプション)	232
5.5.24	CSV 数値データ (固定小数点数) の形式チェックオプション (CSVFRMCHK オプション)	235
5.5.25	選択条件式のブレイクオプション (SELBRK オプション)	238
5.6	-key パラメタ	241
5.7	-sum パラメタ	252
5.8	-inpfle パラメタ	258
5.8.1	fileorg オペランド	258
5.8.2	file オペランド	259
5.8.3	reclen オペランド	259
5.8.4	skiprec オペランド	262
5.8.5	stopaft オペランド	263
5.8.6	startrec オペランド	264
5.8.7	recsepa オペランド	266
5.8.8	cellsepa オペランド	268
5.8.9	quottype オペランド	269
5.8.10	currency オペランド	270
5.8.11	round オペランド	271
5.8.12	nullseq オペランド	273
5.8.13	codetype オペランド	274
5.8.14	reorg オペランド	275
5.8.15	csvreorg オペランド	290
5.8.16	[include   omit] オペランド	299
5.8.17	[csvinclude   csvomit] オペランド	315
5.9	-reorg パラメタ	337
5.10	-outfile パラメタ	338

5.10.1	fileorg オペランド	339
5.10.2	file オペランド	342
5.10.3	reclen オペランド	342
5.10.4	minlen オペランド	347
5.10.5	recsepa オペランド	349
5.10.6	cellsepa オペランド	354
5.10.7	cellnum オペランド	357
5.10.8	quottype オペランド	358
5.10.9	padding オペランド	360
5.10.10	reorg オペランド	362
5.10.11	csvreorg オペランド	362
5.10.12	{include   omit} オペランド	363
5.10.13	{csvinclude   csvomit} オペランド	363
5.11	-sorttmpdir パラメタ	364
5.12	-mergetmpdir パラメタ	366
5.13	-worksize パラメタ	368
5.14	-dateopt パラメタ	372
5.15	-altseq パラメタ	373
5.16	-csvkey パラメタ	375
5.17	-csvsum パラメタ	389
5.18	-csvreorg パラメタ	395
5.19	-iosize パラメタ	396
5.20	-splitfile パラメタ	400
5.20.1	fileno オペランド	400
5.20.2	method オペランド	400
5.21	-splitdata パラメタ	404
5.21.1	recno オペランド	404
5.21.2	spno オペランド	405
5.21.3	nosplitdata オペランド	405
5.21.4	nosplitcsvdata オペランド	413
5.22	-splitkey パラメタ	421
5.22.1	key オペランド	421
5.22.2	csvkey オペランド	422
5.22.3	recno オペランド	423
5.22.4	spno オペランド	423
5.23	-splitinp パラメタ	426

5.23.1	fileorg オペランド	426
5.23.2	file オペランド	427
5.23.3	reclen オペランド	427
5.23.4	skiprec オペランド	427
5.23.5	stopaft オペランド	427
5.23.6	startrec オペランド	427
5.23.7	recsepa オペランド	428
5.23.8	cellsepa オペランド	428
5.23.9	quottype オペランド	428
5.23.10	currency オペランド	428
5.23.11	round オペランド	428
5.23.12	nullseq オペランド	429
5.23.13	codetype オペランド	429
5.23.14	reorg オペランド	429
5.23.15	csvreorg オペランド	429
5.23.16	{include   omit} オペランド	430
5.23.17	{csvinclude   csvomit} オペランド	430
5.24	<b>-splitout パラメタ</b>	431
5.24.1	fileorg オペランド	432
5.24.2	{suffix   prefix} オペランド	432
5.24.3	reclen オペランド	437
5.24.4	minlen オペランド	437
5.24.5	recsepa オペランド	437
5.24.6	cellsepa オペランド	437
5.24.7	cellnum オペランド	438
5.24.8	quottype オペランド	438
5.24.9	padding オペランド	438
5.24.10	reorg オペランド	438
5.24.11	csvreorg オペランド	438
5.24.12	{include   omit} オペランド	439
5.24.13	{csvinclude   csvomit} オペランド	439
5.25	<b>-splittmpdir パラメタ</b>	440
5.26	<b>-catinp パラメタ</b>	441
5.26.1	fileorg オペランド	443
5.26.2	file オペランド	443
5.26.3	{suffix   prefix} オペランド	443
5.26.4	nameseq オペランド (Windows の場合)	445

5.26.5	reclen オペランド	447
5.26.6	skiprec オペランド	447
5.26.7	stopaft オペランド	447
5.26.8	startrec オペランド	447
5.26.9	recsepa オペランド	448
5.26.10	cellsepa オペランド	448
5.26.11	quottype オペランド	448
5.26.12	currency オペランド	448
5.26.13	round オペランド	448
5.26.14	nullseq オペランド	449
5.26.15	codetype オペランド	449
5.26.16	reorg オペランド	449
5.26.17	csvreorg オペランド	449
5.26.18	{include   omit} オペランド	449
5.26.19	{csvinclude   csvomit} オペランド	450
5.27	-catout パラメタ	451
5.27.1	fileorg オペランド	451
5.27.2	file オペランド	451
5.27.3	reclen オペランド	452
5.27.4	minlen オペランド	452
5.27.5	recsepa オペランド	452
5.27.6	cellsepa オペランド	452
5.27.7	cellnum オペランド	453
5.27.8	quottype オペランド	453
5.27.9	padding オペランド	453
5.27.10	reorg オペランド	453
5.27.11	csvreorg オペランド	454
5.27.12	{include   omit} オペランド	454
5.27.13	{csvinclude   csvomit} オペランド	454
5.28	-cpyinp パラメタ	455
5.28.1	fileorg オペランド	455
5.28.2	入力ファイル名	456
5.29	-cpyout パラメタ	457
5.29.1	出力ファイル名	457

## 6

SORT EE の使用例	459
6.1 サンプルファイルの格納ディレクトリ	460
6.2 ソート機能の使用例	461
6.2.1 ソート	461
6.2.2 選択+ソート	461
6.2.3 選択+ソート+集約	462
6.2.4 ソート+ altseq	463
6.2.5 ソート+ csv ファイル	464
6.2.6 ソート+均等分割	465
6.2.7 ソート+定数分割	466
6.2.8 ソート+キー分割	467
6.2.9 ソート+ファイル連結	468
6.2.10 ソート+集約+ CSV ファイル	469
6.2.11 ソート+出力レコード選択+出力レコード編集+ファイル分割（キー分割）+ CSV ファイル	470
6.2.12 ソート+ CSV ファイル+ UCS-2LE	474
6.2.13 ソート+符号なし 2 進数のマスクビット指定キー	475
6.2.14 ソート+ CSV ファイル+ UTF-16LE	476
6.3 マージ機能の使用例	478
6.3.1 マージ	478
6.3.2 マージ+集約	478
6.3.3 マージ+出力選択+出力レコード編集	479
6.4 コピー機能の使用例	481
6.4.1 選択	481
6.4.2 集約	481
6.4.3 ファイルコピー	482
6.4.4 レコード編集	483
6.4.5 レコードの読み飛ばしと打ち切り	484
6.4.6 コピー+選択+ CSV ファイル	484
6.4.7 コピー+ファイル編成変更	485
6.4.8 コピー+レコード長変更	486
6.4.9 簡易コピー	486
6.5 altseq 変換ファイルの作成例	488

## 第3編 SORT

<b>7</b>	<b>ソートコマンドの使い方</b>	<b>489</b>
7.1	ソートコマンドの使い方 (Windows の場合)	490
7.1.1	コマンドの使い方	490
7.1.2	オペランドの意味	490
7.1.3	コマンド指定時の注意事項	496
7.1.4	使用例	498
7.2	ソートコマンドの使い方 (UNIX の場合)	502
7.2.1	コマンドの使い方	502
7.2.2	オペランドの意味	502
7.2.3	コマンド指定時の注意事項	509
7.2.4	使用例	510
<b>8</b>	<b>マージコマンドの使い方</b>	<b>517</b>
8.1	マージコマンドの使い方 (Windows の場合)	518
8.1.1	コマンドの使い方	518
8.1.2	オペランドの意味	518
8.1.3	コマンド指定時の注意事項	522
8.1.4	使用例	523
8.2	マージコマンドの使い方 (UNIX の場合)	527
8.2.1	コマンドの使い方	527
8.2.2	オペランドの意味	527
8.2.3	コマンド指定時の注意事項	532
8.2.4	使用例	534
<b>9</b>	<b>選択コマンドの使い方</b>	<b>537</b>
9.1	選択コマンドの使い方 (Windows の場合)	538
9.1.1	コマンドの使い方	538
9.1.2	オペランドの意味	538
9.1.3	コマンド指定時の注意事項	545
9.1.4	使用例	547
9.2	選択コマンドの使い方 (UNIX の場合)	551
9.2.1	コマンドの使い方	551



9.2.2	オペランドの意味	551
9.2.3	コマンド指定時の注意事項	563
9.2.4	使用例	565

## 10 集約コマンドの使い方 571

10.1	集約コマンドの使い方 (Windows の場合)	572
10.1.1	コマンドの使い方	572
10.1.2	オペランドの意味	572
10.1.3	コマンド指定時の注意事項	578
10.1.4	使用例	579
10.2	集約コマンドの使い方 (UNIX の場合)	581
10.2.1	コマンドの使い方	581
10.2.2	オペランドの意味	581
10.2.3	コマンド指定時の注意事項	586
10.2.4	使用例	588

## 11 拡張 SORT コマンドの使い方 (Windows の場合) 593

11.1	コマンドの使い方	594
11.1.1	拡張 SORT コマンドの形式	594
11.2	オペランドの意味	595
11.2.1	パラメタファイルの形式	595
11.2.2	拡張 SORT コマンドのパラメタの意味	597
11.3	コマンド指定時の注意事項	609
11.3.1	パラメタファイル作成時の注意事項	609
11.4	使用例	610
11.4.1	レコードを選択したあとにソートする例	610
11.4.2	レコードをソートしたあとに集約する例	611

## 12 シェル・スクリプトコマンドの使い方 (UNIX の場合) 613

12.1	コマンドの使い方	614
12.1.1	コマンドの形式	614
12.2	オペランドの意味	615
12.2.1	機能オプション	615
12.2.2	入力ファイルのファイル編成	615
12.2.3	出力ファイルのファイル編成	615

12.2.4	-L レコード長	615
12.2.5	-o 出力ファイル名	615
12.2.6	-y ワークバッファサイズ	615
12.2.7	-T ソート用一時ファイルのディレクトリ名	615
12.2.8	-T2 マージ用一時ファイルのディレクトリ名 2	616
12.2.9	-w 中間ファイルのディレクトリ	616
12.2.10	キー属性	616
12.2.11	+ キー位置 - キー長	616
12.2.12	-a 集約項目情報	616
12.2.13	入力ファイル名	616
12.3	コマンド指定時の注意事項	618
12.3.1	スクリプト・ファイル	618
12.3.2	中間ファイルの削除	618
12.3.3	中間ファイルの容量	618
12.3.4	中間ファイルのエラー	618
12.4	使用例	619
12.4.1	文字項目をキーとしてソートし集約する例	619

## 13 C 言語プログラムでの使い方 (Windows の場合) 621

13.1	関数の機能と形式	622
13.1.1	関数の一覧	622
13.1.2	関数の詳細	624
13.2	コーディング例	640
13.3	コンパイル方法	643
13.4	詳細エラーコード	644

## 第 4 編 運用

## 14 環境変数の設定 647

14.1	SORT EE の環境変数の設定	648
14.1.1	32 ビット版 SORT EE の環境変数	654
14.1.2	64 ビット版 SORT EE の環境変数	674
14.1.3	ワークサイズ設定の優先順位	693
14.1.4	入出力ファイル名の指定	694

14.2	SORT の環境変数の設定 (UNIX の場合)	696
14.2.1	SORT の環境変数	696

<b>15</b>	<b>環境ファイルの設定</b>	<b>701</b>
15.1	SORT EE の環境ファイルの設定	702
15.1.1	環境ファイルに指定できる項目	702
15.2	SORT の環境ファイルの設定 (Windows の場合)	719
15.2.1	SORT の環境ファイルのパラメタの意味	719
15.2.2	SORT の環境ファイルの有効範囲	722
15.2.3	SORT の環境ファイル指定時の注意事項	723

<b>16</b>	<b>一時ファイルの作成条件とファイル容量の算出方法</b>	<b>725</b>
16.1	一時ファイルの種類	726
16.2	SORT EE の一時ファイルの作成条件	728
16.2.1	ソート用一時ファイルおよびマージ用一時ファイルを作成する場合のワークバッファサイズ	728
16.2.2	ソート用一時ファイルだけを作成する場合のワークバッファサイズ	728
16.2.3	一時ファイルを作成しない場合のワークバッファサイズ	728
16.3	SORT EE の一時ファイル容量の算出方法	730
16.4	SORT EE のファイル分割用一時ファイルの作成条件	731
16.5	SORT EE のファイル分割用一時ファイル容量の算出方法	732
16.6	SORT の一時ファイルの作成条件	733
16.6.1	ソート用一時ファイルおよびマージ用一時ファイルを作成する場合のワークバッファサイズ	733
16.6.2	ソート用一時ファイルだけを作成する場合のワークバッファサイズ	733
16.6.3	一時ファイルを作成しない場合のワークバッファサイズ	733
16.7	SORT の一時ファイル容量の算出方法	735

<b>17</b>	<b>性能チューニング方法</b>	<b>737</b>
17.1	ソート機能の処理時間を短くするためには	738
17.1.1	ワークバッファサイズと処理時間	738
17.1.2	SORT EE のワークバッファサイズ自動見積もり機能	742
17.1.3	ファイルの配置	743
17.1.4	I/O サイズ	744
17.1.5	レコード長の指定	744

17.1.6	キー属性の指定	745
17.1.7	コマンドオプションの指定	745
17.1.8	マルチバイト文字データ	745
17.1.9	最適化したソートライブラリの使用【Linux(x64) SORT EE(64) 08-41以降】	745
17.2	ソート機能以外の機能の処理時間を短くするためには	746
17.2.1	ファイルの配置	746
17.2.2	I/O サイズ	746
17.2.3	マルチバイト文字データ	746
17.2.4	最適化したソートライブラリの使用【Linux(x64) SORT EE(64) 08-41以降】	747

## 18 SORT の機能拡張オプションの使い方 749

18.1	SORT の機能拡張オプションの環境設定	750
18.1.1	機能拡張オプションを有効にする	750
18.1.2	使用する SORT EE 機能を有効にする	750
18.2	SORT の機能拡張オプションの注意事項と見直しポイント	754
18.2.1	メモリ所要量	754
18.2.2	メッセージ	755
18.2.3	ディスク使用量	755
18.2.4	選択条件式の文字条件値の円記号 (¥) の扱い	755

## 19 障害時の対処方法 757

19.1	SORT EE の障害情報	758
19.1.1	SORT EE の入出力障害時のファイルの扱い	758
19.1.2	デバッグ情報ファイル	758
19.1.3	ログファイル	762
19.1.4	完了メッセージオプション	766
19.1.5	キーまたは項目の不正による障害の対処方法	767
19.2	SORT の障害情報	769
19.2.1	SORT の入出力障害時のファイルの扱い	769
19.2.2	デバッグ情報ファイル	769
19.2.3	キーまたは項目の不正による障害の場合	771
19.2.4	C 言語で作成したプログラムの不正による障害の場合 (Windows の場合)	773

## 20 SORT から SORT EE への移行ガイド 775

20.1	SORT と SORT EE の処理比較	776
------	----------------------	-----

20.2	UNIX 版 SORT コマンドの移行	778
20.2.1	rsort コマンド	778
20.2.2	rmerge コマンド	779
20.2.3	rselect コマンド	779
20.2.4	rsum コマンド	782
20.2.5	rsortsum コマンド	783
20.2.6	コマンド共通項目	784
20.2.7	SORT 環境変数の移行	788
20.3	Windows 版 SORT コマンドの移行	790
20.3.1	wnsort コマンド	790
20.3.2	wnmerge コマンド	791
20.3.3	wnselect コマンド	791
20.3.4	wsum コマンド	792
20.3.5	拡張 SORT コマンド (wnsortex コマンド)	793
20.3.6	コマンド共通項目	795
20.3.7	SORT 環境ファイルの移行	801
20.4	ワークバッファサイズ	803

## 付録 805

付録 A	JIS8 単位コード体系と EBCDIK コード体系の変換表	806
付録 B	CSV ファイルのデータの扱い (SORT EE の機能)	808
付録 B.1	数値 (固定小数点数) データの端数処理	808
付録 B.2	文字列データの端数処理	810
付録 B.3	日付データの端数処理	811
付録 B.4	引用符のエスケープ	812
付録 B.5	CSV 数値データ (固定小数点数) の除外文字の位置規則	813
付録 C	CSV ファイル形式と CSV データ形式 (SORT EE の機能)	815
付録 C.1	CSV ファイル形式	815
付録 C.2	CSV データ形式	816
付録 D	CSV ファイル内のナル文字列の扱い (SORT EE の機能)	820
付録 D.1	ソートキー・マージキー	820
付録 D.2	集約キー	822
付録 D.3	比較項目	823
付録 D.4	集約項目	825
付録 E	左分離符号付き数字のデータ形式 (SORT EE の機能)	827
付録 F	レコード編集処理で基になるレコード (SORT EE の機能)	830

付録 G altseq 変換ファイル (SORT EE の機能)	833
付録 G.1 altseq 変換ファイルの詳細	833
付録 G.2 altseq 変換ファイルの指定方法	833
付録 H Unicode 操作文字コード一覧 (SORT EE)	835
付録 I マルチバイト文字データの扱い	844
付録 I.1 文字コード体系とエンコード	844
付録 I.2 文字列データの扱い	847
付録 I.3 数値データの扱い	867
付録 I.4 日付データの扱い	869
付録 J 日付時刻データの形式 (SORT EE の機能)	871
付録 K メッセージ	878
付録 K.1 SORT EE メッセージ	878
付録 K.2 SORT EE のメッセージ一覧	880
付録 K.3 SORT メッセージ	966
付録 K.4 SORT のメッセージ一覧	967
付録 L 各バージョンの変更内容	993
付録 M このマニュアルの参考情報	1009
付録 M.1 関連マニュアル	1009
付録 M.2 このマニュアルでの表記	1009
付録 M.3 英略語	1012
付録 M.4 KB (キロバイト) などの単位表記について	1012
付録 N 用語解説	1013

## 索引

1019

# 1

## SORT EE と SORT

ファイルに保管されたデータは、いろいろなケースで利用されています。例えば、社員データファイルのデータは、社員名簿や社員別の合計売上実績データなどに利用されます。このようにファイル中のデータをそれぞれの利用目的に合った形で出力するとき、**SORT EE** および **SORT** で提供するソート機能、マージ機能、選択機能および集約機能が役立ちます。これらの機能のほかにも、**SORT EE** ではコピー機能、編集機能、ファイル分割機能、およびファイル連結機能を提供します。この章では、**SORT EE** と **SORT** の機能の概要について説明します。

---

### 1.1 SORT EE と SORT の概要

---

### 1.2 ソート機能とは

---

### 1.3 マージ機能とは

---

### 1.4 コピー機能とは（SORT EE の機能）

---

### 1.5 選択機能とは

---

### 1.6 集約機能とは

---

### 1.7 編集機能とは（SORT EE の機能）

---

### 1.8 ファイル分割機能とは（SORT EE の機能）

---

### 1.9 ファイル連結機能とは（SORT EE の機能）

---

### 1.10 ファイル編成変更機能とは（SORT EE の機能）

---

## 1. SORT EE と SORT

---

### 1.11 処理の流れの概要

---



## 1.1 SORT EE と SORT の概要

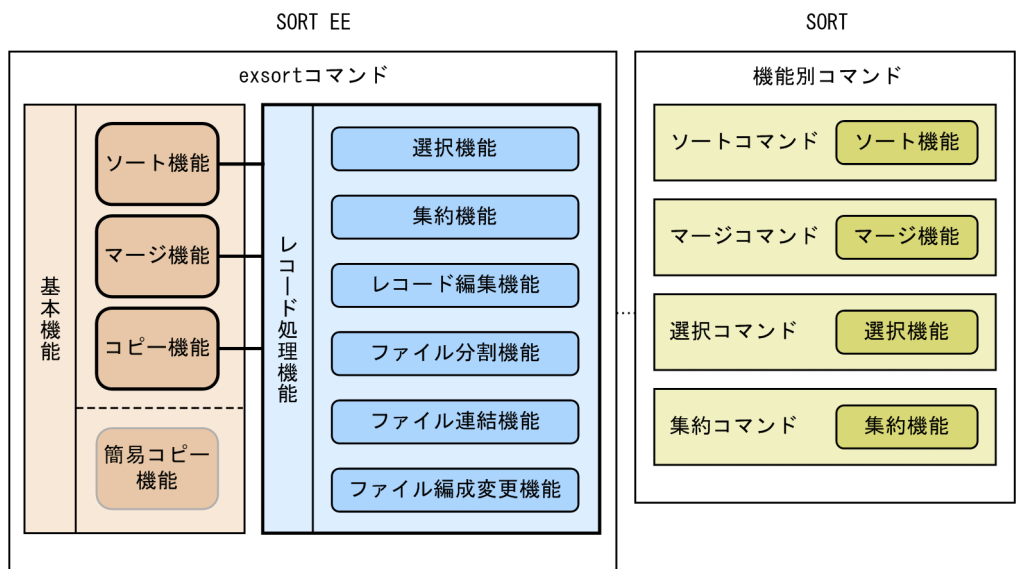
SORT EE は、1 コマンドで、基本機能およびレコード処理機能の両方を使用できます。基本機能には、ソート機能、マージ機能およびコピー機能があります。さらに、コピー機能には、単純なコピーだけを行う簡易コピー機能があります。簡易コピー機能は、レコード処理機能と組み合わせることはできませんが、コマンド指定が簡単でレコード長とレコード件数の制限がありません。

レコード処理機能には、選択機能、集約機能、編集機能、ファイル分割機能、ファイル連結機能およびファイル編成変更機能があります。

SORT には、ソート機能、マージ機能、選択機能および集約機能があり、それぞれの機能ごとにコマンドを提供しています。

SORT EE および SORT の機能比較を次の図に示します。

図 1-1 SORT EE および SORT の機能比較



SORT と SORT EE の入出力構成をそれぞれの図に示します。

# 1. SORT EE と SORT

図 1-2 SORT の入出力構成

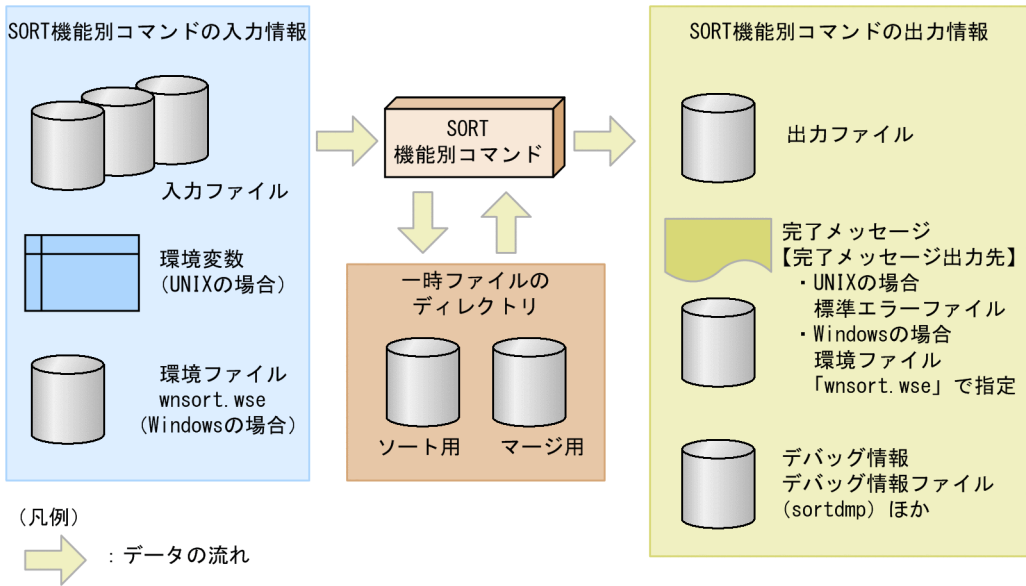
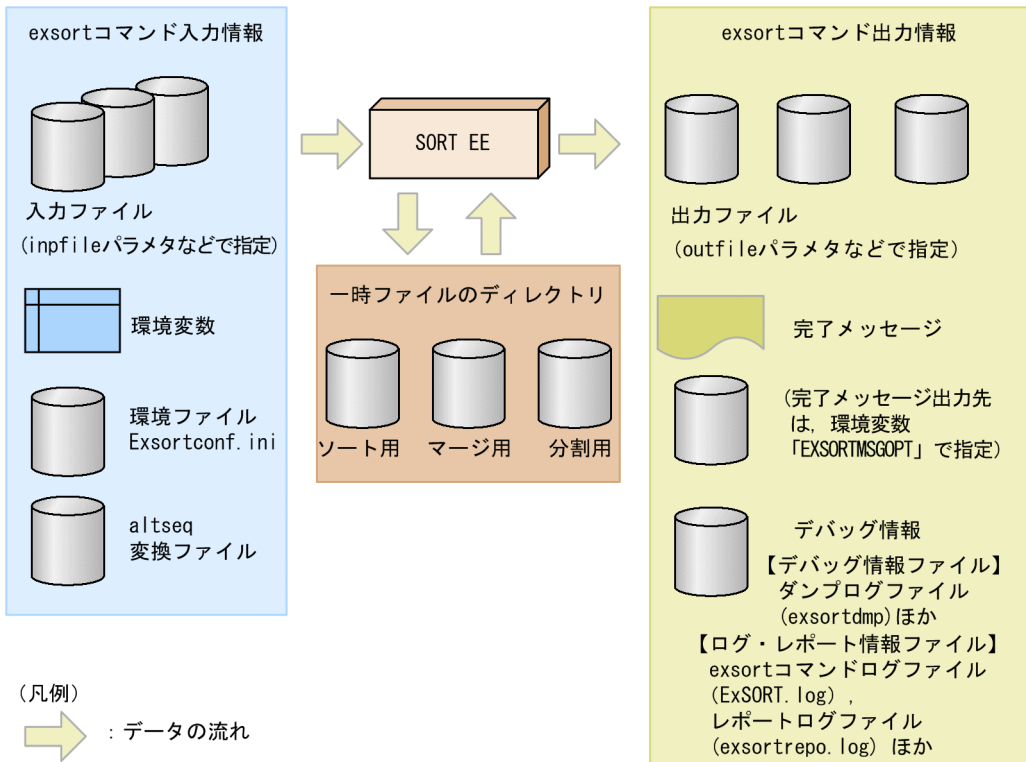


図 1-3 SORT EE の入出力構成



## 1.2 ソート機能とは

ソート機能とは、ファイル内のレコードをデータ中の文字または数字をキーとして、昇順または降順に並べ替える機能です。

例えば、社員番号、氏名、および年齢を登録した社員データファイルを社員番号の昇順に並べ替えることができます。このときのキーは、社員番号です。また、この社員データファイルを年齢をキーにして降順に並べ替えることもできます。

キーにする文字または数字は、任意に指定できます。

社員データファイルを社員番号の昇順にソートした例を次の図に示します。

図 1-4 社員データファイルのソートの例

<ソート前>

社員番号	氏名	年齢
1350	日立	41
1020	鈴木	23
2110	田中	35
2120	山田	33
1930	小林	20
3001	渡辺	27



<ソート後>

社員番号	氏名	年齢
1020	鈴木	23
1350	日立	41
1930	小林	20
2110	田中	35
2120	山田	33
3001	渡辺	27

(凡例)  : ソートのキーを示す。

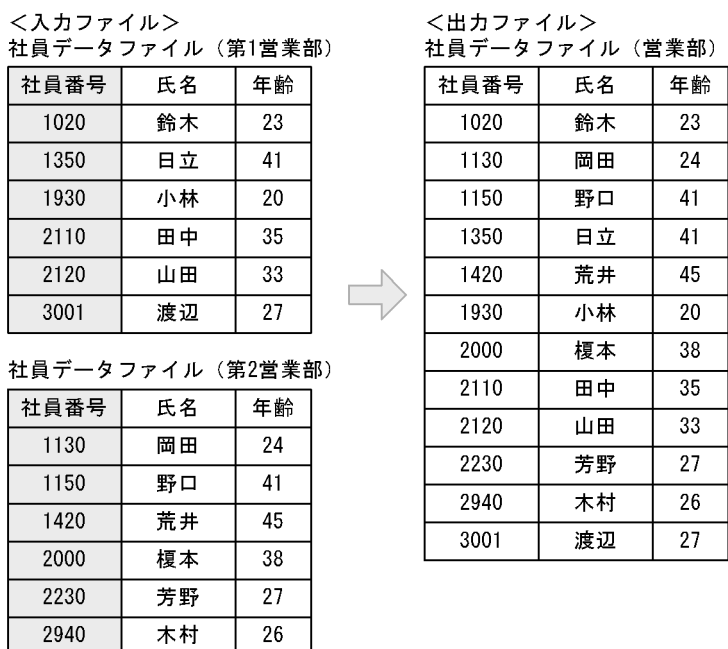
## 1.3 マージ機能とは

マージ機能とは、複数のファイルのデータを一つのファイルに併合する機能です。併合は、データ中の文字または数字をキーとします。

例えば、第1営業部の社員データファイルと第2営業部の社員データファイルを社員番号をキーにして一つの社員データファイルに併合できます。このとき、昇順または降順を任意に指定し、併合できます。ただし、併合する各ファイルはあらかじめ昇順または降順でソートしておく必要があります。

マージの例を次の図に示します。

図 1-5 マージの例



(凡例)  : マージのキーを示す。

注 入力ファイルは、昇順でソート済みである。

## 1.4 コピー機能とは（SORT EE の機能）

---

コピー機能とは、入力ファイルを出力ファイルに複写する機能です。コピー機能は SORT EE の場合だけ使用できます。

例えば、社員番号、氏名および年齢を登録した社員データファイルを社員データバックアップファイルとして複写できます。

社員データファイルを社員データバックアップファイルに複写した例を次の図に示します。

図 1-6 入力ファイルを出力ファイルに複写する例

<入力ファイル>

社員番号	氏名	年齢
1350	日立	41
1020	鈴木	23
2110	田中	35
2120	山田	33
1930	小林	20
3001	渡辺	27



<出力ファイル>

社員番号	氏名	年齢
1350	日立	41
1020	鈴木	23
2110	田中	35
2120	山田	33
1930	小林	20
3001	渡辺	27

## 1.5 選択機能とは

選択機能とは、条件値とデータを比較し、条件に一致したデータを持つレコードをファイルから選択する機能です。選択機能では、条件値と比較されるデータを格納している項目を、比較項目とします。

例えば、社員番号、氏名、および年齢を登録した社員データファイルから、年齢が 22 歳以上のレコードを選択する場合、年齢を比較項目にします。また、社員番号が 1,000 番台のデータを選択する場合は、社員番号を比較項目にします。

比較項目に対する条件値は、文字または数字で任意に指定できます。

年齢が 22 歳から 27 歳までのレコードを選択し、出力した例を次の図に示します。この場合、比較項目は年齢、条件値は 22, 27 です。

図 1-7 選択機能の使用例

<選択機能使用前>

社員番号	氏名	年齢
2000	榎本	38
2940	木村	26
2120	山田	33
3001	渡辺	27
1930	小林	20
1130	岡田	24
1020	鈴木	23
1420	荒井	45
1350	日立	41



<選択機能使用后>

社員番号	氏名	年齢
2940	木村	26
3001	渡辺	27
1130	岡田	24
1020	鈴木	23

(凡例)  : 比較項目を示す。

## 1.6 集約機能とは

集約機能とは、同一ファイル内で同一キーを持つ複数のレコードを一つのレコードに併合し、指定した項目の値を合計する機能です。集約は、データ中の文字または数字をキーとします。

例えば支店コード、担当者名、および売上数量を登録した社員データファイルを支店コードと担当者名をキーとし、売上数量を合計して集約する場合、**集約項目**は、売上数量です。ただし、ファイルはあらかじめ昇順または降順でソートしておく必要があります。ソートしていない場合、集約の結果は保証しません。

支店コードと担当者名をキーとして、売上数量を集約した例を次の図に示します。

図 1-8 集約機能の使用例

<集約機能使用前>

支店コード	担当者名	売上数量
03	田中	7
03	田中	2
04	渡辺	5
05	日立	8
06	山田	6
06	山田	10



<集約機能使用后>

支店コード	担当者名	売上数量
03	田中	9
04	渡辺	5
05	日立	8
06	山田	16

(凡例)  : 集約項目を示す。

## 1.7 編集機能とは（SORT EE の機能）

---

**編集機能**とは、レコード内の任意の部分を選択したり、任意のデータを埋め込んだりすることでレコードを再編成する機能です。編集機能は SORT EE の場合だけ使用できます。

例えば、次の項目を登録した売上マスタファイルから、商品コード、売上金額、売上日、および売上数量だけを抽出して新しい売上マスタファイルを作成できます。

- 支店コード
- 担当者名
- 売上日
- 商品コード
- 売上数量
- 売上金額
- 売上傳票番号

また、項目を抽出すると同時に、売上日、売上数量、商品コード、および売上金額の順番に並べ替えることもできます。

レコード編集した例を次の図に示します。



図 1-9 売上マスタファイルのレコード編集例

&lt;レコード編集前&gt; 売上マスタファイル

支店コード	担当者名	商品コード	伝票番号	売上金額	売上日	売上数量
03	田中	G1375	9002	3500	10/09	7
03	田中	G1246	9003	1400	10/10	2
03	江藤	G1375	9007	500	10/24	1
03	江藤	G1140	9008	1500	10/25	5
04	渡辺	G1246	9000	3500	10/07	5
04	上田	G1140	9005	300	10/15	1
05	日立	G1375	9004	4000	10/15	8
06	山田	G1246	9001	4200	10/08	6
06	山田	G1375	9006	5000	10/23	10



&lt;レコード編集後&gt; 新売上ファイル

売上日	売上数量	商品コード	売上金額
10/09	7	G1375	3500
10/10	2	G1246	1400
10/24	1	G1375	500
10/25	5	G1140	1500
10/07	5	G1246	3500
10/15	1	G1140	300
10/15	8	G1375	4000
10/08	6	G1246	4200
10/23	10	G1375	5000

(凡例)  : 編集フィールドを示す。

## 1.8 ファイル分割機能とは（SORT EE の機能）

ファイル分割機能とは、入力ファイルを指定された分割基準を基に分割し、出力ファイルへ出力する機能です。ファイル分割方法を次の表に示します。

表 1-1 ファイル分割方法

分割方法 (パラメタ)	分割基準	分割 ファイル数	ファイル分割の求め方※
均等分割 (-splitfile)	出力ファイル数 (平均レコード件数)	任意	平均レコード件数=総レコード件数÷分割 ファイル数
	出力ファイル数 (平均ファイルサイズ)		平均ファイルサイズ=総ファイルサイズ÷ 分割ファイル数
定数分割 (-splitdata)	指定レコード件数	不定	分割ファイル数=総レコード件数÷指定レ コード件数
	指定ファイルサイズ		分割ファイル数=総ファイルサイズ÷指定 ファイルサイズ
キー分割 (-splitkey)	分割キー	不定	分割ファイル数は、キーの種類や指定個数 によって変動

### 注※

分割ファイル数は、計算結果の切り上げになります。

例えば、定数分割（分割基準：指定レコード件数）の場合、総レコード件数 100 件、指定レコード件数 3 件のとき、3 レコードのファイルが 33 個と 1 レコードのファイルが 1 個作成されます。

### <分割方法>

#### 均等分割（-splitfile パラメタ指定）

- 指定された分割ファイル数を基に、ファイルを均等に分割して出力します。
- 分割した出力ファイルの平均値は、入力ファイルのレコード件数またはファイルサイズを基に算出します。

#### 定数分割（-splitdata パラメタ指定）

- 指定されたデータ量（レコード件数またはファイルサイズ）で、入力ファイルを分割して出力します。
- 出力する分割ファイルの数は不定です。

#### キー分割（-splitkey パラメタ指定）

- 指定されたキーでファイルを分割して出力します。
- 出力する分割ファイルの数は不定です。

### <オプション>

- 終了メッセージは、出力ファイル数だけ出力します。これらのメッセージを環境変数によって抑止できます。詳細については、「2.24 メッセージ抑止機能（SORT EE の機能）」を参照してください。

- 環境変数によって、出力するレコードが 0 件のファイルについてファイル出力抑止ができます。詳細については、「2.25 0 件ファイル出力抑止機能 (SORT EE の機能)」を参照してください。
- 出力ファイルの逐次オープンができます。詳細については、「2.23 ファイル逐次 OPEN/CLOSE 機能 (SORT EE の機能)」を参照してください。

ファイル分割機能の実行順序については、「4.1.1 機能の実行順序」を参照してください。

## 1.9 ファイル連結機能とは (SORT EE の機能)

---

ファイル連結機能で次の機能が使用できます。

- 指定された複数のファイルを、一つまたは複数のファイルに出力します。
- 入力ファイルは、総称名または格納先ディレクトリ名を指定できます。
- 環境変数で指定した入力ファイルが存在しない場合、エラー終了しないで、存在しない旨のインフォメーションメッセージ (KBLS2100-I) を出力して処理を続行できます。ただし、連結するファイルがない場合は、エラー終了します。
- 終了メッセージは入力ファイル数だけ出力します。出力する終了メッセージを環境変数によって抑止できます。詳細については、「2.24 メッセージ抑止機能 (SORT EE の機能)」を参照してください。
- Windows の場合、ファイル連結する際の入力ファイルの入力順を選択できます。詳細については、「5.26.4 nameseq オペランド (Windows の場合)」を参照してください。

ファイル連結機能の実行順序については、「4.1.1 機能の実行順序」を参照してください。

## 1.10 ファイル編成変更機能とは（SORT EE の機能）

---

ファイル編成変更機能とは、出力ファイルを、入力ファイルと異なるファイル編成に変更する機能です。次の機能があります。

- 固定長順編成ファイル、可変長順編成ファイル、およびテキストファイルのファイル編成を変更できます。CSV ファイルのファイル編成は変更できません。
- ファイル編成を変更しないで、出力レコード長だけの変更ができます。
- 可変長順編成ファイルのレコード長（RDW：レコードヘッダ）について、入力ファイルと出力ファイルで、それぞれ異なるバイトオーダーを指定できます。

ファイル編成変更機能の実行順序については、「4.1.1 機能の実行順序」を参照してください。

ファイル編成の変更については、「5.10.1 fileorg オペランド」を参照してください。

出力レコード長の変更については、「5.10.3 reclen オペランド」および「5.10.4 minlen オペランド」を参照してください。

可変長順編成ファイルのバイトオーダーの指定については、「5.5.4 入力ファイルのエンディアンオプション（BIG | INPBIG | INPLTL オプション）」および「5.5.5 出力ファイルのエンディアン変更オプション（OUTBIG | OUTLTL オプション）」を参照してください。

## 1.11 処理の流れの概要

SORT EE の基本機能には、ソート機能、マージ機能およびコピー機能があります。ここでは、各機能の処理の流れを説明します。

### 1.11.1 ソート機能の処理の流れ

ソート機能の処理には、指定されたワークバッファのサイズにより次の二つの方式があります。

- インコアソート

入力データ量に対して十分なワークバッファサイズを指定した場合、一時ファイルを使用しないでソートする処理のことです。

- アウトコアソート

入力データ量に対してワークバッファサイズが不足した場合、一時ファイルを使用して、入力データを分割しながらソートする処理のことです。

ソートの処理は、次の三つのフェーズで構成されます。

- ソートフェーズ

- 中間マージフェーズ

- 最終マージフェーズ

ソート処理では、三つのフェーズを順次実行します。ただし、インコアソートの場合、中間マージフェーズはありません。各フェーズの処理の概要を次に示します。

表 1-2 ソート機能の処理の概要

フェーズ	インコアソート		アウトコアソート	
	順序	処理概要	順序	処理概要
ソート	1	入力ファイルからレコードを読み込み、ワークバッファ上でソートキーを比較する。	1	入力ファイルからレコードを読み込み、ワークバッファ上でソートキーを比較し、指定された順序に整列したレコードの集合（ストリング）をソート用一時ファイルへ書き出す。
中間マージ	—	—	2	ソートフェーズで作成したストリングがワークバッファに入らない場合に動作する処理。ソート用一時ファイルのストリングをワークバッファ上に読み込み、ワークバッファに入るストリング数になるまでマージする。マージしたストリングは、マージ用一時ファイルへ書き出す。

フェーズ	インコアソート		アウトコアソート	
	順序	処理概要	順序	処理概要
最終マージ	2	ワークバッファ上のレコードを、指定された順序に整列して出力ファイルへ書き出す。	3	ソート用一時ファイル※のストリングをワークバッファ上に読み込み、一つのストリングにマージして、出力ファイルへ書き出す。

(凡例)

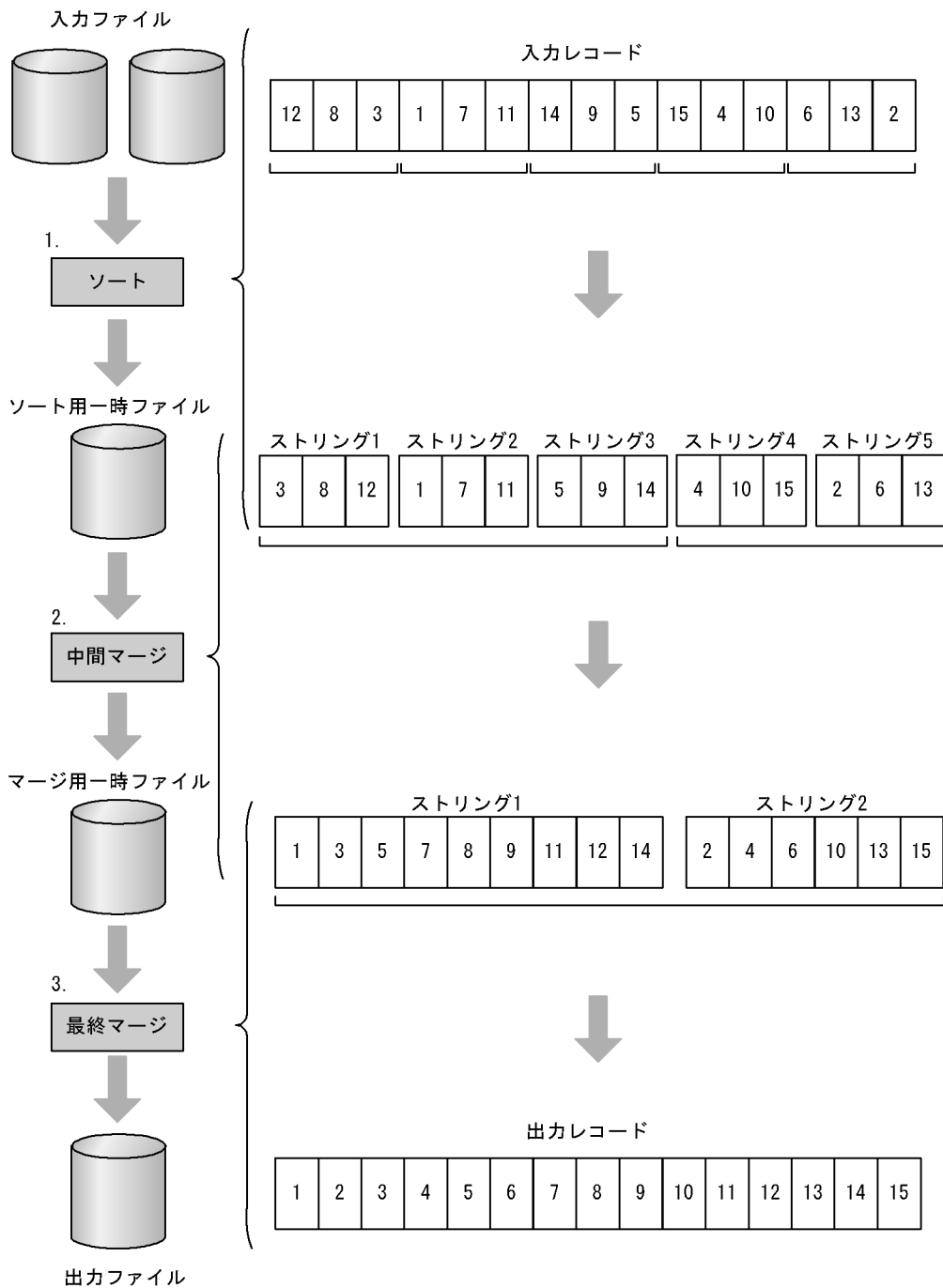
— : なし

注※

中間マージ（順序 2）を実行した場合は、マージ用一時ファイルに出力したストリングを読み込みます。

# 1. SORT EE と SORT

図 1-10 ソート機能の処理の流れ

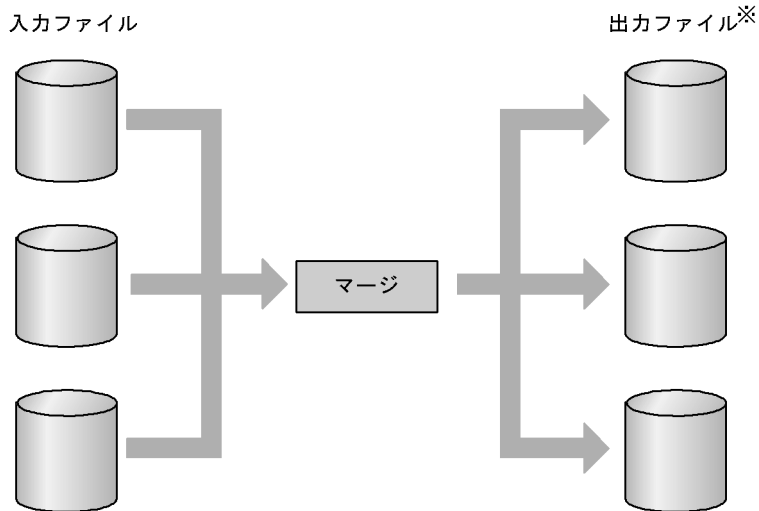




### 1.11.2 マージ機能の処理の流れ

指定されたすべての入力ファイルからレコードを読み込み、マージキーを比較し、指定された順序で出力ファイルへ書き出します。

図 1-11 マージ機能の処理の流れ



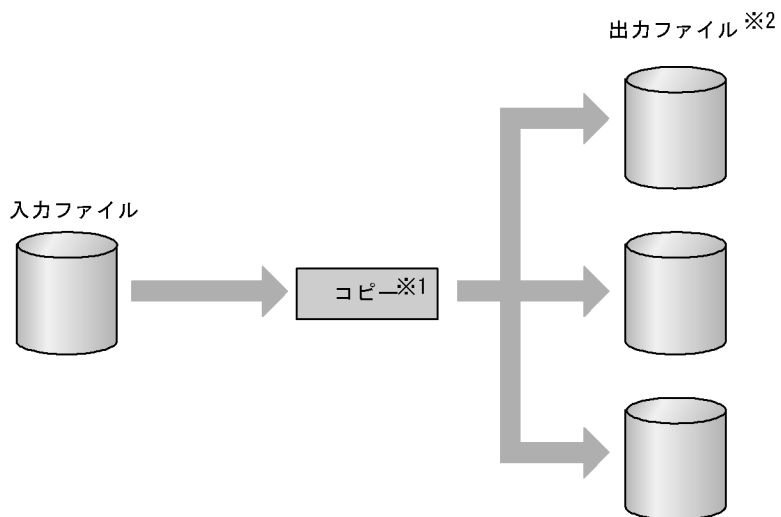
注※ SORT EEでは、最大256個まで出力ファイルを指定可能。  
SORTでは、1個だけ出力ファイルを指定可能。

### 1.11.3 コピー機能の処理の流れ

入力ファイルからレコードを読み込み、出力ファイルへ書き出します。コピー機能は、レコード処理機能（レコード編集、選択、集約など）と組み合わせることもできます。SORTでは選択、または集約機能が、この処理の流れに該当します。選択機能は、指定された条件に合致するレコードだけを出力します。集約機能は、同一キーの複数のレコードを一つのレコードにして出力します。

## 1. SORT EE と SORT

図 1-12 コピー機能の処理の流れ



注※1 コピー機能は、SORT EEの機能。  
SORTでは、選択機能または集約機能となる。

注※2 SORT EEでは、最大256個まで出力ファイルを指定可能。  
SORTでは、1個だけ出力ファイルを指定可能。

# 2

## 便利な機能

ソート機能，マージ機能，コピー機能，選択機能，集約機能，編集機能，ファイル分割機能，およびファイル連結機能のコマンドオプションと組み合わせることができる便利な機能があります。

この章では，SORT EE と SORT の便利な機能について説明します。

---

2.1 重複レコードの入力順序保存機能

---

2.2 EBCDIK コードキー比較機能

---

2.3 強制マージ機能

---

2.4 無条件レコード抽出機能

---

2.5 10 進数オーバーフロー判定機能

---

2.6 エンディアンオプション機能

---

2.7 ファイル排他機能（UNIX の場合）

---

2.8 キャッシュファイルの同期出力機能

---

2.9 内部 10 進数符号保障機能

---

2.10 10 進キーのキーチェック緩和機能（SORT EE の機能）

---

2.11 10 進集約項目のチェック緩和機能（SORT EE の機能）

---

2.12 キー位置のチェック緩和機能（SORT EE の機能）

---

2.13 日付データのチェック緩和機能（SORT EE の機能）

---

2.14 複数ファイル出力機能（SORT EE の機能）

## 2. 便利な機能

- 
- 2.15 入力レコードの読み飛ばし・打ち切り機能 (SORT EE の機能)

---

  - 2.16 文字順序変更機能 (SORT EE の機能)

---

  - 2.17 既存ファイルの書き換え防止機能 (SORT の機能)

---

  - 2.18 I/O サイズ変更機能 (SORT EE の機能)

---

  - 2.19 ワークバッファサイズ自動見積もり機能 (SORT EE の機能)

---

  - 2.20 2GB 超ワークバッファサイズの指定 (64 ビット版 SORT EE の機能)

---

  - 2.21 SORT の機能拡張オプション (SORT の機能)

---

  - 2.22 強制ファイル分割機能 (SORT EE の機能)

---

  - 2.23 ファイル逐次 OPEN/CLOSE 機能 (SORT EE の機能)

---

  - 2.24 メッセージ抑止機能 (SORT EE の機能)

---

  - 2.25 0 件ファイル出力抑止機能 (SORT EE の機能)

---

  - 2.26 キー・項目の上限値拡張機能 (SORT EE の機能)

---

  - 2.27 メッセージ出力オプション (SORT EE の機能)

---

  - 2.28 CSV ファイル関連の便利な機能 (SORT EE の機能)

---

  - 2.29 集約結果オーバーフロー検知オプション (SORT EE の機能)

---

  - 2.30 改行文字の自動判定機能 (SORT EE の機能)

---

  - 2.31 シフト JIS コード比較オプション (SORT EE の機能)

---

  - 2.32 ゼロデータ比較オプション (SORT EE の機能)

---

  - 2.33 空行削除オプション (SORT EE の機能)

---

  - 2.34 ファイルの上書きオプション (SORT EE の機能)

---

  - 2.35 Unicode シグニチャ (BOM) 関連オプション (SORT EE の機能)

---

  - 2.36 マルチバイト文字データの比較 (SORT EE の機能)

---

  - 2.37 内部 10 進数符号オプション (SORT EE の機能)

---

  - 2.38 外部 10 進数データ形式オプション (SORT EE の機能)

---

  - 2.39 符号なし 2 進数のマスクビット指定 (SORT EE の機能)

---

  - 2.40 簡易コピー機能 (SORT EE の機能)

---

  - 2.41 -outfile パラメタ数の制限解除オプション (SORT EE の機能) 【08-50 以降】

---

2.42 exsort パラメタファイルサイズの制限解除オプション (SORT EE の機能) 【08-50 以降】

---

2.43 CSV 数値データ (固定小数点数) の形式チェックオプション (SORT EE の機能) 【08-50 以降】

---

2.44 選択条件式のブレイクオプション (SORT EE の機能) 【08-50 以降】

---

2.45 CSV 日付データの部分指定 (SORT EE の機能) 【08-50 以降】

---

## 2.1 重複レコードの入力順序保存機能

ソートキーの内容の同じレコードが複数ある場合、ソート結果は、ソート時にそのレコードが入力された順に並びます。これを、「重複レコードの入力順序保存機能」といいます。なお、入力ファイルが索引ファイルの場合、この機能は使えません。

製品	オプション名	備考
SORT EE	重複レコードの入力順序保存機能 (DUP   NODUP オプション)	5.5.1
SORT	/d オプション (Windows)	7.1.2(1) 11.2.2(3)
	-d オプション (UNIX)	7.2.2(1)

重複レコードの入力順序保存機能を指定した場合の例を次に示します。

### 重複レコードの入力順序保存機能を指定した場合の例

どちらも年齢が「30」である「田中」および「渡辺」というレコードを「田中」,  
「渡辺」の順で入力したとします。

これらのデータを年齢順でソートした場合、重複レコードの入力順序保存機能を指定しているときは、「田中」,  
「渡辺」の順でソートされます。

重複レコードの入力順序保存機能を指定していないときは、「渡辺」,  
「田中」の順でソートされることがあります。

図 2-1 重複レコードの入力順序保存機能を指定した場合の例


&lt;ソート前&gt;

社員番号	氏名	年齢
1350	日立	41
1020	鈴木	23
2110	田中	30
2120	山田	33
1930	小林	20
3001	渡辺	30



&lt;ソート後&gt;

社員番号	氏名	年齢
1930	小林	20
1020	鈴木	23
2110	田中	30
3001	渡辺	30
2120	山田	33
1350	日立	41

(凡例)  : ソートのキーを示す。

## 2.2 EBCDIK コードキー比較機能

JIS8 単位コードを EBCDIK コードの体系に従ってキーを比較します。

- 入力データの文字コードが JIS8 単位コードの場合、EBCDIK コードの順番にソートできます。
- 入力データの文字コードが EBCDIK コードで、かつ選択条件式の条件値を JIS8 単位コードで記述した場合、条件値を EBCDIK コードとして比較できます。

入力データの文字コード	オプション	説明
JIS8	<ul style="list-style-type: none"> <li>• SORT EE の場合 -cmdopt EBCJ</li> <li>• SORT の場合 -v</li> </ul>	<p>JIS8 の入力データを、EBCDIK コードの順番に正規化して並び替えます。JIS8 でソートすると、次のとおり並び替えます。</p> <ul style="list-style-type: none"> <li>• 記号</li> <li>• 数字</li> <li>• 英大文字</li> <li>• 英小文字</li> <li>• 半角カタカナ</li> </ul> <p>このオプションを指定すると、次のとおり EBCDIK の順番に並び替えます。</p> <ul style="list-style-type: none"> <li>• 記号</li> <li>• 英小文字</li> <li>• 半角カタカナ</li> <li>• 英大文字</li> <li>• 数字</li> </ul> <p>(例) 数字「1」と英大文字「A」を比較する場合 オプションの指定あり： 「1」 &gt; 「A」 オプションの指定なし： 「1」 &lt; 「A」</p>
EBCDIK	<ul style="list-style-type: none"> <li>• SORT EE の場合 -cmdopt EBCE</li> <li>• SORT の場合 -v2</li> </ul>	<p>JIS8 で記述した文字条件値「C'～」を EBCDIK コードに正規化し、入力データと比較します。</p> <p>例えば、文字条件値「C'A」を JIS8 で記述すると、JIS8 コードで「0x41」と表現されます。このオプションを指定すると、EBCDIK コードで「0xA1」に正規化され、EBCDIK の入力データと比較されます。</p>

JIS8 単位コードと EBCDIK コードの対応表は、「付録 A JIS8 単位コード体系と EBCDIK コード体系の変換表」を参照してください。



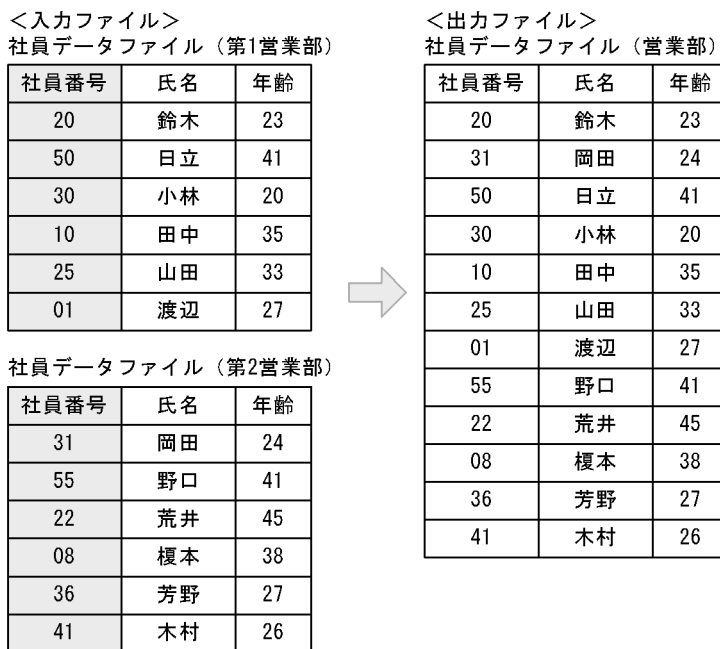
## 2.3 強制マージ機能

マージ機能を使用する場合、ソートされていないファイルを併合することを**強制マージ**といいます。強制マージでは、併合するファイルがソートされていなくても強制的に併合しますが、併合したデータの並びは保証しません。

製品	オプション名	備考
SORT EE	強制マージオプション (FMRG オプション)	5.5.6
SORT	/k オプション (Windows)	8.1.2(1)
	-k オプション (UNIX)	8.2.2(1)

強制マージの例を次の図に示します。

図 2-2 強制マージの例



(凡例)  : マージのキーを示す。

注 入カファイルは、昇順でソートされていない。


## 2.4 無条件レコード抽出機能

集約機能を使用する場合、同じキーを持つ連続したレコードがあるときは、最初に配置されているレコードだけを抽出できます。これを**無条件レコード抽出機能**といいます。この機能を使用するとき、集約項目は指定できません。無条件レコード抽出機能を使用した例を次の図に示します。

製品	コマンド・パラメタ名	備考
SORT EE	-sum パラメタ	5.7
SORT	wnsortsum コマンド /a (Windows)	10.1.2(8)
	wnsortex コマンド sum_field=(Windows)	11.2.2(14)
	rsum コマンド -a (UNIX)	10.2.2(8)
	rsortsum シェルスクリプト -a (UNIX)	12.2.12

図 2-3 無条件レコード抽出機能の使用例

支店コード	担当者名	売上数量
03	田中	2
03	田中	7
04	渡辺	5
05	日立	8
06	山田	6
06	山田	10



支店コード	担当者名	売上数量
03	田中	2
04	渡辺	5
05	日立	8
06	山田	6

(凡例)  : 支店コードをキーとした場合に抽出されるレコードを示す。

## 2.5 10進数オーバーフロー判定機能

集約機能を使用する場合、集約項目の属性が2進数の場合、10進数の桁数でオーバーフロー判定をすることを指定します。

製品	オプション名	備考
SORT EE	10進数オーバーフロー判定オプション (OVFLW オプション)	5.5.7
SORT	/h オプション (Windows)	10.1.2(1) 11.2.2(3)
	-b オプション (UNIX)	10.2.2(1)

集約結果の範囲を次の表に示します。

表 2-1 集約結果の範囲 (固定小数点2進数 (n))

バイト長	10進数オーバーフロー判定機能	
	指定なし	指定あり
2	- 32,768 ~ + 32,767	- 9,999 ~ + 9,999
4	- 2,147,483,648 ~ + 2,147,483,647	- 999,999,999 ~ + 999,999,999
8	- 9,223,372,036,854,775,808 ~ + 9,223,372,036,854,775,807	- 999,999,999,999,999,999 ~ + 999,999,999,999,999,999

表 2-2 集約結果の範囲 (符号なし2進数 (m))

バイト長	10進数オーバーフロー判定機能	
	指定なし	指定あり
2	0 ~ 32,767	0 ~ 9,999
4	0 ~ 2,147,483,647	0 ~ 999,999,999
8	0 ~ 9,223,372,036,854,775,807	0 ~ 999,999,999,999,999,999

## 2.6 エンディアンオプション機能

2進数および内部浮動小数点のキー、比較項目、または集約項目のバイトオーダーを指定できます。

バイトオーダーとは、データの格納順序です。上位バイトを下位番地の数値として扱うバイトオーダーを **big endian** (ビッグエンディアン) といい、上位バイトを上位番地の数値として扱うバイトオーダーを **little endian** (リトルエンディアン) といいます。

**SORT EE** では、可変長順編成ファイルの各レコードの先頭 (レコードヘッダ) に 2進数で格納されているレコード長についても、指定されたバイトオーダーで扱うことができます。

製品	オプション名	備考
SORT EE	入力ファイルのエンディアンオプション (BIG   INPBIG   INPLTL オプション)	5.5.4
	出力ファイルのエンディアン変更オプション (OUTBIG   OUTLTL オプション)	5.5.5
SORT	/b オプション (Windows)	7.1.2(1) 8.1.2(1) 9.1.2(1) 10.1.2(1) 11.2.2(3)
	-B オプション (UNIX)	7.2.2(1) 8.2.2(1) 9.2.2(1) 10.2.2(1)

## 2.7 ファイル排他機能（UNIX の場合）

### 2.7.1 SORT のファイル排他機能

入力ファイルを排他モードでオープンできます。排他モードでオープンする場合は、`-e` オプションを指定します。オプションの指定方法については、「7.2.2(1) ソートオプション」、「8.2.2(1) マージオプション」、「9.2.2(1) 選択オプション」、または「10.2.2(1) 集約オプション」を参照してください。

排他モードでファイルをオープンすると、ほかの処理ではそのファイルをオープンできなくなります。ただし、ファイル排他機能が有効になるのは、次に示すプログラムの処理でファイルをオープンした場合だけです。

- SORT EE
- SORT
- COBOL85 または COBOL2002
- ISAM

なお、入力ファイルが索引ファイル（COBOL85、COBOL2002 または ISAM で作成したファイル）の場合は、ファイル排他機能を使った方が処理時間が短くなります。

ファイル排他機能を使う場合は、次の点に注意してください。

- 読み出しだけをする場合にも、書き込みの許可が必要です。したがって、書き込みが許可されていないファイルは、排他モードでオープンできません。
- すでにほかの処理でオープンされているファイルは、排他モードでオープンできません。

ファイルの状態とオープンの可否の関係を次の表に示します。

表 2-3 ファイルの状態とオープンの可否

SORT でのオープンモード	ファイルの状態	
	共用	排他
共用	○	×
排他	×	×

(凡例)

- : オープンできる
- × : オープンできない

注

「ファイルの状態」とは、ほかの処理でファイルがどのモードでオープンされているかを示します。なお、出力ファイルは常に排他モードでオープンします。

「共用」とは、一つのファイルを複数の処理で使用できることを示します。

## 2. 便利な機能

「排他」とは、ある処理で使用しているファイルをほかの処理で使用できないことを示します。

### 2.7.2 SORT EE のファイル排他機能

入力ファイルおよび出力ファイルを「共用」、「共用・排他無し」、および「排他」のどれかのモードでオープンできます。コマンドオプションの指定方法については、「5.5.14 入出力ファイル共用・排他オプション (NOLOCK | EXC オプション)」を参照してください。

排他モードでファイルをオープンすると、ほかの処理ではそのファイルをオープンできなくなります。ファイル排他機能が有効になるのは、次に示すプログラムの処理でファイルをオープンした場合だけです。

- SORT EE
- SORT
- COBOL85 または COBOL2002
- ISAM

ファイル排他機能を使う場合は、次の点に注意してください。

- 「共用・排他なし」を指定した場合、SORT EE、SORT、COBOL、および ISAM 間でのファイル排他機能が無効となります。4 製品で同一ファイルを指定して、同時実行した場合、入出力ファイルの内容は保証できません。
- 読み出しだけをする場合にも、書き込みの許可が必要です。したがって、書き込みが許可されていないファイルは、排他モードでオープンできません。
- すでにほかの処理でオープンされているファイルは、排他モードでオープンできません。

入力ファイルの状態とオープンの可否の関係を次の表に示します。

表 2-4 入力ファイルの状態とオープンの可否

SORT EE でのオープンモード	入力ファイルの状態	
	共用	排他
共用・排他なし	○	○
共用	○	×
排他	×	×

表 2-5 出力ファイルの状態とオープンの可否

SORT EE でのオープンモード	出力ファイルの状態	
	共用	排他
共用・排他なし	○	○
排他	×	×

(凡例)

- : オープンできる
- × : オープンできない

注

「ファイルの状態」とは、ほかの処理でファイルがどのモードでオープンされているかを示します。

「共用」とは、一つのファイルを複数の処理で使用できることを示します。

「排他」とは、ある処理で使用しているファイルをほかの処理で使用できないことを示します。

## 2.8 キャッシュファイルの同期出力機能

キャッシュファイルの同期出力機能とは、出力データがキャッシュファイルに残っている場合にすべてのデータをファイルに出力したあと、コマンドを終了する機能です。

OS による使用可否を次に示します。

OS	SORT	SORT EE
UNIX	○	○
Windows	×	○

(凡例)

- : 使用できる
- × : 使用できない

製品	オプション名	備考
SORT EE	キャッシュファイルの同期出力オプション (SYNC オプション)	5.5.15
SORT	-q オプション (UNIX)	7.2.2(1) 8.2.2(1) 9.2.2(1) 10.2.2(1)



## 2.9 内部 10 進数符号保障機能

集約機能を使用する場合、集約項目の属性が内部 10 進数について、最初の集約項目の演算符号が  $(F)_{16}$  (絶対値) であるとき、以降の集約項目の集約結果に対して同じ演算符号 (絶対値) を適用します。

最初の集約項目の演算符号が  $(C)_{16}$  (正) である場合は、以降の集約項目に対して  $(C)_{16}$  (正) を適用します。また、このオプションを指定しない場合は、演算符号が  $(F)_{16}$  (絶対値) である集約項目に対して  $(C)_{16}$  (正) を適用します。

最初の集約項目の符号	内部 10 進数符号保障機能	
	指定あり	指定なし
F	F	C
C	C	C

製品	オプション名	備考
SORT EE	内部 10 進数符号保障オプション (ABS オプション)	5.5.8
SORT	/if オプション (Windows)	10.1.2(1)
	-if オプション (UNIX)	10.2.2(1)

## 2.10 10進キーのキーチェック緩和機能 (SORT EE の機能)

ソート機能，マージ機能，選択機能，または集約機能を使用する場合，キーまたは比較項目の属性が外部 10 進数（CSV ファイルの場合は固定小数点数）または内部 10 進数のとき，入力データの形式が不正であってもエラーにしません。コマンドオプションの指定方法については，「5.5.9 10 進キーのチェックオプション (KEYNOCHK オプション)」を参照してください。

10 進キーのキーチェック緩和機能	不正データの扱い※	
指定なし	exsort コマンドが異常終了する。	
指定あり	内部 10 進数	<ul style="list-style-type: none"> <li>• 数値部が 0～9 以外の場合，JIS8 単位コード体系に従って比較する。</li> <li>• 符号部が (C)<sub>16</sub> (正)，(D)<sub>16</sub> (負)，(F)<sub>16</sub> (絶対値) 以外の場合は，内部 10 進数を正値として扱う。</li> </ul>
	外部 10 進数	<ul style="list-style-type: none"> <li>• ゾーン部は，比較対象外。</li> <li>• 数値部が 0～9 以外の場合，JIS8 単位コード体系に従って比較する。</li> <li>• 符号部が 3，7 以外の場合は，外部 10 進数を正値として扱う。</li> </ul>
	固定小数点数 (CSV 数値データ)	<ul style="list-style-type: none"> <li>• データ中に引用符，符号，通貨記号，空白，小数点，コンマが含まれている場合は，これらの文字を除外し，残った文字を外部 10 進数として扱う。</li> <li>• ゾーン部に相当する部分は比較対象外。</li> <li>• 符号部に相当する部分は無視され，常に正値として扱う。</li> <li>• 数値部に相当する部分が 0～9 以外の場合，JIS8 単位コード体系に従って比較する。</li> </ul>

### 注※

不正データと見なされる条件を次の表に示します。

ファイル編成	不正データの条件
固定長順編成ファイル 可変長順編成ファイル テキストファイル	<p>キーまたは比較項目のデータ形式が，次のどれかの場合は不正データと見なされます。</p> <ul style="list-style-type: none"> <li>• 数値部が 0～9 以外の値。</li> <li>• 符号部が不正な場合 内部 10 進数の場合は，(C)<sub>16</sub> (正)，(D)<sub>16</sub> (負)，(F)<sub>16</sub> (絶対値) 以外の値。 外部 10 進数の場合は，3 (正)，7 (負) 以外の値。</li> <li>• 外部 10 進数の場合，ゾーン部が 3 以外の値。</li> </ul>

ファイル編成	不正データの条件
CSV ファイル	<p>キーまたは比較項目のデータ形式が、次のどちらかの場合は不正データと見なされます。</p> <ul style="list-style-type: none"><li>引用符、数字、符号、通貨記号、空白、小数点、コンマ以外の文字を含む場合。</li><li>データ中に含まれる引用符、符号、通貨記号、空白、小数点、コンマの位置が不正な場合。除外文字の位置規則については「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。</li></ul>

## 2.11 10進集約項目のチェック緩和機能 (SORT EE の機能)

集約機能を使用する場合、集約項目の属性が外部 10 進数 (CSV ファイルの場合は固定小数点数) または内部 10 進数のとき、入力データの形式が不正であってもエラーにしません。コマンドオプションの指定方法については、「5.5.10 10 進集約項目のチェックオプション (SUMNOCHK オプション)」を参照してください。

10 進集約項目の チェック緩和機能	不正データの扱い※	
指定なし	exsort コマンドが異常終了する。	
指定あり	集約キーに該当するレコード が 1 件の場合	該当するレコードを出力ファイルに書き、 exsort コマンド処理を続行する。
	集約キーに該当するレコード が 2 件以上の場合	exsort コマンドが異常終了する。

### 注※

不正データと見なされる条件を次の表に示します。

ファイル編成	不正データの条件
固定長順編成ファイル 可変長順編成ファイル テキストファイル	集約項目のデータ形式が、次のどれかの場合は不正データと見なされます。 <ul style="list-style-type: none"> <li>• 数値部が 0 ~ 9 以外の値</li> <li>• 符号部が不正な場合 内部 10 進数の場合は、(C)<sub>16</sub> (正), (D)<sub>16</sub> (負), (F)<sub>16</sub> (絶対値) 以外の値 外部 10 進数の場合は、3 (正), 7 (負) 以外の値</li> <li>• 外部 10 進数の場合、ゾーン部が 3 以外の値</li> </ul>
CSV ファイル	集約項目のデータ形式が、次のどちらかの場合は不正データと見なされます。 <ul style="list-style-type: none"> <li>• 引用符、数字、符号、通貨記号、空白、小数点、およびコンマ以外の文字を含む場合</li> <li>• データ中に含まれる引用符、符号、通貨記号、空白、小数点、およびコンマの位置が不正な場合 除外文字の位置規則については「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。</li> </ul>

## 2.12 キー位置のチェック緩和機能（SORT EE の機能）

ソート機能、マージ機能、または集約機能を使用する場合、可変長順編成ファイル、およびテキストファイルのキーの指定範囲が実レコード長を超えていてもエラーにしません。コマンドオプションの指定方法については、「5.5.11 キー位置のチェックオプション（POSNOCHK オプション）」を参照してください。

実レコード長を超えたキー部分には、NULL 値を仮定してキーを比較します。

キー位置のチェック緩和機能	キーの扱い	
指定なし	キーの範囲が実レコード長を超えている場合、 <code>exsort</code> コマンドが異常終了する。	
指定あり※	1. キー位置+キー長が実レコード長以下の場合	実レコード内のキーが、ソートキー、マージキー、または集約キーとなる。
	2. キー位置が実レコード長より小さく、キー位置+キー長が実レコード長より大きく最大レコード長以下の場合	実レコード内のキーと、実レコードを超えている部分に NULL 値を仮定したキーがソートキー、マージキー、または集約キーとなる。
	3. キー位置が実レコード長より小さく、キー位置+キー長が最大レコード長より大きい場合	
	4. キー位置が実レコード長より大きく、キー位置+キー長が最大レコード長以下の場合	NULL 値を仮定したキーが、ソートキー、マージキー、または集約キーとなる。
	5. キー位置が最大レコード長より小さく、キー位置+キー長が最大レコード長より大きい場合	
	6. キー位置が最大レコード長より大きい場合	

(凡例)

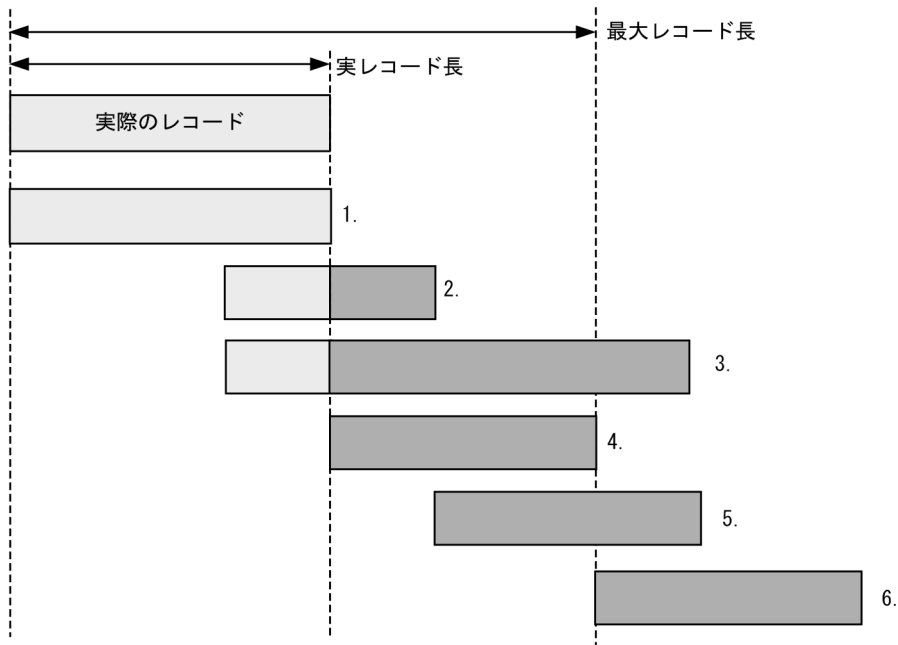
レコード長の種類	可変長順編成ファイル	テキストファイル
最小レコード長	COBOL でファイル作成時にユーザが指定した最小レコード長	1
実レコード長	実際の入力レコード長	実際の入力レコード長（改行文字を除く）
最大レコード長	COBOL でファイル作成時にユーザが指定した最大レコード長	<code>-inpfile</code> 、 <code>-splitinp</code> 、または <code>-catinp</code> パラメタの <code>reclen</code> オペランドで指定したレコード長（改行文字を除く）

## 2. 便利な機能

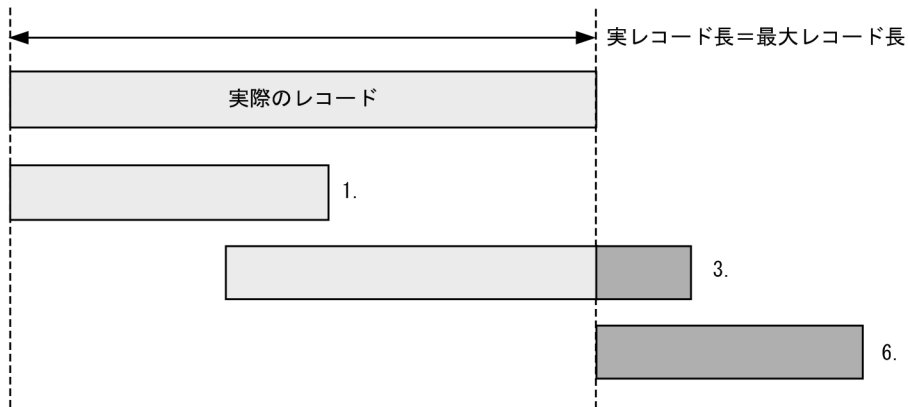
注※

1. ～ 6. は、次の図の 1. ～ 6. に対応しています。

### ●実レコード長<最大レコード長の場合



### ●実レコード長=最大レコード長の場合



(凡例)

□ : 実レコード内の指定された範囲をキーとする。

■ : NULL値(0x00)をキーとする。

## 2.13 日付データのチェック緩和機能（SORT EE の機能）

西暦下 2 桁年号に空白 (X'2020'), Low\_value (X'0000'), High\_value (X'FFFF') が設定されていても、エラーとしないで文字として処理されます。

コマンドオプションの指定方法については、「5.5.12 日付データのチェックオプション (DATENOCK オプション)」を参照してください。

CSV ファイルの場合、西暦下 2 桁年号または西暦 4 桁年号に数字以外の文字が設定されていても、エラーとしないで文字として処理されます。

日付データのチェック緩和機能	不正データの扱い※
指定なし	コマンドが異常終了する。
指定あり	JIS8 単位コード体系でソートキーの大小を比較する（西暦下 2 桁年号であっても基準年を適用しないで、文字として処理されます）。

### 注※

不正データと見なされる条件を次の表に示します。

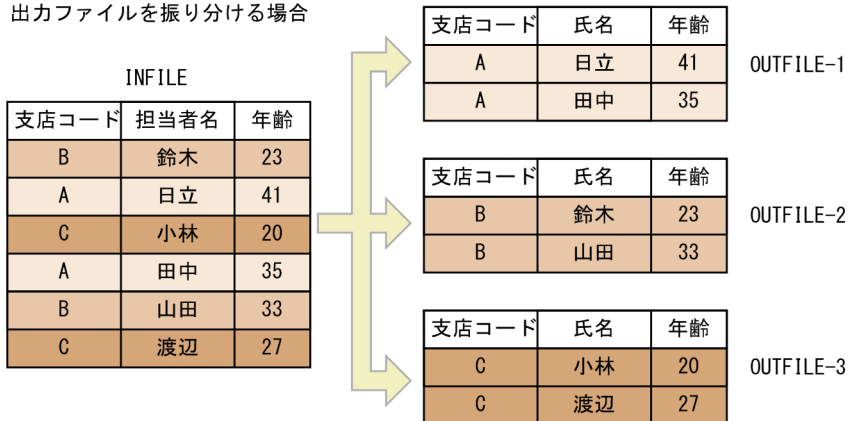
ファイル編成	不正データの条件
可変長順編成ファイル 固定長順編成ファイル テキストファイル	キーの先頭 2 文字（西暦下 2 桁の数値）が指すデータ内容が次のどれかを満たす場合は不正データと見なされます。 <ul style="list-style-type: none"> <li>ゾーン部が 3 以外の値</li> <li>数値部が 0～9 以外の値</li> </ul>
CSV ファイル	キーの年号（西暦下 2 桁年号または西暦 4 桁年号）部分のデータ内容が、0～9 の数字 (X'30'～X'39') 以外の文字の場合は、不正データと見なされます。

## 2.14 複数ファイル出力機能（SORT EE の機能）

選択したレコードを指定した複数のファイルへ振り分けて出力する機能です。複数ファイル出力機能を使用した例を次に示します。

図 2-4 複数ファイル出力機能の例

支店コードを選択キーにして、  
出力ファイルを振り分ける場合



exsortコマンド指定

```
exsort
-function copy
-inpfile fileorg=V file=INFILE
-outfile file=OUTFILE-1 include="c+0-1 = C'A'"
-outfile file=OUTFILE-2 include="c+0-1 = C'B'"
-outfile file=OUTFILE-3 include="c+0-1 = C'C'"
```

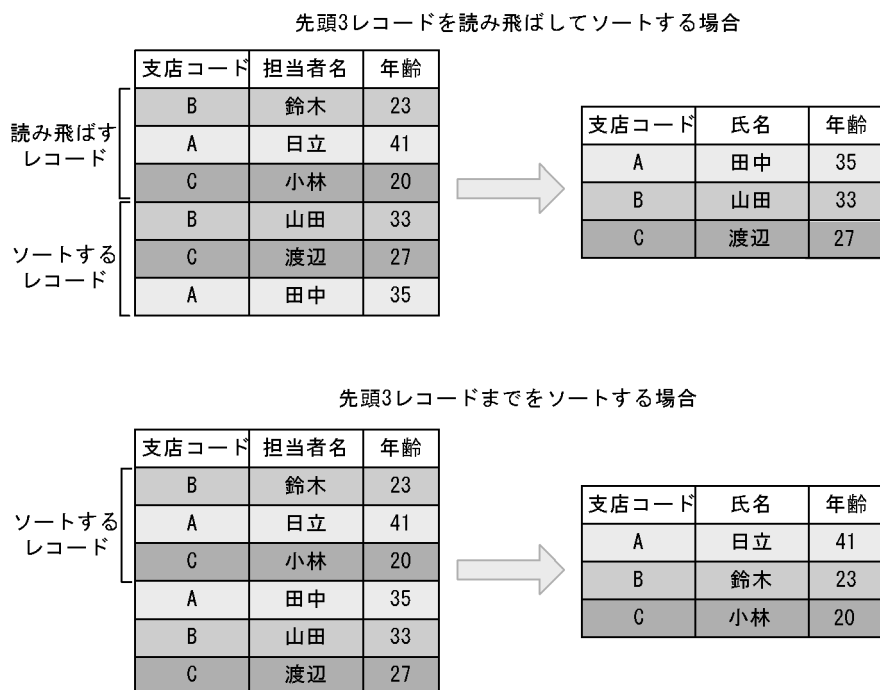


## 2.15 入力レコードの読み飛ばし・打ち切り機能 (SORT EE の機能)

ソート機能またはコピー機能を使用する場合、入力レコードの対象レコードの範囲を指定できる機能です。入力レコードの読み飛ばし機能については「5.8.4 skiprec オペランド」を、入力レコードの打ち切り機能については「5.8.5 stopaft オペランド」を参照してください。

入力レコードの読み飛ばし・打ち切り機能を使用した例を次に示します。

図 2-5 入力レコードの読み飛ばし・打ち切り機能の例

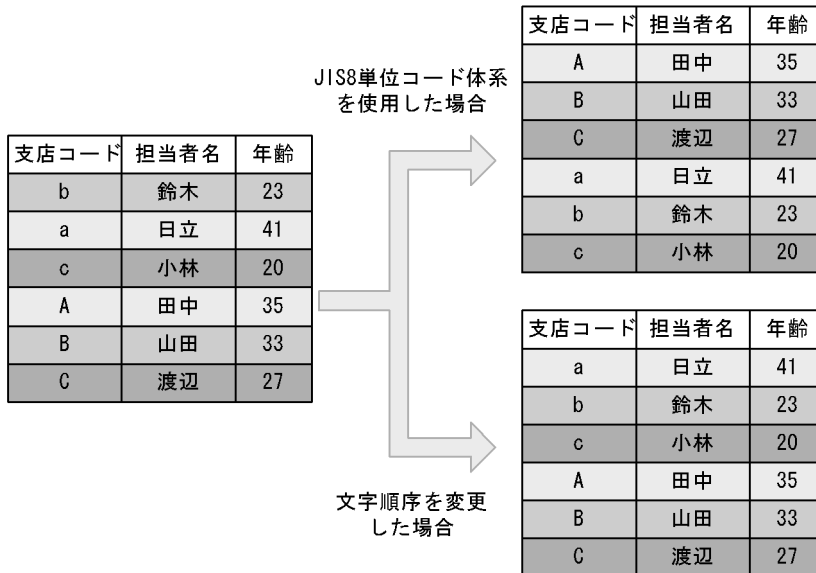


## 2.16 文字順序変更機能（SORT EE の機能）

JIS8 単位コード体系以外の順序でキーを比較できる機能です。文字順序変更機能の詳細については、「5.15 -altseq パラメタ」を参照してください。

文字順序変更機能の概要を次の図に示します。

図 2-6 文字順序変更機能の例



## 2.17 既存ファイルの書き換え防止機能 (SORT の機能)

---

出力先として指定した出力ファイル名と同じ名前のファイルがすでにあるとき、そのファイルの内容が書き換えられるのを防ぐことができます。この機能を使った場合、出力ファイル名と同じ名前のファイルがあると、ソート結果を出力しないで処理が終了し、メッセージが表示されます。

オプション名	備考
/i オプション (Windows)	7.1.2(1) 8.1.2(1) 9.1.2(1) 10.1.2(1) 11.2.2(3)
-i オプション (UNIX)	7.2.2(1) 8.2.2(1) 9.2.2(1) 10.2.2(1)

## 2.18 I/O サイズ変更機能（SORT EE の機能）

---

ユーザが、入力ファイル、出力ファイル、ソート用一時ファイル、マージ用一時ファイル、およびファイル分割用一時ファイルに対する 1 回の I/O サイズを指定できます。詳細は、「5.19 `-iosize` パラメタ」を参照してください。

## 2.19 ワークバッファサイズ自動見積もり機能 (SORT EE の機能)

---

製品がソート機能で使用するワークバッファのサイズを自動で見積もる機能です。

ワークバッファサイズの指定がない場合に、製品がアウトコアソート、またはインコアソートのワークバッファサイズを見積もって割り当てます。また、64 ビット版の SORT EE では、ユーザが 2GB を超えるワークバッファサイズを指定できます。詳細は、「5.13 -worksize パラメタ」を参照してください。

## 2.20 2GB 超ワークバッファサイズの指定（64ビット版 SORT EE の機能）

---

64ビット版 SORT EE では、2GB を超えるワークバッファサイズを指定できます。指定できるサイズは、1MB ～ 2,048,000MB（2,000GB）までです。

## 2.21 SORT の機能拡張オプション（SORT の機能）

---

ユーザが SORT EE 固有の機能を SORT で使用できるようになるオプションです。使用できる SORT EE の機能は次の 8 機能です。

- ファイル排他機能の無効化（UNIX の場合）  
ファイル排他機能の無効化（UNIX の場合）については、「2.7.2 SORT EE のファイル排他機能」を参照してください。
- 10 進キーのキーチェック緩和機能  
10 進キーのキーチェック緩和機能については、「2.10 10 進キーのキーチェック緩和機能（SORT EE の機能）」を参照してください。
- 10 進集約項目のチェック緩和機能  
10 進集約項目のチェック緩和機能については、「2.11 10 進集約項目のチェック緩和機能（SORT EE の機能）」を参照してください。
- キー位置のチェック緩和機能  
キー位置のチェック緩和機能については、「2.12 キー位置のチェック緩和機能（SORT EE の機能）」を参照してください。
- 日付データのチェック緩和機能  
日付データのチェック緩和機能については、「2.13 日付データのチェック緩和機能（SORT EE の機能）」を参照してください。
- I/O サイズ変更機能  
I/O サイズ変更機能については、「2.18 I/O サイズ変更機能（SORT EE の機能）」を参照してください。
- ワークバッファサイズの自動見積もり機能  
ワークバッファサイズの自動見積もり機能については、「2.19 ワークバッファサイズ自動見積もり機能（SORT EE の機能）」を参照してください。
- 2GB 超ワークバッファサイズの指定（64 ビット版 SORT EE の機能）  
2GB 超ワークバッファサイズの指定（64 ビット版 SORT EE の機能）については、「2.20 2GB 超ワークバッファサイズの指定（64 ビット版 SORT EE の機能）」を参照してください。

なお、上記にある機能を使用する上での注意事項を次に示します。

- ファイル編成が固定長順、可変長順、およびテキストファイルだけ適用できます。
- 前提 PP として、SORT EE が必要となります。
- メモリ所要量が変更となります。
- 完了メッセージが SORT EE のメッセージに変更となります。

## 2. 便利な機能

- コマンドログファイルが出力されるようになります。
- デバッグ情報ファイルの出力形態が変更となります。

SORT の機能拡張オプションの使い方については、「18. SORT の機能拡張オプションの使い方」を参照してください。

SORT と SORT EE の両方に指定できる機能の場合、優先順位は次のとおりです。

1. SORT コマンドの指定
2. UNIX の場合 : SORT の環境変数  
Windows の場合 : SORT の環境ファイル
3. SORT EE の環境変数および環境ファイル

SORT と SORT EE の両方に指定できる機能については、「14.1 SORT EE の環境変数の設定」の表 14-1 または表 14-2 を参照してください。



## 2.22 強制ファイル分割機能（SORT EE の機能）

---

分割キーによるファイル分割時、分割する入力ファイルが昇順または降順にソートされていなくても、エラーとならずにファイルを分割できます。詳細については、「5.5.16 強制ファイル分割機能（FSPLIT オプション）」を参照してください。

## 2.23 ファイル逐次 OPEN/CLOSE 機能 (SORT EE の機能)

---

ファイル分割時の出力ファイルおよびファイル連結時の入力ファイルに対するオープンとクローズを一括で行わず、入出力するファイルだけで行うことを指定します。この機能によって、ファイル分割機能およびファイル連結機能で、扱うファイルの増加に伴うシステムのファイルディスクリプタの制限によるエラーを回避できます。ただし、この機能はソート機能とコピー機能使用時だけに有効です。マージ機能使用時は有効になりません。詳細については、「5.5.17 ファイル逐次 OPEN/CLOSE 機能 (SEQOPNCLS オプション)」を参照してください。

## 2.24 メッセージ抑止機能（SORT EE の機能）

---

ファイル分割機能およびファイル連結機能で出力されるメッセージを抑止します。詳細については、「14.1.1(19) EXSORTSUPPRESSSPLITMSG」、「14.1.1(20) EXSORTSUPPRESSCATMSG」、「14.1.2(19) EXSORT64SUPPRESSSPLITMSG」、および「14.1.2(20) EXSORT64SUPPRESSCATMSG」を参照してください。

## 2.25 0件ファイル出力抑止機能（SORT EE の機能）

---

ファイル分割後の出力ファイルに格納するレコード件数が0件の場合、出力ファイルを作成しません。詳細については、「14.1.1(21) EXSORTNOCREATEZEROFILE」および「14.1.2(21) EXSORT64NOCREATEZEROFILE」を参照してください。

## 2.26 キー・項目の上限値拡張機能（SORT EEの機能）

キー、集約項目、および比較項目の上限値を次に示す値まで拡張できます。

- 最大個数：1,280 個
- 最大合計長：262,144 バイト

入力データのエンコード種別によって、最大合計長は変わります。入力データのエンコード種別と最大合計長の関係を次に示します。

入力データのエンコード種別※	キー・項目の最大合計長 (単位：バイト)
ASCII, SJIS, EUC-JP, UTF-8, EUC-HJ	65,536
UCS-2LE, UCS-2BE, UTF-16LE, UTF-16BE	131,072
UCS-4LE, UCS-4BE, UTF-32LE, UTF-32BE	262,144

注※

入力データのエンコードの種別は、`codetype` オペランドで指定できます。

`codetype` オペランドについては、「5.8.13 `codetype` オペランド」を参照してください。

最大個数と合計長の上限値を拡張できる項目と指定方法を次の表に示します。

最大個数と合計長の 上限値を拡張できる 項目	指定方法	
	パラメタ・オペランド	環境ファイルの指定
キー	<code>-key</code> パラメタ	<code>EXS_KEYNUMMAX</code>
集約項目	<code>-sum</code> パラメタ	<code>EXS_SUMNUMMAX</code>
比較項目	<code>include/omit</code> オペランド	<code>EXS_SELNUMMAX</code>
CSV キー	<code>-csvkey</code> パラメタ	<code>EXS_CSVKEYNUMMAX</code>
CSV 集約項目	<code>-csvsum</code> パラメタ	<code>EXS_CSVSUMNUMMAX</code>
CSV 比較項目	<code>csvinclude/csvomit</code> オペランド	<code>EXS_CSVSELNUMMAX</code>

## 2.27 メッセージ出力オプション (SORT EE の機能)

---

`exsort` コマンドのメッセージの出力有無, および出力先を環境変数で指定できます。

出力先は次のファイルから選択できます。

- 標準エラーファイル
- 標準出力ファイル
- 環境変数「`EXSORTMSGFILE` (64 ビット版は「`EXSORT64MSGFILE`)」, または環境ファイルの「`msg_file`」オプションで指定したファイル

メッセージ出力オプションの詳細については, 「14.1.1(5)EXSORTMSGOPT」, または「14.1.2(5)EXSORT64MSGOPT」を参照してください。

## 2.28 CSV ファイル関連の便利な機能（SORT EE の機能）

---

CSV ファイルを指定する処理の場合、次の便利な機能を使用できます。

- 入力開始レコード番号の指定
- 連続区切り文字オプション
- ナル文字列の集約オプション（SUMNULKEY, SUMNONULL オプション）
- 出力ファイルのデリミタ変更
- 出力ファイルの引用符変更
- 出力ファイルのセル数調整

### 2.28.1 入力開始レコード番号の指定

データの入力を開始するレコード番号を指定できます。

指定したレコード番号よりも前のレコードは、ソート処理、マージ処理、またはコピー処理の対象になりません。

入力開始レコード番号の指定の詳細については、「5.8.6 startrec オペランド」を参照してください。

### 2.28.2 連続区切り文字オプション

連続した区切り文字（ConsecutiveDelimiter）を1つのデリミタ（区切り文字）として扱います。

連続区切り文字オプションの詳細については、「5.5.18 連続区切り文字オプション（INPCONSEPA, OUTCONSEPA オプション）」を参照してください。

### 2.28.3 ナル文字列の集約オプション（SUMNULKEY, SUMNONULL オプション）

ナル文字列（空のセル）の集約キーおよび集約項目の扱いを指定するオプションです。

ナル文字列の集約キーが複数存在する場合、それらを同値のキーとして扱うか、個別に扱い集約しないかを指定します。また、ナル文字列の集約項目が複数存在する場合、それらを集約するか、個別に扱い集約しないかを指定します。

ナル文字列の集約オプションの詳細については、「5.5.19 ナル文字列の集約オプション（SUMNULKEY, SUMNONULL オプション）」を参照してください。

## 2.28.4 出力ファイルのデリミタ変更

入力レコード中のデリミタを、指定したデリミタに変更して出力できます。

出力ファイルのデリミタ変更の詳細については、「5.10.6 cellsepa オペランド」を参照してください。

## 2.28.5 出力ファイルの引用符変更

入力レコード中の引用符を、指定した引用符に変更して出力できます。

出力ファイルの引用符変更の詳細については、「5.10.8 quottype オペランド」を参照してください。

## 2.28.6 出力ファイルのセル数調整

すべての出力レコードのセル数を一致させて出力できます。

出力ファイルのセル数調整の詳細については、「5.10.7 cellnum オペランド」を参照してください。



## 2.29 集約結果オーバーフロー検知オプション (SORT EE の機能)

---

集約機能を使用する場合、集約項目の合計でオーバーフローが発生したときは、KBL51412-E エラーで `exsort` コマンドを異常終了させます。

集約結果オーバーフロー検知オプションの詳細については、「5.5.20 集約結果オーバーフロー検知オプション (OVFERR オプション)」を参照してください。

## 2.30 改行文字の自動判定機能（SORT EE の機能）

---

テキストファイルおよび CSV ファイルの入力レコードごとの改行文字（LF または CRLF）を自動的に判定する機能です。入力ファイルの改行文字の種別が不明な場合、または LF と CRLF の改行文字が混在している入力ファイルの場合に有効な機能です。

入力ファイルの改行文字の指定については、「5.8.7 recsepa オペランド」を参照してください。

また、出力ファイルの改行文字を、LF または CRLF のどちらかに変更できます。出力ファイルの改行文字の変更については、「5.10.5 recsepa オペランド」を参照してください。

## 2.31 シフト JIS コード比較オプション (SORT EE の機能)

---

シフト JIS コードの体系に従って、キーおよび比較項目を文字単位で比較します。

キーおよび比較項目にシングルバイト文字（英数字，半角カナ），およびマルチバイト文字（漢字などの全角文字）が混在している場合，シングルバイト文字はマルチバイト文字より先に比較します。コマンドオプションの指定方法については，「5.5.21 シフト JIS コード比較オプション (SJIS オプション)」を参照してください。

## 2.32 ゼロデータ比較オプション (SORT EE の機能)

---

SORT EE では、左分離符号付き数字「d」のキーおよび比較項目において、負符号付きゼロ (-0)、正符号付きゼロ (+0)、符号なしゼロ (0) を、すべて同値 (ゼロ値) として扱います。

このオプションを指定すると、負符号付きゼロ (-0) を、正符号付きゼロ (+0) および符号なしゼロ (0) よりも小さい値として扱います。コマンドオプションの指定方法については、「5.5.22 ゼロデータ比較オプション (DIFFZERO オプション)」を参照してください。

## 2.33 空行削除オプション（SORT EE の機能）

---

入力ファイルがテキストファイルの場合、改行だけのデータがないレコード（空行，空レコード）を入力できません（KBL1636-E エラーとなります）。

入力ファイルが CSV ファイルの場合，空行を入力することはできますが，ソート対象外のレコードとして先頭または末尾に配置されます（配置位置は，`nullseq` オペランドで指定できます）。`nullseq` オペランドの詳細は「5.8.12 `nullseq` オペランド」を参照してください。

このオプションを指定すると，空行を削除して入力します。テキストファイルの場合はエラーになりません。また，出力ファイルには空行は出力されません。

コマンドオプションの指定方法については，「5.5.23 空行削除オプション（`DELBLANKLINE` オプション）」を参照してください。

## 2.34 ファイルの上書きオプション（SORT EEの機能）

---

### 2.34.1 既存ファイルの上書きオプション

出力先として指定した出力ファイル名と同じ名前のファイルがすでにあるとき、そのファイルの内容を上書き出力することを許可するオプションです。

コマンドオプションの指定方法については、「5.5.2 ファイルの上書きオプション (OVWRT | INPOVWRT オプション)」を参照してください。

### 2.34.2 入力ファイルの上書きオプション

ソート機能では、入力ファイルと出力ファイルに同じファイルを指定できます。ただし、選択機能や集約機能と一緒に指定した場合は、レコード件数が変わることで、入力データを保持できなくなるため、この場合は入力と出力に同じファイルを指定することはできません。レコード編集機能も同様に、入力データを保持できないという理由から、同じファイルを指定できません。

このオプションを指定すると、レコード処理機能（選択、集約、または編集機能）を指定していても、入力と出力に同じファイルを指定できます（入力ファイルを出力ファイルで上書きします）。

コマンドオプションの指定方法については、「5.5.2 ファイルの上書きオプション (OVWRT | INPOVWRT オプション)」を参照してください。

## 2.35 Unicode シグニチャ (BOM) 関連オプション (SORT EE の機能)

### 2.35.1 Unicode シグニチャ読み飛ばし・付加機能

テキストファイルおよび CSV ファイルを入力する際に、ファイルの先頭にある Unicode シグニチャ (以降、BOM と呼びます) を読み飛ばして入力できます。また、出力ファイルの先頭に BOM を出力することもできます。

環境変数名 (64 ビット版)	機能概要	備考
EXSORTBOMLNGIN (EXSORT64BOMLNGIN)	すべての入力ファイルの BOM を読み飛ばします。	14.1.1(15) 14.1.2(15)
EXSORTBOMLNGOUT (EXSORT64BOMLNGOUT)	この環境変数で指定した BOM を、出力ファイルに出力します。	14.1.1(16) 14.1.2(16)
EXSORTBOMGETPUT (EXSORT64BOMGETPUT)	すべての入力ファイルの BOM を読み飛ばします。 また、入力ファイルの BOM と同じ形式の BOM を、すべての出力ファイルに出力します。	14.1.1(17) 14.1.2(17)

### 2.35.2 Unicode シグニチャ自動判別機能

入力ファイルの BOM 形式を判別し、入力データのエンコードに適用する機能です。この機能を使用する場合は、`codetype` オペランドに「UNICODE」と指定します。`codetype` オペランドについては、「5.8.13 `codetype` オペランド」を参照してください。

なお、この機能を指定しても BOM の読み飛ばしと付加は行われません。BOM の読み飛ばしと付加を行う場合は、「2.35.1 Unicode シグニチャ読み飛ばし・付加機能」の指定も必要です。

## 2.36 マルチバイト文字データの比較 (SORT EE の機能)

---

入力ファイル編成がテキストファイルおよび CSV ファイルの場合は、マルチバイト文字に対応した文字比較ができます。詳細は、「付録 I マルチバイト文字データの扱い」を参照してください。

- UCS-2, UCS-4 のワイド文字対応
- UTF-16 サロゲートペア文字対応
- SJIS, EUC-JP, EUC-HJ, UTF-8, UTF-16, UTF-32 のマルチバイト文字コードに対応
- Unicode の IVS 対応



## 2.37 内部 10 進数符号オプション (SORT EE の機能)

---

SORT EE で扱う内部 10 進数の符号は、C (正)、D (負) および F (絶対値) で、これ以外の符号は不正データ形式としています。このオプションを指定すると、C、D、F 以外の符号も、正常データ形式として扱います。

オプションの指定方法については、「14.1.1(30) EXSORTPACKSIGNTYPE」および「14.1.2(30) EXSORT64PACKSIGNTYPE」を参照してください。

## 2.38 外部 10 進数データ形式オプション (SORT EE の機能)

---

SORT EE で扱う外部 10 進数の形式は、数字 0x30 ~ 0x39、正直 0x30 ~ 0x39、および負値 0x70 ~ 0x79 で、これ以外の数値は不正データ形式としています。このオプションを指定すると、EBCDIK コード形式の数値と、EBCDIK コードを ASCII コードに変換した数値も、正常データ形式として扱います。

オプションの指定方法については、「14.1.1(31) EXSORTUNPACKDATATYPE」および「14.1.2(31) EXSORT64UNPACKDATATYPE」を参照してください。

## 2.39 符号なし 2 進数のマスクビット指定 (SORT EE の機能)

---

キーデータに対して、指定したマスクビットで AND 演算を行った結果でソートできます。

属性コード「a」または「b」で、16 進形式のマスクビットを指定します。この属性は、ソートキーだけに指定できます。

キーの指定方法については、「5.6 -key パラメタ」を参照してください。

## 2.40 簡易コピー機能（SORT EE の機能）

入力ファイルを出力ファイルに複写する機能です。次の表に示すように、コピー機能では複写できないファイルも複写できます。ただし、コピー機能で可能なレコード処理機能（選択、集約、編集、ファイル分割、およびファイル連結）との組み合わせは、簡易コピー機能ではできません。

項目	コピー機能		簡易コピー機能	
	入力レコード長	最小 1 バイト	最大 65,535 バイト	最小 0 バイト
入力レコード件数	最小 0 件	最大 2,147,483,647 件	最小 0 件	最大は制限なし (システム側の制限あり)
改行文字だけの空のレコード	テキストファイルは入力できない		ファイル編成に関係なく入力できる	

機能の指定方法については、「5.28 `-cpyinp` パラメタ」および「5.29 `-cpyout` パラメタ」を参照してください。

## 2.41 -outfile パラメタ数の制限解除オプション (SORT EE の機能) 【08-50 以降】

---

指定できる `-outfile` パラメタの個数制限 (最大 64 個) を解除できます。

制限解除オプションの指定方法については、環境変数「14.1.1(33)

`EXSORTOUTFILEENUMUNLIMIT` 【08-50 以降】、「14.1.2(33)

`EXSORT64OUTFILEENUMUNLIMIT` 【08-50 以降】、または環境ファイル「15.1.1(24)

`EXS_OUTFILEENUMUNLIMIT` 【08-50 以降】」を参照してください。

指定できる `-outfile` パラメタの最小値および最大値については、「表 3-5 SORT EE のパラメタの最小値および最大値」を参照してください。

## 2.42 exsort パラメタファイルサイズの制限解除オプション（SORT EE の機能）【08-50以降】

---

exsort パラメタファイルのサイズ制限（256KB 以内）を解除できます。

制限解除オプションの指定方法については、環境変数「14.1.1(34)

EXSORTPRMSIZEUNLIMIT 【08-50 以降】」, 「14.1.2(34)

EXSORT64PRMSIZEUNLIMIT 【08-50 以降】」, または環境ファイル「15.1.1(25)

EXS\_PRMSIZEUNLIMIT 【08-50 以降】」を参照してください。

exsort パラメタファイルの形式については、「4.4.1 ファイル形式」を参照してください。

## 2.43 CSV 数値データ（固定小数点数）の形式 チェックオプション（SORT EE の機能） 【08-50 以降】

次に示す条件の場合、指定されたキーまたは比較項目範囲外の入力データの形式が不正でもエラーにはなりません。

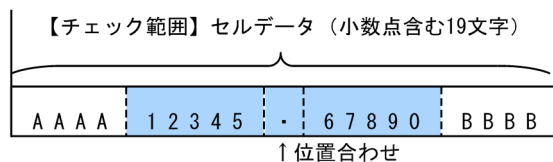
- ソート機能、マージ機能、選択機能、または集約機能を使用する場合、キーまたは比較項目の属性が CSV 数値データ（固定小数点数）である。

コマンドオプションの指定方法については、「5.5.24 CSV 数値データ（固定小数点数）の形式チェックオプション（CSVFRMCHK オプション）」を参照してください。

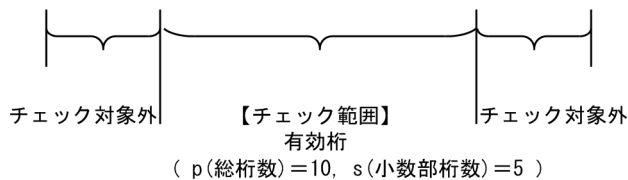
CSV のデータ形式は、「付録 C.2 CSV データ形式」および「付録 B.1 数値（固定小数点数）データの端数処理」を参照してください。

図 2-7 CSV 数値データ（固定小数点数）の形式チェックオプションの例

CSVFRMCHK 指定なし



CSVFRMCHK 指定あり



CSV の数値データ（固定小数点数、属性 z）は、次の文字を含むことができます。

- 0～9 の数字
- データを囲んでいる引用符
- 符号
- 通貨記号
- 桁修飾文字（コンマ、小数点、空白）

これらの文字以外がセル内に含まれた場合は不正データとして扱います。

CSVFRMCHK オプションの指定がない場合、指定された有効桁に関係なく、セル内の文字すべてについて「CSV 数値データ（固定小数点数）の形式」であるかをチェックします。

## 2. 便利な機能

CSVFRMCHK オプションが指定された場合、指定された有効桁範囲だけ「CSV 数値データ（固定小数点数）の形式」であるかをチェックします。指定された有効桁範囲以外のデータに不正があってもエラーにはなりません。



## 2.44 選択条件式のブレイクオプション (SORT EE の機能) 【08-50 以降】

---

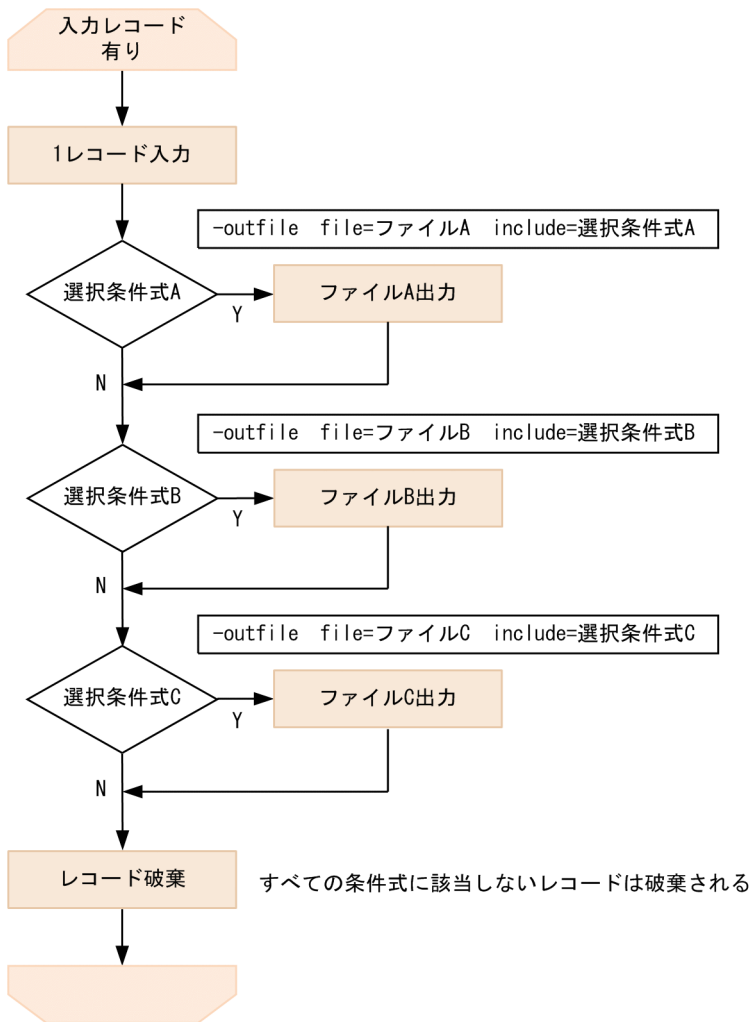
出力ファイルパラメタ (-outfile, -splitout, -catout パラメタ) に指定した選択条件式を、指定した順に評価、選択（出力ファイルへの出力）し、選択条件式に合致した時点で評価を終了（ブレイク）します。

次に選択機能の処理の流れを示します。

選択条件式のブレイクオプションの指定がない場合、-outfile, -catout, または -splitout パラメタに指定した選択条件式を、指定した順にすべて評価し選択（出力ファイルへの出力）します。選択条件に合致すれば選択（出力ファイルへの出力）するため、選択条件によっては1つの入力レコードが複数の出力ファイルへ出力されることがあります。

## 2. 便利な機能

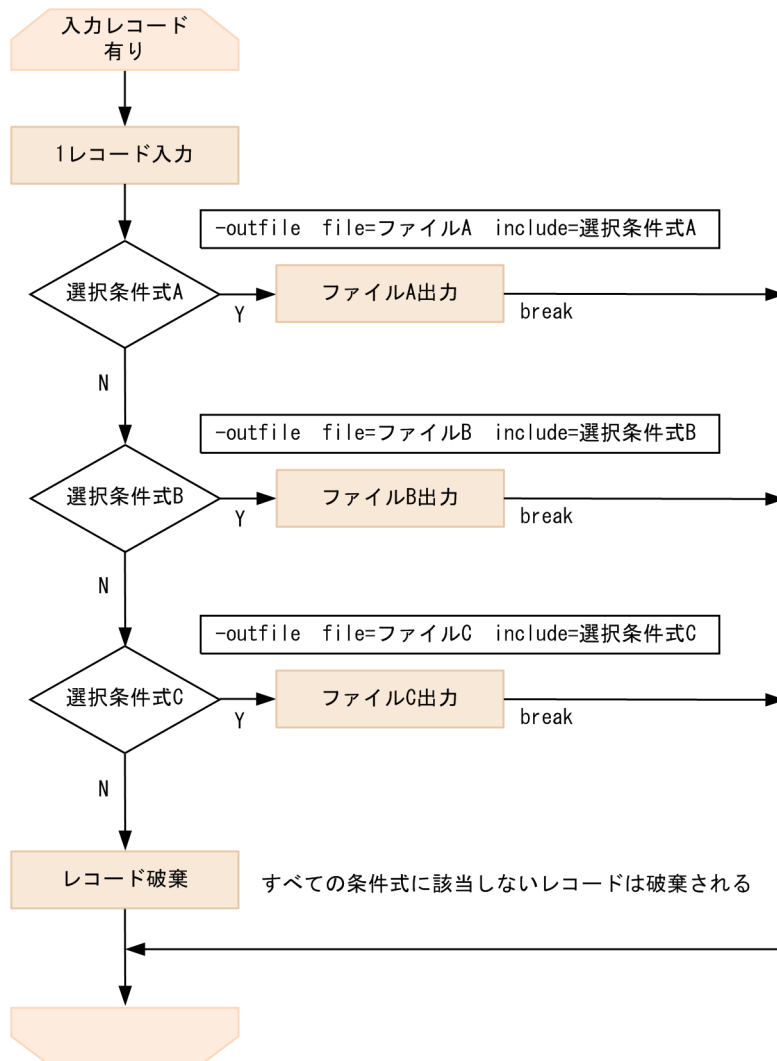
図 2-8 選択条件式のブレイクオプションの指定がない場合の処理の流れ



選択条件式のブレイクオプションを指定した場合、`-outfile` パラメタに指定された選択条件式を、指定した順に評価、選択（出力ファイルへの出力）し、選択条件式に合致した時点で評価を終了（ブレイク）します。選択条件に合致した時点で評価を終了（ブレイク）するため、合致した条件式以降に指定された条件式は評価しません。

なお、選択条件式の指定がない `-outfile` パラメタについては、オプションの有無に関係なくレコードが出力されます。

図 2-9 選択条件式のブレイクオプションを指定した場合の処理の流れ



コマンドオプションの指定方法については、「5.5.25 選択条件式のブレイクオプション (SELBRK オプション)」を参照してください。

## 2.45 CSV 日付データの部分指定（SORT EE の機能）【08-50 以降】

---

比較項目の属性が CSV 日付データ、または CSV 日付時刻データの場合、選択条件値にマスク文字を指定すれば、年 (YYYY) 月 (MM) 日 (DD)、日付時刻データは時 (hh) 分 (mm) 秒 (ss) ミリ秒 (fff) だけ比較、選択できます。

マスク文字の指定方法については、「5.8.17 {csvinclude | csvomit} オペランド」を参照してください。

# 3

## 利用できるファイルの種類 とデータ形式

SORT EE と SORT では、利用できるファイルの種類と、キーまたは項目の属性に指定できるデータ形式が決まっています。また、利用できる入力ファイル数、キー、項目個数などにも指定できる範囲が決まっています。

この章では、利用できるファイルの種類、キー属性と形式、指定できる値の範囲について説明します。

---

3.1 利用できるファイルの種類

---

3.2 キー属性と形式

---

3.3 指定できる値の範囲

---

## 3.1 利用できるファイルの種類

SORT EE と SORT で利用できるファイルの種類を、次の表に示します。

表 3-1 SORT EE と SORT で利用できるファイルの種類

ファイル編成の種類	説明	レコード長 (単位:バイト)		SORT EE	SORT
		最小値	最大値		
固定長順編成ファイル	COBOL85 または COBOL2002 で作成したファイルです。または、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す。	1	65,535	○	○
可変長順編成ファイル	COBOL85 または COBOL2002 で作成したファイルです。	1	65,535	○	○
固定長相対ファイル	COBOL85 または COBOL2002 で作成したファイルです。	1	65,535	×	○
可変長相対ファイル	COBOL85 または COBOL2002 で作成したファイルです。	1	65,535	×	○
固定長索引ファイル	COBOL85, COBOL2002, または ISAM で作成したファイルです。	1	65,503	×	○※2
可変長索引ファイル	COBOL85, COBOL2002, または ISAM で作成したファイルです。	1	65,503	×	○※2
テキストファイル	<p>COBOL85, COBOL2002, またはテキストエディタによって作成されたファイルです。</p> <ul style="list-style-type: none"> <li>• SORT で扱うテキストファイルは、行の区切り (改行) 文字は LF だけです。</li> <li>• SORT EE で扱うテキストファイルは、行の区切り (改行) 文字として、LF, CRLF, または LF/CRLF 混在のどれかを指定できます。</li> <li>• SORT または、SORT EE で改行文字を LF と指定※3 した場合、実際の改行文字が CRLF のレコードについては、CR をテキストデータとして扱います。</li> <li>• SORT では、データがないレコード (改行文字だけの空のレコード) は入力できません。</li> <li>• SORT EE には、空のレコードを削除して入力するオプション※5 があります。このオプションの指定がある場合は、空のレコードを入力してもエラーになりません。</li> </ul>	1	65,535	○※6	○※7

ファイル編成の種類	説明	レコード長 (単位:バイト)		SORT EE	SORT
		最小値	最大値		
CSV ファイル※1	<p>表計算プログラムのアクティブシートだけをデータごとに区切り文字（デリミタ）で区切って出力した形式のテキストファイルです。</p> <ul style="list-style-type: none"> <li>アクティブシートの行の位置はレコードの順番に対応し、列（セル）の位置はレコード中に区切り文字で区切られた位置に対応します。</li> <li>改行文字を LF と指定※3した場合、実際の改行文字が CRLF のレコードについては、CR をテキストデータとして扱います。</li> <li>テキストファイルと異なり、データがないレコード（改行文字だけの空のレコード）を入力できます。この場合、列（セル）のデータは、ナル文字列※4として扱います。</li> <li>空のレコードを削除して入力するオプション※5があります。</li> </ul>	0	65,535	○※6	×

(凡例)

○：使用できる    ×：使用できない

注※1

CSV ファイルのファイル形式の詳細は、「付録 C.1 CSV ファイル形式」を参照してください。

注※2

集約機能をソート機能および選択機能と組み合わせないで単独で使用した場合は、索引ファイルは使用できません。

注※3

SORT の場合は、改行文字は必ず LF です。SORT EE の場合は、改行文字の種類を指定できません。SORT EE の改行文字の指定については、「5.8.7 recsepa オペランド」を参照してください。

注※4

ナル文字列については、「付録 D CSV ファイル内のナル文字列の扱い (SORT EE の機能)」を参照してください。

注※5

空のレコードを削除して入力するオプションについては、「2.33 空行削除オプション (SORT EE の機能)」を参照してください。

注※6

codetype オペランドで指定したエンコードによって、行の区切り（改行）文字の長さや最大レコード長が異なります。なお、改行文字の長さはレコード長に含めません。改行文字の詳細については、「5.8.7 recsepa オペランド」を参照してください。

注※7

テキストファイルの行の区切り（改行）文字の長さは、レコード長に含めません。

## 3.2 キー属性と形式

キーまたは項目の属性として指定できるデータの種類を、次の表に示します。

内部浮動小数点の形式については、使用する C 言語のマニュアルを参照してください。

なお、指定できる内部浮動小数点は、次に示す形式だけです。

- 単精度浮動小数点 (single precision floating point number)
- 倍精度浮動小数点 (double precision floating point number)

表 3-2 SORT で指定できるデータの種類

キー・項目の属性		属性コード	キー	比較項目	集約項目
文字	JIS8 単位コード※	j	○	○	×
	文字	c	○	○	×
2 進数	符号付き固定小数点 2 進数	n	○	○	○
	符号なし 2 進数	m	○	○	○
	内部浮動小数点	f	○	○	×
10 進数	内部 10 進数	p	○	○	○
	外部 10 進数	z	○	○	○
	左符号付き外部 10 進数	x	○	○	○
	右分離符号付き外部 10 進数	s	○	○	○
	左分離符号付き外部 10 進数	t	○	○	○
日付 (英数字)		g	○	○	×

(凡例)

- : 指定できる
- × : 指定できない

注※

JIS8 単位コード (0x00 ~ 0xFF) の文字を意味します。EBCDIK コードの体系に従ってキーを比較する「EBCDIK コードキー比較機能」を使用できます。



表 3-3 SORT EE で指定できるデータの種類

キー・項目の属性		属性コード	キー		比較項目		集約項目	
			-key	-csvkey (CSV)	include/omit	csvinclude/csvomit (CSV)	-sum	-csvsum (CSV)
文字	JIS8 単位コード※1	j	○	○	○	○	×	×
	シングルバイト文字※2	c	○	○	○	○	×	×
	文字順序変更機能で指定した文字※2	q	○	○	×	×	×	×
	ワイド文字※3	w	○	○	○	○	×	×
	マルチバイト文字※5, ※6	u	○	○	○	○	×	×
		v	○	○	○	○	×	×
左分離符号付き数字※4	d	○	○	○	○	×	×	
2進数	符号付き固定小数点2進数※7, ※8	n	○	×	○	×	○	×
	符号なし2進数(0から始まる整数)※7, ※8	m	○	×	○	×	○	×
		a	○	×	×	×	×	×
	符号なし2進数(バイトデータ)※8	b	○	×	×	×	×	×
内部浮動小数点数※7, ※8	f	○	×	×	×	×	×	
10進数	内部10進数	p	○	×	○	×	○	×
	外部10進数(CSV以外)※4	z	○	○	○	○	○	○
			固定小数点数(CSV)※4	○	○	○	○	○
	左符号付き外部10進数※4	x	○	×	○	×	○	×
	右分離符号付き外部10進数※4	s	○	×	○	×	○	×
	左分離符号付き外部10進数※4	t	○	×	○	×	○	×
日付(英数字)※4	g	○	○	○	○	×	×	
日付時刻(数字)※4	h	×	○	×	○	×	×	

(凡例)

- ：指定できる
- ×

×：指定できない

### 3. 利用できるファイルの種類とデータ形式

#### 注※ 1

JIS8 単位コード (0x00 ~ 0xFF) の文字を意味します。JIS8 単位コードを、EBCDIK コードの体系に従ってキーを比較する「EBCDIK コードキー比較機能」を使用できます。

#### 注※ 2

入力データのエンコードの種別には対応しません。入力データの文字を構成するバイト長を意識しないで、常に 1 バイト長のシングルバイト文字として扱います。ASCII コードや JIS8 単位コード向けの属性です。

#### 注※ 3

入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定した場合、1 文字 2 バイトのワイド文字として扱います。また、サロゲートペア文字は 2 文字として扱います。

入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定した場合、1 文字 4 バイトのワイド文字として扱います。

文字のエンコードの指定方法については、「5.8.13 codeltype オペランド」を参照してください。なお、属性コード「w」は、固定長順編成ファイルおよび可変長順編成ファイルの入力ファイルに対して指定できないので注意してください。

#### 注※ 4

入力データのエンコードで解釈します。次のとおりの長さで文字を扱います。

- ・「ASCII」、「SJIS」、「EUC-JP」、「EUC-HJ」または「UTF-8」を指定した場合：1 バイト
  - ・「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」を指定した場合：2 バイト
  - ・「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」を指定した場合：4 バイト
- 文字のエンコードの指定方法については、「5.8.13 codeltype オペランド」を参照してください。

#### 注※ 5

入力データのエンコードで解釈します。1 文字の長さは可変長として扱います。

文字のエンコードの指定方法については、「5.8.13 codeltype オペランド」を参照してください。

なお、属性「u」および「v」は固定長順編成ファイルと可変長順編成ファイルの入力ファイルに対して指定できません。

#### 注※ 6

入力データのエンコードに「UTF-8」、「UTF-16LE」、「UTF-16BE」、「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE、UTF-16BE、UTF-32LE または UTF-32BE)」のどれかを指定した場合、Unicode の IVS (Ideographic Variation Sequence) を次のとおり解釈します。IVS の扱いについては、「付録 I.2(6) IVS 対応」を参照してください。

- ・属性コード「u」は、Unicode の IVS の異体字を同じ文字として扱います。
- ・属性コード「v」は、Unicode の IVS の異体字を異なる文字として扱います。

#### 注※ 7

属性コード「n」、「m」、「a」および「f」は数値として比較するため、バイトオーダーが関係します。バイトオーダーについては、「2.6 エンディアンオプション機能」および「5.5.4 入力ファイルのエンディアンオプション (BIG | INPBIG | INPLTL オプション)」を参照してください。

- ・符号付き固定小数点数「n」: 2バイト (signed short int), 4バイト (signed int), 8バイト (signed long long int)
- ・符号なし2進数「m」, 「a」: 2バイト (unsigned short int), 4バイト (unsigned int), 8バイト (unsigned long long int)
- ・内部浮動小数点数「f」: 4バイト (float), 8バイト (double)

注※8

属性コード「n, m, f」および「a, b」は、どちらも「2進数」データですが、キーの長さの指定方法が異なります。

- ・属性コード「n」, 「m」, および「f」は、キーの長さを「バイト長」で指定します。
- ・属性コード「a」と「b」は、キーの長さ分の「マスクビット」を指定します。

CSVファイルのデータ形式は、「付録 C.2 CSV データ形式」を参照してください。

キーまたは項目の属性として指定できるデータの形式を、SORT/SORT EE 共通と SORT EE 固有に分けて次に示します。丸括弧内は、SORT EE および SORT の属性コードです。

● SORT/SORT EE 共通のデータ形式

●文字 (j, c)

文字	文字	文字
----	----	----

≪1バイト≫

CSVファイルの場合は、引用符でデータの先頭および末尾を囲むことができます。CSVのデータ形式は、「付録 C CSVファイル形式と CSVデータ形式 (SORT EE の機能)」および「付録 B.2 文字列データの端数処理」を参照してください。

●符号付き固定小数点2進数 (n)

符号	整数
----	----

≪1ビット≫

符号は、0 (正) または 1 (負) のどちらかのビット値を指定します。

●符号なし2進数 (m)

ビット	ビット	ビット
-----	-----	-----

≪1ビット≫

●内部10進数 (p)

数値	数値	数値	数値	数値	符号
----	----	----	----	----	----

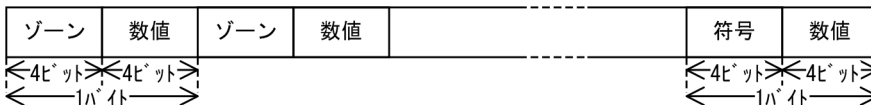
≪4ビット≫ ≪4ビット≫ ≪4ビット≫ ≪4ビット≫ ≪4ビット≫ ≪4ビット≫  
 ≪1バイト≫ ≪1バイト≫

数値は、0x0 ~ 0x9 の値を指定します。

符号は、0xC (正), 0xD (負) または 0xF (絶対値) のどれかの数値を指定します。

### 3. 利用できるファイルの種類とデータ形式

#### ●外部10進数 (z) …CSVファイル以外



ゾーンには 0x3, 数値には 0x0 ~ 0x9 の値を指定します。

符号は, 0x3 (正) または 0x7 (負) のどちらかの数値を指定します。

なお, SORT EE で「外部 10 進数データ形式オプション」を指定している場合, 符号と数字はオプションで指定した形式となります。オプションの詳細については, 「14.1.1(31) EXSORTUNPACKDATATYPE」および「14.1.2(31) EXSORT64UNPACKDATATYPE」を参照してください。

#### ●左符号付き外部10進数 (x)

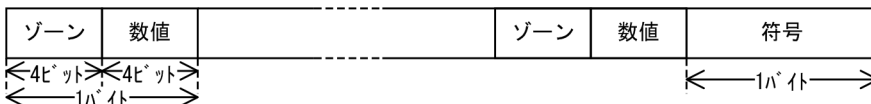


ゾーンには 0x3, 数値には 0x0 ~ 0x9 の値を指定します。

符号は, 0x3 (正) または 0x7 (負) のどちらかの数値を指定します。

なお, SORT EE で「外部 10 進数データ形式オプション」を指定している場合, 符号と数字はオプションで指定した形式となります。オプションの詳細については, 「14.1.1(31) EXSORTUNPACKDATATYPE」および「14.1.2(31) EXSORT64UNPACKDATATYPE」を参照してください。

#### ●右分離符号付き外部10進数 (s)



ゾーンには 0x3, 数値には 0x0 ~ 0x9 の値を指定します。

符号は, + (正) または - (負) のどちらかの文字を指定します。

なお, SORT EE で「外部 10 進数データ形式オプション」を指定している場合, 符号と数字はオプションで指定した形式となります。オプションの詳細については, 「14.1.1(31) EXSORTUNPACKDATATYPE」および「14.1.2(31) EXSORT64UNPACKDATATYPE」を参照してください。

#### ●左分離符号付き外部10進数 (t)



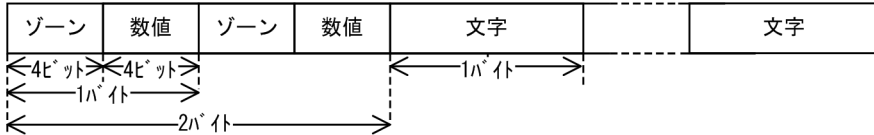
符号は, + (正) または - (負) のどちらかの文字を指定します。

ゾーンには 0x3, 数値には 0x0 ~ 0x9 の値を指定します。

なお, SORT EE で「外部 10 進数データ形式オプション」を指定している場合, 符号

と数字はオプションで指定した形式となります。オプションの詳細については、「14.1.1(31) EXSORTUNPACKDATATYPE」および「14.1.2(31) EXSORT64UNPACKDATATYPE」を参照してください。

●日付(g)・・・CSVファイル以外

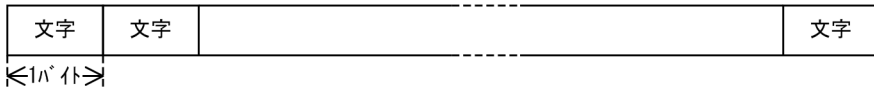


ゾーンには 0x3, 数値には 0x0 ~ 0x9 の値を指定します。

なお、SORT EE で「外部 10 進数データ形式オプション」を指定している場合、数字はオプションで指定した形式となります。オプションの詳細については、「14.1.1(31) EXSORTUNPACKDATATYPE」および「14.1.2(31) EXSORT64UNPACKDATATYPE」を参照してください。

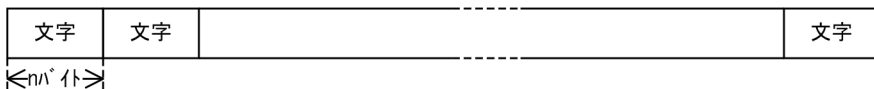
● SORT EE 固有のデータ形式

●文字順序変更機能で指定した文字 (q)



CSV ファイルの場合は、引用符でデータの先頭および末尾を囲むことができます。CSV のデータ形式は、「付録 C.2 CSV データ形式」を参照してください。

●ワイド文字 (w)・・・テキストファイルまたはCSVファイル



1 文字の長さ (n バイト) は、入力ファイルのエンコードによって異なります。次にその種類を示します。

- 「ASCII」, 「SJIS」, 「EUC-JP」, 「EUC-HJ」または「UTF-8」を指定した場合：1 バイト
- 「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」を指定した場合：2 バイト
- 「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」を指定した場合：4 バイト

なお、サロゲートペア文字は 2 文字として扱います。ワイド文字については、「付録 I マルチバイト文字データの扱い」を参照してください。

CSV ファイルの場合は、引用符でデータの先頭および末尾を囲めます。

CSV のデータ形式は、「付録 C CSV ファイル形式と CSV データ形式 (SORT EE の機能)」を参照してください。

### 3. 利用できるファイルの種類とデータ形式

#### ●マルチバイト文字 (u, v) …テキストファイルまたはCSVファイル

文字	文字		文字
----	----	--	----

≪nバイト≫

1文字の長さ (nバイト) は入力ファイルのエンコードによって異なります。次にその種類を示します。

- 「ASCII」を指定した場合：1バイト
- 「SJIS」または「EUC-HJ」を指定した場合：1または2バイト
- 「EUC-JP」を指定した場合：1～3バイト
- 「UTF-8」を指定した場合：1～4バイト
- 「UCS-2LE」または「UCS-2BE」を指定した場合：2バイト (サロゲートペアは2文字で扱う)
- 「UTF-16LE」または「UTF-16BE」を指定した場合：2または4バイト (サロゲートペアは1文字で扱う)
- 「UCS-4LE」または「UCS-4BE」を指定した場合：4バイト
- 「UTF-32LE」または「UTF-32BE」を指定した場合：4バイト

CSVファイルの場合は、引用符でデータの先頭および末尾を囲めます。

マルチバイト文字の詳細については、「付録 I.2 文字列データの扱い」を参照してください。

CSVのデータ形式は、「付録 C CSVファイル形式とCSVデータ形式 (SORT EEの機能)」を参照してください。

#### ●左分離符号付き数字 (d)

(符号あり)

≪符号部≫				≪数値部≫			
符号	数字	数字					数字
≪nバイト≫	≪nバイト≫	≪nバイト≫					≪nバイト≫

(符号なし)

≪数値部≫			
数字	数字		数字
≪nバイト≫	≪nバイト≫		≪nバイト≫

符号は、+ (正) または- (負) のどちらかの文字を指定します。

数字には、0～9を指定します。

なお、入力ファイルのエンコードが「ASCII」、「SJIS」、「EUC-JP」、「EUC-HJ」または「UTF-8」で「外部10進数データ形式オプション」を指定している場合、符号と数字はオプションで指定した形式となります。オプションの詳細については、「14.1.1(31) EXSORTUNPACKDATATYPE」および「14.1.2(31)

EXSORT64UNPACKDATATYPE」を参照してください。

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は、1文字の長さ (nバイ

ト) を1バイトで扱います。

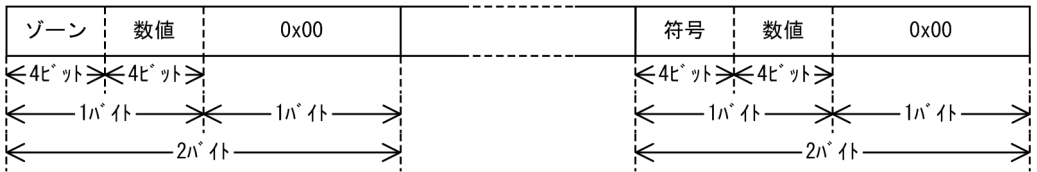
テキストファイルまたは CSV ファイルの場合は1桁の長さ (n バイト) は、入力ファイルのエンコードによって異なります。次にその種類を示します。

- 「ASCII」, 「SJIS」, 「EUC-JP」, 「EUC-HJ」または「UTF-8」を指定した場合：1 バイト
- 「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」を指定した場合：2 バイト
- 「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」を指定した場合：4 バイト

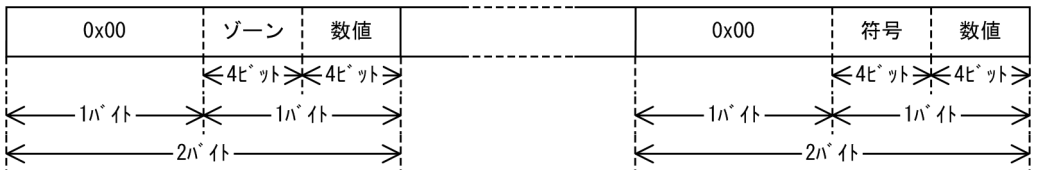
データ形式については、「付録 E 左分離符号付き数字のデータ形式 (SORT EE の機能)」を参照してください。CSV ファイルの場合は、引用符でデータの先頭および末尾を囲むことができます。CSV のデータ形式は、「付録 C.2 CSV データ形式」および「付録 B.1 数値 (固定小数点数) データの端数処理」を参照してください。

●外部10進数 (z) …テキストファイル

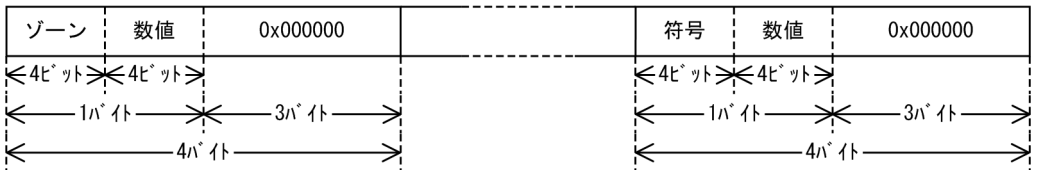
(UCS-2LE, UTF-16LE)



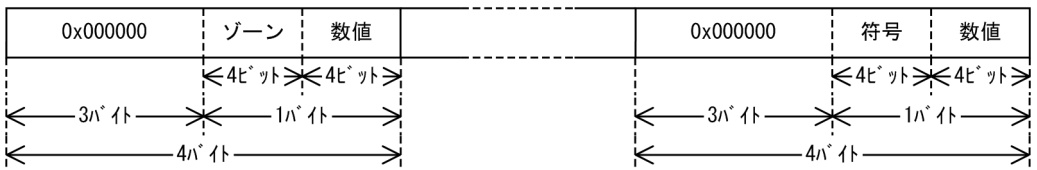
(UCS-2BE, UTF-16BE)



(UCS-4LE, UTF-32LE)



(UCS-4BE, UTF-32BE)



ゾーンには 0x3, 数値には 0x0 ~ 0x9 の値を指定します。

### 3. 利用できるファイルの種類とデータ形式

符号は、0x3（正）または0x7（負）のどちらかの数値を指定します。

#### ●固定小数点数（z）…CSVファイル

（左符号付き）

符号	数字		小数点		数字
≪nバイト≫	≪nバイト≫		≪nバイト≫		≪nバイト≫

（右符号付き）

数字		小数点		数字	符号
≪nバイト≫		≪nバイト≫		≪nバイト≫	≪nバイト≫

数字は、0～9の数字を指定します。また、0～9の数字以外に、次の文字をデータに含むことができます。

- 引用符（データの先頭および末尾の引用符）
- 符号（+、または-）
- 通貨記号
- コンマ
- 小数点
- 空白

1桁の長さ（nバイト）は、入力ファイルのエンコードによって異なります。次にその種類を示します。

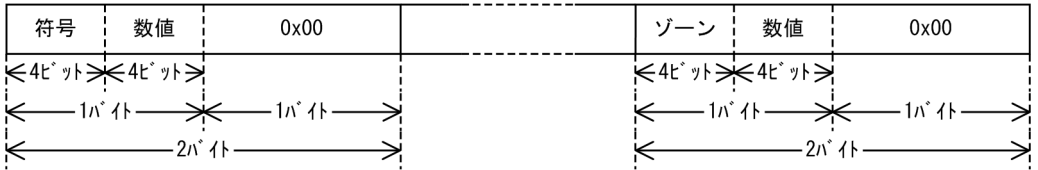
- 「ASCII」、「SJIS」、「EUC-JP」、「EUC-HJ」または「UTF-8」を指定した場合：1バイト
- 「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」を指定した場合：2バイト
- 「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」を指定した場合：4バイト

CSVのデータ形式には、詳細な規則があります。「付録C.2 CSVデータ形式」および「付録B.1 数値（固定小数点数）データの端数処理」を参照してください。

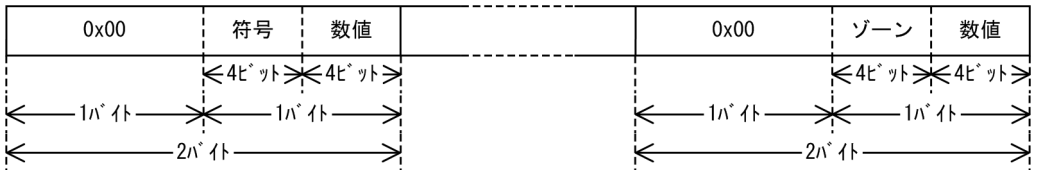


●左符号外部10進数 (x) …テキストファイル

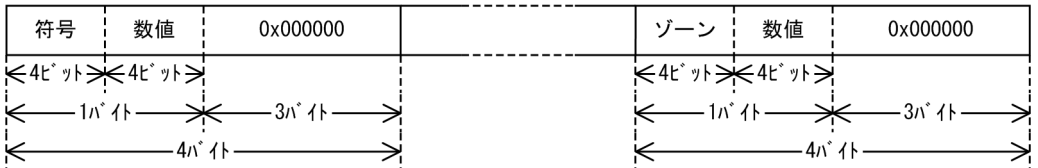
(UCS-2LE, UTF-16LE)



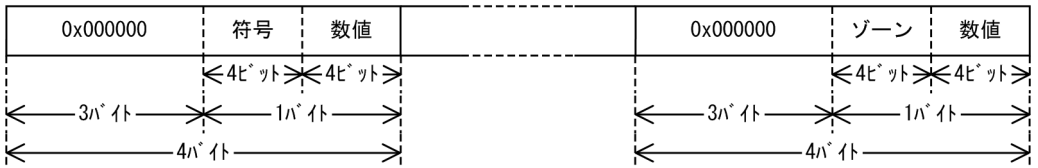
(UCS-2BE, UTF-16BE)



(UCS-4LE, UTF-32LE)



(UCS-4BE, UTF-32BE)



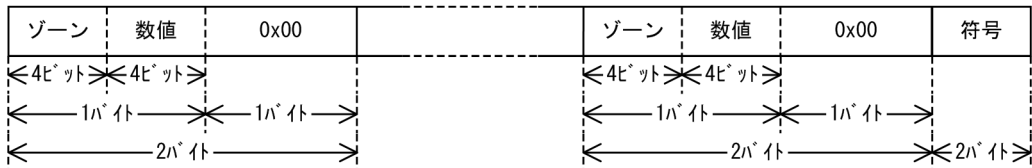
ゾーンには 0x3, 数値には 0x0 ~ 0x9 の値を指定します。

符号は, 0x3 (正) または 0x7 (負) のどちらかの数値を指定します。

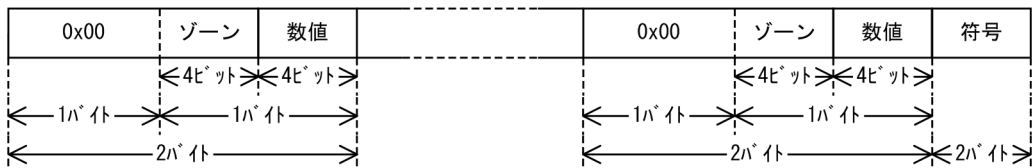
### 3. 利用できるファイルの種類とデータ形式

● 右分離符号付き外部10進数 (s) … テキストファイル

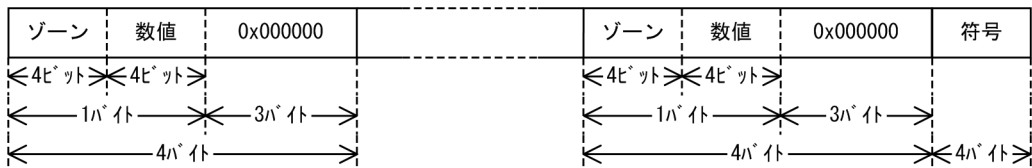
(UCS-2LE, UTF-16LE)



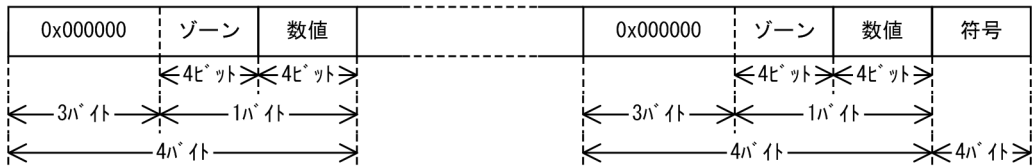
(UCS-2BE, UTF-16BE)



(UCS-4LE, UTF-32LE)



(UCS-4BE, UTF-16BE)

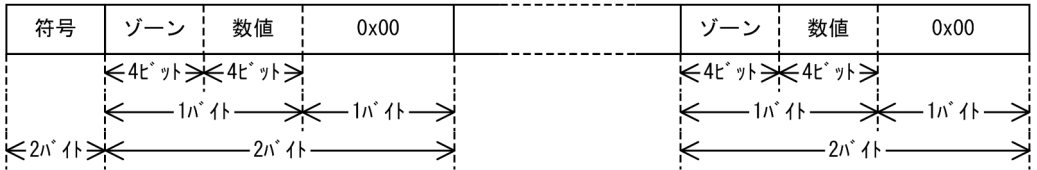


ゾーンには 0x3, 数値には 0x0 ~ 0x9 の値を指定します。

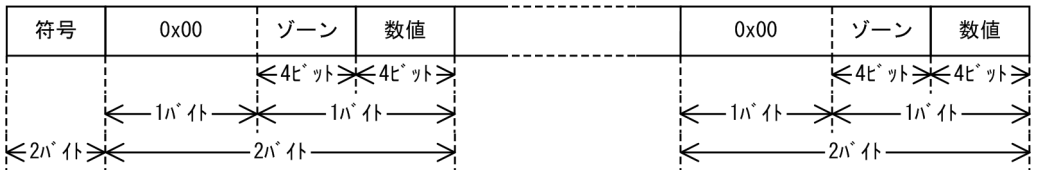
符号は, + (正) または - (負) のどちらかの文字を指定します。

●左分離符号付き外部10進数 (t) …テキストファイル

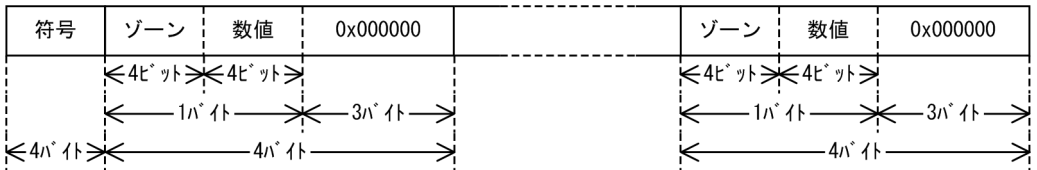
(UCS-2BE, UTF-16LE)



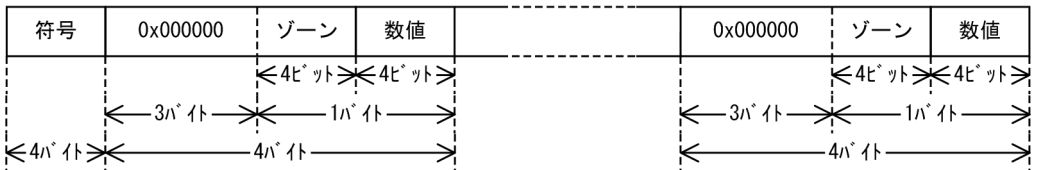
(UCS-2BE, UTF-16BE)



(UCS-4LE, UTF-32LE)



(UCS-4BE, UTF-32BE)

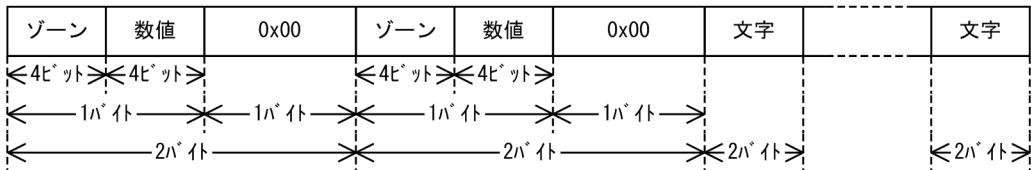


符号は、+ (正) または- (負) のどちらかの文字を指定します。  
 ゾーンには 0x3, 数値には 0x0 ~ 0x9 の値を指定します。

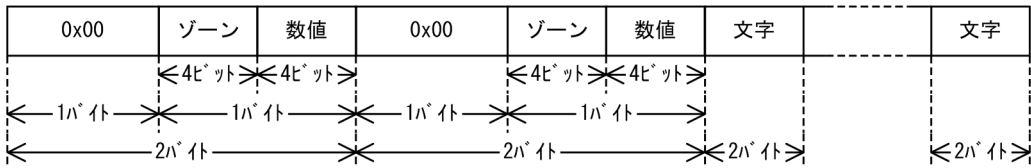
### 3. 利用できるファイルの種類とデータ形式

●日付 (g) …テキストファイル

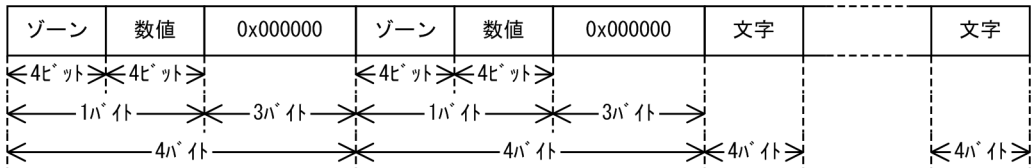
(UCS-2LE, UTF-16LE)



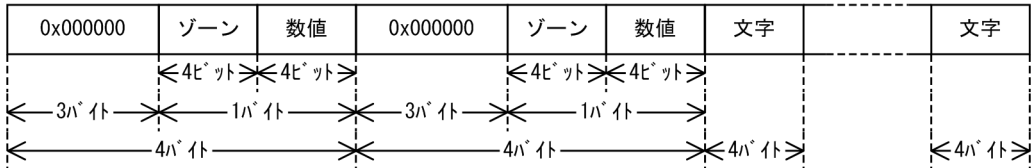
(UCS-2BE, UTF-16BE)



(UCS-4LE, UTF-32LE)



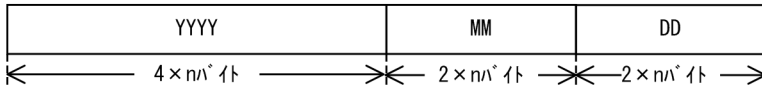
(UCS-4BE, UTF-32BE)



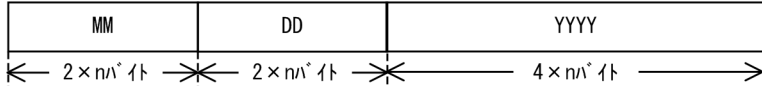
ゾーンには 0x3, 数値には 0x0 ~ 0x9 の値を指定します。

## ●日付 (g) …CSVファイル

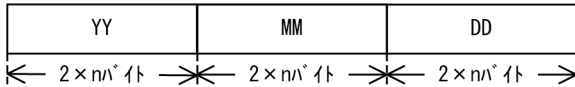
(前置4桁年号)



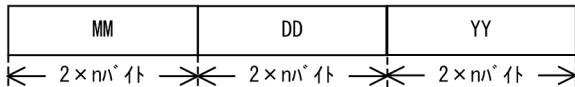
(後置4桁年号)



(前置2桁年号)



(後置2桁年号)



Y (年), M (月), D (日) には, それぞれ 0 ~ 9 の数字を指定します。

- YYYY : 西暦 4 桁年号
- YY : 西暦下 2 桁年号
- MM : 月
- DD : 日

また, 0 ~ 9 の数字以外に, 次の文字をデータに含むことができます。

- 引用符 (データの先頭および末尾の引用符)
- スラント (年月日の区切り)
- ハイフン (年月日の区切り)
- ピリオド (年月日の区切り)

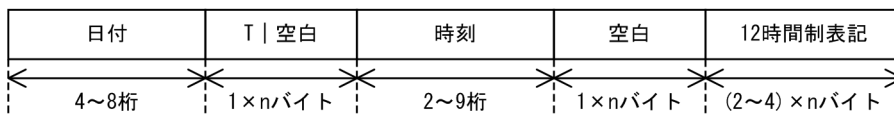
1 桁の長さ (n バイト) は, 入力ファイルのエンコードによって異なります。次にその種類を示します。

- 「ASCII」, 「SJIS」, 「EUC-JP」, 「EUC-HJ」または「UTF-8」を指定した場合 : 1 バイト
- 「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」を指定した場合 : 2 バイト
- 「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」を指定した場合 : 4 バイト

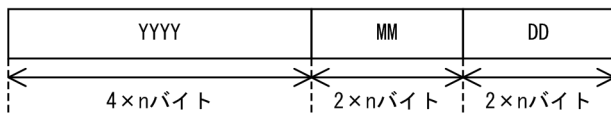
CSV のデータ形式には, 詳細な規則があります。「付録 B.3 日付データの端数処理」および「付録 C.2 CSV データ形式」を参照してください。

### 3. 利用できるファイルの種類とデータ形式

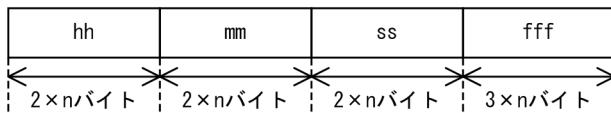
#### ●日付時刻 (h) …CSVファイル



(日付)



(時刻)



日付時刻のデータは、日付、時刻、12時間制表記の順に並びます。

日付の Y (年), M (月), D (日), および時刻の h (時), m (分), s (秒), f (ミリ秒) には、それぞれ 0 ~ 9 の数字を指定します。

- YYYY : 西暦 4 桁年号
- MM : 月
- DD : 日
- hh : 時
- mm : 分
- ss : 秒
- fff : ミリ秒
- 12 時間表記 :  
午前を示す表記「AM」, 「am」, 「A.M.」, または「a.m.」  
午後を示す表記「PM」, 「pm」, 「P.M.」, または「p.m.」

1 桁の長さ (n バイト) は、入力ファイルのエンコードによって異なります。

また、0 ~ 9 の数字以外に、次の文字をデータに含められます。

- 引用符 (データの先頭および末尾の引用符)
- スラント (年月日の区切り)
- ハイフン (年月日の区切り)
- ピリオド (年月日および時分秒の区切り)
- コロン (時分秒の区切り)

1 桁の長さ (n バイト) は、入力ファイルのエンコードによって異なります。次にその種類を示します。

- 「ASCII」, 「SJIS」, 「EUC-JP」, 「EUC-HJ」 または 「UTF-8」 を指定した場合 : 1 バイト
- 「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」 または 「UTF-16BE」 を指定した場合 : 2 バイト

- 「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」 または 「UTF-32BE」 を指定した場合 :  
4 バイト

データ形式については、「付録 J 日付時刻データの形式 (SORT EE の機能)」を参照してください。

CSV のデータ形式は、「付録 C.2 CSV データ形式」を参照してください。

## 3.3 指定できる値の範囲

パラメタ、メモリ量、および入力レコード件数とファイルサイズに指定できる値の範囲について説明します。

### 3.3.1 パラメタ

パラメタ指定時に必要な値として指定できる最小値および最大値を、次の表に示します。

表 3-4 SORT のパラメタの最小値および最大値

項目	最小値	最大値		単位	備考	
		UNIX	Windows			
ソートキーの個数	1	64	64	個	—	
マージキーの個数	1	64	64	個	—	
集約キーの個数	1	64	64	個	—	
選択条件式の個数	1	32	128	個	—	
集約項目の個数	0	64	64	個	—	
キーおよび項目の位置	0	レコード長－1	レコード長－1	バイト	—	
キーおよび項目の長さ	文字 (c, j)	1	4,096	4,096	バイト	集約項目は指定できません。
	2進数 (n, m)	2	8	8	バイト	2, 4, 8以外は指定できません。
	内部浮動小数点数 (f)	4	8	8	バイト	4, 8以外は指定できません。
	内部 10 進数 (p)	1	10	10	バイト	—
	外部 10 進数 (z)	1	18	18	バイト	—
	左符号外部 10 進数 (x)	1	18	18	バイト	—
	右分離符号外部 10 進数 (s)	2	19	19	バイト	—
	左分離符号外部 10 進数 (t)	2	19	19	バイト	—
日付 (g)	2	8	8	バイト	集約項目は指定できません。	



項目	最小値	最大値		単位	備考	
		UNIX	Windows			
条件値の長さ	文字条件値	1	4,096	4,096	桁	—
	数値条件値	1	19	19	桁	—
	16進数条件値	2	—	8,192	桁	—
	日付条件値	2	8	8	桁	—
キーの合計長	1	4,096	4,096	バイト	—	
比較項目の合計長	1	4,096	4,096	バイト	—	
集約項目の合計長	0	4,096	4,096	バイト	—	
入力ファイル数	1	制限なし (システム側の制限あり)	256	個	集約機能使用時の最大値は1個です。	
出力ファイルの数	1	1	1	個	—	
レコードの長さ	順編成ファイル	1	65,535	65,535	バイト	—
	相対編成ファイル	1	65,535	65,535	バイト	—
	索引編成ファイル	1	65,503	65,503	バイト	—
	テキストファイル	1	65,535	65,535	バイト	—

(凡例)

— : なし

表 3-5 SORT EE のパラメタの最小値および最大値

項目	最小値	最大値	単位	備考
キーの個数	1	128 × n	個	最大値の初期値は 128 個 (n=1) です。 環境ファイル「EXS_KEYNUMMAX」で最大 1,280 個 (n=10) まで拡張できます。
CSV キーの個数	1	128 × n	個	最大値の初期値は 128 個 (n=1) です。 環境ファイル「EXS_CSVKEYNUMMAX」で最大 1,280 個 (n=10) まで拡張できます。

### 3. 利用できるファイルの種類とデータ形式

項目	最小値	最大値	単位	備考
選択条件式の個数	1	128 × n	個	最大値の初期値は 128 個 (n=1) です。 環境ファイル 「EXS_SELNUMMAX」で最大 1,280 個 (n=10) まで拡張できます。
CSV 選択条件式の個数	1	128 × n	個	最大値の初期値は 128 個 (n=1) です。 環境ファイル 「EXS_CSVSELNUMMAX」で最大 1,280 個 (n=10) まで拡張できます。
集約項目の個数	0	128 × n	個	最大値の初期値は 128 個 (n=1) です。 環境ファイル 「EXS_SUMNUMMAX」で最大 1,280 個 (n=10) まで拡張できます。
CSV 集約項目の個数	0	128 × n	個	最大値の初期値は 128 個 (n=1) です。 環境ファイル 「EXS_CSVSUMNUMMAX」で最大 1,280 個 (n=10) まで拡張できます。
編集フォーマットの個数	1	128	個	—
CSV 編集フォーマットの個数	1	128	個	—
分割キーの個数	1	64	個	—
分割 CSV キーの個数	1	64	個	—
非分割データの個数	1	64	個	—
非分割 CSV データの個数	1	64	個	—
文字順序変更機能に指定する文字コードの個数	1	256	対	変換対象の文字コードと、変換後の順序を対で指定します。
キーおよび項目の位置	0	レコード長 - 1	バイト	—
CSV キーおよび項目の位置	1	16,384	セル番号	—

### 3. 利用できるファイルの種類とデータ形式

	項目	最小値	最大値	単位	備考	
キーおよび項目の長さ	文字 (c, j, q, w)	1※1	16,384※1	バイト	集約項目は指定できません。	
	マルチバイト文字 (u, v)	1	4,096	文字	—	
	左分離符号付き数字 (d)	1	256	桁	—	
	2進数 (n, m)	2	8	バイト	2, 4, 8 以外は指定できません。	
	2進数 (a)	4	16	桁	2, 4, 8 バイトに対応するマスクビットを 4, 8, 16 桁の 16 進数で指定します。 キー以外は指定できません。	
	2進数 (b)	2	8,192	桁	1 ~ 4,096 バイトに対応するマスクビットを 2 ~ 8,192 桁の 16 進数で指定します。 キー以外は指定できません。	
	内部浮動小数点数 (f)	4	8	バイト	4, 8 以外は指定できません。	
	内部 10 進数 (p)	1	10	バイト	—	
	外部 10 進数 (z)	1	18	桁	—	
	左符号外部 10 進数 (x)	1	18	桁	—	
	右分離符号外部 10 進数 (s)	2	19	桁	—	
	左分離符号外部 10 進数 (t)	2	19	桁	—	
	日付 (g)	2	8	桁	集約項目は指定できません。	
CSV キーおよび項目の長さ	文字 (c, j, q, w)	1※1	16,384※1	バイト	集約項目は指定できません。	
	マルチバイト文字 (u, v)	1	4,096	文字	—	
	左分離符号付き数字 (d)	1	256	桁	集約項目は指定できません。	
	固定小数点数 (z)	1	38	桁	—	
	西暦 2/4 桁年号 (g)	1	8	桁	集約項目は指定できません。	
	日付時刻 (h)	日付	0	8	桁	集約項目は指定できません。
		時刻	0	9	桁	

### 3. 利用できるファイルの種類とデータ形式

項目		最小値	最大値	単位	備考
条件値の長さ	文字条件値	1	4,096	桁	—
	10進数条件値	1	19	桁	符号なしは最大 18 桁です。
	数字条件値	1	256	桁	左分離符号付き数字 (d) の条件値です。
	16進数条件値	2 <sup>※2</sup>	32,768 <sup>※2</sup>	桁	—
	日付条件値	2	8	桁	—
CSV 条件値の長さ	文字条件値	1	4,096	桁	—
	固定小数点数条件値	1	39	桁	符号なしは最大 38 桁です。
	数値条件値	1	256	桁	左分離符号付き数字 (d) の条件値です。
	16進数条件値	2 <sup>※2</sup>	32,768 <sup>※2</sup>	桁	—
	日付条件値	2	8	桁	—
	日付時刻 (h)	日付	0	8	桁
時刻		0	9	桁	—
キーの合計長		1	16,384 × m <sup>※3</sup>	バイト	最大値の初期値は m=1 です。環境ファイル「EXS_KEYNUMMAX」で最大 m=16 まで拡張できます。
CSV キーの合計長		1	16,384 × m <sup>※3</sup>	バイト	最大値の初期値は m=1 です。環境ファイル「EXS_CSVKEYNUMMAX」で最大 m=16 まで拡張できます。
比較項目の合計長		1	16,384 × m <sup>※3</sup>	バイト	最大値の初期値は m=1 です。環境ファイル「EXS_SELNUMMAX」で最大 m=16 まで拡張できます。
CSV 比較項目の合計長		1	16,384 × m <sup>※3</sup>	バイト	最大値の初期値は m=1 です。環境ファイル「EXS_CSVSELNUMMAX」で最大 m=16 まで拡張できます。
集約項目の合計長		0	4,096 × m	バイト	最大値の初期値は m=1 です。環境ファイル「EXS_SUMNUMMAX」で最大 m=16 まで拡張できます。
CSV 集約項目の合計長		0	4,096 × m	バイト	最大値の初期値は m=1 です。環境ファイル「EXS_CSVSUMNUMMAX」で最大 m=16 まで拡張できます。
分割キーの合計長		1	16,384 <sup>※3</sup>	バイト	—

### 3. 利用できるファイルの種類とデータ形式

項目	最小値	最大値	単位	備考	
分割 CSV キーの合計長	1	16,384 ※3	バイト	—	
非分割データの合計長	1	16,384 ※3	バイト	—	
非分割 CSV データの合計長	1	16,384 ※3	バイト	—	
入力ファイルパラメタの個数 ( <code>-inpfile</code> , <code>-catinp</code> , <code>-splitinp</code> , <code>-cpyinp</code> パラメタ)	1	1※4	個	—	
入力ファイル数	<code>-inpfile</code> パラメタ	1	256 (システム側の制限あり) ※4	個	<code>-function copy</code> と組み合わせた集約機能を使用する場合、最大値は1個です。
	<code>-catinp</code> , <code>-splitinp</code> パラメタ	1	制限なし (システム側の制限あり) ※4	個	<code>-function copy</code> と組み合わせた集約機能を使用する場合、最大値は1個です。
	<code>-cpyinp</code> パラメタ	1	制限なし (システム側の制限あり) ※4	個	—
出力ファイルパラメタの個数	<code>-outfile</code> パラメタ	1	64※4	個	—
			制限なし ※4		
	<code>-catout</code> , <code>-splitout</code> パラメタ	1	制限なし ※4	個	—
	<code>-cpyout</code> パラメタ	1	1※4	個	—

### 3. 利用できるファイルの種類とデータ形式

項目	最小値	最大値	単位	備考	
すべての出力ファイルパラメタに指定できる出力ファイルの合計数	-outfile パラメタ	1	256 (システム側の制限あり) ※4	個	1つの -outfile パラメタに指定できる出力ファイル数は最大4個です。
			制限なし (システム側の制限あり) ※4		-outfile パラメタ数の制限解除オプション指定で制限なく指定できます。 ※7 1つの -outfile パラメタに指定できる出力ファイル数は最大4個です。
	-catout パラメタ	1	制限なし (システム側の制限あり) ※4	個	1つの -catout パラメタに指定できる出力ファイル数は最大4個です。
	-splitout パラメタ	1	制限なし (システム側の制限あり) ※4	個	1つの -splitout パラメタに指定できる出力ファイル数は最大4個です。
	-cpyout パラメタ	1	4 (システム側の制限あり) ※4	個	—
レコード長※6	順編成ファイル	1	65,535	バイト	—
	テキストファイル	1	65,535 ※5	バイト	—
	CSV ファイル	0	65,535 ※5	バイト	—

(凡例)

— : なし

注※1

codetype オペランドで指定したエンコードによって、文字属性のキー・項目に指定できる長さが異なります。

エンコード種別	文字 (c, j, q, w) に指定できる長さ (単位 : バイト)	
	最小長	最大長
ASCII, SJIS, EUC-JP, EUC-HJ または UTF-8	1	4,096
UCS-2LE, UCS-2BE, UTF-16LE または UTF-16BE	2	8,192

エンコード種別	文字 (c, j, q, w) に指定できる長さ (単位: バイト)	
	最小長	最大長
UCS-4LE, UCS-4BE, UTF-32LE または UTF-32BE	4	16,384

## 注※ 2

16 進数条件値は、1 バイトを 2 桁で指定します。また、比較項目が文字属性の場合は、codetype オペランドで指定したエンコードによって、指定できる桁数が異なります。

エンコード種別	16 進数条件値に指定できる桁数	
	最小長	最大長
ASCII, SJIS, EUC-JP, EUC-HJ または UTF-8	2	8,192
UCS-2LE, UCS-2BE, UTF-16LE または UTF-16BE	4	16,384
UCS-4LE, UCS-4BE, UTF-32LE または UTF-32BE	8	32,768

## 注※ 3

codetype オペランドで指定したエンコードによって、キー・項目の最大合計長が異なります。

エンコード種別	キー・項目の最大合計長 (単位: バイト)
ASCII, SJIS, EUC-JP, EUC-HJ または UTF-8	4,096 × m
UCS-2LE, UCS-2BE, UTF-16LE または UTF-16BE	8,192 × m
UCS-4LE, UCS-4BE, UTF-32LE または UTF-32BE	16,384 × m

## 注※ 4

- `-inpfile` パラメタに指定できる入力ファイルの最大数は 256 個です。
- `-catinp`, `-splitinp`, および `-cpyinp` パラメタは、入力ファイル数に制限はありません。
- `-outfile` パラメタ数の制限解除オプションを指定しない場合、すべての `-outfile` パラメタに指定できる出力ファイルの最大数は、「`-outfile` パラメタ 64 個 × 指定ファイル 4 個 = 256 個」です。
- `-outfile` パラメタ数の制限解除オプションを指定した場合、指定する `-outfile` パラメタの個数に制限はありません。ただし、1 つの `-outfile` パラメタに指定できる出力ファイルの最大数は 4 個です。
- `-cpyout` パラメタに指定できる出力ファイルの最大数は 4 個です。
- `-catout`, `-splitout`, および `-outfile` パラメタ数の制限解除オプションを指定した `-outfile` パラメタで、実際に使用できるファイル数は、システム側のファイル数の制限までなので注意してください。

### 3. 利用できるファイルの種類とデータ形式

#### 注※ 5

`codetype` オペランドで指定したエンコードによって、改行文字の長さや最大レコード長が異なります。なお、改行文字の長さはレコード長に含めません。

エンコード種別	改行文字のコード		テキスト・CSV ファイルの最大レコード長 (単位: バイト)	
	CR	LF	改行文字が LF の場合	改行文字が CRLF の場合
ASCII	0x0d	0x0a	65,535	65,534
SJIS				
EUC-JP				
EUC-HJ				
UTF-8				
UCS-2LE	0x0d00	0x0a00	65,534	65,532
UTF-16LE				
UCS-2BE	0x000d	0x000a		
UTF-16BE				
UCS-4LE	0x0d000000	0x0a000000	65,532	65,528
UTF-32LE				
UCS-4BE	0x0000000d	0x0000000a		
UTF-32BE				

#### 注※ 6

レコード長が最大値を超えるレコードは入力できません。ただし、簡易コピー機能 (`-copyinp`, `-copyout` パラメタ) ではレコード長の制限はありません。

#### 注※ 7

`-outfile` パラメタ数の制限解除オプションは、SORT EE 08-50 以降で使用できます。使用する場合は、環境変数「`EXSORTOUTFILEENUMUNLIMIT` (64 ビット版は「`EXSORT64OUTFILEENUMUNLIMIT`)」、または環境ファイル「`EXS_OUTFILEENUMUNLIMIT`」を指定してください。

## 3.3.2 メモリ量

ソート機能でワークバッファサイズに指定するメモリ量に指定できる最小値および最大値を次の表に示します。



表 3-6 メモリ量の最小値および最大値

種別		レコード長	最小値	最大値	備考
SORT	32 ビット版	32,767 バイト以下	128KB	2,000MB	最大値とは、ワークバッファサイズに指定できる最大メモリ量です。なお、確保できるメモリ量には OS またはシステム側の制限があります。
		32,768 バイト以上	256KB	2,000MB	
	64 ビット版	32,767 バイト以下	256KB	2,000MB	
		32,768 バイト以上	384KB	2,000MB	
SORT EE	32 ビット版	—	1MB	2,000MB	
	64 ビット版	—	1MB	2,000GB	

(凡例)

— : なし

### 3.3.3 入力レコード件数とファイルサイズ

各コマンドで処理できる入力ファイルの最小レコード件数は 0 件<sup>※1</sup>、最大レコード件数は、2,147,483,647 件<sup>※2</sup>です。なお、入力ファイルを複数指定した場合、レコード件数は合計数となります。

実行時に処理できる入力ファイル、出力ファイル、一時ファイルのファイルサイズは、最小 0 バイト<sup>※3</sup>、最大は制限なし（ただし OS の制限に従う<sup>※4</sup>）です。

注※1

- レコードが 1 件もないファイル（空のファイル）を入力できます。
- テキストファイルの場合は、データがないレコード（改行文字だけの空のレコード）を入力できません。ただし、「空行削除オプション（`-cmdopt DELBLANKLINE`）」を指定すれば、データがないレコードを削除して入力できます。
- 簡易コピー機能を使用する場合は、テキストファイルであってもデータがないレコードも入力できます。
- CSV ファイルの場合は、データがないレコードを入力できます（ナル文字列として扱います）。

また、「空行削除オプション（`-cmdopt DELBLANKLINE`）」を指定すれば、データがないレコードを削除して入力できます。

【例】データの無いレコードの扱い

次の図のレコード 3（データの無いレコード）の扱いは次のとおりです。

### 3. 利用できるファイルの種類とデータ形式

レコード1	ABCD0123EFGH LF	
レコード2	Abcd5555 LF	
レコード3	LF	← 改行文字だけの空のレコード
レコード4	9999WXYZ0000 LF	

テキストファイル（空行削除オプションなし）：入力できない

テキストファイル（空行削除オプションあり）：入力できる（レコード3は削除される）

CSV ファイル（空行削除オプションなし）：入力できる

CSV ファイル（空行削除オプションあり）：入力できる（レコード3は削除される）

簡易コピー機能を使用：入力できる

#### 注※ 2

簡易コピー機能を使用する場合、レコード件数の制限はありません。

#### 注※ 3

固定長順編成ファイル、テキストファイル、および CSV ファイルは、レコード件数が 0 件の場合、ファイルの最小サイズは 0 バイトです。

可変長順編成ファイル、相対編成ファイルおよび索引編成ファイルは、レコード件数が 0 件の場合でも、ファイル情報が格納されるので、ファイルサイズが 0 バイトになることはありません。

#### 注※ 4

UNIX の場合、ラージファイル（2GB 以上）を使用するときは、ファイルシステムがラージファイルに対応していることを確認してください。

# 4

## SORT EE の使い方

SORT EE の機能を使うには、コマンドを入力します。コマンドを使って、処理したいデータが収められているファイルを入力ファイルに指定し、処理結果をどのファイルに出力するのかを指定します。また、キーや項目の属性や位置などを指定します。

SORT EE のコマンドは、1 コマンドで基本機能のソート機能、マージ機能、コピー機能と、レコード処理機能の選択機能、集約機能、編集機能、ファイル分割機能、ファイル連結機能を組み合わせて実行できます。

この章では、SORT EE で使用するコマンドについて説明します。

---

### 4.1 機能の実行順序と実行形態

---

### 4.2 exsort コマンドの形式

---

### 4.3 コマンドの表記規則

---

### 4.4 exsort パラメタファイルの表記規則

---

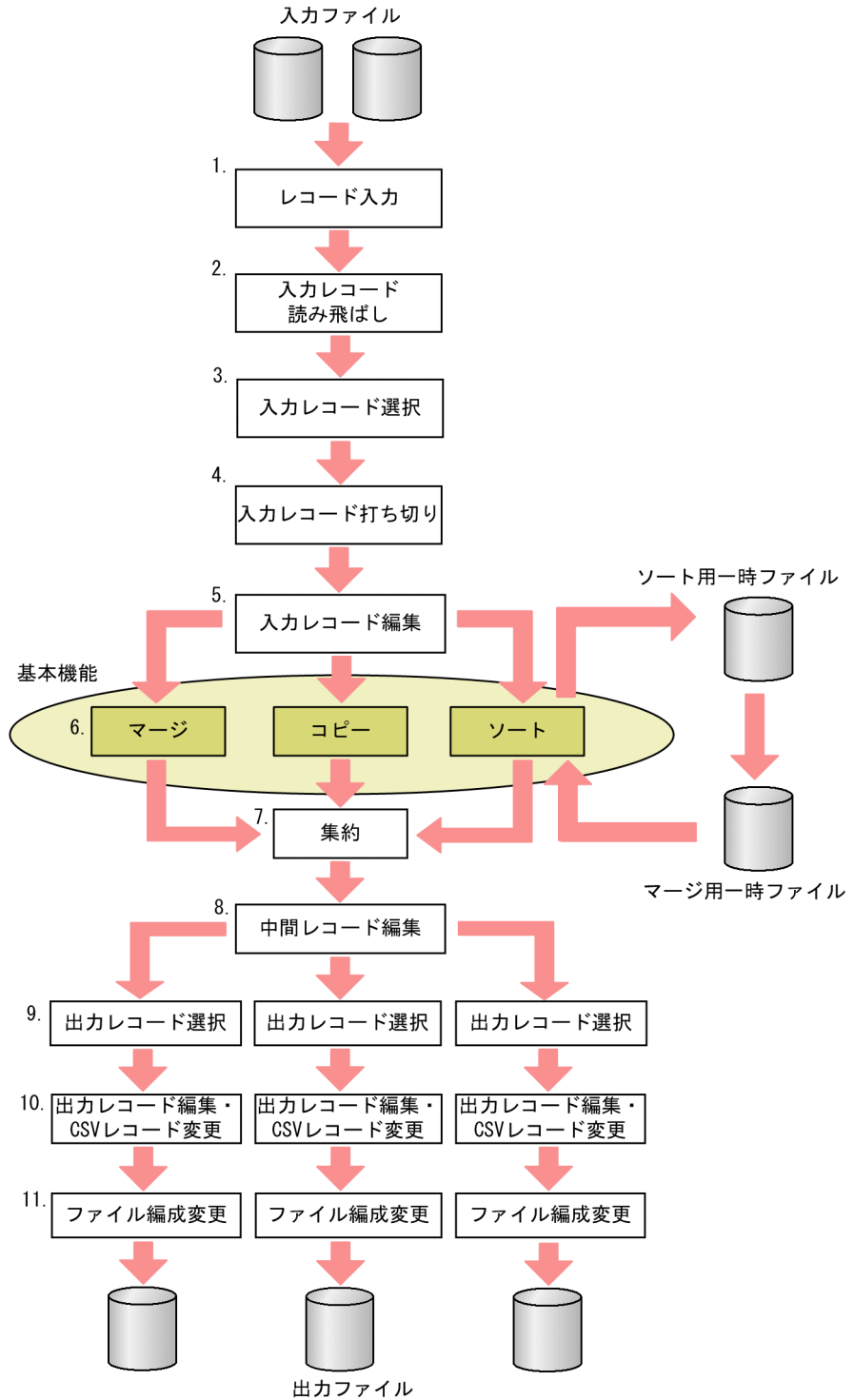
## 4.1 機能の実行順序と実行形態

---

### 4.1.1 機能の実行順序

基本機能（ファイル分割・連結なし）、ファイル分割機能、およびファイル連結機能を使用した場合の実行順序を次の図および表に示します。

図 4-1 機能の実行順序（ファイル分割・連結なし）



4. SORT EE の使い方

図 4-2 機能の実行順序（ファイル分割機能使用時）

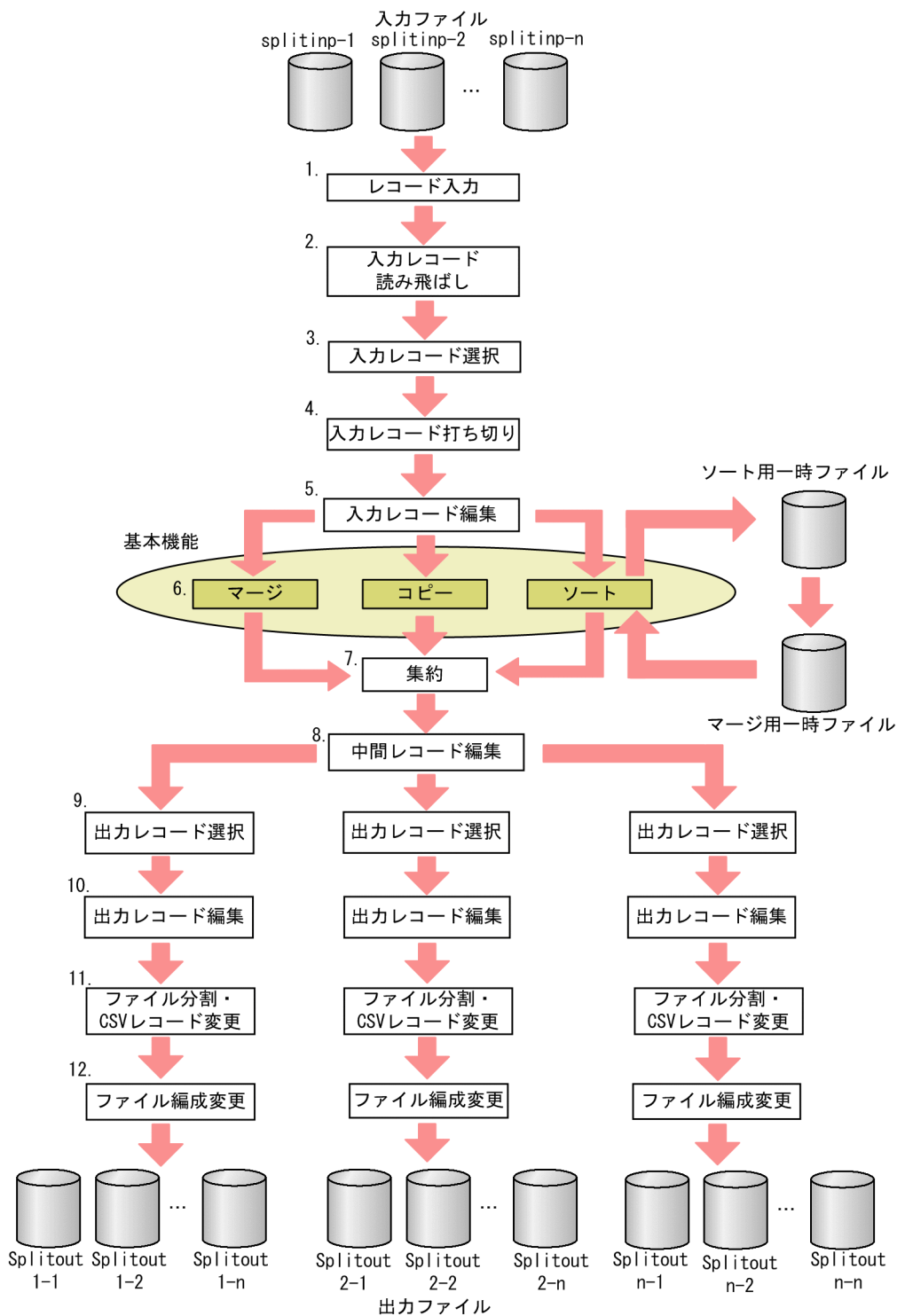
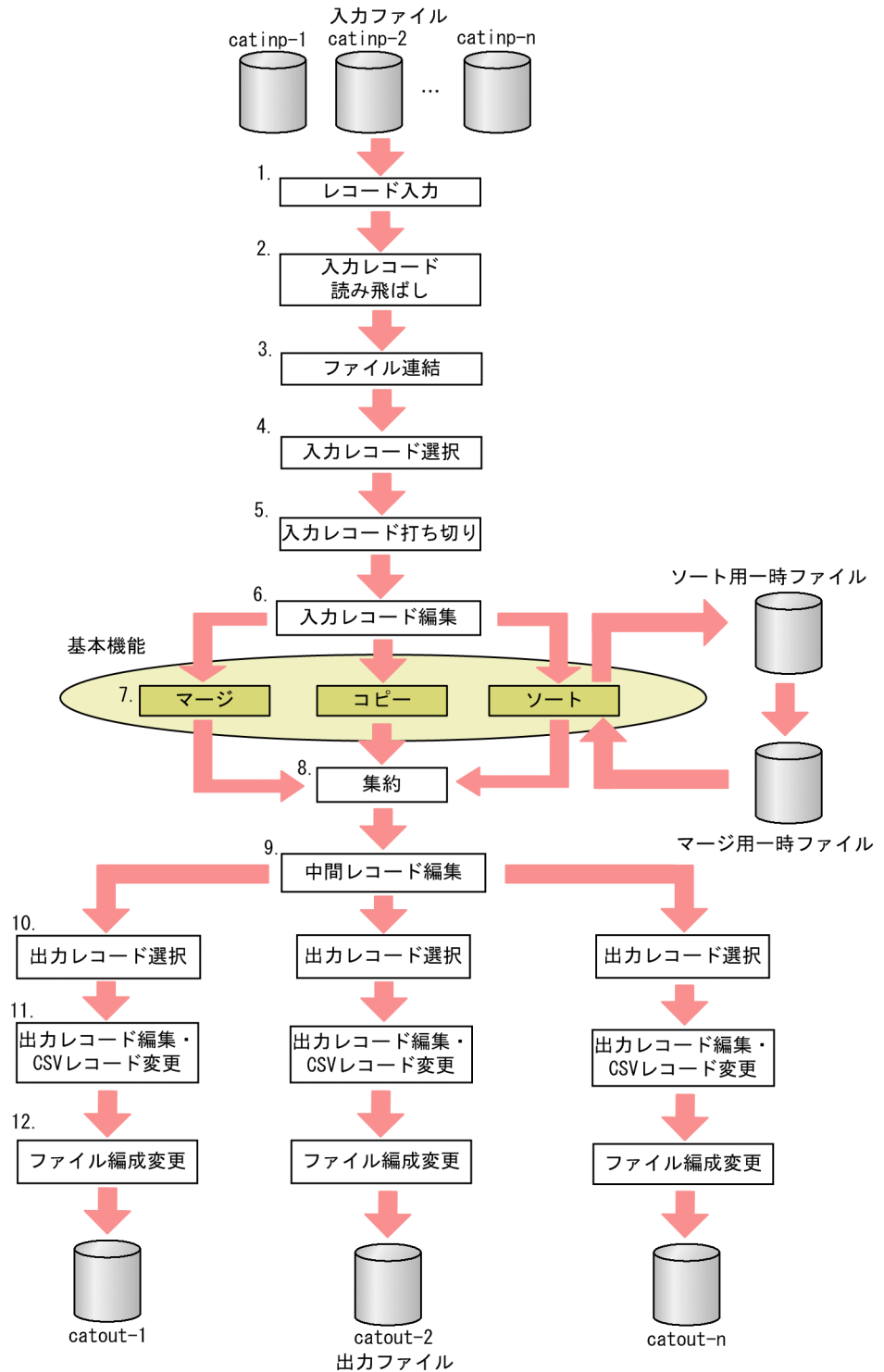


図 4-3 機能の実行順序（ファイル連結機能使用時）



#### 4. SORT EE の使い方

表 4-1 機能の実行順序

実行順序			機能	説明
ファイル 分割・連 結なし	ファイル 分割	ファイル 連結		
1	1	1	レコード入力	(レコード処理機能) <ul style="list-style-type: none"> <li>入力ファイルのエンディアンオプション (INPBIG, INPLTL)</li> <li>連続区切り文字オプション (INPCONSEPA)</li> <li>入力ファイル編成 (fileorg オペランド)</li> <li>入力レコード長 (reclen オペランド)</li> <li>改行文字 (recsepa オペランド)</li> </ul>
2	2	2	入力レコード読み飛ばし	(レコード処理機能) 指定されたレコード範囲で入力する。 <ul style="list-style-type: none"> <li>入力レコードの読み飛ばし (skiprec オペランド)</li> <li>入力レコードの開始位置 (startrec オペランド)</li> </ul>
—	—	3	ファイル連結	(レコード処理機能) 総称ファイル名または指定ディレクトリ下のファイルを連結する。
3	3	4	入力レコード選択	(レコード処理機能) ソート、マージ、またはコピーの対象とする入力レコードを選択する。
4	4	5	入力レコード打ち切り	(レコード処理機能) 指定されたレコード範囲で入力する。 <ul style="list-style-type: none"> <li>入力レコードの打ち切り (stopaft オペランド)</li> </ul>
5	5	6	入力レコード編集	(レコード処理機能) ソート、マージ、またはコピーの対象レコードを編集する。
6	6	7	<ul style="list-style-type: none"> <li>ソート</li> <li>マージ</li> <li>コピー</li> </ul>	(基本機能) ソート、マージ、またはコピーのうち、どれかを実行する。
7	7	8	集約	(レコード処理機能) ソート、マージ、またはコピーした結果に対して集約する。
8	8	9	中間レコード編集	(レコード処理機能) ソート、マージ、またはコピーした結果のレコードを編集する。
9	9	10	出力レコード選択	(レコード処理機能) 出力ファイル別に出力レコードを選択する。



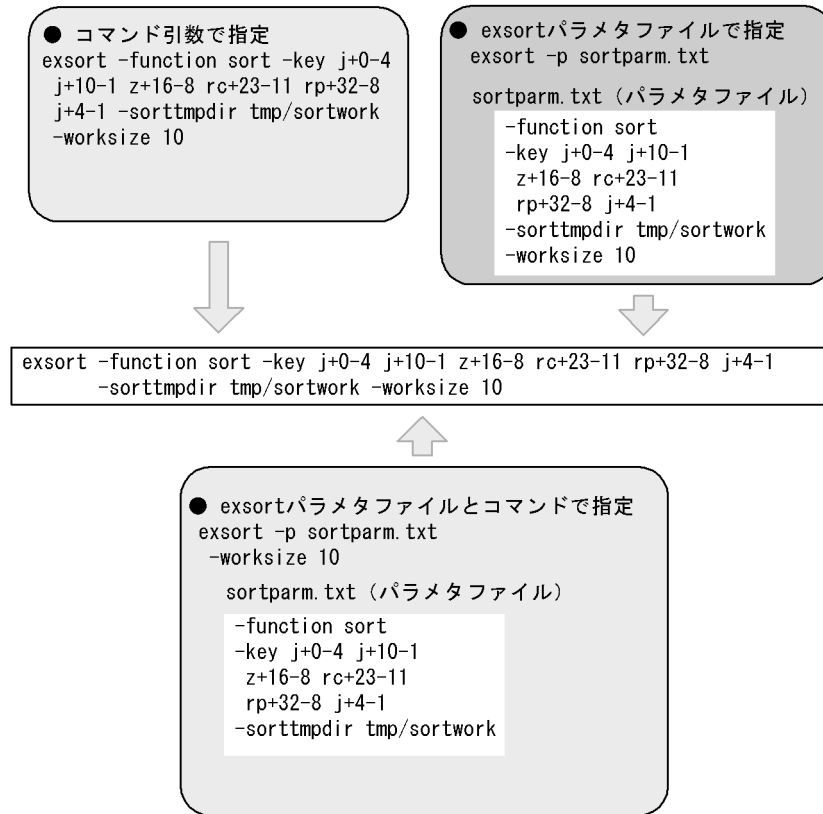
実行順序			機能	説明
ファイル 分割・連 結なし	ファイル 分割	ファイル 連結		
10	10	11	出力レコード編集	(レコード処理機能) 出力ファイル別に出力レコードを編集する。
			CSV レコード変更	(レコード処理機能) 出力ファイル別に出力 CSV レコード形式を変更する。 <ul style="list-style-type: none"> <li>• 連続区切り文字オプション (OUTCONSEPA)</li> <li>• デリミタ変更 (cellsepa オペランド)</li> <li>• 引用符変更 (quottype オペランド)</li> <li>• セル数調整 (cellnum オペランド)</li> </ul>
—	11	—	ファイル分割	(レコード処理機能) 入力ファイルを指定された分割基準を基に分割し、出力ファイルへ出力する。 <ul style="list-style-type: none"> <li>• 均等分割 (-splitfile パラメタ)</li> <li>• 定数分割 (-splitdata パラメタ)</li> <li>• キー分割 (-splitkey パラメタ)</li> </ul>
11	12	12	ファイル編成変更	(レコード処理機能) 出力ファイルへ出力する際に、出力ファイル別にファイル編成を変更する。 <ul style="list-style-type: none"> <li>• 出力ファイルのエンディアンオプション (OUTBIG, OUTLTL)</li> <li>• 出力ファイル編成 (fileorg オペランド)</li> <li>• 出力レコード長 (reclen, minlen オペランド)</li> <li>• 改行文字変更 (recsepa オペランド)</li> </ul>

### 4.1.2 機能の実行形態

exsort コマンドは、コマンド引数を直接指定する方法以外に、引数を指定したパラメタファイルを指定することができます。また、パラメタファイルとコマンド引数を組み合わせて指定することもできます。

#### 4. SORT EE の使い方

図 4-4 機能の実行形態



## 4.2 exsort コマンドの形式

---

SORT EE のコマンド名を次に示します。

- exsort (32 ビット版 SORT EE でコマンドを使う場合)
- exsort64 (64 ビット版 SORT EE でコマンドを使う場合) ※

注※

使用している SORT EE が 64 ビット版対応の場合、「exsort コマンド」を「exsort64 コマンド」と読み替えてください。

exsort コマンドの形式を機能ごとに示します。指定方法の詳細は、「5.1 パラメタおよびオペランドの組み合わせ」の表 5-1 を参照してください。

## 4.2.1 ソート機能

## (1) exsort コマンド形式

```

exsort*
  [Δ-pΔparameter-file-name ]
  [Δ-functionΔsort]
  [Δ-locatajst]
  [Δ-cmdoptΔcommand-option [:command-option ... ] ]
  [Δ { -keyΔkey-definition [Δkey-definition ... ]
    | -csvkeyΔcsv-key-definition [Δcsv-key-definition ... ] } ]
  [Δ { -sum [Δfield-definition [Δfield-definition ... ] ]
    | -csvsum [Δcsv-field-definition [Δcsv-field-definition ... ] ] } ]
  [Δ-inpfile
    [Δfileorg=organization-type ]
    [Δfile=file-name [Δfile-name ... ] ]
    [Δreclen=record-length]
    [Δ {skiprec=skip-record-number | startrec=start-record-number} ]
    [Δstopaft=stop-record-number ]
    [Δrecsepa=record-separator-type]
    [Δcellsepa=" {TAB | 'cell-separator'}
      [Δ {TAB | 'cell-separator'} ... ] " ]
    [Δquotttype=quotation-type]
    [Δcurrency=currency-symbol]
    [Δround=option-code]
    [Δnullseq=option-code]
    [Δcodetype=encode]
    [Δ { reorg="reorganization-format [Δreorganization-format ... ] "
      | csvreorg="csv-reorganization-format
        [Δcsv-reorganization-format ... ] " } ]
    [Δ { {include | omit} ="condition-expression"
      | {csvinclude | csvomit} ="csv-condition-expression" } ] ]
  [Δ { -reorgΔ"reorganization-format [Δreorganization-format ... ] "
    | -csvreorgΔ"csv-reorganization-format
      [Δcsv-reorganization-format ... ] " } ]
  [Δ-outfile
    [Δfileorg=organization-type]
    [Δfile=file-name [Δfile-name ... ] ]
    [Δreclen=record-length]
    [Δminlen=minimum-record-length]
    [Δrecsepa=record-separator-type]
    [Δ { cellsepa=" {TAB | 'cell-separator'} "
      | cellsepa=" {TAB | 'cell-separator-in_1'}
        Δ {TAB | 'cell-separator-out_1'}
          Δ {TAB | 'cell-separator-in_2'}
            Δ {TAB | 'cell-separator-out_2'} ... } " } ]
    [Δcellnum= {ADJUST | cell-number} ]
    [Δquotttype=option-code]
    [Δpadding= {X | Z | "C'padding-character'" | "X'hex-digit'" } ]
    [Δ { reorg="reorganization-format [Δreorganization-format ... ] "
      | csvreorg="csv-reorganization-format
        [Δcsv-reorganization-format ... ] " } ]
    [Δ { {include | omit} ="condition-expression"
      | {csvinclude | csvomit} ="csv-condition-expression" } ] ]
  [Δ-outfile ... ]
  [Δ-sorttmpdirΔtemporary-directory1]
  [Δ-mergetmpdirΔtemporary-directory2]
  [Δ-worksizeΔ { buffer-size [unit]
    | auto [,max-buffer-size [unit] ]
    | incore [,max-buffer-size [unit] ] } ]
  [Δ-dateoptΔbase-year]
  [Δ-altseqΔcode=convert-code [Δconvert-code ... ] ]

```

```
[Δ-iosize { Δiosize
| [Δinpiosize=input-file-iosize]
  [Δoutiosize=output-file-iosize]
  [Δsorttmpiosize=sort-temporary-file-iosize]
  [Δmergetmpiosize=merge-temporary-file-iosize] } ]
```

(凡例)

*Δ* : 1 個以上のスペース (空白), またはタブを示す

[ ] : 省略できることを示す

… : 繰り返して指定できることを示す

{ } : この記号で囲まれている項目のうち 1 組の項目を選択することを示す

| : { } で囲んだ項目の区切りを示す

*斜体文字*: 値を任意に指定できることを示す

注※

使用している SORT EE が 64 ビット版対応の場合, コマンド名は「*exsort64*」となります。

#### 4. SORT EE の使い方

##### (2) exsort パラメタファイル形式

```
[<code=character-code>] ※
[-function△sort]
[-locatajst]
[-cmdopt△command-option [:command-option ... ] ]
[ { -key△key-definition [△key-definition ... ] } ]
  | -csvkey△csv-key-definition [△csv-key-definition ... ] } ]
[ { -sum [△field-definition [△field-definition ... ] ]
  | -csvsum [△csv-field-definition [△csv-field-definition ... ] ] } ] ]
[-inpfiler]
  [△fileorg=organization-type]
  [△file=file-name [△file-name ... ] ]
  [△reclen=record-length ]
  [△ { skiprec=skip-record-number | startrec=start-record-number } ]
  [△stopaft=stop-record-number]
  [△recsepa=record-separator-type]
  [△cellsepa= {TAB | 'cell-separator'} [△ {TAB | 'cell-separator'} ... ] ]
  [△quottype=quotation-type]
  [△currency=currency-symbol]
  [△round=option-code]
  [△nullseq=option-code]
  [△codetype=encode]
  [△ { reorg=reorganization-format [△reorganization-format ... ]
    | csvreorg=csv-reorganization-format [△csv-reorganization-format ... ] } ]
  [△ { {include | omit} =condition-expression
    | {csvinclude | csvomit} =csv-condition-expression } ] ]
[ { -reorg△reorganization-format [△reorganization-format ... ]
  | -csvreorg△csv-reorganization-format [△csv-reorganization-format ... ] } ] ]
[-outfile]
  [△fileorg=organization-type ]
  [△file=file-name [△file-name ... ] ]
  [△reclen=record-length]
  [△minlen=minimum-record-length]
  [△recsepa=record-separator-type]
  [△ { cellsepa= {TAB | 'cell-separator'}
    | cellsepa= {TAB | 'cell-separator-in_1'}
      △ {TAB | 'cell-separator-out_1'}
      [△ {TAB | 'cell-separator-in_2'}
        △ {TAB | 'cell-separator-out_2'} ... ] } ] ]
  [△cellnum= {ADJUST | cell-number} ]
  [△quottype=option-code]
  [△padding= {X | Z | C'padding-character' | X'hex-digit'} ]
  [△ { reorg=reorganization-format [△reorganization-format ... ]
    | csvreorg=csv-reorganization-format [△csv-reorganization-format ... ] } ]
  [△ { {include | omit} =condition-expression
    | {csvinclude | csvomit} =csv-condition-expression } ] ]
[-outfile ... ]
[-sorttmpdir△temporary-directory1]
[-mergetmpdir△temporary-directory2]
[-worksize△ {buffer-size [unit ]
  | auto [,max-buffer-size [unit ] ]
  | incore [,max-buffer-size [unit ] ] } ] ]
[-dateopt△base-year]
[-altseq△code=convert-code [△convert-code ... ] ]
[-iosize { △iosize
  | [△inpioffset=input-file-iosize]
  [△outpioffset=output-file-iosize]
  [△sorttmpiosize=sort-temporary-file-iosize]
  [△mergetmpiosize=merge-temporary-file-iosize] } ] ]
```

(凡例)

- △ : 1 個以上のスペース (空白), またはタブを示す
- [ ] : 省略できることを示す
- … : 繰り返して指定できることを示す
- { } : この記号で囲まれている項目のうち 1 組の項目を選択することを示す
- | : { } で囲んだ項目の区切りを示す
- 斜体文字* : 値を任意に指定できることを示す

**注※**

**exsort** コマンドのパラメタ・オペランドではありません。パラメタファイルを記述した文字コードの種類を指定します。

## 4.2.2 マージ機能

## (1) exsort コマンド形式

```

exsort*
  [Δ-pΔparameter-file-name]
  [Δ-functionΔmerge]
  [Δ-locatajst]
  [Δ-cmdoptΔcommand-option [:command-option ... ] ]
  [Δ { -keyΔkey-definition [Δkey-definition ... ]
    | -csvkeyΔcsv-key-definition [Δcsv-key-definition ... ] } ]
  [Δ { -sum [Δfield-definition [Δfield-definition ... ] ]
    | -csvsum [Δcsv-field-definition [Δcsv-field-definition ... ] ] } ]
  [Δ-inpfile
    [Δfileorg=organization-type ]
    [Δfile=file-name [Δfile-name ... ] ]
    [Δreclen=record-length]
    [Δstartrec=start-record-number]
    [Δstopaft=stop-record-number ]
    [Δrecsepa=record-separator-type]
    [Δcellsepa=" {TAB | 'cell-separator'}
      [Δ {TAB | 'cell-separator'} ... ] " ]
    [Δquottype=quotation-type]
    [Δcurrency=currency-symbol]
    [Δround=option-code]
    [Δnullseq=option-code]
    [Δcodetype=encode]
    [Δ { reorg="reorganization-format [Δreorganization-format ... ] "
      | csvreorg="csv-reorganization-format
        [Δcsv-reorganization-format ... ] " } ]
    [Δ { {include | omit} ="condition-expression"
      | {csvinclude | csvomit} ="csv-condition-expression" } ] ]
  [Δ { -reorgΔ"reorganization-format [Δreorganization-format ... ] "
    | -csvreorgΔ"csv-reorganization-format
      [Δcsv-reorganization-format ... ] " } ]
  [Δ-outfile
    [Δfileorg=organization-type]
    [Δfile=file-name [Δfile-name ... ] ]
    [Δreclen=record-length]
    [Δminlen=minimum-record-length]
    [Δrecsepa=record-separator-type]
    [Δ { cellsepa=" {TAB | 'cell-separator'} "
      | cellsepa=" {TAB | 'cell-separator-in_1'}
        Δ {TAB | 'cell-separator-out_1'}
          Δ {TAB | 'cell-separator-in_2'}
            Δ {TAB | 'cell-separator-out_2'} ... } " ] ]
    [Δcellnum= {ADJUST | cell-number} ]
    [Δquottype=option-code]
    [Δpadding= {X | Z | "C'padding-character'" | "X'hex-digit'" } ]
    [Δ { reorg="reorganization-format [Δreorganization-format ... ] "
      | csvreorg="csv-reorganization-format
        [Δcsv-reorganization-format ... ] " } ]
    [Δ { {include | omit} ="condition-expression"
      | {csvinclude | csvomit} ="csv-condition-expression" } ] ]
  [Δ-outfile ... ]
  [Δ-dateoptΔbase-year]
  [Δ-altseqΔcode=convert-code [Δconvert-code ... ] ]
  [Δ-iosize { Δiosize
    | [Δinpiosize=input-file-iosize]
      [Δoutiosize=output-file-iosize] } ]

```

(凡例)



- △ : 1 個以上のスペース (空白), またはタブを示す
- [ ] : 省略できることを示す
- … : 繰り返して指定できることを示す
- { } : この記号で囲まれている項目のうち 1 組の項目を選択することを示す
- | : { } で囲んだ項目の区切りを示す
- 斜体文字* : 値を任意に指定できることを示す

注※

使用している SORT EE が 64 ビット版対応の場合, コマンド名は「`exsort64`」となります。

#### 4. SORT EE の使い方

##### (2) exsort パラメタファイル形式

```
[<code=character-code>] ※
[-function△merge ]
[-locatajst ]
[-cmdopt△command-option [ :command-option ... ] ]
[ { -key△key-definition [△key-definition ... ]
  | -csvkey△csv-key-definition [△csv-key-definition ... ] } ]
[ { -sum [△field-definition [△field-definition ... ] ]
  | -csvsum [△csv-field-definition [△csv-field-definition ... ] ] } ]
[-infile
 [△fileorg=organization-type]
 [△file=file-name [△file-name ... ] ]
 [△reclen=record-length ]
 [△startrec=start-record-number]
 [△stopaft=stop-record-number]
 [△recsepa=record-separator-type]
 [△cellsepa= {TAB | 'cell-separator'} [△ {TAB | 'cell-separator'} ... ] ]
 [△quottype=quotation-type]
 [△currency=currency-symbol]
 [△round=option-code]
 [△nullseq=option-code]
 [△codetype=encode]
 [△ { reorg=reorganization-format [△reorganization-format ... ]
  | csvreorg=csv-reorganization-format [△csv-reorganization-format ... ] } ]
 [△ { {include | omit} =condition-expression
  | {csvinclude | csvomit} =csv-condition-expression } ] ]
[ { -reorg△reorganization-format [△reorganization-format ... ]
  | -csvreorg△csv-reorganization-format [△csv-reorganization-format ... ] } ]
[-outfile
 [△fileorg=organization-type ]
 [△file=file-name [△file-name ... ] ]
 [△reclen=record-length]
 [△minlen=minimum-record-length]
 [△recsepa=record-separator-type]
 [△ { cellsepa= {TAB | 'cell-separator'}
  | cellsepa= {TAB | 'cell-separator-in_1'}
  Δ {TAB | 'cell-separator-out_1'}
  [△ {TAB | 'cell-separator-in_2'}
  Δ {TAB | 'cell-separator-out_2'} ... } ] ]
 [△cellnum= {ADJUST | cell-number} ]
 [△quottype=option-code]
 [△padding= {X | Z | C'padding-character' | X'hex-digit'} ]
 [△ { reorg=reorganization-format [△reorganization-format ... ]
  | csvreorg=csv-reorganization-format [△csv-reorganization-format ... ] } ]
 [△ { {include | omit} =condition-expression
  | {csvinclude | csvomit} =csv-condition-expression } ] ]
[-outfile ... ]
[-dateopt△base-year]
[-altseq△code=convert-code [△convert-code ... ] ]
[-iosize { Δiosize
  | [△inpiosize=input-file-iosize]
  [△outiosize=output-file-iosize] } ]
```

(凡例)

- △ : 1個以上のスペース (空白), またはタブを示す
- [ ] : 省略できることを示す
- … : 繰り返して指定できることを示す
- { } : この記号で囲まれている項目のうち1組の項目を選択することを示す
- | : { } で囲んだ項目の区切りを示す

*斜体文字*: 値を任意に指定できることを示す

注※

**exsort** コマンドのパラメタ・オペランドではありません。パラメタファイルを記述した文字コードの種類を指定します。

## 4.2.3 コピー機能

## (1) exsort コマンド形式

```

exsort*
  [Δ-pΔparameter-file-name]
  [Δ-functionΔcopy]
  [Δ-locatajst]
  [Δ-cmdoptΔcommand-option [:command-option ... ] ]
  [Δ { -keyΔkey-definition [Δkey-definition ... ]
    | -csvkeyΔcsv-key-definition [Δcsv-key-definition ... ] } ]
  [Δ { -sum [Δfield-definition [Δfield-definition ... ] ]
    | -csvsum [Δcsv-field-definition [Δcsv-field-definition ... ] ] } ]
  [Δ-inpfile
    [Δfileorg=organization-type ]
    [Δfile=file-name [Δfile-name ... ] ]
    [Δreclen=record-length]
    [Δ { skiprec=skip-record-number | startrec=start-record-number } ]
    [Δstopaft=stop-record-number ]
    [Δrecsepa=record-separator-type]
    [Δcellsepa=" {TAB | 'cell-separator'} [Δ {TAB | 'cell-separator'} ... ] " ]
    [Δquottype=quotation-type]
    [Δcurrency=currency-symbol]
    [Δround=option-code]
    [Δnullseq=option-code]
    [Δcodetype=encode]
    [Δ { reorg="reorganization-format [Δreorganization-format ... ] "
      | csvreorg="csv-reorganization-format
        [Δcsv-reorganization-format... ] " } ]
    [Δ { {include | omit} ="condition-expression"
      | {csvinclude | csvomit} ="csv-condition-expression" } ] ]
  [Δ { -reorgΔ"reorganization-format [Δreorganization-format ... ] "
    | -csvreorgΔ"csv-reorganization-format
      [Δcsv-reorganization-format ... ] " } ]
  [Δ-outfile
    [Δfileorg=organization-type]
    [Δfile=file-name [Δfile-name ... ] ]
    [Δreclen=record-length]
    [Δminlen=minimum-record-length]
    [Δrecsepa=record-separator-type]
    [Δ { cellsepa=" {TAB | 'cell-separator'} "
      | cellsepa=" {TAB | 'cell-separator-in_1'}
        Δ {TAB | 'cell-separator-out_1'}
          [Δ {TAB | 'cell-separator-in_2'}
            Δ {TAB | 'cell-separator-out_2'} ... ] " } ]
    [Δcellnum= {ADJUST | cell-number} ]
    [Δquottype=option-code]
    [Δpadding= {X | Z | "C'padding-character'" | "X'hex-digit'"} ]
    [Δ { reorg="reorganization-format [ Δreorganization-format ... ] "
      | csvreorg="csv-reorganization-format
        [Δcsv-reorganization-format ... ] " } ]
    [Δ { {include | omit} ="condition-expression"
      | {csvinclude | csvomit} ="csv-condition-expression" } ] ]
  [Δ-outfile ... ]
  [Δ-dateoptΔbase-year]
  [Δ-altseqΔcode=convert-code [Δconvert-code ... ] ]
  [Δ-iosize { Δiosize
    | [Δinpiosize=input-file-iosize]
      [Δoutiosize=output-file-iosize] } ]

```

(凡例)

- △ : 1 個以上のスペース (空白), またはタブを示す
- [ ] : 省略できることを示す
- … : 繰り返して指定できることを示す
- { } : この記号で囲まれている項目のうち 1 組の項目を選択することを示す
- | : { } で囲んだ項目の区切りを示す
- 斜体文字* : 値を任意に指定できることを示す

注※

使用している SORT EE が 64 ビット版対応の場合, コマンド名は「`exsort64`」となります。

#### 4. SORT EE の使い方

##### (2) exsort パラメタファイル形式

```
[<code=character-code>] ※
[-function△copy]
[-locatajst]
[-cmdopt△command-option [:command-option ... ] ]
[ { -key△key-definition [△key-definition ... ] } ]
  | -csvkey△csv-key-definition [△csv-key-definition ... ] } ]
[ { -sum [△field-definition [△field-definition ... ] ]
  | -csvsum [△csv-field-definition [△csv-field-definition ... ] ] } ]
[-infile
 [△fileorg=organization-type]
 [△file=file-name [△file-name ... ] ]
 [△reclen=record-length ]
 [△ {skiprec=skip-record-number | startrec=start-record-number} ]
 [△stopaft=stop-record-number]
 [△recsepa=record-separator-type]
 [△cellsepa= {TAB | 'cell-separator'} [△ {TAB | 'cell-separator'} ... ] ]
 [△quottype=quotation-type]
 [△currency=currency-symbol]
 [△round=option-code]
 [△nullseq=option-code]
 [△codetype=encode]
 [△ { reorg=reorganization-format [△reorganization-format ... ]
  | csvreorg=csv-reorganization-format [△csv-reorganization-format ... ] } ]
 [△ { {include | omit} =condition-expression
  | {csvinclude | csvomit} =csv-condition-expression} ] ]
[ { -reorg△reorganization-format [△reorganization-format ... ]
  | -csvreorg△csv-reorganization-format [△csv-reorganization-format ... ] } ]
[-outfile
 [△fileorg=organization-type ]
 [△file=file-name [△file-name ... ] ]
 [△reclen=record-length]
 [△minlen=minimum-record-length]
 [△recsepa=record-separator-type]
 [△ { cellsepa= {TAB | 'cell-separator'}
  | cellsepa= {TAB | 'cell-separator-in_1'}
  Δ {TAB | 'cell-separator-out_1'}
  [△ {TAB | 'cell-separator-in_2'}
  Δ {TAB | 'cell-separator-out_2'} ... ] } ]
 [△cellnum= {ADJUST | cell-number} ]
 [△quottype=option-code]
 [△padding= {X | Z | C'padding-character' | X'hex-digit'} ]
 [△ {reorg=reorganization-format [△reorganization-format ... ]
  | csvreorg=csv-reorganization-format [△csv-reorganization-format ... ] } ]
 [△ { {include | omit} =condition-expression
  | {csvinclude | csvomit} =csv-condition-expression} ] ]
[-outfile ... ]
[-dateopt△base-year]
[-altseq△code=convert-code [△convert-code ... ] ]
[-iosize { Δiosize
  | [△inpiosize=input-file-iosize]
  [△outiosize=output-file-iosize] } ]
```

(凡例)

- △ : 1個以上のスペース (空白), またはタブを示す
- [ ] : 省略できることを示す
- … : 繰り返して指定できることを示す
- { } : この記号で囲まれている項目のうち1組の項目を選択することを示す
- | : { } で囲んだ項目の区切りを示す

*斜体文字*: 値を任意に指定できることを示す

注※

`exsort` コマンドのパラメタ・オペランドではありません。パラメタファイルを記述した文字コードの種類を指定します。

## 4.2.4 ファイル分割機能

## (1) exsort コマンド形式

```

exsort*
  [Δ-pΔparameter-file-name]
  [Δ-functionΔ {sort | merge | copy} ]
  [Δ { -splitfile
    [Δfileno=split-file-number]
    [Δmethod= {reccnt | filesize} ]
    | -splitdata
    [Δ {recno=split-record-number | spno=split-file-size} ]
    [Δ { nosplitdata=no-split-data-definition
      [Δno-split-data-definition
        | nosplitcsvdata=no-split-csv-data-definition
          [Δno-split-csv-data-definition ... ] } ] ]
    | -splitkey
    [Δ { key=split-key-definition [Δsplit-key-definition ... ]
      | csvkey=split-csv-key-definition
        [Δsplit-csv-key-definition ... ] } ]
    [Δ {recno=split-record-number | spno=split-file-size} ] } ] ]
  [Δ-splitinp
    [Δfileorg=organization-type]
    [Δfile=file-name [Δfile-name ... ] ]
    [Δreclen=record-length]
    [Δ {skiprec=skip-record-number | startrec=start-record-number} ]
    [Δstopaft=stop-record-number]
    [Δrecsepa=record-separator-type]
    [Δcellsepa=" {TAB | 'cell-separator'}
      [Δ {TAB | 'cell-separator'} ... ] " ]
    [Δquottype=quotation-type]
    [Δcurrency=currency-symbol]
    [Δround=option-code]
    [Δnullseq=option-code]
    [Δcodetype=encode]
    [Δ { reorg="reorganization-format [Δreorganization-format ... ] "
      | csvreorg="csv-reorganization-format
        [Δcsv-reorganization-format ... ] " } ] ]
    [Δ { {include | omit} ="condition-expression"
      | {csvinclude | csvomit} ="csv-condition-expression" } ] ]
  [Δ-splitout
    [Δfileorg=organization-type]
    [Δ {suffix | prefix} = {output-generic-name | output-directory-name} ]
    [Δreclen=record-length]
    [Δminlen=minimum-record-length]
    [Δrecsepa=record-separator-type]
    [Δ { cellsepa=" {TAB | 'cell-separator'} "
      | cellsepa=" {TAB | 'cell-separator-in_1'}
        Δ {TAB | 'cell-separator-out_1'}
          Δ {TAB | 'cell-separator-in_2'}
            Δ {TAB | 'cell-separator-out_2'} ... } " ] ]
    [Δcellnum= {ADJUST | cell-number} ]
    [Δquottype=option-code]
    [Δpadding= {X | Z | "C'padding-character'" | "X'hex-digit'" } ]
    [Δ { reorg="reorganization-format [Δreorganization-format ... ] "
      | csvreorg="csv-reorganization-format
        [Δcsv-reorganization-format ... ] " } ] ]
    [Δ { {include | omit} ="condition-expression"
      | {csvinclude | csvomit} ="csv-condition-expression" } ] ]

```



```

[△-splitout ... ]
[△-splittmpdir△split-temporary-directory]
[△-locatajst]
[△-cmdopt△command-option [:command-option ... ] ]
[△ { -key△key-definition [△key-definition ... ]
  | -csvkey△csv-key-definition [△csv-key-definition ... ] } ]
[△ { -sum [△field-definition [△field-definition ... ] ]
  | -csvsum [△csv-field-definition [△csv-field-definition ... ] ] } ]
[△ { -reorg△"reorganization-format [△reorganization-format ... ] "
  | -csvreorg△"csv-reorganization-format
    [△csv-reorganization-format ... ] " } ]
[△-sorttmpdir△temporary-directory1]
[△-mergetmpdir△temporary-directory2]
[△-worksize△ { buffer-size [unit]
  | auto [,max-buffer-size [unit] ]
  | incore [,max-buffer-size [unit] ] } ]
[△-dateopt△base-year]
[△-altseq△code=convert-code [△convert-code ... ] ]
[△-iosize { △iosize
  | [△inpiosize=input-file-iosize]
  | [△outiosize=output-file-iosize]
  | [△sorttmpiosize=sort-temporary-file-iosize]
  | [△mergetmpiosize=merge-temporary-file-iosize]
  | [△splittmpiosize=split-temporary-file-iosize] } ]

```

## (凡例)

△ : 1 個以上のスペース (空白), またはタブを示す

[ ] : 省略できることを示す

... : 繰り返して指定できることを示す

{ } : この記号で囲まれている項目のうち 1 組の項目を選択することを示す

| : { } で囲んだ項目の区切りを示す

斜体文字: 値を任意に指定できることを示す

## 注※

使用している SORT EE が 64 ビット版対応の場合、コマンド名は「exsort64」となります。

#### 4. SORT EE の使い方

##### (2) exsort パラメタファイル形式

```
[<code=character-code>] ※
[-function△ {sort | merge | copy} ]
[ { -splitfile
  [△fileno=split-file-number]
  [△method= {recCNT | filesize} ]
  | -splitdata
  [△ {recno=split-record-number | spno=split-file-size} ]
  [△ { nosplitdata=no-split-data-definition
    [△no-split-data-definition ... ]
    | nosplitcsvdata=no-split-csv-data-definition
    [△no-split-csv-data-definition ... ] } ]
  | -splitkey
  [△ { key=split-key-definition [△split-key-definition ... ]
    | csvkey=split-csv-key-definition
    [△split-csv-key-definition ... ] } ]
  [△ {recno=split-record-number | spno=split-file-size} ] } ]
[-splitinp
  [△fileorg=organization-type]
  [△file=file-name [△file-name ... ] ]
  [△reclen=record-length]
  [△ {skiprec=skip-record-number | startrec=start-record-number} ]
  [△stopaft=stop-record-number]
  [△recsepa=record-separator-type]
  [△cellsepa= {TAB | 'cell-separator'} [△ {TAB | 'cell-separator'} ... ] ]
  [△quottype=quotation-type]
  [△currency=currency-symbol]
  [△round=option-code]
  [△nullseq=option-code]
  [△codetype=encode]
  [△ { reorg=reorganization-format [△reorganization-format ... ]
    | csvreorg=csv-reorganization-format [△csv-reorganization-format ... ] } ]
  [△ { {include | omit} =condition-expression
    | {csvinclude | csvomit} =csv-condition-expression} ] ]
[-splitout
  [△fileorg=organization-type]
  [△ {suffix | prefix} = {output-generic-name | output-directory-name} ]
  [△reclen=record-length]
  [△minlen=minimum-record-length]
  [△recsepa=record-separator-type]
  [△ { cellsepa= {TAB | 'cell-separator'}
    | cellsepa= {TAB | 'cell-separator-in_1'}
    △ {TAB | 'cell-separator-out_1'}
    [△ {TAB | 'cell-separator-in_2'}
    △ {TAB | 'cell-separator-out_2'} ... ] } ]
  [△cellnum= {ADJUST | cell-number} ]
  [△quottype=option-code]
  [△padding= {X | Z | C'padding-character' | X'hex-digit'} ]
  [△ { reorg=reorganization-format [△reorganization-format ... ]
    | csvreorg=csv-reorganization-format [△csv-reorganization-format ... ] } ]
  [△ { {include | omit} =condition-expression
    | {csvinclude | csvomit} =csv-condition-expression} ] ]
[-splitout ... ]
[-splittmpdir△split-temporary-directory]
[-locatajst]
[-cmdopt△command-option [:command-option ... ] ]
[ { -key△key-definition [△key-definition ... ]
  | -csvkey△csv-key-definition [△csv-key-definition ... ] } ]
[ { -sum [△field-definition [△field-definition ... ] ]
  | -csvsum [△csv-field-definition [△csv-field-definition ... ] ] } ]
[ { -reorg△reorganization-format [△reorganization-format ... ]
  | -csvreorg△csv-reorganization-format [△csv-reorganization-format ... ] } ]
```

```

[-sorttmpdir△temporary-directory1]
[-mergetmpdir△temporary-directory2]
[-worksize△ {buffer-size [unit ]
              | auto [,max-buffer-size [unit ] ]
              | incore [,max-buffer-size [unit ] ] } ]
[-dateopt△base-year]
[-altseq△code=convert-code [△convert-code ... ] ]
[-iosize {      △iosize
              | [△inpioffset=input-file-iosize]
                [△outpioffset=output-file-iosize]
                [△sorttmpiosize=sort-temporary-file-iosize]
                [△mergetmpiosize=merge-temporary-file-iosize]
                [△splittmpiosize=split-temporary-file-iosize] } ]

```

(凡例)

△ : 1 個以上のスペース (空白), またはタブを示す

[ ] : 省略できることを示す

… : 繰り返して指定できることを示す

{ } : この記号で囲まれている項目のうち 1 組の項目を選択することを示す

| : { } で囲んだ項目の区切りを示す

*斜体文字*: 値を任意に指定できることを示す

注※

**exsort** コマンドのパラメタ・オペランドではありません。パラメタファイルを記述した文字コードの種類を指定します。

## 4.2.5 ファイル連結機能

## (1) exsort コマンド形式

```

exsort*
[△-p△parameter-file-name]
[△-function△ {sort | merge | copy} ]
[△-catinp
  [△fileorg=organization-type]
  [△ { file=file-name [△file-name ... ]
    | {suffix | prefix} = {input-generic-name | input-directory-name}
      [△ {input-generic-name | input-directory-name} ... ]
    [△nameseq=input-order-code] } ]
  [△reclen=record-length]
  [△ {skiprec=skip-record-number | startrec=start-record-number} ]
  [△stopaft=stop-record-number]
  [△recsepa=record-separator-type]
  [△cellsepa=" {TAB | 'cell-separator'} [△ {TAB | 'cell-separator'} ... ] " ]
  [△quottype=quotation-type]
  [△currency=currency-symbol]
  [△round=option-code]
  [△nullseq=option-code]
  [△codetype=encode]
  [△ { reorg="reorganization-format [△reorganization-format ... ] "
    | csvreorg="csv-reorganization-format
      [△csv-reorganization-format ... ] " } ]
  [△ { {include | omit} ="condition-expression"
    | {csvinclude | csvomit} ="csv-condition-expression" } ] ]
[△-catout
  [△fileorg=organization-type]
  [△file=file-name [△file-name ... ] ]
  [△reclen=record-length]
  [△minlen=minimum-record-length]
  [△recsepa=record-separator-type]
  [△ { cellsepa=" {TAB | 'cell-separator'} "
    | cellsepa=" {TAB | 'cell-separator-in_1'}
      △ {TAB | 'cell-separator-out_1'}
      [△ {TAB | 'cell-separator-in_2'}
        △ {TAB | 'cell-separator-out_2'} ... ] " } ]
  [△cellnum= {ADJUST | cell-number} ]
  [△quottype=option-code]
  [△padding= {X | Z | "C'padding-character'" | "X'hex-digit'" } ]
  [△ { reorg="reorganization-format [△reorganization-format ... ] "
    | csvreorg="csv-reorganization-format
      [△csv-reorganization-format ... ] " } ]
  [△ { {include | omit} ="condition-expression"
    | {csvinclude | csvomit} ="csv-condition-expression" } ] ]
[△-catout ... ]
[△-locatajst]
[△-cmdopt△command-option [:command-option ... ] ]
[△ { -key△key-definition [△key-definition ... ]
  | -csvkey△csv-key-definition [△csv-key-definition ... ] } ]
[△ { -sum [△field-definition [△field-definition ... ] ]
  | -csvsum [△csv-field-definition [△csv-field-definition ... ] ] } ] ]
[△ { -reorg△"reorganization-format [△reorganization-format ... ] "
  | -csvreorg△"csv-reorganization-format
    [△csv-reorganization-format ... ] " } ]
[△-sorttmpdir△temporary-directory1]
[△-mergetmpdir△temporary-directory2]
[△-worksize△ { buffer-size [unit]
  | auto [,max-buffer-size [unit] ]
  | incore [,max-buffer-size [unit] ] } ]

```

```
[△-dateopt△base-year]
[△-altseq△code=convert-code [△convert-code ... ] ]
[△-iosize {   △iosize
              | [△inpiosize=input-file-iosize]
                [△outiosize=output-file-iosize]
                [△sorttmpiosize=sort-temporary-file-iosize]
                [△mergetmpiosize=merge-temporary-file-iosize] } ]
```

(凡例)

△ : 1 個以上のスペース (空白), またはタブを示す

[ ] : 省略できることを示す

… : 繰り返して指定できることを示す

{ } : この記号で囲まれている項目のうち 1 組の項目を選択することを示す

| : { } で囲んだ項目の区切りを示す

*斜体文字* : 値を任意に指定できることを示す

注※

使用している SORT EE が 64 ビット版対応の場合, コマンド名は「*exsort64*」となります。

#### 4. SORT EE の使い方

##### (2) exsort パラメタファイル形式

```
[<code=character-code>] ※
[-function△ {sort | merge | copy} ]
[-catinp
  [△fileorg=organization-type]
  [△ { file=file-name [△file-name ... ]
    | {suffix | prefix} = {input-generic-name | input-directory-name}
      [△ {input-generic-name | input-directory-name} ... ]
    [△nameseq=input-order-code} ] ]
  [△reclen=record-length]
  [△ {skiprec=skip-record-number | startrec=start-record-number} ]
  [△stopaft=stop-record-number]
  [△recsepa=record-separator-type]
  [△cellsepa= {TAB | 'cell-separator'} [△ {TAB | 'cell-separator'} ... ] ]
  [△quottype=quotation-type]
  [△currency=currency-symbol]
  [△round=option-code]
  [△nullseq=option-code]
  [△codetype=encode]
  [△ { reorg=reorganization-format [△reorganization-format ... ]
    | csvreorg=csv-reorganization-format [△csv-reorganization-format ... ] } ]
  [△ { {include | omit} =condition-expression
    | {csvinclude | csvomit} =csv-condition-expression} ] ]
[-catout
  [△fileorg=organization-type]
  [△file=file-name [△file-name ... ] ]
  [△reclen=record-length]
  [△minlen=minimum-record-length]
  [△recsepa=record-separator-type]
  [△ { cellsepa= {TAB | 'cell-separator'}
    | cellsepa= {TAB | 'cell-separator-in_1'}
      △ {TAB | 'cell-separator-out_1'}
      [△ {TAB | 'cell-separator-in_2'}
        △ {TAB | 'cell-separator-out_2'} ... ] } ]
  [△cellnum= {ADJUST | cell-number} ]
  [△quottype=option-code]
  [△padding= {X | Z | C'padding-character' | X'hex-digit'} ]
  [△ { reorg=reorganization-format [△reorganization-format ... ]
    | csvreorg=csv-reorganization-format [△csv-reorganization-format ... ] } ]
  [△ { {include | omit} =condition-expression
    | {csvinclude | csvomit} =csv-condition-expression} ] ]
[-catout ... ]
[-locatajst]
[-cmdopt△command-option [:command-option ... ] ]
[ { -key△key-definition [△key-definition ... ] } ]
[ { -csvkey△csv-key-definition [△csv-key-definition ... ] } ]
[ { -sum [△field-definition [△field-definition ... ] ]
  | -csvsum [△csv-field-definition [△csv-field-definition ... ] ] } ]
[ { -reorg△reorganization-format [△reorganization-format ... ]
  | -csvreorg△csv-reorganization-format [△csv-reorganization-format ... ] } ]
[-sorttmpdir△temporary-directory1]
[-mergetmpdir△temporary-directory2]
[-worksize△ {buffer-size [unit]
  | auto [,max-buffer-size [unit] ]
  | incore [,max-buffer-size [unit] ] } ]
[-dateopt△base-year]
[-altseq△code=convert-code [△convert-code ... ] ]
[-iosize { △iosize
  | [△inpiosize=input-file-iosize]
  [△outpiosize=output-file-iosize]
  [△sorttmpiosize=sort-temporary-file-iosize]
  [△mergetmpiosize=merge-temporary-file-iosize] } ]
```

(凡例)

- △ : 1 個以上のスペース (空白), またはタブを示す
- [ ] : 省略できることを示す
- … : 繰り返して指定できることを示す
- { } : この記号で囲まれている項目のうち 1 組の項目を選択することを示す
- | : { } で囲んだ項目の区切りを示す
- 斜体文字*: 値を任意に指定できることを示す

注※

exsort コマンドのパラメタ・オペランドではありません。パラメタファイルを記述した文字コードの種類を指定します。

## 4.2.6 簡易コピー機能 (コマンド形式だけ)

```
exsort※
  △-function△copy
  [△-cmdopt△command-option [:command-option … ] ]
  △-cpyinp
    △fileorg=organization-type
    [△file-name [△file-name … ] ]
  [△-cpyout
    [△file-name [△file-name … ] ] ]
  [△-iosize { △iosize
    | [△inpiosize=input-file-iosize]
    [△outiosize=output-file-iosize] } ]
```

(凡例)

- △ : 1 個以上のスペース (空白), またはタブを示す
- [ ] : 省略できることを示す
- … : 繰り返して指定できることを示す
- { } : この記号で囲まれている項目のうち 1 組の項目を選択することを示す
- | : { } で囲んだ項目の区切りを示す
- 斜体文字*: 値を任意に指定できることを示す

注※

使用している SORT EE が 64 ビット版対応の場合, コマンド名は「exsort64」となります。

## 4.3 コマンドの表記規則

---

### 4.3.1 パラメタ形式

ハイフン (-) が付いたコマンド予約語を**パラメタ**といい、コマンドに対するオプションを指定します。

パラメタの基本形式は、次の 4 種類となります。

```
-パラメタ  
-パラメタ△値  
-パラメタ△値△ . . . △値  
-パラメタ△オペランド△ . . . △オペランド  
(凡例) △ : スペース (空白) を示す。
```

パラメタとパラメタ、パラメタと値、パラメタとオペランド、オペランドとオペランド、および値と値は、1 個以上の空白またはタブで区切って指定してください。

### 4.3.2 オペランド形式

パラメタに付随するコマンド予約語を**オペランド**といい、パラメタに対するオプションを指定します。

オペランドの基本形式は、次の 2 種類となります。

```
オペランド=値  
オペランド=値△ . . . △値  
(凡例) △ : スペース (空白) , またはタブを示す。
```

オペランドとオペランド、および値と値は、1 個以上の空白またはタブで区切って指定してください。また、イコール (=) の前後には、空白およびタブを入れなくて、連続で指定してください。

### 4.3.3 パラメタおよびオペランドの指定順序

パラメタおよびオペランドの指定順序について次に示します。

- 各パラメタの指定順序は任意です。
- 一つのパラメタに対して複数のオペランドを指定する場合は、そのパラメタに後続させて指定してください。各オペランドの指定順序は任意です。オペランドの指定例を次に示します。

```
exsort -inpfile△fileorg=C△file=file1△reclen=65535
```



### 4.3.4 コマンド予約語の表記

コマンド予約語は、大文字と小文字のどちらで記述するか、あらかじめ決められています。コマンド予約語を次に示します。

表 4-2 コマンド予約語の文字区別

文字区別	予約語の種類別	備考
大文字	コマンドオプション	DUP, FMRG など
	ファイル編成	S, V など
	挿入データ形式 (レコード編集)	X, Z, C'nn', X'nn' (シングルクォーテーションで囲んだ値 nn は任意の文字列)
	境界調整 (レコード編集)	H, F, D
	条件値 (選択条件式)	C'nn', N'nn', X'nn', D'nn' (シングルクォーテーションで囲んだ値 nn は任意の文字列)
	論理演算子 (選択条件式)	AND, OR
小文字	コマンド名	exsort
	パラメタ名	-p, -chk, -function など
	-function パラメタ指定値	sort, merge, copy
	オペランド名	fileorg=, file=, reclen= など
	キー属性, または項目属性	j, z, p など

### 4.3.5 ファイル名およびディレクトリ名の指定形式

ファイル名およびディレクトリ名の指定形式について次に示します。

- 空白文字を含んだファイル名またはディレクトリ名を指定する場合、ファイル名またはディレクトリ名ごとにダブルクォーテーションで囲んで指定してください。ファイル名の指定例を次に示します。

「a △ b」および「c △ d」の、二つのファイルを指定する場合

```
-inpfile file="a△b" "c△d"
```

ディレクトリ名の指定例を次に示します。

```
-sorttmpdir "C:¥Documents△and△Settings¥sortwork"
```

(凡例)

△ : 半角スペース (空白) を示します。

- ダブルクォーテーション ("), セミコロン (;), イコール (=), およびパーセント

## 4. SORT EE の使い方

(%) は、`exsort` コマンドでは引用符、区切り、予約語、および環境変数名に使用しているため、ファイル名およびディレクトリ名に使用しないでください。これらの文字を使用したファイル名またはディレクトリ名を指定した場合、次のように解釈されません。

- **Windows** ではセミコロン (;) を含んだファイル名およびディレクトリ名の指定ができます。ただし、環境変数に関連づけるファイル名の区切りとして解釈するため、ファイル名およびディレクトリ名への使用は推奨できません。詳細は「4.4.6 ファイル名およびディレクトリ名」を参照してください。
- **UNIX** ではダブルクォーテーション (") を含んだファイル名およびディレクトリ名の指定ができます。ただし、パラメタファイルでは引用符として解釈するため、ファイル名およびディレクトリ名への使用は推奨できません。詳細は「4.4.6 ファイル名およびディレクトリ名」を参照してください。
- **UNIX** ではコロン (:) を含んだファイル名およびディレクトリ名の指定ができます。ただし、環境変数に関連づけるファイル名の区切りとして解釈するため、ファイル名およびディレクトリ名への使用は推奨できません。詳細は「4.4.6 ファイル名およびディレクトリ名」を参照してください。
- **UNIX** では先頭にパーセント (%) を付けたファイル名を指定すると環境変数名と解釈し、「**KBLS1622-E** 入力ファイル名の指定が正しくありません。」で異常終了となります。
- イコール (=) を含んだファイル名を指定すると、オペランドと解釈されます。例えば、ファイル名に「`abc=def`」と指定すると、「**KBLS1100-E** 不正なパラメタが指定されました。」というメッセージが表示され、コマンドが異常終了します。メッセージが表示されたときの原因と対処方法の詳細は、「付録 K メッセージ」を参照してください。
- ファイル名、およびディレクトリ名には漢字を指定できます。

### 4.3.6 選択条件式の書き方

選択条件式の書き方について次に示します。

- 選択条件式の中に指定する比較演算子にはリダイレクト演算子 (<>) と同じ形式のものが含まれます。このためコマンドで指定する場合は、選択条件式全体をダブルクォーテーション (") で囲んで指定してください。選択条件式の指定例を次に示します。

10 バイト目から 4 バイト (外部 10 進数) が 1,000 を超えるレコードを選択する場合

```
include="z+10-4△>△N'1000'"
```

(凡例)

△ : 半角スペース (空白) を示します。

- コマンドで文字条件値 (C'*nn'*) の中に、シングルクォーテーション (')、ダブルクォーテーション (")、または円記号 (¥) のどれかを指定する場合、次の表に示すと

おりに指定してください。

表 4-3 選択条件式の文字条件値

項番	指定する記号	指定方法		
		UNIX		Windows
		B シェル系の場合	C シェルの場合	
1	シングルクォーテーション (')	文字として扱いたいシングルクォーテーション (') の前に、円記号 (¥) を一つ指定してください。 例えば、「C'a¥'b'」と指定すると、「a"b」と解釈されます。なお、シングルクォーテーション (') を 16 進数条件値で指定する場合は「X'27'」と指定してください。	文字として扱いたいシングルクォーテーション (') の前に、円記号 (¥) を一つ指定してください。 例えば、「C'a¥'b'」と指定すると、「a"b」と解釈されます。なお、シングルクォーテーション (') を 16 進数条件値で指定する場合は「X'27'」と指定してください。	文字として扱いたいシングルクォーテーション (') の前に、円記号 (¥) を一つ指定してください。 例えば、「C'a¥'b'」と指定すると、「a"b」と解釈されます。なお、シングルクォーテーション (') を 16 進数条件値で指定する場合は「X'27'」と指定してください。
2	ダブルクォーテーション (")	文字として扱いたいダブルクォーテーション (") の前に、円記号 (¥) を一つ指定してください。 例えば、「C'¥'"」と指定すると、「"」と解釈されます。なお、ダブルクォーテーション (") を 16 進数条件値で指定する場合は「X'22'」と指定してください。	文字として扱いたいダブルクォーテーション (") の直前でいったんダブルクォーテーション (") 囲みを閉じます。 文字として扱いたいダブルクォーテーション (") の前に、円記号 (¥) を一つ指定します。その直後から再度ダブルクォーテーション (") 囲みを指定します。 例 1 <code>include="j+0-3 ≡ C'a"¥"b"</code> は、 <code>include=j+0-3 ≡ C'a"b'</code> と解釈されます。  例 2 <code>include="j+0-2 ≡ C"¥"¥"'"</code> は <code>include=j+0-2 = C''''</code> と解釈されます。 16 進数条件値で指定する場合は「X'22'」と指定してください。	16 進数条件値で「X'22'」と指定してください。

#### 4. SORT EE の使い方

項番	指定する記号	指定方法		
		UNIX		Windows
		B シェル系の場合	C シェルの場合	
3	円記号 (¥) ※	文字として扱いたい円記号 (¥) を「¥¥¥¥」と指定してください。例えば、「C'¥¥¥¥100'」と指定すると、「¥100」と解釈されます。なお、円記号 (¥) を 16 進数条件値で指定する場合は「X'5C'」と指定してください。	文字として扱いたい円記号 (¥) を「¥¥」と指定してください。例えば、「C'¥¥100'」と指定すると、「¥100」と解釈されます。なお、円記号 (¥) を 16 進数条件値で指定する場合は「X'5C'」と指定してください。	文字として扱いたい円記号 (¥) を「¥¥」と指定してください。例えば、「C'¥¥100'」と指定すると、「¥100」と解釈されます。なお、円記号 (¥) を 16 進数条件値で指定する場合は「X'5C'」と指定してください。

#### 注※

円記号 (¥) を一つ指定し、このあとに、次の三つの記号以外を指定した場合、円記号 (¥) は文字として扱われます。ただし、円記号 (¥) を一つだけ指定した場合は、KBL51100-E のエラーとなります。

- ・シングルクォーテーション (')
- ・ダブルクォーテーション (")
- ・円記号 (¥)

ただし、文字条件値 (C'nn') にこれらの記号を指定すると、指定が複雑でわかりにくくなります。このため、16 進数条件値 (X'nn') で指定することを推奨します。パラメタファイルの指定方法については、「4.4.3 選択条件式の書き方」を参照してください。

### 4.3.7 レコード編集の挿入データの書き方

レコード編集の挿入データの書き方について次に示します。

- ・挿入データ (C'nn' または X'nn') を指定する場合は、編集フィールド全体をダブルクォーテーション (") で囲んで指定してください。
- ・挿入文字データ (C'nn') の中にシングルクォーテーション ('), ダブルクォーテーション (") または円記号 (¥) のどれかを指定する場合は、選択条件式の文字条件値と同じように指定してください。選択条件式の書き方については、「4.3.6 選択条件式の書き方」を参照してください。

### 4.3.8 コマンド実行時のロケール (UNIX の場合)

選択条件式の文字条件値、およびレコード編集の文字列挿入データに日本語などのマルチバイト文字を「C'文字列'」形式で指定する場合、コマンド実行時のロケールを次のとおり指定します。

## 【SORT EE 08-11 以前】

コマンド実行時のロケール（環境変数 LANG）は、Shift\_JIS を設定してください。  
ただし、入力データが Shift\_JIS 以外の文字コードである場合は、16 進数の文字コード（「X'16 進数'」形式）で、条件値および挿入データを指定してください。

OS	LANG 環境変数の値 (Shift_JIS)
HP-UX	ja_JP.SJIS
AIX	Ja_JP Ja_JP.IBM-943

## 【SORT EE 08-20】

入力データの文字コードと同じロケールを指定してください。  
入力データの文字コードと同じロケールを指定できない場合は、条件値および挿入データのマルチバイト文字を、16 進数の文字コード（「X'16 進数'」形式）で指定してください。  
なお、SORT EE 08-20 以前のバージョンでは、「UCS-2LE」、「UCS-2BE」、「UCS-4LE」、「UCS-4BE」、「UTF-16LE」、「UTF-16BE」、「UTF-32LE」および「UTF-32BE」の入力データには対応していません。

OS	LANG 環境変数の値		
	Shift_JIS	日本語 EUC	UTF-8
HP-UX	ja_JP.SJIS	ja_JP.eucJP	—
AIX	Ja_JP Ja_JP.IBM-943	ja_JP ja_JP.IBM-eucJP	—
Linux	—	ja_JP.eucJP ja_JP ja_JP.ujis	ja_JP.UTF-8 ja_JP.utf8

(凡例)

— : なし

## 【SORT EE 08-30】

入力データの文字コードと同じロケールを指定してください。  
次のどちらかの場合は、条件値および挿入データのマルチバイト文字を、16 進数の文字コード（「X'16 進数'」形式）で指定してください。

- 入力データの文字コードと同じロケールを指定できない
- 入力データの文字コードが、「UCS-2LE」、「UCS-2BE」、「UCS-4LE」、「UCS-4BE」、「UTF-16LE」、「UTF-16BE」、「UTF-32LE」または「UTF-32BE」である

## 【SORT EE 08-40, 08-41】

- 入力ファイル編成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

## 4. SORT EE の使い方

入力データの文字コードと同じロケールを指定してください。

次のどちらかの場合は、条件値および挿入データのマルチバイト文字を、16進数の文字コード（「X'16進数'」形式）で指定してください。

- 入力データの文字コードと同じロケールを指定できない
- 入力データの文字コードが「UCS-2LE」、 「UCS-2BE」、 「UCS-4LE」、 「UCS-4BE」、 「UTF-16LE」、 「UTF-16BE」、 「UTF-32LE」 または 「UTF-32BE」 である

### ・入力ファイル編成がテキストファイル、または CSV ファイルの場合

入力データの文字コードと異なるロケールであっても、入力データのエンコードの種別を `exsort` コマンドに指定すれば、「C' 文字列'」形式のマルチバイト文字の条件値および挿入データを指定できます。指定できるロケールは、前述の【SORT EE 08-20】の表「LANG 環境変数の値」に記載されたロケールです。

### 【SORT EE 08-50 以降】

### ・入力ファイル編成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

入力データの文字コードと同じロケールを指定してください。次のどちらかの場合は、条件値および挿入データのマルチバイト文字を、16進数の文字コード（「X'16進数'」形式）で指定してください。

- 入力データの文字コードと同じロケールを指定できない
- 入力データの文字コードが「UCS-2LE」、 「UCS-2BE」、 「UCS-4LE」、 「UCS-4BE」、 「UTF-16LE」、 「UTF-16BE」、 「UTF-32LE」 または 「UTF-32BE」 である

### ・入力ファイル編成がテキストファイル、または CSV ファイルの場合

入力データの文字コードが「EUC-HJ」以外の場合は、異なるロケールであっても、入力データのエンコードの種別を `exsort` コマンドに指定すれば、「C' 文字列'」形式のマルチバイト文字の条件値および挿入データを指定できます。

入力データの文字コードが「EUC-HJ」の場合は、条件値および挿入データのマルチバイト文字を、16進数の文字コード（「X'16進数'」形式）で指定してください。指定できるロケールは、前述の【SORT EE 08-20】の表「LANG 環境変数の値」に記載されたロケールです。

SORT EE で指定できる入力データのエンコードの種別については、「付録 I.1 文字コード体系とエンコード」を参照してください。また、エンコードの指定方法については、「5.8.13 `codetype` オペランド」を参照してください。

## 4.3.9 コマンド実行時のコードページ（Windows の場合）

「Microsoft コードページ 932（以降、CP932 と呼びます）」でコマンドを実行してください。

選択条件式の文字条件値、およびレコード編集の文字列挿入データに日本語などのマル

マルチバイト文字を「C' 文字列'」形式で指定する場合の指定方法を次に示します。

**【SORT EE 08-30 以前】**

入力データが CP932 以外の文字コードの場合は、16 進数の文字コード（「X'16 進数'」形式）で条件値および挿入データを指定してください。

なお、SORT EE 08-20 以前のバージョンでは、「UCS-2LE」、 「UCS-2BE」、 「UCS-4LE」、 「UCS-4BE」、 「UTF-16LE」、 「UTF-16BE」、 「UTF-32LE」 および 「UTF-32BE」 の入力データには対応していません。

**【SORT EE 08-40】**

・ **入力ファイル編成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**

入力データが CP932 以外の文字コードである場合は、16 進数の文字コード（「X'16 進数'」形式）で条件値および挿入データを指定してください。

・ **入力ファイル編成がテキストファイル、または CSV ファイルの場合**

CP932 以外に入力データであっても、入力データのエンコードの種別を `exsort` コマンドに指定すれば、「C' 文字列'」形式のマルチバイト文字の条件値および挿入データを指定できます。

**【SORT EE 08-50 以降】**

・ **入力ファイル編成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**

入力データが CP932 以外の文字コードである場合は、16 進数の文字コード（「X'16 進数'」形式）で条件値および挿入データを指定してください。

・ **入力ファイル編成がテキストファイル、または CSV ファイルの場合**

CP932 以外に入力データであっても、入力データのエンコードの種別を `exsort` コマンドに指定すれば、「C' 文字列'」形式のマルチバイト文字の条件値および挿入データを指定できます。

ただし、入力データの文字コードが「EUC-HJ」の場合は、条件値および挿入データのマルチバイト文字を、16 進数の文字コード（「X'16 進数'」形式）で指定してください。

SORT EE で指定できる入力データのエンコードの種別については、「付録 I.1 文字コード体系とエンコード」を参照してください。また、エンコードの指定方法については、「5.8.13 `codetype` オペランド」を参照してください。

## 4.4 exsort パラメタファイルの表記規則

コマンドの表記規則と異なる点、およびパラメタファイル独自の規則について説明します。コマンドの表記規則については、「4.3 コマンドの表記規則」を参照してください。

### 4.4.1 ファイル形式

exsort パラメタファイルの形式を次の表に示します。

項目	exsort パラメタファイルサイズの制限解除オプション※	
	指定なし	指定あり
ファイルサイズ	256KB 以内	制限なし (システム側の制限あり)
ファイルに記述できる 1 行の長さ	50KB 以内 (改行文字は含まない)	50KB 以内 (改行文字は含まない)

#### 注※

exsort パラメタファイルサイズの制限解除オプションは、SORT EE 08-50 以降で使用できません。使用する場合は、環境変数「EXSORTPRMSIZEUNLIMIT (64 ビット版は「EXSORT64PRMSIZEUNLIMIT)」」、または環境ファイル「EXS\_PRMSIZEUNLIMIT」を指定してください。

- exsort パラメタファイルサイズの制限解除オプションの指定がない、または SORT EE 08-50 より前の場合、ファイルサイズは 256KB 以内です。256KB を超えるサイズのファイルを指定した場合、KBLS1607-E エラーとなります。
- exsort パラメタファイルサイズの制限解除オプションを指定した場合、ファイルサイズの制限はありません。
- 1 行の長さが 50KB を超えるデータを含むファイルを指定した場合、KBLS1608-E エラーとなります。
- ファイルサイズが 2GB 以上の exsort パラメタファイルを使用するときは、ファイルシステムがラージファイルに対応していることを確認してください。

### 4.4.2 コーディング規則

コーディング規則について次に示します。

- 1 行に複数のパラメタ、または複数のオペランドを記述できません。1 行に記述できるパラメタ、またはオペランドはそれぞれ一つだけです。ただし、-iosize パラメタは 1 行に複数のオペランドを記述できます。

正しい指定例と誤った指定例をそれぞれ次に示します。

正しい指定例 1 (1 行に一つのパラメタ、またはオペランドを指定)



```
-infile△fileorg=S
△file=FILEA
```

正しい指定例 2 (1 行に `-iosize` パラメタと複数オペランドを指定)

```
-iosize△inpiosize=1024△outiosize=512
```

誤った指定例 1 (1 行に `-function` パラメタと `-cmdopt` パラメタを指定)

```
-function sort△-cmdopt DUP
-inpfile△fileorg=S
△file=FILEA
```

誤った指定例 2 (1 行に `fileorg` オペランドと `file` オペランドを指定)

```
-inpfile△fileorg=S△file=FILEA
△include=j+0-4 = C'ABCD'
```

(凡例)

△ : 半角スペース (空白) を示します。

- パラメタは必ず行の先頭から記述してください。また、パラメタの前に 1 個以上の空白またはタブを指定しないでください。
- オペランドを指定する場合は、必ずオペランドの前に 1 個以上の空白またはタブを指定してください。
- 行の先頭に「//」を指定すると、その行全体をコメントとして扱います。コメント行は、code 文を記述しない場合はパラメタファイル内のどの位置でも記述できます。code 文を記述する場合は code 文の後ろであればどの位置でも記述できます。なお、行の途中でコメントを記述できません。
- 行の途中で改行する場合は、オペランドの終わり、または区切り (空白、またはタブ) の位置に改行文字を指定してください。改行の指定例を次に示します。

ファイル名の区切り位置で改行する場合

```
-inpfile△file=FILEA↓
△FILEB↓
```

パラメタ、またはオペランドの終わり位置で改行する場合

```
-inpfile↓
△fileorg=S↓
△file=FILEA↓
△include=j+0-4 = C'ABCD'↓
```

(凡例)

△ : 半角スペース (空白) を示します。

↓ : 改行文字を示します。

### 4.4.3 選択条件式の書き方

選択条件式の書き方を次に示します。

- パラメタファイルで指定する場合は、選択条件式をダブルクォーテーション (") で囲まないでください。
- 比較の条件となる値 (条件値) を環境変数名で指定できます。詳細については、「4.4.5 環境変数」を参照してください。
- パラメタファイルで文字条件値 (C'*nn*') の中にシングルクォーテーション (') または円記号 (¥) を指定する場合、次の表に示すとおりに指定してください。なお、ダブルクォーテーション (") は、文字条件値の中にそのまま指定してください。

表 4-4 選択条件式の文字条件値

項番	指定する記号	指定方法
1	シングルクォーテーション (')	文字として扱いたいシングルクォーテーション (') の前に、円記号 (¥) を一つ指定してください。 例えば、「C'a¥'b'」と指定すると、「a"b」と解釈されます。
2	円記号 (¥)	「¥¥」と指定してください。 例えば、「C'¥¥100'」と指定すると、「¥100」と解釈されます。 円記号 (¥) を一つ指定し、このあとに、次の二つの記号以外を指定した場合、円記号 (¥) は文字として扱われます。ただし、円記号 (¥) を一つだけ指定した場合は、KBLS1100-E のエラーとなります。 <ul style="list-style-type: none"> <li>• シングルクォーテーション (')</li> <li>• 円記号 (¥)</li> </ul>

コマンドの指定方法については、「4.3.6 選択条件式の書き方」を参照してください。

### 4.4.4 レコード編集の挿入データの書き方

- パラメタファイルで指定する場合は、編集フィールドをダブルクォーテーション (") で囲まないでください。
- 挿入文字データ (C'*nn*') の中にシングルクォーテーション (')、ダブルクォーテーション (") または円記号 (¥) のどれかを指定する場合は、選択条件式の文字条件値と同じように指定してください。選択条件式の書き方については、「4.4.3 選択条件式の書き方」を参照してください。

### 4.4.5 環境変数

入出力ファイル名、一時ファイルを作成するディレクトリ名、または比較の条件となる値 (条件値) を、環境変数名で指定できます。環境変数名はパーセント (%) で囲んで指定してください。また、空白文字を含んだ環境変数名を指定する場合は、環境変数名ごとにダブルクォーテーション (") で囲んで指定してください。なお、UNIX の B シェル系では、空白を含む環境変数名を指定できません。指定例を次に示します。

```
-infile file="%SORTIN△1%" "%SORTIN△2%"
-outfile file=%SORTOUT%
-sorttmpdir "%dir△sorttemp%"
-inpfile csvinclude=h+1-8,9 > H' "%TIME△VALUE%"'
```

(凡例)

△：半角スペース（空白）を示します。

環境変数を指定できるパラメタおよびオペランドを次の表に示します。

項目	パラメタ	オペランド	備考
入出力ファイル名	-infile -splitinp -outfile -catout	file	—
	-catinp -splitout	file  {suffix   prefix}	—  —
一時ファイルを作成するディレクトリ名	-sorttmpdir		—
	-mergetmpdir		—
	-splittmpdir		—
比較の条件となる値 (条件値)	-infile -splitinp -catinp -outfile -splitout -catout	{include   omit}  {csvinclude   csvomit}	指定できる条件値を次に示します。 <ul style="list-style-type: none"> <li>• S' 文字列 '</li> <li>• N' 固定小数点数</li> <li>• X'16 進数 '</li> <li>• D' 日付 '</li> <li>• D' マスク日付 (CSV ファイルだけ) '</li> <li>• H' 日付 T 時刻' (CSV ファイルだけ)</li> <li>• H' マスク日付 T マスク時刻' (CSV ファイルだけ)</li> </ul>

### (1) 入出力ファイル名

- 入出力ファイル名に「% 環境変数名 %」が指定された場合、指定された環境変数に定義された値をファイル名として扱います。
- 環境変数名に複数のファイル名を関連づける場合、Windows ではセミコロン (;) で、UNIX ではコロン (:) で区切って指定してください。また、ファイル名が空白を含む場合、UNIX ではダブルクォーテーション (") で囲ってください。指定例を次に示します。

(Windows)

```
set SORTIN△1=D:¥master¥uriage△01;D:¥master¥uriage△02;D:¥master¥uriage△03
set SORTOUT=E:¥newmaster¥uritotal
```

(UNIX B シェル系)

#### 4. SORT EE の使い方

```
SORTIN="/master/uriage△01:/master/uriage△02:/master/uriage△03"  
SORTOUT=/newmaster/uritotal
```

(UNIX C シェル)

```
setenv "SORTIN△1" "/master/uriage△01:/master/uriage△02:/master/uriage△03"  
setenv SORTOUT /newmaster/uritotal
```

(凡例)

△ : 半角スペース (空白) を示します。

#### (2) 一時ファイルを作成するディレクトリ名

- SORT EE 08-50 以降, 一時ファイルを作成するディレクトリ名を「% 環境変数名 %」で指定できます。
- 一時ファイルを作成するディレクトリ名に「% 環境変数名 %」を指定し, かつディレクトリ名 "% 環境変数名 %" が存在した場合, 「% 環境変数名 %」は一時ファイル作成用ディレクトリ名として扱われます。ディレクトリ名 "% 環境変数名 %" が存在しない場合, 指定された環境変数に定義された値が, 一時ファイルを作成するディレクトリ名として扱われます。  
例を次に示します。

環境変数「ENVSORT」

/tmp/sorttmp

-sorttmpdirパラメタ

-sorttmpdir %ENVSORT%

ソート用一時ファイルの割り当てディレクトリ

名称が"%ENVSORT%" のディレクトリが存在した場合

%ENVSORT%

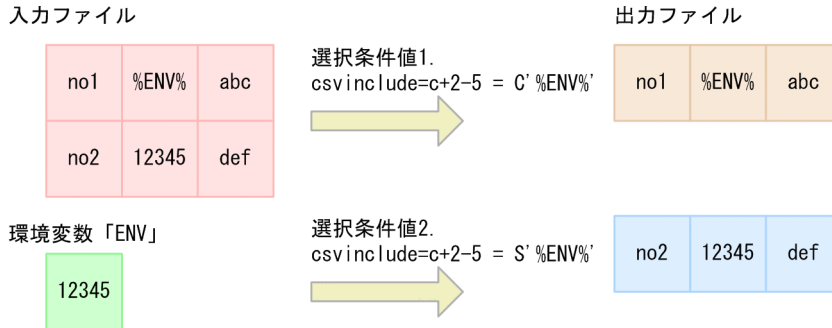
名称が"%ENVSORT%" のディレクトリが存在しなかった場合

/tmp/sorttmp

#### (3) 比較の条件となる値 (条件値)

- SORT EE 08-30 以降, 条件値「H' 日付 T 時刻' (CSV ファイルだけ)」を「% 環境変数名 %」で指定できます。
- SORT EE 08-50 以降, 次の条件値を「% 環境変数名 %」で指定できます。
  - S' 文字列 '
  - N' 固定小数点数 '
  - X'16 進数 '
  - D' 日付 '
  - D' マスク日付' (CSV ファイルだけ)

- H' マスク日付 T' マスク時刻' (CSV ファイルだけ)
- 文字条件値を環境変数で指定する場合は「S' 文字列'」を指定してください。条件値「C' 環境変数名 %'」と指定した場合、「% 環境変数名 %」は文字列として扱われず。  
例を次に示します



環境変数「ENV」に文字列「12345」を設定した場合、選択条件値 1 の「C'%ENV%」は文字列として扱われるため、値が「%ENV%」のレコードが選択されます。選択条件値 2 の「S'%ENV%」は環境変数として扱われるため、値が「12345」のレコードが選択されます。

- 次の条件値に指定した環境変数名を、パーセント (%) で囲んでいない場合は、KBL51314-E エラーとなります。
  - N' 固定小数点数'
  - X'16 進数'
  - D' 日付'
  - D' マスク日付' (CSV ファイルだけ)
  - H' 日付 T' 時刻' (CSV ファイルだけ)
  - H' マスク日付 T' マスク時刻' (CSV ファイルだけ)
- 条件値「S' 文字列'」に指定した環境変数名を、パーセント (%) で囲んでいない場合はエラーとなりません。指定された値が文字列として扱われます。例えば、条件値「S'%s\_value'」と指定した場合は、入力データが「%s\_value」のレコードが選択されます。

設定する条件値の詳細は、「5.8.16 {include | omit} オペランド」または「5.8.17 {csvinclude | csvomit} オペランド」を参照してください。

#### 4.4.6 ファイル名およびディレクトリ名

ダブルクォーテーション ("), セミコロン (;), イコール (=), パーセント (%) および コロン (: ) は、`exsort` コマンドでは引用符, 区切り, 予約語, および環境変数名に使用しているため、ファイル名およびディレクトリ名に使用しないでください。これらの文字を使用したファイル名またはディレクトリ名を指定した場合、次のように解釈されま

## 4. SORT EE の使い方

す。

- ダブルクォーテーション (") を含んだファイル名を指定すると、「KBLS1100-E 不正なパラメタが指定されました。」というメッセージが表示され、コマンドが異常終了します。メッセージが表示されたときの原因と対処方法の詳細は、「付録 K メッセージ」を参照してください。
- Windows でセミコロンを含んだファイル名「abc:def」を環境変数に指定すると、「abc」と「def」の2個のファイル名と解釈されます。
- イコール (=) を含んだファイル名を指定すると、オペランドと解釈されます。例えば、「abc=def」と指定すると、「KBLS1100-E 不正なパラメタが指定されました。」というメッセージが表示され、コマンドが異常終了します。メッセージが表示されたときの原因と対処方法の詳細は、「付録 K メッセージ」を参照してください。
- パーセント (%) で囲んだファイル名を指定すると、環境変数名と解釈されます。例えば、「%abc%」と指定すると、環境変数「abc」に関連づけられたファイル名を取得します。
- UNIX でコロンを含んだファイル名「ghi:jkl」を環境変数に指定すると、「ghi」と「jkl」の2個のファイル名と解釈します。

### 4.4.7 パラメタファイルを記述する文字コード

`exsort` コマンドのパラメタ、オペランドおよび指定値は、ASCII コード (8 ビット文字コード) で記述します。ただし、選択条件式の文字条件値、およびレコード編集の文字列挿入データに日本語などのマルチバイト文字を「C'文字列'」形式で指定する場合は、次の注意事項に従って記述してください。

#### 【SORT EE 08-11 以前】

Shift\_JIS (Windows の場合は CP932) で記述します。

ただし、入力データが Shift\_JIS (CP932) 以外の文字コードである場合は、16 進数の文字コード (「X'16 進数'」形式) で、条件値および挿入データを指定してください。

#### 【SORT EE 08-20 ~ 08-30】

入力データと同じ文字コードで記述します。記述した文字コードの種類を、パラメタファイルの先頭行で、`code` 文を使って定義します。

ただし、パラメタファイルを記述するコードは、Shift\_JIS (Windows の場合は CP932)、EUC-JP (Windows の場合は CP51932) または UTF-8 (Windows の場合は CP65001) のどれかでなければなりません。

入力データが Shift\_JIS (CP932)、EUC-JP (CP51932) および UTF-8 (CP65001) 以外の文字コードの場合は、16 進数の文字コード (「X'16 進数'」形式) で、条件値および挿入データを記述してください。

#### 【SORT EE 08-40, 08-41】

- ・入力ファイル編成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  
入力データと同じ文字コードで記述します。記述した文字コードの種類を、パラメータファイルの先頭行で、code 文を使って定義します。  
ただし、パラメータファイルを記述するコードは、Shift\_JIS (CP932)、EUC-JP (CP51932) または UTF-8 (CP65001) のどれかでなければなりません。  
入力データが Shift\_JIS (CP932)、EUC-JP (CP51932) および UTF-8 (CP65001) 以外の文字コードである場合は、16 進数の文字コード（「X'16 進数'」形式）で、条件値および挿入データを記述してください。
- ・入力ファイル編成がテキストファイル、または CSV ファイルの場合  
入力データが Shift\_JIS (CP932)、EUC-JP (CP51932) および UTF-8 (CP65001) 以外の文字コードであっても、入力データのエンコードの種別を exsort コマンドに指定すれば、「C' 文字列'」形式で条件値および挿入データを記述できます。

#### 【SORT EE 08-50 以降】

- ・入力ファイル編成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  
入力データと同じ文字コードで記述します。記述した文字コードの種類を、パラメータファイルの先頭行で、code 文を使って定義します。  
ただし、パラメータファイルを記述するコードは、Shift\_JIS (CP932)、EUC-JP (CP51932)、または UTF-8 (CP65001) のどれかでなければなりません。  
入力データが Shift\_JIS (CP932)、EUC-JP (CP51932)、および UTF-8 (CP65001) 以外の文字コードである場合は、16 進数の文字コード（「X'16 進数'」形式）で、条件値および挿入データを記述してください。
- ・入力ファイル編成がテキストファイル、または CSV ファイルの場合  
入力データが Shift\_JIS (CP932)、EUC-JP (CP51932) および UTF-8 (CP65001) 以外の文字コードであっても、入力データのエンコードの種別を exsort コマンドに指定すれば、「C' 文字列'」形式で条件値および挿入データを記述できます。  
ただし、入力データの文字コードが「EUC-HJ」の場合、条件値および挿入データのマルチバイト文字は、16 進数の文字コード（「X'16 進数'」形式）で指定してください。

SORT EE で指定できる入力データのエンコードの種別については、「付録 I.1 文字コード体系とエンコード」を参照してください。また、エンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。

### (1) code 文の書式

```
<code=指定値>
```

code 文の指定値

#### 4. SORT EE の使い方

指定値	意味	
	UNIX	Windows
SJIS	Shift_JIS	CP932
EUC-JP	日本語 EUC	CP51932
UTF-8	UTF-8	CP65001

#### (2) code 文の注意事項

- SORT EE 08-11 以前のバージョンでは、code 文は記述できません。
- code 文を省略した場合、Shift\_JIS (CP932) でパラメタ解析します。
- code 文は必ずパラメタファイルの先頭行に記述します。コメント行を記述する場合でも、code 文の後ろに記述します。code 文を先頭行以外に記述した場合、exsort コマンドは「KBLS1100-E 不正なパラメタが指定されました」というメッセージを出力して異常終了します。
- code 文は行の先頭から記述します。code 文の前に 1 個以上の空白またはタブを挿入しないでください。
- <code= 指定値 > は続けて記述してください。それぞれの文字の間に空白やタブを挿入しないでください。

#### (3) パラメタファイルの記述例

```
<code=SJIS>
-function copy
-key c+0-4
-sum z+8-4 z+16-4
-inpfile
fileorg=A
file=inp.txt
codetype=UCS-2LE
-outfile
file=out.txt
include=w+20-10 = C'Tokyo'
```

< code=SJIS > : パラメタファイルの記述コードは Shift\_JIS (CP932)

codetype=UCS-2LE : 入力データの文字コードは UCS-2 little endian

include=w+20-10 = C'Tokyo' : 選択条件式の文字条件値は ASCII コード対応の文字 (英小文字)

### 4.4.8 Unicode シグニチャ (BOM) 付きパラメタファイルの扱い

#### 【SORT EE 08-11 以前】

Unicode シグニチャには対応していません。

Unicode シグニチャ付きのパラメタファイルを指定した場合、exsort コマンドは



「KBLS1100-E 不正なパラメタが指定されました」というメッセージを出力して異常終了します。

**【SORT EE 08-20 以降】**

Unicode シグニチャ付きのパラメタファイルを指定した場合、Unicode シグニチャを無視し、パラメタファイルの先頭行に指定されている文字コードでパラメタ解析をします。

文字コードの指定がない場合は、Shift\_JIS でパラメタ解析をします。

文字コードの詳細については、「4.4.7 パラメタファイルを記述する文字コード」を参照してください。



# 5

## exsort コマンドのパラメタおよびオペランドの詳細

この章では、exsort コマンドのパラメタおよびオペランドの詳細について説明します。

---

5.1 パラメタおよびオペランドの組み合わせ

---

5.2 -p パラメタ

---

5.3 -function パラメタ

---

5.4 -locatajst パラメタ

---

5.5 -cmdopt パラメタ

---

5.6 -key パラメタ

---

5.7 -sum パラメタ

---

5.8 -infile パラメタ

---

5.9 -reorg パラメタ

---

5.10 -outfile パラメタ

---

5.11 -sorttmpdir パラメタ

---

5.12 -mergetmpdir パラメタ

---

5.13 -worksize パラメタ

---

5.14 -dateopt パラメタ

---

5.15 -altseq パラメタ

---

## 5. exsort コマンドのパラメタおよびオペランドの詳細

---

5.16 -csvkey パラメタ

---

5.17 -csvsum パラメタ

---

5.18 -csvreorg パラメタ

---

5.19 -iosize パラメタ

---

5.20 -splitfile パラメタ

---

5.21 -splitdata パラメタ

---

5.22 -splitkey パラメタ

---

5.23 -splitinp パラメタ

---

5.24 -splitout パラメタ

---

5.25 -splittmpdir パラメタ

---

5.26 -catinp パラメタ

---

5.27 -catout パラメタ

---

5.28 -cpyinp パラメタ

---

5.29 -cpyout パラメタ

---

## 5.1 パラメタおよびオペランドの組み合わせ

各パラメタ，およびオペランドの組み合わせを次の表に示します。

表 5-1 パラメタおよびオペランドの組み合わせ

項 番	パラメタ・オペランド (意味)	指定可否		-function		
		コマ ンド	パラ メタ ファ イル	sort	merg e	copy
1	-p (exsort パラメタファイル名)	●	×	△	△	△
2	-function (機能オプション)	●	●	◎	◎	◎
3	-locatajst (キー位置調整オプション)	●	●	△	△	△
4	-cmdopt (コマンドオプション)	●	●	△	△	△
5	{-key   -csvkey} (ソート・マージ・集約キー)	●	●	◎ ※1, ※2	◎ ※1, ※2	○ ※1, ※2
6	{-sum   -csvsum} (集約項目)	●	●	△※ 2	△※ 2	○※ 2

5. exsort コマンドのパラメタおよびオペランドの詳細

項番	パラメタ・オペランド (意味)		指定可否		-function			
			コマンド	パラメタファイル	sort	merge	copy	
7	{-inpfile   -catinp   -splitinp}	fileorg= (入力ファイル編成)	●	●	◎	◎	◎	
8		file= (入力ファイル名)	●	●	△※ 3	△※ 3	△※ 3	
9		{suffix   prefix} = (連結ファイル名)	●	●	◎※ 3	◎※ 3	◎※ 3	
10		reclen= (入力レコード長)	●	●	◎※ 8	◎※ 8	◎※ 8	
11		skiprec= (入力レコードスキップ件数)	●	●	△※ 9	—	△※ 9	
12		startrec= (入力開始レコード番号)	●	●	△※ 9	△※ 9	△※ 9	
13		stopaft= (入力レコード打ち切り件数)	●	●	△	—	△	
14		recsepa= (テキスト・CSV ファイルの改行文字)	●	●	△※ 13	△※ 13	△※ 13	
15		cellsepa= (CSV ファイルのデリミタ)	●	●	△※ 14	△※ 14	△※ 14	
16		quottype= (CSV ファイルの引用符)	●	●	△※ 14	△※ 14	△※ 14	
17		currency= (CSV ファイルの通貨記号)	●	●	△※ 14	△※ 14	△※ 14	
18		round= (CSV ファイルの数値データの丸め方式)	●	●	△※ 14	△※ 14	△※ 14	
19		nullseq= (CSV ファイルのナル文字列のソート順序)	●	●	△※ 14	△※ 14	△※ 14	
20		codetype= (CSV ファイルの文字コード)	●	●	△※ 14, ※ 18	△※ 14, ※ 18	△※ 14, ※ 18	
21		{reorg   csvreorg} = (入力レコード編集)	●	●	△※ 16	△※ 16	△※ 16	
22		{{include   omit}   {csvinclude   csvomit}} = (入力レコードの選択条件式)	●	●	△※ 17	△※ 17	△※ 17	
23		-copyinp	fileorg= (入力ファイル編成)	●	×	×	×	◎
24			(入力ファイル名) ※19	●	×	×	×	△
25		{-reorg   -csvreorg} (中間レコード編集)		●	●	△※ 17	△※ 17	△※ 17

## 5. exsort コマンドのパラメタおよびオペランドの詳細

項番	パラメタ・オペランド (意味)	指定可否		-function				
		コマンド	パラメタファイル	sort	merge	copy		
26	{-outfile   -catout   -splitout}	fileorg= (出力ファイル編成)	●	●	△	△	△	
27		file= (出力ファイル名)	●	●	△※ 4	△※ 4	△※ 4	
28		{suffix   prefix} = (分割ファイル名)	●	●	◎※ 4	◎※ 4	◎※ 4	
29		reclen= (出力レコード長)	●	●	○※ 10	○※ 10	○※ 10	
30		minlen= (最小出力レコード長)	●	●	○※ 10	○※ 10	○※ 10	
31		recsepa= (テキスト・CSV ファイルの改行文字)	●	●	△※ 13	△※ 13	△※ 13	
32		cellsepa= (CSV ファイルのデリミタ)	●	●	△※ 14	△※ 14	△※ 14	
33		quottype= (CSV ファイルの引用符)	●	●	△※ 14	△※ 14	△※ 14	
34		cellnum= (CSV ファイル出力セル数)	●	●	△※ 14	△※ 14	△※ 14	
35		padding= (パディング文字)	●	●	△※ 15	△※ 15	△※ 15	
36		{reorg   csvreorg} = (出力レコード編集)	●	●	△※ 16	△※ 16	△※ 16	
37		{{include   omit}   {csvinclude   csvomit}} = (出力レコードの選択条件式)	●	●	△※ 17	△※ 17	△※ 17	
38		-cpyout	(出力ファイル名) ※19	●	×	×	×	△
39		-splitfile	fileno= (ファイルの分割数)	●	●	△	△	◎※ 5
40	method= (分割方式)		●	●	△	△	◎※ 5	
41	-splitdata	{recno   spno} = (ファイルの分割量)	●	●	△	△	◎※ 5	
42		{nosplitdata   nosplitcsvdata} = (非分割データ)	●	●	△※ 6	△※ 6	△※ 5, ※6	
43	-splitkey	{key   csvkey} = (分割キー)	●	●	△※ 7	△※ 7	◎※ 5, ※7	
44		{recno   spno} = (ファイルの分割量)	●	●	△	△	△※ 5	

5. exsort コマンドのパラメタおよびオペランドの詳細

項番	パラメタ・オペランド (意味)	指定可否		-function			
		コマンド	パラメタファイル	sort	merge	copy	
45	-sorttmpdir (ソートフェーズ用一時ファイルのディレクトリ)	●	●	◎※ 11	—	—	
46	-mergetmpdir (マージフェーズ用一時ファイルのディレクトリ)	●	●	△	—	—	
47	-splittmpdir (分割用一時ファイルのディレクトリ)	●	●	◎※ 12	◎※ 12	◎※ 12	
48	-worksize (作業用バッファサイズ)	●	●	△	—	—	
49	-dateopt (日付項目の基準年)	●	●	△	△	△	
50	-altseq (文字コードの比較順序の変更)	●	●	△	△	△	
51	-iosize (I/O バッファサイズ)	●	●	△	△	△	
52		inpiosize= (入力ファイル用 I/O サイズ)	●	●	△	△	△
53		outiosize= (出力ファイル用 I/O サイズ)	●	●	△	△	△
54		sorttmpiosize= (ソートフェーズ用一時ファイル用 I/O サイズ)	●	●	△	—	—
55		mergetmpiosize= (マージフェーズ用一時ファイル用 I/O サイズ)	●	●	△	—	—
56		splittmpiosize= (分割用一時ファイル用 I/O サイズ)	●	●	△	—	—

(凡例)

- : 指定できる
- × : 指定できない
- ◎ : 必須パラメタまたはオペランド
- : 場合によって指定が必要なパラメタまたはオペランド
- △ : 任意のパラメタまたはオペランド
- : 指定しても無視されるパラメタまたはオペランド

注※1

- 固定長順編成ファイル, 可変長順編成ファイル, またはテキストファイルを入力ファイルとする場合は, -key パラメタを指定してください。
- CSV ファイルを入力ファイルとする場合は, -csvkey パラメタを指定してください。

注※2

- -sum パラメタと組み合わせて指定する場合は, -key パラメタを必ず指定してください。



- CSV ファイルを入力ファイルとし、`-csvsum` パラメタと組み合わせて指定する場合は、`-csvkey` パラメタを必ず指定してください。

## 注※ 3

- ファイル分割およびファイル連結機能を使用しない場合は、`-inpfile` パラメタの `file` オペランドで入力ファイル名を指定します。
- `-inpfile` パラメタの `file` オペランドを省略した場合、標準入力ファイルから入力します。ファイル分割機能を使用する場合は、`-splitinp` パラメタの `file` オペランドで入力ファイル名を指定します。`-splitinp` パラメタの `file` オペランドは省略できません。
- ファイル連結機能を使用する場合は、`-catinp` パラメタの `file`, `suffix`, または `prefix` オペランドで連結ファイル名を指定します。`-catinp` パラメタは、`file`, `suffix`, および `prefix` オペランドのうち、どれか 1 つを必ず指定してください。

## 注※ 4

- ファイル分割およびファイル連結機能を使用しない場合は、`-outfile` パラメタの `file` オペランドで出力ファイル名を指定します。`-outfile` パラメタの `file` オペランドを省略した場合、標準出力ファイルへ出力します。
- ファイル分割機能を使用する場合は、`-splitout` パラメタの `suffix` または `prefix` オペランドで分割ファイル名を指定します。`-splitout` パラメタは、`suffix` および `prefix` オペランドのどちらかを必ず指定してください。
- ファイル連結機能を使用する場合は、`-catout` パラメタの `file` オペランドで出力ファイル名を指定します。`-catout` パラメタの `file` オペランドを省略した場合、標準出力ファイルへ出力します。

## 注※ 5

- 集約機能を使用しないコピー機能とファイル分割機能を使用する場合、`-splitfile`, `-splitdata`, または `-splitkey` パラメタのどれか 1 つを必ず指定してください。

## 注※ 6

- 固定長順編成ファイル、可変長順編成ファイル、またはテキストファイルを入力ファイルとする場合は、`nosplitdata` オペランドを指定してください。
- CSV ファイルを入力ファイルとする場合は、`nosplitcsvdata` オペランドを指定してください。

## 注※ 7

- 固定長順編成ファイル、可変長順編成ファイル、またはテキストファイルを入力ファイルとする場合は、`key` オペランドを指定してください。
- CSV ファイルを入力ファイルとする場合は、`csvkey` オペランドを指定してください。

## 注※ 8

- 固定長順編成ファイルを入力ファイルとする場合は、`reclen` オペランドを必ず指

## 5. exsort コマンドのパラメタおよびオペランドの詳細

定してください。

### 注※ 9

- CSV ファイルを入力ファイルとする場合だけ、**startrec** オペランドの指定が有効となります。
- **skiprec** オペランドと **startrec** オペランドを同時に指定できません。同時に指定した場合、**KBLS1105-E** のエラーとなります。また、**startrec** オペランドは、CSV ファイルを指定した場合だけ有効です。

### 注※ 10

- レコード編集機能を使用する場合、可変長順編成ファイルを入力ファイルとするときは、**reclen** および **minlen** オペランドを必ず指定してください。また、固定長順編成ファイルを入力ファイルとするときは、**reclen** オペランドを必ず指定してください。

### 注※ 11

- コマンド、パラメタファイル、環境変数「**EXSORTSORTTMPDIR** (64 ビット版は「**EXSORT64SORTTMPDIR**）」のどれかを使用して、ソートフェーズ用一時ファイルのディレクトリを指定してください。

### 注※ 12

- ファイル分割機能を使用する場合、コマンド、パラメタファイル、環境変数「**EXSORTSPLITTMPDIR** (64 ビット版は「**EXSORT64SPLITTMPDIR**）」のどれかを使用して、分割用一時ファイルのディレクトリを指定してください。

### 注※ 13

- 入力ファイルまたは出力ファイルに、テキストファイルまたは CSV ファイルを指定した場合だけ有効です。

### 注※ 14

- 入力ファイルまたは出力ファイルに CSV ファイルを指定した場合だけ有効です。

### 注※ 15

- レコード編集機能を使用しない場合で、かつ **-outfile**、**-catout** または **-splitout** パラメタの **reclen** または **minlen** オペランドを指定している場合だけ有効です。

### 注※ 16

- 固定長順編成ファイル、可変長順編成ファイル、またはテキストファイルを入力ファイルとする場合は、**reorg** オペランドまたは **-reorg** パラメタを指定してください。
- CSV ファイルを入力ファイルとする場合は、**csvreorg** オペランドまたは **-csvreorg** パラメタを指定してください。

### 注※ 17

- 固定長順編成ファイル、可変長順編成ファイル、またはテキストファイルを入力

ファイルとする場合は、**include** または **omit** オペランドを指定してください。

- CSV ファイルを入力ファイルとする場合は、**csvinclude** または **csvomit** オペランドを指定してください。

注※ 18

- **codetype** オペランドに指定したエンコード（文字コード）で、CSV ファイルの区切り文字（デリミタ）、文字属性コード「**w**」、**u** および **v** を解析します。

注※ 19

- **-cpyinp** パラメタに指定する入力ファイル、および **-cpyout** パラメタに指定する出力ファイルは、オペランド名「**file=**」は不要です。直接、ファイル名を指定してください。

それぞれのパラメタおよびオペランドの意味を次に示します。

## 5.2 -p パラメタ

### 機能

**exsort** パラメタファイル名を指定します。

**exsort** パラメタファイルには、**exsort** コマンドに指定するパラメタおよびオペランド値を記述します。共通のパラメタ値や、長いパラメタ値などを **exsort** パラメタファイルに指定してください。

### 形式

`-p parameter-file-name`

### 説明

- このパラメタは省略できます。省略した場合は、**exsort** コマンドに指定した引数だけで実行します。
- このパラメタは **exsort** コマンドだけに指定できます。**exsort** パラメタファイルには指定できません。
- `-cpyinp`、および `-cpyout` パラメタと同時に指定できません。

### パラメタの優先順位およびパラメタの指定内容が有効になる範囲について

パラメタの優先順位、およびパラメタに指定した内容が有効になる範囲について説明します。

#### パラメタの優先順位

**exsort** パラメタファイルと **exsort** コマンドの両方に同じパラメタを指定した場合、**exsort** コマンドに指定したパラメタの指定内容が有効になります。

パラメタの優先順位を次の表に示します。

表 5-2 パラメタの優先順位

優先順位	exsort パラメタファイル	exsort コマンド	説明
1	◎	—	<b>exsort</b> パラメタファイルに指定したパラメタの内容が有効になります。
2	○	◎	<b>exsort</b> パラメタファイルと <b>exsort</b> コマンドの両方に同じパラメタを指定した場合、 <b>exsort</b> コマンドに指定したパラメタの内容が有効になります。
3	—	◎	<b>exsort</b> コマンドに指定したパラメタの内容が有効になります。

#### (凡例)

- ◎：指定あり（有効）
- ：指定あり（無効）
- ：指定なし

パラメタまたはオペランドの指定内容が有効になる範囲

パラメタまたはオペランドに指定した内容が有効になる範囲を次の表に示します。

表 5-3 パラメタまたはオペランドに指定した内容が有効になる範囲

項番	パラメタ	オペランド	パラメタまたはオペランドに指定した内容が有効になる範囲
1	<ul style="list-style-type: none"> <li>• -function</li> <li>• -locatajst</li> <li>• -cmdopt</li> <li>• -key</li> <li>• -csvkey</li> <li>• -sum</li> <li>• -csvsum</li> <li>• -reorg</li> <li>• -csvreorg</li> <li>• -sorttmpdir</li> <li>• -mergetmpdir</li> <li>• -splittmpdir</li> <li>※</li> <li>• -worksize</li> <li>• -dateopt</li> <li>• -altseq</li> <li>• -iosize</li> </ul>	—	それぞれのパラメタごとに有効になります。 例えば、exsort パラメタファイルに「-key j+0-4 j+8-1」、exsort コマンドに「-key z+16-8」と指定すると、exsort コマンドに指定した -key パラメタの内容 (z+16-8) が有効になります。exsort パラメタファイルに指定した -key パラメタの内容 (j+0-4, および j+8-1) はどちらも無効になります。
2	<ul style="list-style-type: none"> <li>-inpfile</li> </ul>	<ul style="list-style-type: none"> <li>• fileorg</li> <li>• file</li> <li>• reclen</li> <li>• skiprec</li> <li>• startrec</li> <li>• stopaft</li> <li>• recsepa</li> <li>• cellsepa</li> <li>• quottype</li> <li>• currency</li> <li>• round</li> <li>• nullseq</li> <li>• codetype</li> <li>• reorg</li> <li>• csvreorg</li> <li>• {include   omit}</li> <li>• {csvinclude   csvomit}</li> </ul>	それぞれのオペランドごとに有効になります。 例えば、exsort パラメタファイルに「-inpfile fileorg=S reclen=100 file=INFILE1」、exsort コマンドに「-inpfile file=INFILE2」と指定した場合、exsort コマンドに指定した file=INFILE2 オペランドだけが置き換えられます。 置き換えられた結果は「-inpfile fileorg=S reclen=100 file=INFILE2」となります。
	<ul style="list-style-type: none"> <li>-splitfile</li> </ul>	<ul style="list-style-type: none"> <li>• fileno</li> <li>• method</li> </ul>	
	<ul style="list-style-type: none"> <li>-splitdata</li> </ul>	<ul style="list-style-type: none"> <li>• {recno   spno}</li> <li>• {nosplitdata   nosplitcsvdata}</li> </ul>	
	<ul style="list-style-type: none"> <li>-splitkey</li> </ul>	<ul style="list-style-type: none"> <li>• {key   csvkey}</li> <li>• {recno   spno}</li> </ul>	

5. exsort コマンドのパラメタおよびオペランドの詳細

項番	パラメタ	オペランド	パラメタまたはオペランドに指定した内容が有効になる範囲
	-splitinp	<ul style="list-style-type: none"> <li>• fileorg</li> <li>• file</li> <li>• reclen</li> <li>• skiprec</li> <li>• startrec</li> <li>• stopaft</li> <li>• recsepa</li> <li>• cellsepa</li> <li>• quottype</li> <li>• currency</li> <li>• round</li> <li>• nullseq</li> <li>• codetype</li> <li>• reorg</li> <li>• csvreorg</li> <li>• {include   omit}</li> <li>• {csvinclude   csvomit}</li> </ul>	
	-catinp	<ul style="list-style-type: none"> <li>• fileorg</li> <li>• file</li> <li>• {suffix   prefix}</li> <li>• reclen</li> <li>• skiprec</li> <li>• startrec</li> <li>• stopaft</li> <li>• recsepa</li> <li>• cellsepa</li> <li>• quottype</li> <li>• currency</li> <li>• round</li> <li>• nullseq</li> <li>• codetype</li> <li>• reorg</li> <li>• csvreorg</li> <li>• {include   omit}</li> <li>• {csvinclude   csvomit}</li> <li>• nameseq<sup>※</sup></li> </ul>	

## 5. exsort コマンドのパラメタおよびオペランドの詳細

項番	パラメタ	オペランド	パラメタまたはオペランドに指定した内容が有効になる範囲
3	-outfile	<ul style="list-style-type: none"> <li>• fileorg</li> <li>• file</li> <li>• reclen</li> <li>• minlen</li> <li>• recsepa</li> <li>• cellsepa</li> <li>• cellnum</li> <li>• quottype</li> <li>• padding</li> <li>• reorg</li> <li>• csvreorg</li> <li>• {include   omit}</li> <li>• {csvinclude   csvomit}</li> </ul>	<ul style="list-style-type: none"> <li>• exsort パラメタファイルと exsort コマンドの両方に -outfile パラメタを一つだけ指定した場合は、オペランドごとに有効になります。また、-splitout パラメタおよび -catout パラメタも同様に、パラメタを一つだけ指定した場合は、オペランドごとに有効になります。例えば、exsort パラメタファイルに「-outfile fileorg=S file=OUTFILE1」、exsort コマンドに「-outfile file=OUTFILE2」と指定した場合、exsort コマンドに指定した file=OUTFILE2 オペランドだけが置き換えられます。置き換えられた結果は、「-outfile fileorg=S file=OUTFILE2」となります。</li> <li>• exsort パラメタファイルと exsort コマンドの両方、またはどちらかに -outfile パラメタを複数指定した場合は「KBLS1101-E -outfile パラメタが複数個指定されています。」というメッセージが表示されて、エラーとなります。また、-splitout パラメタおよび -catout パラメタも同様に、パラメタを複数指定した場合は「KBLS1101-E -outfile パラメタが複数個指定されています。」というメッセージが表示されて、エラーとなります。メッセージが表示されたときの原因と対処方法の詳細は、「付録 K メッセージ」を参照してください。</li> </ul>
	-splitout	<ul style="list-style-type: none"> <li>• fileorg</li> <li>• {prefix   suffix}</li> <li>• reclen</li> <li>• minlen</li> <li>• recsepa</li> <li>• cellsepa</li> <li>• cellnum</li> <li>• quottype</li> <li>• padding</li> <li>• reorg</li> <li>• csvreorg</li> <li>• {include   omit}</li> <li>• {csvinclude   csvomit}</li> </ul>	

## 5. exsort コマンドのパラメタおよびオペランドの詳細

項番	パラメタ	オペランド	パラメタまたはオペランドに指定した内容が有効になる範囲
	-catout	<ul style="list-style-type: none"> <li>• fileorg</li> <li>• file</li> <li>• reclen</li> <li>• minlen</li> <li>• recsepa</li> <li>• cellsepa</li> <li>• cellnum</li> <li>• quottype</li> <li>• padding</li> <li>• reorg</li> <li>• csvreorg</li> <li>• {include   omit}</li> <li>• {csvinclude   csvomit}</li> </ul>	

(凡例)

— : 該当しません。

注※

Windows だけ有効です。



## 5.3 -function パラメタ

### 意味

exsort コマンドで実行する基本機能を指定します。

### 形式

```
-function△ {sort | merge | copy}
```

### 説明

- このパラメタは省略できません。exsort コマンド、または exsort パラメタファイルのどちらかに必ずこのパラメタを指定してください。exsort コマンドと exsort パラメタファイルの両方に指定した場合は、exsort コマンドの指定が有効となります。
- sort, merge, および copy を同時に指定できません。どれか一つだけ指定してください。

### 指定値

指定値	意味	備考
sort	ソート処理を実行します。	次に示すパラメタを必ず指定してください。 <ul style="list-style-type: none"> <li>• -key</li> <li>• -sorttmpdir</li> </ul>
merge	マージ処理を実行します。	-key パラメタを必ず指定してください。
copy	コピー処理を実行します。	-sum パラメタを指定する場合は、-key パラメタを必ず指定してください。
	簡易コピー処理を実行します。	選択、集約、編集、分割、および連結機能を使用できません。次のパラメタを必ず指定してください。 <ul style="list-style-type: none"> <li>• -cpyinp パラメタ</li> <li>• -cpyout パラメタ</li> </ul>

### 関連パラメタ

- -key パラメタ
- -sum パラメタ
- -sorttmpdir パラメタ
- -cpyinp パラメタ
- -cpyout パラメタ

## 5.4 -locatajst パラメタ

### 意味

キー、集約項目、比較項目、およびレコード編集フィールドの位置を日立メインフレーム系ソートプログラム (XSORT) の指定位置と同じように解釈します。

### 形式

-locatajst

### 説明

このパラメタを指定した場合、次の項目を「1」から始まるものとして扱います。

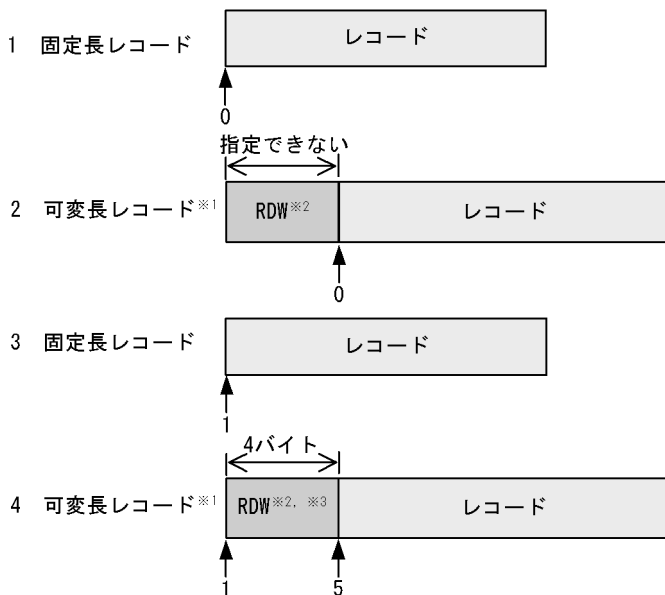
- キーの位置
- 比較項目の位置
- 集約項目の位置
- レコード編集での編集フィールドの位置
- レコード編集でのデータ挿入の位置

レコードの先頭位置の扱いを次の表および図に示します。なお、表の項番は図中の番号に対応しています。

表 5-4 レコードの先頭位置の扱い

項番	-locatajst パラメタ	レコード形式	レコードの先頭位置の扱い
1	指定なし	固定長	0
2		可変長	0 (RDW は含みません)
3	指定あり	固定長	1
4		可変長	1 (RDW の先頭を 1 とします)

図 5-1 レコードの先頭位置の扱い



注※1 日立メインフレームの可変長順編成ファイルの形式。

注※2 RDWは「レコード記述語」を示す。

注※3 RDWをキー、および項目として指定できない。

#### 関連パラメタ

- -key パラメタ
- -sum パラメタ
- -infile パラメタの reorg オペランド
- -outfile パラメタの reorg オペランド
- -reorg パラメタ
- -infile パラメタの {include | omit} オペランド
- -outfile パラメタの {include | omit} オペランド

## 5.5 -cmdopt パラメタ

### 意味

exsort コマンドのオプションを指定します。

### 形式

```
-cmdopt command-option [:command-option ... ]
```

### 注意事項

- 複数のコマンドオプションを指定する場合は、コロン (:) で区切って指定します。
- コロンの前後に空白やタブを指定しないでください。DUP:BIG のように、複数のコマンドオプションをつなげて指定します。
- 同じコマンドオプションを複数指定しないでください。

### 指定値

*command-option* に指定する値を表 5-5 に示します。また、コマンドオプションと機能の関係を表 5-6 に示します。

表 5-5 コマンドオプションの指定値

項番	指定値	意味	備考
1	DUP   NODUP	重複レコードの入力順序保存機能	ソート機能にだけ有効なオプションです。マージとコピー機能の場合は、オプションの指定有無に関わらず、必ず重複レコードの入力順序が保存されます。
2	OVWRT   INPOVWRT	ファイルの上書きオプション	<ul style="list-style-type: none"> <li>• ファイル分割機能を使用する場合、環境変数「EXSORTNOCREATEZEROFILE (64 ビット版は「EXSORT64NOCREATEZEROFILE」)」と同時に指定できません。</li> <li>• 出力ファイル名重複チェック抑止オプション (環境変数「EXSORTOUTFILENODUPCHK (64 ビット版は「EXSORT64OUTFILENODUPCHK」)」) を使用する場合、必ず OVWRT オプションを指定してください。</li> </ul>
3	EBCJ, EBCE	EBCDIK コード比較オプション	<ul style="list-style-type: none"> <li>• EBCJ オプション ソートキー、マージキー、集約キー、分割キーおよび比較項目が対象です。</li> <li>• EBCE オプション 比較項目だけが対象です。</li> </ul>

## 5. exsort コマンドのパラメタおよびオペランドの詳細

項番	指定値	意味	備考
4	BIG   INPBIG   INPLTL	入力ファイルのエンディアン オプション	<ul style="list-style-type: none"> <li>• BIG オプション Windows および Linux だけ有効なオプションです。2 進数（ただし、2 進数バイトデータ（属性 b）は除く）、または内部浮動小数点のソートキー、マージキー、集約キー、集約項目、および比較項目が対象です。</li> <li>• INPBIG, INPLTL オプション SORT EE 08-20 以降で有効なオプションです。 可変長順編成ファイルのレコード長のバイトオーダー、および 2 進数（ただし、2 進数バイトデータ（属性 b）は除く）、または内部浮動小数点のソートキー、マージキー、集約キー、集約項目、および比較項目が対象です。</li> </ul>
5	OUTBIG   OUTLTL	出力ファイルのエンディアン 変更オプション	SORT EE 08-20 以降で有効なオプション です。 可変長順編成ファイルのレコード長のバ イトオーダーだけが対象です。
6	FMRG	強制マージ機能	マージ機能にだけ有効なオプションです。
7	OVFLW	10 進数オーバーフロー判定オ プション	2 進数の集約項目だけが対象です。
8	ABS	内部 10 進数符号保障オプショ ン	内部 10 進数の集約項目だけが対象です。
9	KEYNOCHK	10 進キーのチェックオプショ ン	内部 10 進数、外部 10 進数、および固定 小数点数 (CSV) のソートキー、マージ キー、集約キー、分割キー、および比較 項目が対象です。
10	SUMNOCHK	10 進集約項目のチェックオプ ション	内部 10 進数、外部 10 進数、および固定 小数点数 (CSV) の集約項目だけが対象 です。
11	POSNOCHK	キー位置のチェックオプショ ン	可変長順編成ファイルおよびテキスト ファイルのソートキー、マージキー、分 割キー、および集約キーが対象です。
12	DATENOCCHK	日付データのチェックオプ ション	日付のソートキー、マージキー、集約 キー、分割キー、および比較項目が対象 です。
13	DATETIMENO CHK	日付時刻データのチェックオ プション	日付時刻のソートキー、マージキー、集 約キー、分割キーおよび比較項目が対象 です。
14	NOLOCK   EXC	入出力ファイル共用・排他機 能	UNIX にだけ有効なオプションです。
15	SYNC	ファイルの同期出力オプショ ン	—
16	FSPLIT	強制ファイル分割機能	分割キーにだけ有効なオプションです。

5. exsort コマンドのパラメタおよびオペランドの詳細

項番	指定値	意味	備考
17	SEQOPNCLS	ファイル逐次 OPEN/CLOSE 機能	ファイル分割機能およびファイル連結機能を使用する場合だけ有効なオプションです。
18	INPCONSEPA, OUTCONSEPA	連続区切り文字オプション	CSV ファイルの連続した区切り文字 (デリミタ) を 1 つのデリミタとして扱うオプションです。 CSV ファイルにだけ有効なオプションです。
19	SUMNULKEY, SUMNONULL	ナル文字列の集約オプション	ナル文字列のキー, および集約項目のオプションです。 CSV ファイルにだけ有効なオプションです。
20	OVFERR	集約結果オーバーフロー検知オプション	集約機能にだけ有効なオプションです。
21	SJIS	シフト JIS コード比較オプション	文字「c」にだけ有効なオプションです。
22	DIFFZERO	ゼロデータ比較オプション	左分離符号付き数字「d」にだけ有効なオプションです。
23	DELBLANKLINE	空行削除オプション	テキストファイルと CSV ファイルに有効なオプションです。
24	CSVFRMCHK	CSV 数値データ (固定小数点数) の形式チェックオプション	<ul style="list-style-type: none"> <li>• SORT EE 08-50 以降で有効なオプションです。</li> <li>• CSV 数値データ (固定小数点数) のソートキー, マージキー, 集約キー, 分割キー, 非分割データおよび比較項目が対象です。</li> <li>• KEYNOCHK オプションおよび環境変数「EXSORTMGTOPT (64 ビット版は「EXSORT64MGTOPT」)」の KEY_INVALID_NOCHK オプションと同時に指定できません。</li> </ul>
25	SELBRK	選択条件式のブレイクオプション	<ul style="list-style-type: none"> <li>• SORT EE 08-50 以降で有効なオプションです。</li> <li>• 出力ファイルパラメタ (-outfile, -catout, -splitout パラメタ) に指定された選択条件式 (include, omit, csvinclude, csvomit オペランド) にだけ有効なオプションです。</li> </ul>

(凡例)

— : 該当なし

表 5-6 コマンドオプションと機能の関係

項番	指定値	基本機能				レコード処理機能				
		ソート	マージ	コピー	簡易コピー	選択	集約	編集	分割	連結
1	DUP	○	—	—	—	—	—	—	—	—
	NODUP	○	—	—	—	—	—	—	—	—
2	OVWRT	○	○	○	○	—	—	—	—	—
	INPOVWRT	○	—	—	—	—	—	—	—	—
3	EBCJ	○	○	—	—	○	○※1	—	○※2	—
	EBCE	—	—	—	—	○	—	—	—	—
4	BIG	○	○	○	—	○	○	—	○※2	—
	INPBIG	○	○	○	—	○	○	—	○※2	—
	INPLTL	○	○	○	—	○	○	—	○※2	—
5	OUTBIG	○	○	○	—	—	—	—	—	—
	OUTLTL	○	○	○	—	—	—	—	—	—
6	FMRG	—	○	—	—	—	—	—	—	—
7	OVFLW	—	—	—	—	—	○	—	—	—
8	ABS	—	—	—	—	—	○	—	—	—
9	KEYNOCHK	○	○	—	—	○	○※1	—	○※2	—
10	SUMNOCHK	—	—	—	—	—	○	—	—	—
11	POSNOCHK	○	○	—	—	—	○※1	—	○※2	—
12	DATENOCHK	○	○	—	—	○	○※1	—	○※2	—
13	DATETIMENOCHK	○	○	—	—	○	○※1	—	○※2	—
14	NOLOCK	○	○	○	○	—	—	—	—	—
	EXC	○	○	○	○	—	—	—	—	—
15	SYNC	○	○	○	○	—	—	—	—	—

5. exsort コマンドのパラメタおよびオペランドの詳細

項番	指定値	基本機能				レコード処理機能				
		ソート	マージ	コピー	簡易コピー	選択	集約	編集	分割	連結
16	FSPLIT	-	-	-	-	-	-	-	○※3	-
17	SEQOPNCLS	-	-	-	-	-	-	-	○	○
18	INPCONSEPA	○	○	○	-	-	-	-	-	-
	OUTCONSEPA	○	○	○	-	-	-	-	-	-
19	SUMNULKEY	-	-	-	-	-	○※1	-	-	-
	SUMNONULL	-	-	-	-	-	○	-	-	-
20	OVFERR	-	-	-	-	-	○	-	-	-
21	SJIS	○	○	-	-	-	○※1	-	-	-
22	DIFFZERO	○	○	-	-	-	○※1	-	-	-
23	DELBLANKLINE	○	○	○	-	-	-	-	-	-
24	CSVFRMCHK	○	○	-	-	○	○※1	-	○※2	-
25	SELBRK	-	-	-	-	○	-	-	-	-

(凡例)

○：有効

-：無効

注※1

集約キーが有効となります。

注※2

分割キー，および非分割データが有効となります。

注※3

分割キーが有効となります。

それぞれの指定値の詳細を次に示します。

## 5.5.1 重複レコードの入力順序保存機能 (DUP | NODUP オプション)

DUP

入力ファイル中に重複したキーのレコードがある場合，重複したデータを入力順に



並べて出力します。

#### NODUP

入力ファイル中に重複したキーのレコードがある場合、重複したデータを入力順に並べないで出力します。

#### 注意事項

- DUP と NODUP を同時に指定しないでください。
- DUP および NODUP のどちらも指定しなかった場合、環境変数「EXSORTDUPOPT (64 ビット版は「EXSORT64DUPOPT)」」に YES が指定されていると、DUP が指定されたものとして動作します。YES 以外を指定、または環境変数がないと、NODUP が指定されたものとして動作します。
- NODUP を指定してソートを実行した場合のソート結果は、一意ではありません。このため、ソート結果がデータ入力順と同じになる場合もあります。

## 5.5.2 ファイルの上書きオプション (OVWRT | INPOVWRT オプション)

#### OVWRT

出力ファイルと同じ名前のファイルがすでにある場合、そのファイルを上書きしたいときに指定します。

このオプションを指定しない場合、出力ファイルと同じ名前のファイルがすでにあると、ファイルが書き換えられません。このとき、exsort コマンドは異常終了します。

#### INPOVWRT

ソート機能を使用する場合、ソート機能以外に選択、集約、または編集機能を指定しても、入力ファイルと出力ファイルに同じファイルを指定できます。

オプションの指定がない場合、入力ファイルと出力ファイルに同じファイルを指定できるのは、選択、集約、および編集機能を指定していないソート機能だけです。

#### 注意事項

- OVWRT, および INPOVWRT を同時に指定できません。
- OVWRT, および INPOVWRT オプションは、ファイル分割機能使用時、環境変数「EXSORTNOCREATEZEROFIL (64 ビット版は「EXSORT64NOCREATEZEROFIL)」」と同時に指定できません。
- INPOVWRT オプションを指定し、ソート機能と選択、集約、または編集機能を指定した場合、入力ファイルと出力ファイルに同じファイルを指定すると、入力ファイルのデータが変更され元データが消失することがあります。そのため、事前に入力ファイルのバックアップを作成してください。
- 環境変数「EXSORTOUTFILENODUPCHK (64 ビット版は「EXSORT64OUTFILENODUPCHK)」」を指定する場合、OVWRT オプションを必ず指定してください。OVWRT オプションの指定がない場合は、KBLS1122-E

## 5. exsort コマンドのパラメタおよびオペランドの詳細

エラーとなります。

### 5.5.3 EBCDIK コード比較オプション (EBCJ, EBCE オプション)

EBCDIK コードの体系に従って、キーおよび比較項目を比較します。

EBCJ および EBCE オプションは、SORT では -v パラメタに相当します。-v パラメタについての詳細は、「9.2.2(1)(a) -v パラメタについて」を参照してください。

このオプションを指定しない場合は、JIS8 単位コード体系に従って比較します。コード変換の詳細は、「付録 A JIS8 単位コード体系と EBCDIK コード体系の変換表」を参照してください。

入力データと指定値の対応を次の表に示します。

入力データと指定値の対応

入力データ	指定値
キーまたは比較項目が JIS8 単位コード	EBCJ
比較項目が EBCDIK コード	EBCE

なお、EBCE は比較項目にだけ有効なオプションです。キーに対して指定しても無効になります。

指定値が有効になる範囲を次の表に示します。

指定値	ソートキー	マージキー	集約キー	比較項目
EBCJ	○	○	○	○
EBCE	—	—	—	○
EBCJ : EBCE	○ (EBCJ) ※	○ (EBCJ) ※	○ (EBCJ) ※	○ (EBCE) ※

(凡例)

- : 有効
- : 無効

注※

丸括弧内に示した値が有効になります。

注意事項

- キー属性、または比較項目属性に「j (英数字)」を指定していない場合、このオプションは無効となります。
- 文字コードの順序を EBCDIK コードの並びに合わせて比較を実行するだけで、入力レコードのデータそのものが変更されることはありません。

- 文字が存在しない JIS コード (X'80' ~ X'9F' および X'E0' ~ X'FF') については、X'80' を「X'00'」と見なします。また、X'81' ~ X'9F' および X'E0' ~ X'FF' を「X'FF'」と見なします。このため、X'81' ~ X'9F' および X'E0' ~ X'FF' は同じ文字として処理されます。例えば、選択条件式に「include=j+0-1 > X'B3'」, コマンドオプションに「-cmdopt EBCJ」と指定した場合、X'81' ~ X'9F' および X'E0' ~ X'FF' はすべて X'FF' と見なされるため、選択対象となります。

## 5.5.4 入力ファイルのエンディアンオプション (BIG | INPBIG | INPLTL オプション)

入力ファイルのエンディアンを指定します。

### BIG

キー、比較項目、または集約項目の属性が 2 進数 (n, m, a) または内部浮動小数点 (f) の場合、入力データを **big endian** (上位バイトを下位番地とする) の数値として扱います。

なお、可変長順編成ファイルの場合、レコードヘッダ<sup>※</sup>に格納されているレコード長のバイトオーダーには適用されません。レコード長には、プラットフォームのバイトオーダーが適用されます。

### INPBIG

キー、比較項目、または集約項目の属性が 2 進数 (n, m, a) または内部浮動小数点 (f) の場合、入力データを **big endian** (上位バイトを下位番地とする) の数値として扱います。

また、可変長順編成ファイルの場合は、レコードヘッダ<sup>※</sup>に格納されているレコード長のバイトオーダーについても、**big endian** として扱います。

### INPLTL

キー、比較項目、または集約項目の属性が 2 進数 (n, m, a) または内部浮動小数点 (f) の場合、入力データを **little endian** (上位バイトを上位番地とする) の数値として扱います。

また、可変長順編成ファイルの場合は、レコードヘッダ<sup>※</sup>に格納されているレコード長のバイトオーダーについても、**little endian** として扱います。

### 注※

可変長順編成ファイルは、各レコードの先頭にレコード長を 2 進数で格納したレコードヘッダを持ちます。

### 注意事項

- BIG, INPBIG, および INPLTL を同時に指定できません。
- オプションの指定によって入力レコードのデータそのものが変更されることはありません。
- オプションを指定しない場合は、プラットフォームに依存したエンディアンで扱

## 5. exsort コマンドのパラメタおよびオペランドの詳細

います。プラットフォームのエンディアンについては、「5.5.5 出力ファイルのエンディアン変更オプション (OUTBIG | OUTLTL オプション)」を参照してください。

- BIG オプションは Windows および Linux だけで有効なオプションです。
- INPBIG, および INPLTL オプションは SORT EE 08-20 以降で有効なオプションです。08-20 より前のバージョンでは指定できません。

### 5.5.5 出力ファイルのエンディアン変更オプション (OUTBIG | OUTLTL オプション)

出力ファイルのエンディアンを変更します。

#### OUTBIG

可変長順編成ファイルのレコードヘッダ<sup>※</sup>に格納されているレコード長のバイトオーダーを、`big endian` (上位バイトを下位番地とする) に変更して出力します。

#### OUTLTL

可変長順編成ファイルのレコードヘッダ<sup>※</sup>に格納されているレコード長のバイトオーダーを、`little endian` (上位バイトを上位番地とする) に変更して出力します。

#### 注※

可変長順編成ファイルは、各レコードの先頭にレコード長を 2 進数で格納したレコードヘッダを持ちます。

#### 注意事項

- OUTBIG と OUTLTL を同時に指定できません。
- 出力ファイルのファイル編成が可変長順編成ファイルの場合だけ有効です。固定長順編成ファイル、テキストファイル、および CSV ファイルの場合、指定は無効になります。
- オプションを指定しない場合、または入力のエンディアンと同じエンディアンを指定した場合、レコード長のバイトオーダーは変更されません。
- オプション指定時、変換するのはレコード長だけです。2 進数 (n, m, a) および内部浮動小数点 (f) 属性のキー、比較項目、または集約項目は変更されません。
- SORT EE 08-20 以降で有効なオプションです。08-20 より前のバージョンでは指定できません。
- エンディアン関連のオプションの組み合わせを次の表に示します。

プラットフォーム	エンディアン	-cmdopt の指定		可変長順ファイルの入力レコード長の扱い	入力レコードデータの扱い	可変長順ファイルの出力レコード長の変更
AIX または HP-UX	big endian	-	-	big endian	big endian	変更しない
			OUTBIG	big endian	big endian	変更しない
			OUTLTL	big endian	big endian	変更する (little)
		BIG	-	big endian	big endian	変更しない
			OUTBIG	big endian	big endian	変更しない
			OUTLTL	big endian	big endian	変更する (little)
		INBIG	-	big endian	big endian	変更しない
			OUTBIG	big endian	big endian	変更しない
			OUTLTL	big endian	big endian	変更する (little)
		INPLTL	-	little endian	little endian	変更しない
			OUTBIG	little endian	little endian	変更する (big)
			OUTLTL	little endian	little endian	変更しない
Linux または Windows	little endian	-	-	little endian	little endian	変更しない
			OUTBIG	little endian	little endian	変更する (big)
			OUTLTL	little endian	little endian	変更しない
		BIG	-	little endian ※	big endian	変更しない
			OUTBIG	little endian ※	big endian	変更する (big)
			OUTLTL	little endian ※	big endian	変更しない
		INBIG	-	big endian	big endian	変更しない
			OUTBIG	big endian	big endian	変更しない
			OUTLTL	big endian	big endian	変更する (little)
		INPLTL	-	little endian	little endian	変更しない
			OUTBIG	little endian	little endian	変更する (big)
			OUTLTL	little endian	little endian	変更しない

(凡例) - : 指定なし

注※ BIG オプションは、可変長順編成ファイルのレコード長のバイトオーダーには適用されません。レコード長は little endian で扱います。

### 5.5.6 強制マージオプション (FMRG オプション)

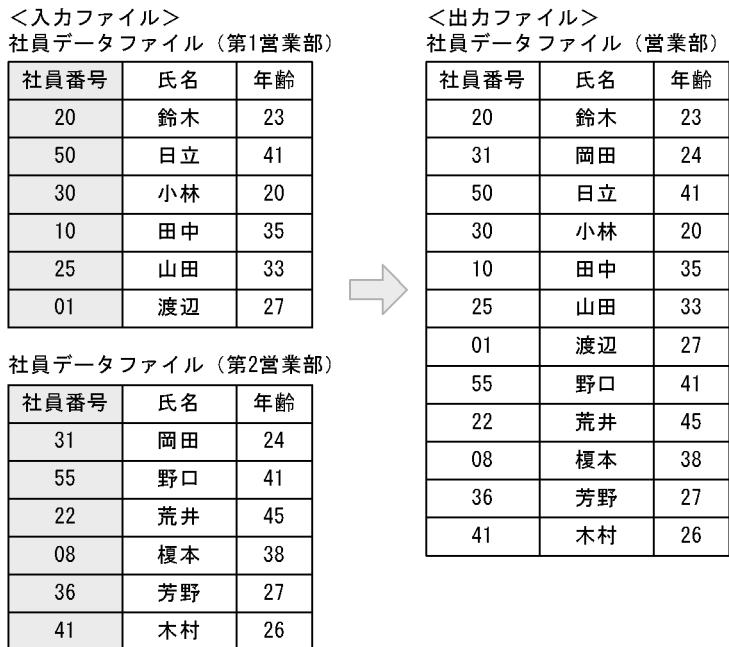
通常のマージでは、マージするそれぞれのファイルがあらかじめ昇順または降順に配列されていることが前提となっています。このため、マージ後も昇順または降順の指定に

## 5. exsort コマンドのパラメタおよびオペランドの詳細

従って配列されます。

しかし、マージするそれぞれのファイルが昇順または降順に配列されていなくてもマージできます。これを「強制マージ」といいます。ただし、強制マージではマージ後のデータの配列は保証できません。強制マージの例を次の図に示します。

図 5-2 強制マージの例



(凡例)  : マージのキーを示す。

注 入力ファイルは、昇順でソートされていない。

### 5.5.7 10進数オーバーフロー判定オプション (OVFLW オプション)

集約項目の属性が2進数の場合、10進数の桁数でオーバーフロー判定をすることを指定します。

集約結果の範囲を次の表に示します。

表 5-7 集約結果の範囲 (固定小数点2進数 (n))

バイト長	OVFLW オプション	
	指定なし	指定あり
2	- 32,768 ~+ 32,767	- 9,999 ~+ 9,999
4	- 2,147,483,648 ~+ 2,147,483,647	- 999,999,999 ~+ 999,999,999

バイト長	OVFLW オプション	
	指定なし	指定あり
8	− 9,223,372,036,854,775,808 ~ + 9,223,372,036,854,775,807	− 999,999,999,999,999,999 ~ + 999,999,999,999,999,999

表 5-8 集約結果の範囲（符号なし 2 進数 (m)）

バイト長	OVFLW オプション	
	指定なし	指定あり
2	0 ~ 32,767	0 ~ 9,999
4	0 ~ 2,147,483,647	0 ~ 999,999,999
8	0 ~ 9,223,372,036,854,775,807	0 ~ 999,999,999,999,999,999

### 5.5.8 内部 10 進数符号保障オプション (ABS オプション)

集約項目の属性が内部 10 進数である場合、最初の集約項目の演算符号が (F)<sub>16</sub> (絶対値) であるとき、以降の集約項目の集約結果に対して同じ演算符号 (絶対値) を適用します。

最初の集約項目の演算符号が (C)<sub>16</sub> (正) であるときは、以降の集約項目に対して (C)<sub>16</sub> (正) を適用します。また、このオプションを指定しないときは、演算符号が (F)<sub>16</sub> (絶対値) である集約項目に対して (C)<sub>16</sub> (正) を適用します。

最初の集約項目の符号	ABS オプション	
	指定あり	指定なし
F	F	C
C	C	C

### 5.5.9 10 進キーのチェックオプション (KEYNOCHK オプション)

キーまたは比較項目の属性が外部 10 進数 (CSV ファイルの場合は固定小数点)、または内部 10 進数の場合、入力データの形式が不正であってもエラーにしません。

このオプションを指定しない場合、環境変数「EXSORTMGTOPT (64 ビット版は「EXSORT64MGTOPT)」」に KEY\_INVALID\_NOCHK オプションが指定されていれば KEYNOCHK 指定ありと同じ扱いとなります。

CSVFRMCHK オプションと同時に指定できません。同時に指定した場合は KBLS1120-E エラーとなります。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

ソート、マージ、選択、および集約機能でキーまたは比較項目が不正な場合の例を次に示します。

### (a) ソート機能でキーが不正な場合

売上数量をキーにしてソートする例を示します。

#### (1) KEYNOCHKの指定がない場合

入力ファイル

品番	売上数量
03	1596
05	2396
04	159a
06	2397



exsortコマンドエラー

「KBLS1211-E キーにエラーデータを検出しました。」

#### (2) KEYNOCHKの指定がある場合

入力ファイル

品番	売上数量
03	1596
05	2396
04	159a
06	2397



出力ファイル

品番	売上数量
03	1596
04	159a
05	2396
06	2397

(凡例) ■ : 不正なデータを示す。

注 品番と売上数量は、内部10進数。

### (b) マージ機能でキーが不正な場合

売上数量をキーにしてマージする例を示します。



## (1) KEYNOCHKの指定がない場合

入力ファイルA

品番	売上数量
03	1596
05	1597



exsortコマンドエラー

「KBLS1211-E キーにエラーデータを検出しました。」

入力ファイルB

品番	売上数量
02	1595
03	1596

## (2) KEYNOCHKの指定がある場合

入力ファイルA

品番	売上数量
03	1596
05	1597



出力ファイル

品番	売上数量
02	1595
03	1596
05	1597
03	1596

入力ファイルB

品番	売上数量
02	1595
03	1596

(凡例) ■ : 不正なデータを示す。

注 品番と売上数量は、内部10進数。

## (c) 選択機能で比較項目が不正な場合

売上数量を比較項目として、売り上げ数量が 2,000 以上のレコードを選択する例を示します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

### (1) KEYNOCHKの指定がない場合

入力ファイル

品番	売上数量
02	1595
03	1596
04	200a
05	2005



exsortコマンドエラー

「KBLS1320-E 比較項目にエラーデータを検出しました。」

### (2) KEYNOCHKの指定がある場合

入力ファイル

品番	売上数量
02	1595
03	1596
04	200a
05	2005



出力ファイル

品番	売上数量
04	200a
05	2005

(凡例) ■ : 不正なデータを示す。

注 品番と売上数量は、内部10進数。

### (d) 集約機能でキーが不正な場合

品番をキーに、売上数量を集約項目にして集約する例を示します。

## (1) KEYNOCHKの指定がない場合

入力ファイル

品番	売上数量
03	10
03	20
05	30
0a	20
0a	30



exsortコマンドエラー

「KBLS1211-E キーにエラーデータを検出しました。」

## (2) KEYNOCHKの指定がある場合

入力ファイル

品番	売上数量
03	10
03	20
05	30
0a	20
0a	30



出力ファイル

品番	売上数量
03	30
05	30
0a	50

(凡例) ■ : 不正なデータを示す。

注 品番と売上数量は、内部10進数。

## 注意事項

キー属性、または比較項目属性に、内部 10 進数「p」、外部 10 進数「z」、左符号外部 10 進数「x」、左分離符号外部 10 進数「s」、右分離符号外部 10 進数「t」、または CSV ファイルの固定小数点数「z」を指定している場合だけ、このオプションは有効です。

KEYNOCHK	不正データの扱い※1
指定なし	KBLS1211-E または KBLS1320-E メッセージを表示し、exsort コマンドが異常終了する。

5. exsort コマンドのパラメタおよびオペランドの詳細

KEYNOCHK	不正データの扱い※1	
指定あり	内部 10 進数	<ul style="list-style-type: none"> <li>• 数値部が 0～9 以外の場合、JIS8 単位コード体系に従って比較する。</li> <li>• 符号部が (C)<sub>16</sub> (正), (D)<sub>16</sub> (負), (F)<sub>16</sub> (絶対値) 以外の場合は、内部 10 進数を正値として扱う。</li> </ul>
	外部 10 進数 左符号付き外部 10 進数 (x)	<ul style="list-style-type: none"> <li>• ゾーン部は、比較対象外※2。</li> <li>• 数値部が 0～9 以外の場合、JIS8 単位コード体系に従って比較する。</li> <li>• 符号部が 3, 7 以外の場合は、外部 10 進数を正値として扱う※3。</li> </ul>
	左分離符号付き外部 10 進数 (s) 右分離符号付き外部 10 進数 (t)	<ul style="list-style-type: none"> <li>• ゾーン部に相当する部分は比較対象外。</li> <li>• 数値部に相当する部分が 0～9 以外の場合、JIS8 単位コード体系に従って比較をする。</li> <li>• 分離符号に相当する部分は無視され、常に正値として扱う。</li> </ul>
	固定小数点 (CSV 数値データ)	<ul style="list-style-type: none"> <li>• データ中に引用符, 符号, 通貨記号, 空白, 小数点, コマが含まれている場合は、これらの文字を除外し、残った文字を外部 10 進数として扱う。</li> <li>• ゾーン部に相当する部分は比較対象外。</li> <li>• 符号部に相当する部分は無視され、常に正値として扱う。</li> <li>• 数値部に相当する部分が 0～9 以外の場合、JIS8 単位コード体系に従って比較をする。</li> </ul>

注※1

キー、または比較項目のデータ形式が、次の表のどれかに該当する場合、不正データと見なされます。

属性	不正データの条件	備考
内部 10 進数 (p)	<ul style="list-style-type: none"> <li>• 数値部が 0～9 以外の値</li> <li>• 符号部が C, D, F 以外の値</li> </ul>	<p>「内部 10 進数符号オプション」で符号の範囲を変更できます。 オプションを指定した場合の不正データ形式については、「14.1.1(30) EXSORTPACKSIGNTYPE」および「14.1.2(30) EXSORT64PACKSIGNTYPE」を参照してください。</p>
外部 10 進数 (z) 左符号付き外部 10 進数 (x)	<ul style="list-style-type: none"> <li>• 数値部が 0～9 以外の値</li> <li>• ゾーン部が 3 以外の値</li> <li>• 符号部が 3, 7 以外の値</li> </ul>	<p>「外部 10 進数データ形式オプション」でデータ形式を変更できます。 オプションを指定した場合の不正データ形式については、「14.1.1(31) EXSORTUNPACKDATATYPE」および「14.1.2(31) EXSORT64UNPACKDATATYPE」を参照してください。</p>

属性	不正データの条件	備考
右分離符号付き外部 10 進数 (s) 左分離符号付き外部 10 進数 (t)	<ul style="list-style-type: none"> <li>数字が 0 ～ 9 (0x30 ～ 0x39) 以外の値</li> <li>符号が + (0x2B), - (0x2D) 以外の値</li> </ul>	「外部 10 進数データ形式オプション」でデータ形式を変更できます。オプションを指定した場合の不正データ形式については、「14.1.1(31) EXSORTUNPACKDATATYPE」および「14.1.2(31) EXSORT64UNPACKDATATYPE」を参照してください。
固定小数点 (z) (CSV 数値データ)	<ul style="list-style-type: none"> <li>引用符, 数字, 符号, 通貨記号, 空白, 小数点, コンマ以外の文字を含む場合</li> <li>データ中に含まれる引用符, 符号, 通貨記号, 空白, 小数点, コンマの位置が不正な場合</li> </ul>	除外文字の位置規則については、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。

## 注※ 2

ソートおよびマージ機能において、ゾーン部はキーの大小比較対象外となるため、数値部と符号部が同じレコードの場合、ソートおよびマージの結果を保証しません。例を次に示します。

## 外部10進数を16進数で表した例

ソート前	ソート後
30303030---(1)	30303030---(1)
20202020---(2)	20202020---(2)

## (凡例)

■ : ゾーン部を示す。

(1) : 10進数で「0000」を示す。

(2) : 不正データを示す。

ソート前後でレコードの並びは変わりません。

## 注※ 3

ソートおよびマージ機能において、数値部が同じレコードの場合、ソートおよびマージの結果を保証しません。例を次に示します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

外部10進数を16進数で表した例

ソート前	ソート後
39303030 --- (1)	39303030 --- (1)
39303020 --- (2)	39303020 --- (2)

(凡例)

- : 符号部を示す。
- (1) : 10進数で「9000」を示す。
- (2) : 不正データを示す。

ソート前後でレコードの並びは変わりません。

集約機能において、キーが外部 10 進数で、同一キーの先頭レコードの符号部に 3, 7 以外が設定されている場合、出力ファイルに出力されるレコードの符号部は、同一キーの先頭レコードの符号部となります。例を次に示します。

外部10進数のキーを16進数で表した例

入力ファイル		出力ファイル	
集約キー	集約項目	集約キー	集約項目
39303020 --- (1)	10	39303020 --- (1)	30
39303030 --- (2)	20		

(凡例)

- : 符号部を示す。
- (1) : 数値部が正しく、符号部が「2」の不正な10進数「9000」を示す。
- (2) : 10進数で「9000」を示す。

キー「9000」を出力ファイルに書く場合、符号部が「2」となります。

### 5.5.10 10進集約項目のチェックオプション (SUMNOCHK オプション)

集約項目の属性が外部 10 進数 (CSV ファイルの場合は固定小数点数) または内部 10 進数の場合、入力データの形式が不正であってもエラーにしません。

このオプションを指定しない場合、環境変数「EXSORTMGTOPT (64 ビット版は「EXSORT64MGTOPT)」に SUM\_INVALID\_NOCHK オプションが指定されていれば SUMNOCHK 指定ありと同じ扱いとなります。

支店コードを集約キーに、売上数量を集約項目にして集約機能を実行する場合の例を次に示します。

- (a) SUMNOCHK の指定がなく、集約項目に不正データがある場合

入力ファイル

支店コード	売上数量
03	10
03	20
04	不正データ
05	20
06	30



exsortコマンドエラー

「KBLS1411-E 集約項目にエラーデータを検出しました。」

(凡例) ■ : 不正な集約項目データを含むレコードを示す。

注 売上数量は、内部10進数。

- (b) SUMNOCHK の指定があり、キーに該当するレコードが1件で集約項目に不正データがある場合

入力ファイル

支店コード	売上数量
03	10
03	20
04	不正データ
05	20
05	30



出力ファイル

支店コード	売上数量
03	30
04	不正データ
05	50

(凡例) ■ : 不正な集約項目データを含むレコードを示す。

注 売上数量は、内部10進数。

5. exsort コマンドのパラメタおよびオペランドの詳細

(c) SUMNOCHK の指定があり、キーに該当するレコードが 2 件以上で集約項目に不正データがある場合

入カファイル		
支店コード	売上数量	
03	10	exsortコマンドエラー 「KBLS1411-E 集約項目にエラーデータを検出しました。」
03	20	
04	15	
04	不正データ	
05	30	

(凡例) ■ : 不正な集約項目データを含むレコードを示す。

注 売上数量は、内部10進数。

注意事項

集約項目の属性に、内部 10 進数「p」、外部 10 進数「z」、または CSV ファイルの固定小数点数「z」を指定している場合だけ、このオプションは有効です。

また、無条件レコード抽出機能では、このオプションは無効です。

SUMNOCHK	不正データの扱い※1	
指定なし	KBLS1411-E メッセージを表示し、exsort コマンドが異常終了する。	
指定あり	集約キーに該当するレコードが 1 件の場合	該当するレコードを出力ファイルに書き、exsort コマンド処理を続行する。
	集約キーに該当するレコードが 2 件以上の場合	KBLS1411-E メッセージを表示し※2、exsort コマンドが異常終了する。

注※1

不正データと見なされる条件を次の表に示します。

ファイル編成	不正データの条件
固定長順編成ファイル 可変長順編成ファイル テキストファイル	集約項目のデータ形式が、次のどれかの場合は不正データと見なされます。 <ul style="list-style-type: none"> <li>数値部が 0 ~ 9 以外の値</li> <li>符号部が不正な場合 内部 10 進数の場合は、(C)<sub>16</sub> (正)、(D)<sub>16</sub> (負)、(F)<sub>16</sub> (絶対値) 以外の値 外部 10 進数の場合は、3 (正)、7 (負) 以外の値</li> <li>外部 10 進数の場合、ゾーン部が 3 以外の値</li> </ul>
CSV ファイル	集約項目のデータ形式が、次のどちらかの場合は不正データと見なされません。 <ul style="list-style-type: none"> <li>引用符、数字、符号、通貨記号、空白、小数点、コンマ以外の文字を含む場合</li> <li>データ中に含まれる引用符、符号、通貨記号、空白、小数点、コンマの位置が不正な場合。除外文字の位置規則については「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。</li> </ul>



## 注※ 2

KBLS1411-E メッセージにエラー要因となったレコード番号を表示します。例を次に示します。

- キーに該当するレコードが 2 件以上あり、キーの先頭レコードの集約項目が不正データの場合

入力ファイル	
支店コード	売上数量
03	10
03	20
04	不正データ
04	15
05	30

exsortコマンドエラー  
「KBLS1411-E 集約項目にエラーデータを検出しました。」  
.....エラー要因となったレコード

(凡例) ■ : 不正な集約項目データを含むレコードを示す。

注 売上数量は、内部10進数。

- キーに該当するレコードが 2 件以上あり、キーの中間または、最終レコードの集約項目が不正データの場合

入力ファイル	
支店コード	売上数量
03	10
03	20
04	15
04	不正データ
05	30

exsortコマンドエラー  
「KBLS1411-E 集約項目にエラーデータを検出しました。」  
.....エラー要因となったレコード

(凡例) ■ : 不正な集約項目データを含むレコードを示す。

注 売上数量は、内部10進数。

### 5.5.11 キー位置のチェックオプション (POSNOCHK オプション)

可変長順編成ファイルのキーの指定範囲が最小レコード長を超えていてもエラーにしません。また、テキストファイルのキーの指定範囲が実レコード長を超えていてもエラーにしません。

実レコード長を超えたキー部分には、NULL 値 (0x00) を仮定してキーを比較します。

このオプションを指定しない場合、環境変数「EXSORTMGTOPT (64 ビット版は

## 5. exsort コマンドのパラメタおよびオペランドの詳細

「EXSORT64MGTOPT」)に VKEY\_POS\_NOCHK オプションが指定されていれば POSNOCHK 指定ありと同じ扱いとなります。

### 注意事項

- キー属性に外部 10 進数または内部 10 進数を指定した場合に、仮定した NULL 値が不正データとして扱われないように、KEYNOCHK オプションの指定を推奨します。
- テキストファイルの場合、入力レコード中の改行文字は、入力データおよびキーとして扱いません。キーの範囲内の改行文字については、NULL 値を仮定します。
- CSV ファイルの場合は、POSNOCHK の指定に関係なく次のとおり扱います。  
 -csvkey パラメタに指定した入力セルのデータがない場合、ナル文字列 (null キー) として扱います。  
 -csvkey パラメタに指定した有効桁数が不足している場合、桁数を調整します。詳細については、「付録 D CSV ファイル内のナル文字列の扱い (SORT EE の機能)」を参照してください。

POSNOCHK オプションのキーの扱いについて次の表に示します。

POSNOCHK	キーの扱い	
指定なし	可変長順編成ファイルのキーの範囲が最小レコード長を超えている場合、KBLS1208-E メッセージを表示し、exsort コマンドが異常終了します。 テキストファイルのキーの範囲が実レコード長を超えている場合、KBLS1208-E メッセージを表示し、exsort コマンドが異常終了します。	
指定あり※1	1. キー位置+キー長が実レコード長以下の場合	実レコード内のキーが、ソートキー、マージキー、または集約キーとなります。
	2. キー位置が実レコード長より小さく、キー位置+キー長が実レコード長より大きく最大レコード長以下の場合 3. キー位置が実レコード長より小さく、キー位置+キー長が最大レコード長より大きい場合	実レコード内のキーと、実レコードを超えている部分に NULL 値を仮定したキーがソートキー、マージキー、または集約キーとなります。
	4. キー位置が実レコード長より大きく、キー位置+キー長が最大レコード長以下の場合 5. キー位置が最大レコード長より小さく、キー位置+キー長が最大レコード長より大きい場合 6. キー位置が最大レコード長より大きい場合	NULL 値を仮定したキーが、ソートキー、マージキー、または集約キーとなります。

(凡例)

レコード長※2	可変長順編成ファイル	テキストファイル
最小レコード長	COBOL でファイル作成時にユーザが指定した最小レコード長	1
実レコード長	実際の入力レコード長	実際の入力レコード長 (改行文字を除く)

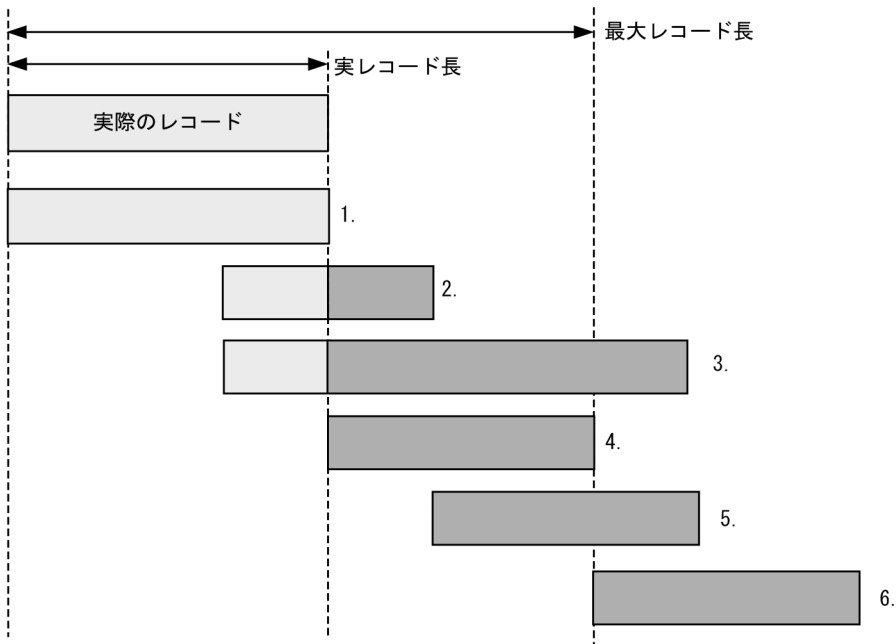
レコード長※2	可変長順編成ファイル	テキストファイル
最大レコード長	COBOL でファイル作成時にユーザが指定した最大レコード長	<code>-inpfile</code> , <code>-splitinp</code> または <code>-catinp</code> パラメタの <code>reclen</code> オペランドで指定したレコード長（改行文字を除く）※3

## 注※1

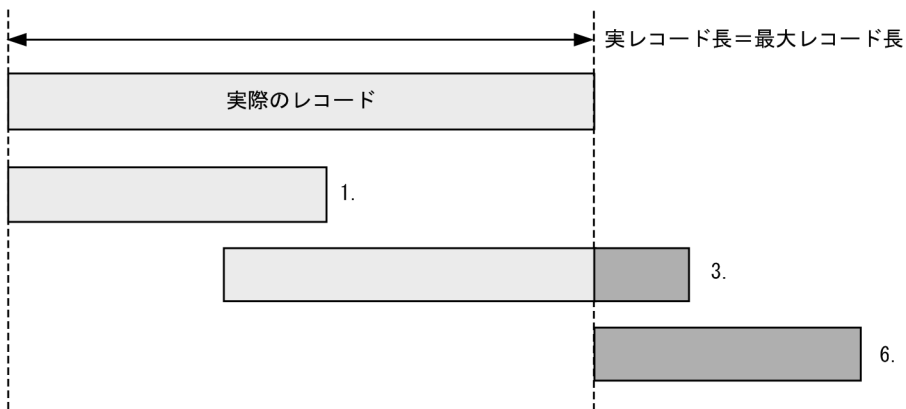
1. ～ 6. は、次の図の 1. ～ 6. に対応しています。

5. exsort コマンドのパラメタおよびオペランドの詳細

●実レコード長<最大レコード長の場合



●実レコード長=最大レコード長の場合



(凡例)

□ : 実レコード内の指定された範囲をキーとする。

■ : NULL値(0x00)をキーとする。

注※ 2

入力レコード編集 (-infile, -splitinp または -catinp パラメタの reorg オペランド) を指定した場合は、入力レコード編集後のレコード長となります。



## 5. exsort コマンドのパラメタおよびオペランドの詳細

- キーの範囲は最小レコード長以下なので、POSNOCHK 指定なしと同じソート結果となります。

**【例 2】** キー位置が最小レコード長より小さく、キー位置+キー長が最大レコード長以下の場合

(exsort コマンド指定内容)

- cmdopt POSNOCHK:DUP : 重複レコードの入力順序保存機能を指定
- key c+2-4 : キー位置 +2 ~ 5 キー長 4 バイト 昇順ソート
- inpfiler fileorg=A reclen=8 : テキストファイル (最大レコード長 8 バイト)

(ソート結果)

	0	1	← キーの範囲 →				6	7	8	(実レコード長)
レコード3	a	b	c	d	LF				4バイト (最小)	
レコード2	a	b	c	d	e	LF			5バイト	
レコード4	a	b	c	d	e	f	LF		6バイト	
レコード5	a	b	c	d	e	f	g	h	LF	8バイト (最大)
レコード6	a	b	c	d	e	f	g	LF	7バイト	
レコード1	u	v	w	x	y	z	LF		6バイト	

- 重複レコードの入力順序保存機能の指定によって、レコード 4 ~ 6 の順番は変わりません。

**【例 3】** キー位置が最小レコード長より小さく、キー位置+キー長が最大レコード長より大きい場合

(exsort コマンド指定内容)

- cmdopt POSNOCHK
- key c+2-8 : キー位置 +2 ~ 9 キー長 8 バイト 昇順ソート
- inpfiler fileorg=A reclen=8 : テキストファイル (最大レコード長 8 バイト)

(ソート結果)

	0	1	← キーの範囲 →								10	(実レコード長)
レコード3	a	b	c	d	LF							4バイト (最小)
レコード2	a	b	c	d	e	LF						5バイト
レコード4	a	b	c	d	e	f	LF					6バイト
レコード6	a	b	c	d	e	f	g	LF				7バイト
レコード5	a	b	c	d	e	f	g	h	LF			8バイト (最大)
レコード1	u	v	w	x	y	z	LF					6バイト

**【例 4】** キー位置が最小レコード長より大きく、キー位置+キー長が最大レコード長以下の場合

(exsort コマンド指定内容)

-cmdopt POSNOCHK

-key c+2-6 : キー位置 +4 ~ 9 キー長 6 バイト 昇順ソート

-inpf fileorg=A : テキストファイル (最大レコード長 65,535 バイトを仮定)

(ソート結果)

	0	1	2	3	4	5	6	7	8	9	10 (実レコード長)
レコード3	a	b	c	d	LF						4バイト (最小)
レコード2	a	b	c	d	e	LF					5バイト
レコード4	a	b	c	d	e	f	LF				6バイト
レコード6	a	b	c	d	e	f	g	LF			7バイト
レコード5	a	b	c	d	e	f	g	h	LF		8バイト (最大)
レコード1	u	v	w	x	y	z	LF				6バイト

← キーの範囲 →

**【例 5】** キー位置が最大レコード長より小さく、キー位置+キー長が最大レコード長より大きい場合

(exsort コマンド指定内容)

-cmdopt POSNOCHK:DUP : 重複レコードの入力順序保存機能を指定

-key c+6-4 : キー位置 +6 ~ 9 キー長 4 バイト 昇順ソート

-inpf fileorg=A : テキストファイル (最大レコード長 65,535 バイトを仮定)

(ソート結果)

	0	1	2	3	4	5	6	7	8	9	10 (実レコード長)
レコード1	u	v	w	x	y	z	LF				6バイト
レコード2	a	b	c	d	e	LF					5バイト
レコード3	a	b	c	d	LF						4バイト (最小)
レコード4	a	b	c	d	e	f	LF				6バイト
レコード6	a	b	c	d	e	f	g	LF			7バイト
レコード5	a	b	c	d	e	f	g	h	LF		8バイト (最大)

← キーの範囲 →

- 重複レコードの入力順序保存機能の指定によって、レコード 4 ~ 6 の順番は変わりません。

**【例 6】** キー範囲が最大レコード長より大きい場合

(exsort コマンド指定内容)

-cmdopt POSNOCHK:DUP : 重複レコードの入力順序保存機能を指定

-key c+8-4088 : キー位置 +8 ~ 4,095 キー長 4,088 バイト 昇順ソート

-inpf fileorg=A : テキストファイル (最大レコード長 65,535 バイトを仮定)

## 5. exsort コマンドのパラメタおよびオペランドの詳細

(ソート結果)

	0	1	2	3	4	5	6	7	← キーの範囲 →				
									8	9	~	4095	(実レコード長)
レコード1	u	v	w	x	y	z	LF						6バイト
レコード2	a	b	c	d	e	LF							5バイト
レコード3	a	b	c	d	LF								4バイト (最小)
レコード4	a	b	c	d	e	f	LF						6バイト
レコード5	a	b	c	d	e	f	g	h	LF				8バイト (最大)
レコード6	a	b	c	d	e	f	g	LF					7バイト

- 重複レコードの入力順序保存機能の指定によって、レコード1～6の順番は変わりません。

### 5.5.12 日付データのチェックオプション (DATENOCHK オプション)

西暦下2桁年号に空白 (X'2020'), Low\_value (X'0000'), High\_value (X'FFFF') が設定されていても、エラーとしないで文字として処理されます。

CSV ファイルの場合、西暦下2桁年号または西暦4桁年号に数字以外の文字が設定されていても、エラーとしないで文字として処理されます。

ソート、マージ、選択、および集約機能でキーまたは比較項目が不正な場合の例を次に示します。

#### (a) ソート機能でキーが不正な場合

売上日付をキーにしてソートする例を示します。



## (1) DATENOCHKの指定がない場合

入力ファイル

売上日付	品番
00/10/20	AAA
58/02/07	BBB
74/12/21	CCC
75/05/30	DDD
空白	EEE
LOW-VALUE	FFF
HIGH-VALUE	GGG



exsortコマンドエラー

「KBLS1211-E キーにエラーデータを検出しました。」

## (2) DATENOCHKの指定がある場合

入力ファイル

売上日付	品番
00/10/20	AAA
58/02/07	BBB
74/12/21	CCC
75/05/30	DDD
空白	EEE
LOW-VALUE	FFF
HIGH-VALUE	GGG



出力ファイル

売上日付	品番
LOW-VALUE	FFF
空白	EEE
75/05/30	DDD
00/10/20	AAA
58/02/07	BBB
74/12/21	CCC
HIGH-VALUE	GGG

(凡例)

■ : 不正なデータを示す。

空白 : (x' 20')

HIGH-VALUE : (x' ff')

LOW-VALUE : (x' 00')

注 基準年度は「75」とする。

## (b) マージ機能でキーが不正な場合

売上日付をキーにしてマージする例を示します。

5. exsort コマンドのパラメタおよびオペランドの詳細

(1) DATENOCHKの指定がない場合

入力ファイルA

売上日付	品番
LOW-VALUE	AAA
75/05/30	BBB
74/12/21	CCC
HIGH-VALUE	DDD



exsortコマンドエラー  
「KBLS1211-E キーにエラーデータを検出しました。」

入力ファイルB

売上日付	品番
空白	EEE
00/10/20	FFF
58/02/07	GGG

(2) DATENOCHKの指定がある場合

入力ファイルA

売上日付	品番
LOW-VALUE	AAA
75/05/30	BBB
74/12/21	CCC
HIGH-VALUE	DDD



出力ファイル

売上日付	品番
LOW-VALUE	AAA
空白	EEE
75/05/30	BBB
00/10/20	FFF
58/02/07	GGG
74/12/21	CCC
HIGH-VALUE	DDD

入力ファイルB

売上日付	品番
空白	EEE
00/10/20	FFF
58/02/07	GGG

(凡例)

■ : 不正なデータを示す。

空白 : (x' 20')

HIGH-VALUE : (x' ff')

LOW-VALUE : (x' 00')

注 基準年度は「75」とする。

(c) 選択機能で比較項目が不正な場合

売上日付を比較項目にして選択する例を示します。

(1) DATENOCHKの指定がない場合

条件値

売上日付&gt;00年以降

入力ファイル

売上日付	品番
LOW-VALUE	AAA
空白	BBB
75/05/30	CCC
00/10/20	DDD
58/02/07	EEE
HIGH-VALUE	FFF



exsortコマンドエラー

「KBLS1320-E 比較項目にエラーデータを検出しました。」

(2) DATENOCHKの指定がある場合

条件値

売上日付&lt;=1999年

入力ファイル

売上日付	品番
LOW-VALUE	AAA
空白	BBB
75/05/30	CCC
00/10/20	DDD
58/02/07	EEE
HIGH-VALUE	FFF



出力ファイル

売上日付	品番
LOW-VALUE	AAA
空白	BBB
75/05/30	CCC

条件値

売上日付&gt;=2000年

入力ファイル

売上日付	品番
LOW-VALUE	AAA
空白	BBB
75/05/30	CCC
00/10/20	DDD
58/02/07	EEE
HIGH-VALUE	FFF



出力ファイル

売上日付	品番
00/10/20	DDD
58/02/07	EEE
HIGH-VALUE	FFF

(凡例)

■ : 不正なデータを示す。

空白 : (x' 20')

HIGH-VALUE : (x' ff')

LOW-VALUE : (x' 00')

注 基準年度は「75」とする。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

### (d) 集約機能でキーが不正な場合

売上日付をキーに、売上数量を集約項目にして集約する例を示します。

#### (1) DATENOCHKの指定がない場合

入力ファイル

売上日付	売上数量
LOW-VALUE	30
LOW-VALUE	40
空白	99
75/05/30	10
75/05/30	79
98/12/31	74
74/12/21	25
74/12/21	18
HIGH-VALUE	75



exsortコマンドエラー

「KBLS1211-E キーにエラーデータを検出しました。」

#### (2) DATENOCHKの指定がある場合

入力ファイル

売上日付	売上数量
LOW-VALUE	30
LOW-VALUE	40
空白	99
75/05/30	10
75/05/30	79
98/12/31	74
74/12/21	25
74/12/21	18
HIGH-VALUE	75



出力ファイル

売上日付	売上数量
LOW-VALUE	70
空白	99
75/05/30	89
98/12/31	74
74/12/21	43
HIGH-VALUE	75

#### (凡例)

■ : 不正なデータを示す。

LOW-VALUE : (x' 00')

空白 : (x' 20')

HIGH-VALUE : (x' ff')

注 基準年度は「75」とする。

#### 注意事項

- キー属性または比較項目属性で、「g (日付)」を指定していない場合は、このオプションは無効となります。
- このオプションを指定しない場合、環境変数「EXSORTMGTOPT (64ビット版は「EXSORT64MGTOPT」)」に DATE\_INVALID\_NOCHK オプションが指定さ

れていれば DATENOCHK オプション指定ありと同じ扱いとなります。

DATENOCHK	不正データの扱い※
指定なし	KBLS1211-E メッセージを表示し、exsort コマンドが異常終了する。
指定あり	JIS8 単位コード体系でソートキーの大小比較をする（西暦下 2 桁年号であっても基準年を適用しないで、文字として処理されます）。

#### 注※

不正データと見なされる条件を次の表に示します。

ファイル編成	不正データの条件	備考
可変長順編成ファイル 固定長順編成ファイル テキストファイル	キーの先頭 2 文字（西暦下 2 桁の数値）が指すデータ内容が次にどれかを満たす場合は不正データと見なされます。 <ul style="list-style-type: none"> <li>ゾーン部が 3 以外の値</li> <li>数値部が 0 ~ 9 以外の値</li> </ul>	「外部 10 進数データ形式オプション」でデータ形式を変更できます。オプションを指定した場合の不正データ形式については、「14.1.1(31) EXSORTUNPACKDATATYPE」および「14.1.2(31) EXSORT64UNPACKDATATYPE」を参照してください。
CSV ファイル	キーの年号（西暦下 2 桁年号または西暦 4 桁年号）部分のデータ内容が、0 ~ 9 の数字 (X'30' ~ X'39') 以外の文字の場合は、不正データと見なされます。	除外文字の位置規則については、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。

### 5.5.13 日付時刻データのチェックオプション (DATETIMENOCHK オプション)

キーまたは比較項目の属性が日付時刻の場合、入力データの形式が不正でも、エラーにしません。

ソートキーが不正な場合の例を次に示します。

5. exsort コマンドのパラメタおよびオペランドの詳細

(a) DATETIMENOCHKの指定がない場合

入力ファイル

売上日時	品番
2013/08/19T16:39	AAA
2013/00/00T00:00	BBB
2013/01/01T13:25	CCC
2013	DDD
△△△	EEE
2013/aaa	FFF
201/03/04T05:08	GGG



exsortコマンドエラー (KBLS1211-E)

(b) DATETIMENOCHKの指定がある場合

入力ファイル

売上日時	品番
2013/08/19T16:39	AAA
2013/00/00T00:00	BBB
2013/01/01T13:25	CCC
2013	DDD
△△△	EEE
2013/aaa	FFF
201/03/04T05:08	GGG



出力ファイル

売上日時	品番
△△△	EEE
201/03/04T05:08	GGG
2013/00/00T00:00	BBB
2013	DDD
2013/01/01T13:25	CCC
2013/08/19T16:39	AAA
2013/aaa	FFF

(凡例)

■ : 不正なデータを示す。

△ : 空白を示す。

「T」, 「/」, 「:」 : 区切り, または桁修飾文字を示す。

注意事項

- CSV ファイルだけに有効なオプションです。
- キー属性または比較項目属性で、「h (日付時刻)」を指定していない場合は、このオプションは無効となります。
- このオプションを指定しない場合、環境変数「EXSORTMGTOPT (64ビット版は「EXSORT64MGTOPT)」」に DATETIME\_INVALID\_NOCHK オプションが指定されていれば、コマンドオプション「-cmdopt DATETIMENOCHK」指定ありと同じ扱いとなります。

DATETIMENOCHK	不正データの扱い※
指定なし	KBLS1211-E, または KBLS1320-E メッセージを表示し, exsort コマンドが異常終了します。

DATETIMENOCHK	不正データの扱い※
指定あり	エンコードに従って文字として解析し、ソートキーおよび比較項目の大小比較を実施します。 <ul style="list-style-type: none"> <li>・年の桁が 4 桁未満の場合、不足の桁数分、0（文字）を埋める。</li> <li>・年の設定がなかった場合は 1900（文字）を仮定する。</li> <li>・月日の設定がなかった場合は 01（文字）を仮定する。</li> <li>・時分秒ミリ秒の設定がなかった場合は 0（文字）を仮定する。</li> <li>・日付、時刻またがりは考慮しない。</li> </ul>

## 注※

不正データと見なされる条件を次に示します。

- ・日付時刻データ内容が、0～9の数字以外の場合
- ・年データの桁数が、4 未満の場合
- ・日付時刻データが、指定可能な範囲以外の場合
- ・12 時間制表記が指定され、時刻データが 87 を超える場合

### 5.5.14 入出力ファイル共用・排他オプション（NOLOCK | EXC オプション）

SORT, SORT EE, COBOL, ISAM プログラムの処理間での入出力ファイルのオープンモードを指定します。

## EXC

入力ファイルを排他モードでオープンします。

## NOLOCK

入力ファイルおよび出力ファイルを、共用・排他なしでオープンします。

入力ファイルと出力ファイルのファイル状態とオープンの可否を次の表に示します。

（入力ファイル）

command-option	SORT EE での OPEN モード	入力ファイルの状態	
		共用	排他
NOLOCK	共用・排他なし	○	○
指定なし	共用	○	×
EXC	排他	×	×

（出力ファイル）

command-option	SORT EE での OPEN モード	出力ファイルの状態	
		共用	排他
NOLOCK	共用・排他なし	○	○

## 5. exsort コマンドのパラメタおよびオペランドの詳細

command-option	SORT EE での OPEN モード	出力ファイルの状態	
		共用	排他
指定なし	排他	×	×

(凡例)

- ：ほかの処理でオープンできます。
- ×

このオプションを指定する場合は、次の点に注意してください。

- NOLOCK と EXC は背反です。両方のオプションを同時に指定した場合は、NOLOCK が有効となります。
- NOLOCK, EXC の両方を指定しなかった場合、環境変数「EXSORTMGTOPT (64 ビット版は「EXSORT64MGTOPT」)」に LOCK\_INVALID オプションが指定されていれば NOLOCK, 指定がなければ共用と同じ扱いになります。
- NOLOCK を指定した場合、SORT, SORT EE, COBOL, ISAM 間での入出力ファイルの共有・排他機能が無効となります。4 製品で同一ファイルを指定して、同時実行した場合、入出力ファイルの内容は保証できません。
- 書き込みが許可されていないファイルは、排他モードでオープンできません（読み出しだけをする場合にも、書き込みの許可が必要となります）。
- すでにほかの処理でオープンされているファイルは、排他モードでオープンできません。
- NOLOCK オプションは UNIX にだけ有効です。Windows で指定した場合は無視します。

### 5.5.15 キャッシュファイルの同期出力オプション (SYNC オプション)

出力データがキャッシュファイルに残っている場合、すべてのデータをファイル出力したあと、コマンドを終了します。

SORT EE で使用できるすべてのファイル編成（固定長順編成ファイル、可変長順編成ファイル、テキストファイルおよび CSV ファイル）で有効です。

### 5.5.16 強制ファイル分割機能 (FSPLIT オプション)

分割キーによるファイル分割時、分割する入力ファイルは昇順または降順にソートされている必要があります。ソートされていない場合、KBLS2030-E エラーとなります。しかし、このオプションを指定することで、ソートされていないでもエラーとならずにファイルを分割できます。ただし、キー順序の並び替えをしないで分割するので、大量



の分割ファイルが作成されるおそれがあります。注意してください。

### 5.5.17 ファイル逐次 OPEN/CLOSE 機能 (SEQOPNCLS オプション)

exsort コマンドでは、指定されたすべてのファイルを一度にオープンしているため、多くのファイルを指定する、ファイル分割機能またはファイル連結機能では、システムの制限によって I/O エラーとなることがあります。ファイル逐次 OPEN/CLOSE 機能は、ファイル分割時の出力ファイル、およびファイル連結時の入力ファイルに対するオープンとクローズを、全ファイル一括で行わずに入出力するファイルごとに行います。ファイル分割機能およびファイル連結機能使用時に、システムのファイルディスクリプタの制限によってエラーとなっていた場合は、ファイル逐次 OPEN/CLOSE 機能を指定してエラーを回避してください。ただし、SORT EE、SORT、COBOL、および ISAM 間での入出力ファイルの共有・排他機能は、ファイルがオープンされている間だけ有効であるため、exsort コマンド実行中であっても、オープンされていない入出力ファイルの内容は保証できません。

ファイル逐次 OPEN/CLOSE 機能は、ソートとコピー機能使用時だけに有効であり、マージ機能使用時は有効となりません。

### 5.5.18 連続区切り文字オプション (INPCONSEPA, OUTCONSEPA オプション)

連続した区切り文字 (ConsecutiveDelimiter) を 1 つのデリミタ (区切り文字) として扱います。

#### INPCONSEPA

入力レコードの連続したデリミタを 1 つのデリミタとして扱います。

このオプションの指定によって、入力レコードのデータそのものが変更されることはありません。

#### OUTCONSEPA

レコード出力時に、連続したデリミタを 1 つのデリミタに置き換えて出力します。

このオプションを指定すると、INPCONSEPA も同時に有効となります。また、デリミタの置き換えによって、出力レコード長が変わる場合があるので注意してください。

#### 注意事項

- 入力ファイルが CSV ファイルの場合だけ有効なオプションです。CSV ファイル以外に指定しても無効です。
- 複数種類のデリミタを指定している場合、異なるデリミタでも隣接している場合は 1 つのデリミタとして扱います。OUTCONSEPA オプション指定では、最初のデリミタに置き換えます。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- 入力レコードのデリミタの指定については「5.8.8 cellsepa オペランド」を参照してください。
- 出力レコードのデリミタを変更する場合は、`-outfile`、`-catout` または `-splitout` パラメタの `cellsepa` オペランドで変更したいデリミタを指定します。指定方法については「5.10.6 cellsepa オペランド」を参照してください。
- CSV のレコード編集では、セルとセルの間のデリミタは必ず 1 文字だけを配置します。連続しているデリミタの移動はできません。また、連続しているデリミタは 1 文字にまとめられるので注意してください。なお、レコード編集を指定するとデリミタは必ず 1 文字で配置されることになるので、レコード編集と `OUTCONSEPA` オプションを同時に指定した場合、`OUTCONSEPA` オプションは無効となります。CSV のレコード編集機能の詳細については、「5.8.15 csvreorg オペランド」を参照してください。

### 指定例

- 入力ファイルのデリミタに半角空白とタブの 2 種類を指定
- 改行文字は「LF」を指定

(入力レコード)

レコード1	Tokyo△△→△Japan△65723
レコード2	Paris→→France△△△△△858363

(凡例)

- △ : デリミタ (半角空白)
- : デリミタ (タブ)

### 【例 1】入力レコードの連続区切り文字オプション指定例

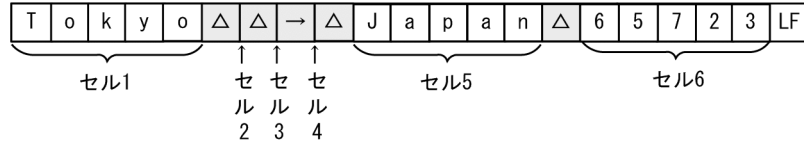
- 「`-cmdopt INPCONSEPA`」指定なし  
「`INPCONSEPA`」の指定がないので、連続したデリミタの間はナル文字列 (空のセル) として扱います。  
入力セル数は、レコード 1 は 6 個 (7, 8 個を参照した場合はナル文字列を仮定)、レコード 2 は 8 個と解釈されます。

(入力セルデータ)

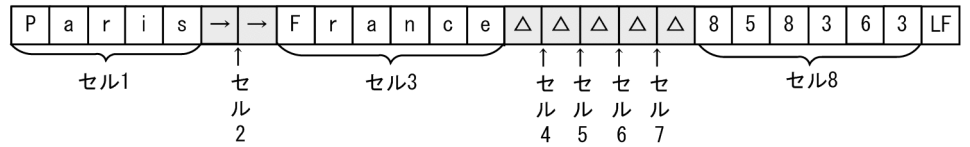
	セル1	セル2	セル3	セル4	セル5	セル6	セル7	セル8
レコード1	Tokyo	(null)	(null)	(null)	Japan	65723	(null)	(null)
レコード2	Paris	(null)	France	(null)	(null)	(null)	(null)	858363

(凡例) (null) : ナル文字列 (空のセル)

(レコード1)



(レコード2)

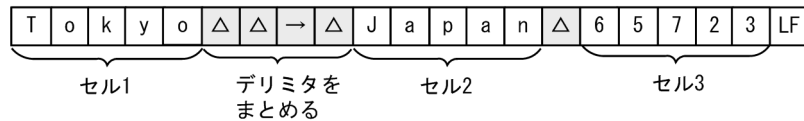


- 「-cmdopt INPCONSEPA」指定あり  
連続のデリミタを1つのデリミタとして扱います。  
入力セル数は3個と解釈されます。

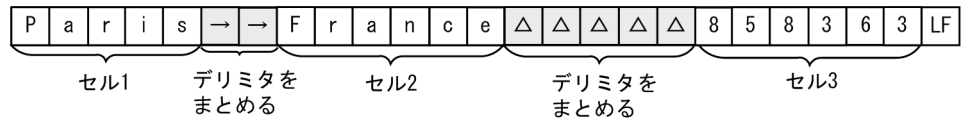
(入力セルデータ)

	セル1	セル2	セル3
レコード1	Tokyo	Japan	65723
レコード2	Paris	France	858363

(レコード1)



(レコード2)



**【例2】出力レコードの連続区切り文字オプション指定例**

- 「-cmdopt OUTCONSEPA」指定あり  
連続したデリミタを、最初に出現したデリミタに置き換えて出力します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

(出力レコード)

レコード1	Tokyo△Japan△65723
レコード2	Paris→France△858363

- 「-cmdopt OUTCONSEPA」および「-outfile cellsepa="'△',' ','TAB '\*'"」指定あり  
連続したデリミタを、最初に出現したデリミタに対で指定されている出力デリミタに置き換えて出力します。

入力デリミタ	出力デリミタ
△ (半角空白)	, (コンマ)
→ (タブ)	* (アスタリスク)

(出力レコード)

レコード1	Tokyo,Japan,65723
レコード2	Paris*France,858363

### 【例 3】 出力レコードの連続区切り文字オプション「OUTCONSEPA」とレコード編集を同時に指定した例

出力レコード編集で、最初に定義したデリミタ (半角空白) が 1 個配置されま  
す。

レコード編集によって連続デリミタではなくなるので、「OUTCONSEPA」は無効  
となります。

(exsort コマンド指定パラメタ)

```
-cmdopt OUTCONSEPA
-outfile
  csvreorg="1+3-1 +2-1 +1-1"
```

(出力レコード編集後のレコード)

レコード1	65723△Japan△Tokyo
レコード2	858363△France△Paris



OUTCONSEPA指定による  
変更はありません。

(出力レコード)

レコード1	65723△Japan△Tokyo
レコード2	858363△France△Paris

### 5.5.19 ナル文字列の集約オプション (SUMNULKEY, SUMNONULL オプション)

集約項目、集約キーがナル文字列の場合の扱いを指定します。

#### SUMNULKEY

ナル文字列 (空のセル) の集約キーを同値と扱いません。

集約キーがナル文字列の集約項目は集約されないで、1 個のレコードで出力されます。

#### SUMNONULL

ナル文字列 (空のセル) の集約項目を集約対象としません。

ナル文字列の集約項目は集約されないで、1 個のレコードで出力されます。

#### 注意事項

- 入力ファイルが CSV ファイルの場合だけ有効なオプションです。CSV ファイル以外に指定しても無効です。  
SUMNULKEY と SUMNONULL は、同時に指定できます。
- SUMNULKEY オプションを指定しない場合、ナル文字列のキーを同値として扱い、集約項目は集約されます。
- SUMNONULL オプションを指定しない場合、ナル文字列の集約項目は 0 (ゼロ) として扱い、集約します。
- 集約キーおよび集約項目のナル文字列の扱いについては、「付録 D CSV ファイル内のナル文字列の扱い (SORT EE の機能)」を参照してください。

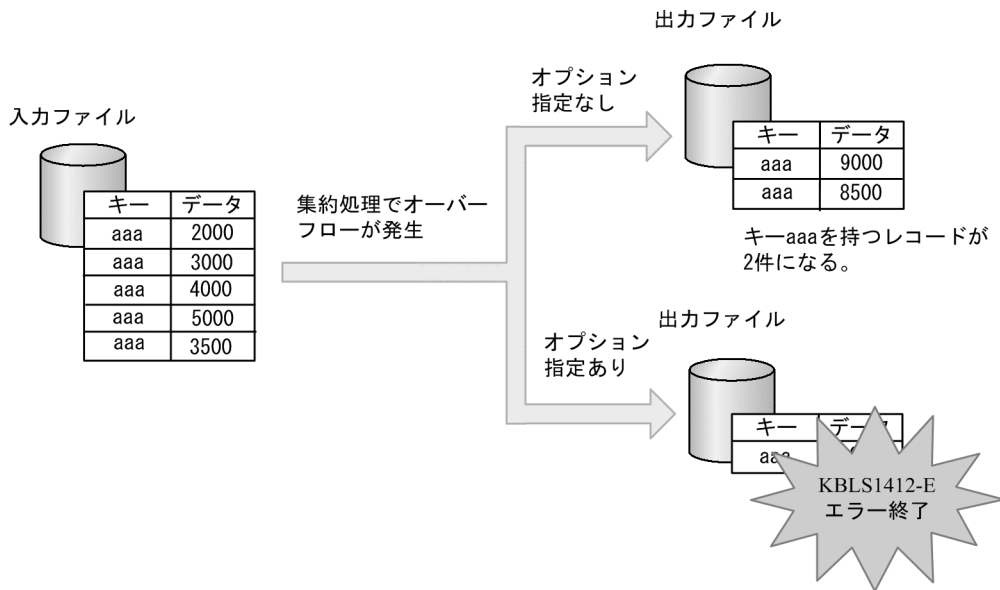
### 5.5.20 集約結果オーバーフロー検知オプション (OVFERR オプション)

集約項目の合計でオーバーフローが発生した場合、KBLS1412-E エラー (集約結果がオーバーフローしました) で exsort コマンドを異常終了させるオプションです。

オプションの指定がない場合は、オーバーフローが発生したところでレコードを分けて出力し、exsort コマンドは正常終了します。

集約結果オーバーフロー検知オプションの概要を次の図に示します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細



### 注意事項

- 集約項目にだけ有効なオプションです。
- 無条件レコード抽出機能では、このオプションは無効です。
- 10進数オーバーフロー判定オプション (OVFLW オプション) と同時に指定できません。
- 集約結果がオーバーフローでエラーとなった場合でも、出力ファイルが作成されます。不要な場合は削除してください。
- 集約結果のオーバーフローの発生によって exsort コマンドが異常終了した場合、トラブルシューティング情報をデバッグ情報ファイルに出力します。デバッグ情報ファイルについては、「19.1 SORT EE の障害情報」を参照してください。

## 5.5.21 シフト JIS コード比較オプション (SJIS オプション)

シフト JIS コードの体系に従って、ソートキー、マージキー、集約キー、分割キー、および比較項目を文字単位で比較します。

キー、比較項目にシングルバイト文字 (英数字、半角カナ)、およびマルチバイト文字 (漢字などの全角文字) が混在している場合、シングルバイト文字はマルチバイト文字より先に比較します。

### 注意事項

- キー、または比較項目の属性に「c」を指定していない場合、このオプションは無効となります。
- シフト JIS コードのマルチバイト文字 (2 バイト文字) の範囲を次の表に示します。

バイト位置	範囲
第1バイト (上位バイト)	0x81 ~ 0x9F, 0xE0 ~ 0xEF
第2バイト (下位バイト)	0x40 ~ 0x7E, 0x80 ~ 0xFC

## 指定例

## 【例1】

-key パラメタに指定されたソートキーを例題に、オプション有無のソート結果を次に示します。

入力レコードの +0 ~ 5 バイト目の長さ 6 バイトをソートキーとして昇順ソートします。網掛けはキーの範囲を示します。

(入カデータ)

	+0	1	2	3	4	5	6
レコード1	E4	95	20	20	20	20	0A
	莓		△	△	△	△	LF
レコード2	B2	C1	BA	DE	20	20	0A
	イ	チ	コ	ゝ	△	△	LF
レコード3	82	A2	82	BF	BA	DE	0A
	い		ち		コ	ゝ	LF
レコード4	82	A2	82	BF	82	B2	0A
	い		ち		ご		LF
レコード5	69	63	68	69	67	6F	0A
	i	c	h	i	g	o	LF
レコード6	49	43	48	49	47	4F	0A
	I	C	H	I	G	O	LF
レコード7	31	35	20	20	20	20	0A
	1	5	△	△	△	△	LF
レコード8	2A	2A	2A	2A	2A	2A	0A
	*	*	*	*	*	*	LF

5. exsort コマンドのパラメタおよびオペランドの詳細

(SJISオプションなしのソート結果)

	+0	1	2	3	4	5	6
レコード8	2A	2A	2A	2A	2A	2A	0A
	*	*	*	*	*	*	LF
レコード7	31	35	20	20	20	20	0A
	1	5	△	△	△	△	LF
レコード6	49	43	48	49	47	4F	0A
	I	G	H	I	G	O	LF
レコード5	69	63	68	69	67	6F	0A
	i	c	h	i	g	o	LF
レコード4	82	A2	82	BF	82	B2	0A
	い		ち		ご		LF
レコード3	82	A2	82	BF	BA	DE	0A
	い		ち	こ	ゝ		LF
レコード2	B2	C1	BA	DE	20	20	0A
	イ	チ	コ	ゝ	△	△	LF
レコード1	E4	95	20	20	20	20	0A
	莓		△	△	△	△	LF

↑  
1バイト単位で比較

SJIS オプションなしの場合、キーデータは「JIS8 単位コード体系」に従って、1 バイト単位で比較します。ソート結果は、半角カナ文字より前にマルチバイト文字が配置されます。



(SJISオプションありのソート結果)

	出力データ							キーデータ内部表現											
	+0	1	2	3	4	5	6	+0	1	2	3	4	5	6	7	8	9	10	11
レコード8	2A	2A	2A	2A	2A	2A	0A	00	2A	00	2A	00	2A	00	2A	00	2A	00	2A
	*	*	*	*	*	*	LF		*		*		*		*		*		*
レコード7	31	35	20	20	20	20	0A	00	31	00	35	00	20	00	20	00	20	00	20
	1	5	△	△	△	△	LF		1		5		△		△		△		△
レコード6	49	43	48	49	47	4F	0A	00	49	00	43	00	48	00	49	00	47	00	4F
	I	C	H	I	G	O	LF		I		C		H		I		G		O
レコード5	69	63	68	69	67	6F	0A	00	69	00	63	00	68	00	69	00	67	00	6F
	i	c	h	i	g	o	LF		i		c		h		i		g		o
レコード2	B2	C1	BA	DE	20	20	0A	00	B2	00	C1	00	BA	00	DE	00	20	00	20
	イ	チ	コ	△	△	△	LF		イ		チ		コ		△		△		△
レコード3	82	A2	82	BF	BA	DE	0A	82	A2	82	BF	00	BA	00	DE	00	00	00	00
	い		ち	コ	△	△	LF		い		ち		コ		△		△		△
レコード4	82	A2	82	BF	82	B2	0A	82	A2	82	BF	82	B2	00	00	00	00	00	00
	い		ち		ご	△	LF		い		ち		ご		△		△		△
レコード1	E4	95	20	20	20	20	0A	E4	95	00	20	00	20	00	20	00	20	00	20
	莓	△	△	△	△	△	LF		莓		△		△		△		△		△

↑  
文字単位で比較

■ 0x00の補正

SJIS オプションありの場合、キーデータは「シフト JIS コード」に従って、文字単位で比較します。シングルバイト文字については、上位バイトに 0x00 を付加した 2 バイト長に、また、文字数が足りない場合（レコード 1, 3, 4）は、不足数分の 2 バイトの 0x00 を補正した内部表現で比較します。

ソート結果は、半角カナ文字より後ろにマルチバイト文字が配置されます。

**【例 2】**

-csvkey パラメタに指定されたソートキーを例題に、オプション有無のソート結果を次に示します。

入力レコードの第 1 セルの有効桁長 6 バイトをキーとして昇順ソートします。

セルのデリミタ（区切り文字）は、コンマ（0x2C）とします。網掛けはキーの範囲を示します。

5. exsort コマンドのパラメタおよびオペランドの詳細

(入力データ)

	+0	1	2	3	4	5	6
レコード1	E4	95	2C	20	20	20	0A
	莓		,	△	△	△	LF
レコード2	B2	C1	BA	DE	2C	20	0A
	イ	チ	コ	ゝ	,	△	LF
レコード3	82	A2	82	BF	BA	DE	0A
	い		ち		コ	ゝ	LF
レコード4	82	A2	82	BF	82	B2	0A
	い		ち		ご		LF
レコード5	69	63	68	69	67	6F	0A
	i	c	h	i	g	o	LF
レコード6	49	43	48	49	47	4F	0A
	I	C	H	I	G	O	LF
レコード7	31	35	2C	20	20	20	0A
	1	5	,	△	△	△	LF
レコード8	2A	2A	2A	2A	2A	2A	0A
	*	*	*	*	*	*	LF

(SJISオプションなしのソート結果)

	+0	1	2	3	4	5	6
レコード8	2A	2A	2A	2A	2A	2A	0A
	*	*	*	*	*	*	LF
レコード7	31	35	2C	20	20	20	0A
	1	5	,	△	△	△	LF
レコード6	49	43	48	49	47	4F	0A
	I	C	H	I	G	O	LF
レコード5	69	63	68	69	67	6F	0A
	i	c	h	i	g	o	LF
レコード4	82	A2	82	BF	82	B2	0A
	い		ち		ご		LF
レコード3	82	A2	82	BF	BA	DE	0A
	い		ち		コ	ゝ	LF
レコード2	B2	C1	BA	DE	2C	20	0A
	イ	チ	コ	ゝ	,	△	LF
レコード1	E4	95	2C	20	20	20	0A
	莓		,	△	△	△	LF

↑  
1バイト単位で比較

SJIS オプションなしの場合、データは「JIS8 単位コード体系」に従って、1 バイト単位で比較します。ソート結果は、半角カナ文字より前にマルチバイト文

字が配置されます。

(SJISオプションありのソート結果)

	出力データ							キーデータ内部表現											
	+0	1	2	3	4	5	6	+0	1	2	3	4	5	6	7	8	9	10	11
レコード8	2A	2A	2A	2A	2A	2A	0A	00	2A	00	2A	00	2A	00	2A	00	2A	00	2A
	*	*	*	*	*	*	LF		*		*		*		*		*		*
レコード7	31	35	2C	20	20	20	0A	00	31	00	35	00	00	00	00	00	00	00	
	1	5	,	△	△	△	LF		1		5								
レコード6	49	43	48	49	47	4F	0A	00	49	00	43	00	48	00	49	00	47	00	
	I	C	H	I	G	O	LF		I		C		H		I		G	O	
レコード5	69	63	68	69	67	6F	0A	00	69	00	63	00	68	00	69	00	67	00	
	i	c	h	i	g	o	LF		i		c		h		i		g	o	
レコード2	B2	C1	BA	DE	2C	20	0A	00	B2	00	C1	00	BA	00	DE	00	00	00	
	イ	チ	コ	ゝ	,	△	LF		イ		チ		コ		ゝ				
レコード3	82	A2	82	BF	BA	DE	0A	82	A2	82	BF	00	BA	00	DE	00	00	00	
	い		ち	コ	ゝ		LF		い		ち		コ		ゝ				
レコード4	82	A2	82	BF	82	B2	0A	82	A2	82	BF	82	B2	00	00	00	00	00	
	い		ち		ご		LF		い		ち		ご						
レコード1	E4	95	2C	20	20	20	0A	E4	95	00	00	00	00	00	00	00	00	00	
	莓	,	△	△	△		LF		莓										

↑  
文字単位で比較

■ 0x00の補正

SJIS オプションありの場合、データは「シフト JIS コード」に従って、文字単位で比較します。

シングルバイト文字については、上位バイトに 0x00 を付加した 2 バイト長に、また、文字数が足りない場合（レコード 1, 3, 4）は、不足数分の 2 バイトの 0x00 を補正した内部表現で比較します。

ソート結果は、半角カナ文字より後ろにマルチバイト文字が配置されます。

(注) CSV データの有効桁の調整について

実際に入力したキーデータに過不足がある場合、桁調整を行います。

- データ長 < 有効桁数  
入力したデータの右側に空白を加え、有効バイト数に調整します。
- データ長 > 有効桁数

## 5. exsort コマンドのパラメタおよびオペランドの詳細

入力したデータの右側の桁あふれ部分を無視します。

詳細については、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。

### 5.5.22 ゼロデータ比較オプション (DIFFZERO オプション)

左分離符号付き数字「d」のキーおよび比較項目において、負符号付きゼロ (-0) を、正符号付きゼロ (+0) より小さい値として扱います。なお、符号なしゼロ (0) と正符号付きゼロ (+0) は同じ値として扱います。

● DIFFZERO オプションなし

$-0 = 0 = +0$

● DIFFZERO オプションあり

$-0 < 0$

$-0 < +0$

$0 = +0$

#### 注意事項

キー属性または比較項目の属性に、「d (左分離符号付き数字 (文字))」を指定していなければ、このオプションは無効となります。

重複レコードの入力順序保存機能 (-cmdopt パラメタの DUP オプション) と同時に指定できます。

#### 指定例

【例 1】

-key パラメタに指定されたソートキーを例題に、オプション有無のソート結果を次に示します。

入力レコードの +0 ~ 3 バイト目の長さ 4 バイトをソートキーとして昇順ソートします。

(入力データ)

レコード1	△△+0
レコード2	+001
レコード3	+012
レコード4	+123
レコード5	0000
レコード6	△△12
レコード7	△123
レコード8	a123
レコード9	あ12
レコード10	7123
レコード11	△*12
レコード12	-000
レコード13	-001
レコード14	-012
レコード15	-123

(凡例)

△ : 半角スペース

5. exsort コマンドのパラメタおよびオペランドの詳細

(DIFFZEROオプションおよびDUPオプション指定なし)

入力データ		内部表現	
レコード15	-123	-0123	
レコード14	-012	-0012	
レコード13	-001	-0001	
レコード1	△△+0	+0000	※1
レコード12	-000	-0000	
レコード11	△*12	+0000	※1, ※2
レコード10	7123	+0000	※2
レコード9	あ12	+0000	※2
レコード8	a123	+0000	※2
レコード5	0000	+0000	※3
レコード2	+001	+0001	
レコード3	+012	+0012	
レコード6	△△12	+0012	※1, ※3
レコード4	+123	+0123	
レコード7	△123	+0123	※1, ※3

「+0」, 「-0」 および 「0」 は  
 同じ値として扱います。  
 また、-cmdopt DUPの指定がないため、  
 同値の並び順は保証されません。

(凡例)

△ : 半角スペース

(DIFFZEROオプション指定なし, DUPオプション指定あり)

	入力データ	内部表現	
レコード15	-123	-0123	
レコード14	-012	-0012	
レコード13	-001	-0001	
レコード1	△△+0	+0000	※1
レコード5	0000	+0000	※3
レコード8	a123	+0000	※2
レコード9	あ12	+0000	※2
レコード10	7123	+0000	※2
レコード11	△*12	+0000	※1, ※2
レコード12	-000	-0000	
レコード2	+001	+0001	
レコード3	+012	+0012	
レコード6	△△12	+0012	※1, ※3
レコード4	+123	+0123	
レコード7	△123	+0123	※1, ※3

「+0」、 「-0」 および 「0」 は  
同じ値として扱います。  
また、 -cmdopt DUPが指定されて  
いるため、 同値の並び順を保証し  
ます。

(凡例)

△ : 半角スペース

5. exsort コマンドのパラメタおよびオペランドの詳細

(DIFFZEROオプション指定あり, DUPオプション指定なし)

	入力データ	内部表現	
レコード15	-123	-0123	
レコード14	-012	-0012	
レコード13	-001	-0001	
レコード12	-000	-0000	
レコード1	△△+0	+0000	※1
レコード11	△*12	+0000	※1, ※2
レコード10	7123	+0000	※2
レコード9	あ12	+0000	※2
レコード8	a123	+0000	※2
レコード5	0000	+0000	※3
レコード2	+001	+0001	
レコード3	+012	+0012	
レコード6	△△12	+0012	※1, ※3
レコード4	+123	+0123	
レコード7	△123	+0123	※1, ※3

「-0」は「+0」より小さい値として扱います。  
 また、-cmdopt DUPの指定がないため、同値の並び順は保証されません。

(凡例)

△ : 半角スペース



(DIFFZEROオプションおよびDUPオプション指定あり)

	入力データ	内部表現	
レコード15	-123	-0123	
レコード14	-012	-0012	
レコード13	-001	-0001	
レコード12	-000	-0000	
レコード1	△△+0	+0000	※1
レコード5	0000	+0000	※3
レコード8	a123	+0000	※2
レコード9	あ12	+0000	※2
レコード10	7123	+0000	※2
レコード11	△*12	+0000	※1, ※2
レコード2	+001	+0001	
レコード3	+012	+0012	
レコード6	△△12	+0012	※1, ※3
レコード4	+123	+0123	
レコード7	△123	+0123	※1, ※3

「-0」は「+0」より小さい値として扱います。  
また、-cmdopt DUPが指定されているため、同値の並び順を保証します。

(凡例)

△：半角スペース

注※1 符号の前、および数値部の前後に存在した空白を無視します。

注※2 先頭文字が、符号、数字、空白、およびタブ以外は、正符号付きゼロ値として扱います。

注※3 符号なしのデータは、正の値として扱います。

**【例2】**

include オペランドに指定された比較項目を例題に、オプション有無の選択結果を次に示します。

比較項目の値が「-0」以下のレコードを選択します。

5. exsort コマンドのパラメタおよびオペランドの詳細

(DIFFZEROオプション指定なし)

入力データ		内部表現	⇒	選択結果	
レコード1	△△+0	+0000 ※1		△△+0	レコード1
レコード2	+001	+0001		0000	レコード5
レコード3	+012	+0012		a123	レコード8
レコード4	+123	+0123		あ12	レコード9
レコード5	0000	+0000 ※3		7123	レコード10
レコード6	△△12	+0012 ※1, ※3		△*12	レコード11
レコード7	△123	+0123 ※1, ※3		-000	レコード12
レコード8	a123	+0000 ※2		-001	レコード13
レコード9	あ12	+0000 ※2		-012	レコード14
レコード10	7123	+0000 ※2		-123	レコード15
レコード11	△*12	+0000 ※1, ※2			
レコード12	-000	-0000			
レコード13	-001	-0001			
レコード14	-012	-0012			
レコード15	-123	-0123			

(凡例)

△ : 半角スペース

「+ 0」、「- 0」および「0」は同じ値として扱います。0 以下（「+ 0」と「- 0」を含む）のデータを選択します。

(DIFFZEROオプション指定あり)

入力データ		内部表現	選択結果
レコード1	△△+0	+0000 ※1	-000 レコード12
レコード2	+001	+0001	-001 レコード13
レコード3	+012	+0012	-012 レコード14
レコード4	+123	+0123	-123 レコード15
レコード5	0000	+0000 ※3	
レコード6	△△12	+0012 ※1, ※3	
レコード7	△123	+0123 ※1, ※3	
レコード8	a123	+0000 ※2	
レコード9	あ12	+0000 ※2	
レコード10	7123	+0000 ※2	
レコード11	△*12	+0000 ※1, ※2	
レコード12	-000	-0000	
レコード13	-001	-0001	
レコード14	-012	-0012	
レコード15	-123	-0123	

(凡例)

△：半角スペース

「-0」は「+0」より小さい値として扱います。「-0」以下（「+0」は含みません）のデータを選択します。

注※1 符号の前、および数値部の前後に存在した空白を無視します。

注※2 先頭文字が、符号、数字、空白、およびタブ以外は、正符号付きゼロ値として扱います。

注※3 符号なしのデータは、正の値として扱います。

**【例3】**

include オペランドに指定された比較項目同士の比較を例題に、オプション有無の選択結果を次に示します。比較項目 A と比較項目 B が同じ値のレコードを選択します。

5. exsort コマンドのパラメタおよびオペランドの詳細

(入力データ)

	比較項目 A	比較項目 B
レコード1	△△-2	-002
レコード2	△△+0	+000
レコード3	△△+0	-000
レコード4	-2a0	-002
レコード5	ああ	-000
レコード6	a123	+△△0
レコード7	777△	△△△0
レコード8	△△△0	-a45
レコード9	+020	-000
レコード10	+2△△	△△+2
レコード11	+269	+043
レコード12	△181	+181
レコード13	+307	+131
レコード14	+△79	△△79
レコード15	-753	+587

(凡例)

△：半角スペース

(DIFFZEROオプション指定なし)

	入力データ		内部表現		選択結果				
	A	B	A	B	A	B			
レコード1	△△-2	-002	-0002	※1	-0002	△△-2	-002	レコード1	
レコード2	△△+0	+000	+0000	※1	+0000	△△+0	+000	レコード2	
レコード3	△+00	-000	+0000	※1	-0000	△+00	-000	レコード3	
レコード4	-2a0	-002	-0002	※2	-0002	-2a0	-002	レコード4	
レコード5	ああ	-000	+0000	※3	-0000	ああ	-000	レコード5	
レコード6	a123	+△△0	+0000	※3	+0000	※1	a123	+△△0	レコード6
レコード7	777△	△△△0	+0000	※3	+0000	※1, ※4	777△	△△△0	レコード7
レコード8	△△△0	-a45	+0000	※1, ※4	+0000	※3	△△△0	-a45	レコード8
レコード9	+020	-000	+0020		-0000		+2△△	△△+2	レコード10
レコード10	+2△△	△△+2	+0002	※1	+0002	※1	△181	+181	レコード12
レコード11	+269	+043	+0269		+0043		+△79	△△79	レコード14
レコード12	△181	+181	+0181	※1, ※4	+0181				
レコード13	+307	+131	+0307		+0131				
レコード14	+△79	△△79	+0079	※1	+0079	※1, ※4			
レコード15	-753	+587	-0753		+0587				

(凡例)

△：半角スペース

「+ 0」、「- 0」および「0」は同じ値として扱います。比較項目 A と比較項目 B が同じ値のレコードを選択します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

(DIFFZEROオプション指定あり)

入力データ			内部表現		選択結果				
A	B		A	B	A	B			
レコード1	△△-2	-002	-0002	※1	-0002	△△-2	-002	レコード1	
レコード2	△△+0	+000	+0000	※1	+0000	△△+0	+000	レコード2	
レコード3	△+00	-000	+0000	※1	-0000	-2a0	-002	レコード4	
レコード4	-2a0	-002	-0002	※2	-0002	a123	+△△0	レコード6	
レコード5	ああ	-000	+0000	※3	-0000	777△	△△△0	レコード7	
レコード6	a123	+△△0	+0000	※3	+0000	※1	△△△0	-a45	レコード8
レコード7	777△	△△△0	+0000	※3	+0000	※1, ※4	+2△△	△△+2	レコード10
レコード8	△△△0	-a45	+0000	※1, ※4	+0000	※3	△181	+181	レコード12
レコード9	+020	-000	+0020		-0000		+△79	△△79	レコード14
レコード10	+2△△	△△+2	+0002	※1	+0002	※1			
レコード11	+269	+043	+0269		+0043				
レコード12	△181	+181	+0181	※1, ※4	+0181				
レコード13	+307	+131	+0307		+0131				
レコード14	+△79	△△79	+0079	※1	+0079	※1, ※4			
レコード15	-753	+587	-0753		+0587				

(凡例)

△：半角スペース

「-0」は「+0」より小さい値として扱います。比較項目 A と比較項目 B が同じ値のレコードを選択します。「-0」と「+0」は選択されません。

注※1 符号の前、および数値部の前後に存在した空白を無視します。

注※2 数値部に数字と数字以外の文字が混在した場合、先頭から連続した数字だけを比較の対象とします。数字以外の文字より後ろは無視します。

注※3 先頭が符号、数字、空白およびタブでない場合、正符号付きゼロ値として扱います。

注※4 符号なしのデータは、正の値として扱います。

### 5.5.23 空行削除オプション (DELBLANKLINE オプション)

改行だけのデータがないレコード (空行) を削除して入力できます。出力ファイルには、空行は出力されません。

このオプションは、入力ファイル編成がテキストファイル、または CSV ファイルのときに有効となります。入力ファイル編成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は、指定しても無視されます。

入力ファイル編成	空行削除オプション	
	指定なし	指定あり
テキストファイル	KBLS1636-E エラー	空行を削除して入力
CSV ファイル	空行を入力	空行を削除して入力

このオプションを省略した場合、環境変数「EXSORTDELBLANKLINE (64 ビット版は「EXSORT64DELBLANKLINE)」」の指定を仮定します。

「EXSORTDELBLANKLINE」については、「14.1.1(32)EXSORTDELBLANKLINE」を、「EXSORT64DELBLANKLINE」については、「14.1.2(32)EXSORT64DELBLANKLINE」を参照してください。

また、入力レコードの読み飛ばし・打ち切り機能との関係を次に示します。

レコード件数の指定	空行削除オプション	
	指定なし	指定あり
skiprec (読み飛ばしレコード件数)	空行を含めた読み飛ばし件数を指定します。	空行を含めない読み飛ばし件数を指定します。
stopaft (打ち切りレコード件数)	空行を含めた打ち切り件数を指定します。	空行を含めない打ち切り件数を指定します。
startrec (入力開始レコード番号)	空行を含めた開始レコード番号を指定します。	空行を含めない開始レコード番号を指定します。

空行削除オプションと、入力レコードの読み飛ばし・打ち切り機能を組み合わせた例を次に示します。

**【例 1】** 改行削除オプションなし

レコード件数の指定 : skiprec=2, stopaft=6

(入力データ)	(件数)	
レコード1	1	} skiprec=2 (空行を含めた2件を読み飛ばす)
(空行)	2	
レコード3	1	} stopaft=6 (空行を含めた6件を入力して打ち切る)
(空行)	2	
レコード5	3	
レコード6	4	
レコード7	5	
(空行)	6	
レコード9		
(空行)		

**【例 2】** 改行削除オプションあり

5. exsort コマンドのパラメタおよびオペランドの詳細

レコード件数の指定 : skiprec=2, stopaft=6

(入力データ)	(件数)	
レコード1	1	} skiprec=2 (空行を除いた2件を読み飛ばす)
(空行)	—	
レコード3	2	}
(空行)	—	
レコード5	1	} stopaft=6 (空行を除いた5件を入力する) (stopaftオペランドに、入力レコードの件数より大きい数を指定した場合、すべてのレコードが入力の対象となる)
レコード6	2	
レコード7	3	
(空行)	—	
レコード9	5	
(空行)	—	

【例 3】 改行削除オプションなし

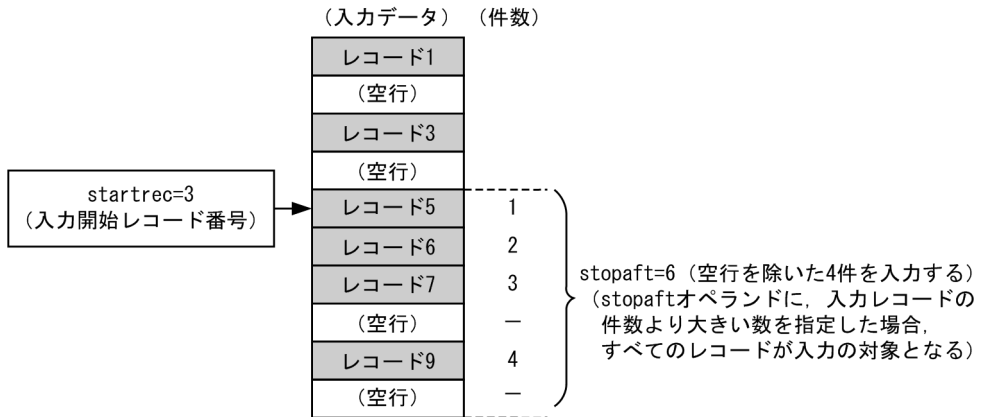
レコード件数の指定 : startrec=3, stopaft=6

	(入力データ)	(件数)	
<div style="border: 1px solid black; padding: 5px; display: inline-block;">                     startrec=3                      (入力開始レコード番号)                 </div> →	レコード1		} stopaft=6 (空行を含めた6件を入力して打ち切る)
	(空行)		
	レコード3	1	
	(空行)	2	
	レコード5	3	
	レコード6	4	
	レコード7	5	
	(空行)	6	
	レコード9		
(空行)			

【例 4】 改行削除オプションあり

レコード件数の指定 : startrec=3, stopaft=6





### 5.5.24 CSV 数値データ（固定小数点数）の形式チェックオプション（CSVFRMCHK オプション）

CSVFRMCHK オプションは、SORT EE 08-50 以降で有効なオプションです。

ソート機能、マージ機能、選択機能、または集約機能を使用する場合、次に示す条件のときは指定したキーまたは比較項目範囲以外を入力データの形式が不正であってもエラーになりません。

- キーまたは比較項目の属性が CSV 数値データ（固定小数点数）である。

指定値「CSVFRMCHK」が有効になる選択条件式の範囲を次の表に示します。

指定パラメタ	選択条件式（指定オペランド）			
	include	omit	csvinclude	csvomit
-infile	—	—	○	○
-outfile	—	—	○	○
-catinp	—	—	○	○
-catout	—	—	○	○
-splitinp	—	—	○	○
-splitout	—	—	○	○

(凡例)

- : 有効
- : 無効

CSV 数値データ（固定小数点数）の比較項目で、有効桁 5 桁（下線部分）を指定した場合のデータの扱い例を次の表に示します。

5. exsort コマンドのパラメタおよびオペランドの詳細

#	セルデータ	オプション 指定なし	KEYNOCHK オ プション指定	CSVFRMCHK オ プション指定	備考
1	1234 <u>5678</u>	45678	45678	45678	全文字が数字
2	AAA4 <u>5678</u>	不正	45678	45678	有効桁外が英字
3	123 <u>BBBBB</u>	不正	22222※	不正	有効桁内が英字

注※

文字コード 'B'(0x42) の下 4 ビットをパック化。

CSV 数値データ（固定小数点数）が不正な場合の例を次に示します。

項目「コード」の下3桁（下線部分）をキーにしてソートする。

(1) CSVFRMCHKの指定がない場合

入力ファイル

項番	コード
01	40868 <u>26</u>
02	Dkt76 <u>73</u>
03	Slk28 <u>26</u>
04	Ypg48 <u>26</u>
05	Ypg64 <u>31</u>



exsortコマンドエラー  
「KBLS1211-E キーにエラーデータを検出しました。」

(凡例)

下線：選択範囲

■：不正レコード

(2) CSVFRMCHKの指定がある場合

入力ファイル

項番	コード
01	40868 <u>26</u>
02	Dkt76 <u>73</u>
03	Slk28 <u>26</u>
04	Ypg48 <u>26</u>
05	Ypg64 <u>31</u>



出力ファイル

項番	コード
05	Ypg64 <u>31</u>
02	Dkt76 <u>73</u>
01	40868 <u>26</u>
03	Slk28 <u>26</u>
04	Ypg48 <u>26</u>

(凡例)

下線：選択範囲

■：不正データ

不正データは、キー範囲外のため項目「コード」の下3桁（下線部分）でレコードがソートされた。

項目「コード」の下3桁（下線部分）が' 826' のレコードを選択する。

(1) CSVFRMCHKの指定がない場合

入力ファイル

項番	コード
01	40868 <u>26</u>
02	Dkt7 <u>673</u>
03	Slk2 <u>826</u>
04	Ypg4 <u>826</u>
05	Ypg64 <u>31</u>



exsortコマンドエラー  
「KBL1320-E 比較項目にエラーデータを検出しました。」

(凡例)

下線：選択範囲

■：不正レコード

(2) CSVFRMCHKの指定がある場合

入力ファイル

項番	コード
01	40868 <u>26</u>
02	Dkt7 <u>673</u>
03	Slk2 <u>826</u>
04	Ypg4 <u>826</u>
05	Ypg64 <u>31</u>



出力ファイル

項番	コード
01	40868 <u>26</u>
03	Slk2 <u>826</u>
04	Ypg4 <u>826</u>

不正データは、キー範囲外のため項目「コード」の下3桁（下線部分）が「826」のレコードが選択された。

(凡例)

下線：選択範囲

■：不正データ

### 注意事項

- CSVFRMCHK オプションは SORT EE 08-50 以降で有効なオプションです。08-50 より前のバージョンでは指定できません。
- CSVFRMCHK オプションは、CSV 数値データ（固定小数点数）「z」属性に有効です。
- CSV 数値データ（固定小数点数）の不正データ条件と不正データの扱いについて次の表に示します。

CSVFRMCHK	不正データの条件	不正データの扱い	備考
指定なし	<ul style="list-style-type: none"> <li>セルデータ中に、引用符、数字、符号、通貨記号、空白、小数点、コンマ以外の文字を含む場合</li> <li>セルデータ中に含まれる引用符、符号、通貨記号、空白、小数点、コンマの位置が不正な場合</li> </ul>	KBL1211-E または KBL1320-E メッセージを表示し、exsort コマンドが異常終了する。	除外文字の位置規則については、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

CSVFRMCHK	不正データの条件	不正データの扱い	備考
指定あり	<ul style="list-style-type: none"> <li>セルデータ中の有効桁範囲に、引用符、数字、符号、通貨記号、空白、小数点、コンマ以外の文字を含む場合</li> <li>セルデータ中の有効桁範囲に含まれる引用符、符号、通貨記号、空白、小数点、コンマの位置が不正な場合※</li> </ul>	有効桁範囲内に不正データを含む場合は、KBL1211-E または KBL1320-E メッセージを表示し、exsort コマンドが異常終了する。	除外文字の位置規則については「付録 B.5 数値データ（固定小数点数）の除外文字の位置規則」を参照してください。

### 注※

CSVFRMCHK オプションが指定された場合、参照するデータは有効桁範囲内です。有効桁範囲外のデータは参照しないため、不正データが存在してもエラーになりません。

- 10 進キーのチェックオプション (-cmdopt コマンドの KEYNOCHK オペランド、および環境変数「EXSORTMGTOPT (64 ビット版は「EXSORT64MGTOPT)」」の KEY\_INVALID\_NOCHK オプション) と同時に指定できません。同時に指定した場合は、KBL1120-E または KBL1107-E エラーとなります。
- 引用符を除いた除外文字が有効桁内の数字に付帯していない場合、エラーになりませんが、除外文字の意味は解釈しない (符号や小数点などとして扱わない) ので、注意してください。

除外文字の位置規則については「付録 B.5 数値データ (固定小数点数) の除外文字の位置規則」を参照してください。

### 5.5.25 選択条件式のブレイクオプション (SELBRK オプション)

SELBRK オプションは、SORT EE 08-50 以降で有効なオプションです。

出力ファイルパラメタ (-outfile, -splitout, -catout パラメタ) に指定された選択条件式 (include, omit, csvinclude, csvomit オペランド) を、指定された順に評価、選択 (出力ファイルへの出力) し、選択条件式に合致した時点で評価を終了 (ブレイク) します。

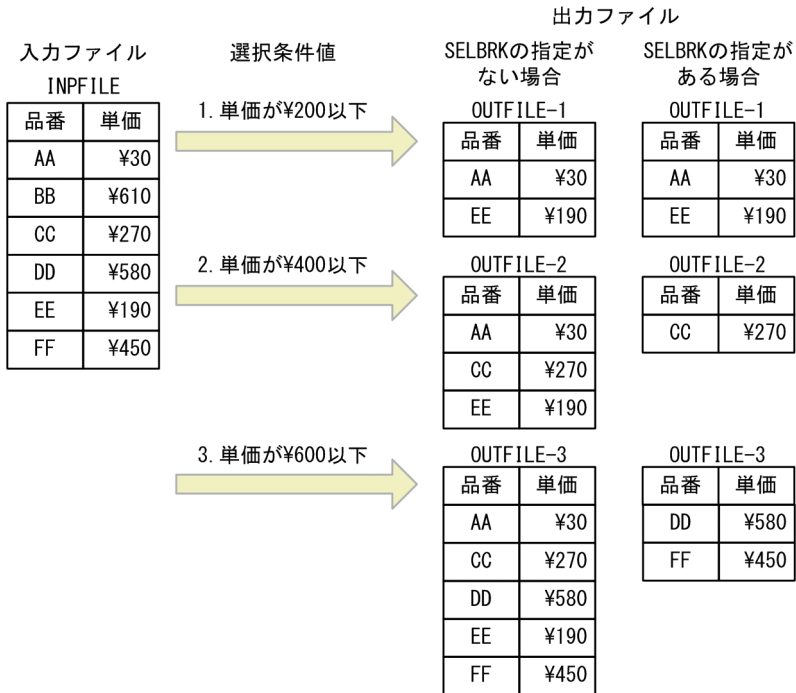
図 5-3 選択条件式のブレイクオプションの例

exsortコマンド指定

```

exsort
-functin copy
[ -cmdopt SELBRK]
-inpfile fileorg=C file=INFILE
-outfile file=OUTFILE-1 csvinclude= "z+2-3 <= N '200' " . . . .1.
-outfile file=OUTFILE-2 csvinclude= "z+2-3 <= N '400' " . . . .2.
-outfile file=OUTFILE-3 csvinclude= "z+2-3 <= N '600' " . . . .3.

```



SELBRK オプションを指定しない場合、指定した選択条件式をすべて評価、選択するため、1レコードは選択条件式に合致した複数のファイルへ出力されます。例えば、1、2、3の選択条件式に合致する「単価が ¥30」のレコードは、出力ファイル OUTFILE-1、OUTFILE-2、OUTFILE-3 へ出力されます。

SELBRK オプションを指定した場合、指定した選択条件式を、指定された順に評価、選択し、選択条件式に合致した時点で評価を終了するため、1レコードは1ファイルだけ出力されます。例えば、「単価 ¥30」のレコードは、1、2、3の選択条件式に合致しますが、1の選択条件式に合致した時点で評価を終了するため、出力ファイル OUTFILE-1 だけ出力されます。

指定値「SELBRK」が有効になる選択条件式の範囲を次の表に示します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

指定パラメタ	選択条件式（指定オペランド）			
	include	omit	csvinclude	csvomit
-inpfile	—	—	—	—
-outfile	○	○	○	○
-catinp	—	—	—	—
-catout	○	○	○	○
-splitinp	—	—	—	—
-splitout	○	○	○	○

（凡例）

- ：有効
- ：無効

### 注意事項

- SELBRK オプションは SORT EE 08-50 以降で有効なオプションです。08-50 より前のバージョンでは指定できません。
- 入力ファイルパラメタ（-inpfile, -catinp, -splitinp パラメタ）に指定された選択条件式に対しては無効です。
- 選択条件式がない出力ファイルパラメタを指定した場合は、オプションの有無に関係なく、すべてのレコードが出力されます。
- 条件が同じとなる選択条件式を、複数の出力ファイルパラメタに指定した場合、指定した条件に合致するレコードは、先に指定した選択条件式のある出力ファイルパラメタへ出力されます。あとに指定した出力ファイルパラメタへレコードは出力されないため、出力ファイルパラメタの記述順序については注意してください。

## 5.6 -key パラメタ

### 意味

ソートキー、マージキー、または集約キーを指定します。

### 形式

`-key key-definition [ $\Delta$ key-definition ... ]`

### 説明

- `-function` パラメタに `sort` または `merge` を指定した場合、もしくは `-sum` パラメタを指定した場合は、このパラメタを必ず指定してください。
- ソート機能では、重複レコードの入力順序保存機能 (`-cmdopt DUP`) を指定することで、重複キーのレコードを入力順に並べて出力できます。重複レコードの入力順序保存機能については、「5.5.1 重複レコードの入力順序保存機能 (DUP | NODUP オプション)」を参照してください。
- マージおよびコピー機能では、重複キーのレコードは入力順に並べて出力されません。
- キーを複数指定した場合、キーの指定順によってソートまたはマージの優先順位が決まります。先に指定したキーほど優先順位が高く、あとに指定したキーほど優先順位が低くなります。
- キー同士の範囲は重なってもかまいません。
- キーの範囲は、集約項目の範囲と重なってははいけません。
- 指定できるキーの個数と合計長は、環境ファイル「`EXS_KEYNUMMAX`」で指定できます。なお、キーの合計長は、入力データのエンコードによって異なります。環境ファイル「`EXS_KEYNUMMAX`」については、「15.1.1(14) `EXS_KEYNUMMAX`」を参照してください。

### 指定値

`key-definition` の形式を次に示します。

#### key-definition 形式 1

`[r]key-attribute+pos-len`

- `[r]key-attribute + pos - len` は必ず続けて指定してください。それぞれの値の間に空白やタブを指定しないでください。
- `[r]key-attribute + pos - len` を複数指定する場合は、1 個以上の空白またはタブで区切って指定します。
- `key-attribute` には、属性コード「j, c, n, m, p, z, x, s, t, f, g, q, d, w, u, v」が指定できます。

#### key-definition 形式 2

`[r]key-attribute+pos-maskbit`

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- [r]*key-attribute* + *pos* - *maskbit* は必ず続けて指定してください。それぞれの値の間に空白やタブを指定しないでください。
- [r]*key-attribute* + *pos* - *maskbit* を複数指定する場合は、1 個以上の空白、またはタブで区切って指定します。
- *key-attribute* には、属性コード「a, b」が指定できます。

*key-definition* に指定するそれぞれの値について次に示します。

[r]

キーのソート順序を指定します。

r オプション	意味
指定なし	キーの昇順でソートまたはマージします。
指定あり	キーの降順でソートまたはマージします。

- 降順でソートまたはマージする場合は、キーの属性を「rc」のようにつなげて指定します。
- 昇順か降順かは、キーごとに指定します。キーを複数指定して、昇順と降順を混在させたソートまたはマージもできます。

*key-attribute*

キーの属性を指定します。

<i>key-attribute</i> に指定する値	意味	<i>key-definition</i> 形式
j	キーが文字 (JIS8 単位コード※1) であることを指定します。	形式 1
c	キーが文字であることを指定します。※3	形式 1
n	キーが符号付き固定小数点 2 進数であることを指定します。※9, ※10	形式 1
m	キーが符号なし 2 進数 (0 から始まる整数) であることを指定します。※9, ※10	形式 1
a	キーが符号なし 2 進数 (0 から始まる整数) であることを指定します。※9, ※10 (マスクビットで指定するビット単位のキーです)	形式 2
b	キーが符号なし 2 進数 (バイトデータ) であることを指定します。※10 (マスクビットで指定するビット単位のキーです)	形式 2
p	キーが内部 10 進数であることを指定します。	形式 1



<i>key-attribute</i> に指定する値	意味	<i>key-definition</i> 形式
z	キーが外部 10 進数であることを指定します。※5	形式 1
x	キーが左符号付き外部 10 進数であることを指定します。※5	形式 1
s	キーが右分離符号付き外部 10 進数であることを指定します。※5	形式 1
t	キーが左分離符号付き外部 10 進数であることを指定します。※5	形式 1
f	キーが内部浮動小数点であることを指定します。※9, ※10	形式 1
g	キーが日付 (英数字) であることを指定します。※5	形式 1
q	キーが文字 (文字順序変更機能※2 で指定した文字) であることを指定します。※3	形式 1
d	キーが文字 (左分離符号付き数字) であることを指定します。※5	形式 1
w	キーが文字 (ワイド文字※4, ※6) であることを指定します (テキストファイルだけ指定できます)。	形式 1
u	キーが文字 (マルチバイト文字※6, ※7, ※8) であることを指定します (テキストファイルだけ指定できます)。	形式 1
v	キーが文字 (マルチバイト文字※6, ※7, ※8) であることを指定します (Unicode の IVS に対応した属性で、テキストファイルだけ指定できます)。	形式 1

## 注※1

ASCII コード (0x00 ~ 0x7F) を含む, JIS8 単位コード (0x00 ~ 0xFF) を意味します。

入力データに日本語などのマルチバイト文字が含まれている場合, マルチバイト文字としてではなく, シングルバイト文字として扱われます。

EBCDIK コードの体系に従ってキーを比較する EBCDIK コード比較オプション機能を使用できます。EBCDIK コード比較オプション機能については, 「5.5.3 EBCDIK コード比較オプション (EBCJ, EBCE オプション)」を参照してください。

## 注※2

文字順序変更機能については, 「5.15 -altseq パラメタ」を参照してください。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

### 注※ 3

入力データのエンコードの種別には対応しません。入力データの文字を構成するバイト長を意識しないで、常に 1 バイト長のシングルバイト文字として扱います。ASCII コードや JIS8 単位コード向けの属性です。

### 注※ 4

入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定した場合、1 文字 2 バイトのワイド文字として扱います。またサロゲートペア文字は 2 文字として扱います。

入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定した場合、1 文字 4 バイトのワイド文字として扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。

なお、属性コード「w」は、固定長順編成ファイルと可変長順編成ファイルの入力ファイルに対して指定できません。指定した場合、KBLS1203-E メッセージが表示されて、エラーになります。

### 注※ 5

入力データのエンコードで解釈します。

「ASCII」、「SJIS」、「EUC-JP」、「EUC-HJ」または「UTF-8」を指定した場合は 1 バイト、「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」を指定した場合は 2 バイト、「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」を指定した場合は 4 バイトの長さで文字を扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。

### 注※ 6

キー範囲の入力データがすべて ASCII 対応コード (英数字) の場合は、属性コード「c」を指定した方が処理効率が向上します。属性コード「c」は 1 文字 1 バイトで扱うため、エンコードに対応したバイト長を指定してください。

ASCII 対応コードについては、「付録 H Unicode 操作文字コード一覧 (SORT EE)」を参照してください。

### 注※ 7

属性コード「u」と「v」は、入力データのエンコードで解釈します。1 文字の長さは可変長として扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参

照してください。また、マルチバイト文字の扱いについては、「付録 I.2 文字列データの扱い」を参照してください。

#### 注※ 8

入力データのエンコードに「UTF-8」,「UTF-16LE」,「UTF-16BE」,「UCS-4LE」,「UCS-4BE」,「UTF-32LE」,「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE, UTF-16BE, UTF-32LE または UTF-32BE)」のどれかを指定した場合、Unicode の IVS (Ideographic Variation Sequence) を次のとおり解釈します。Unicode の IVS については、「付録 I.2(6) IVS 対応」を参照してください。

- 属性コード「u」は、UCS コードが一致する文字であれば、Unicode の IVS の異体字であっても同じ文字として扱います。
- 属性コード「v」は、UCS コードが一致する文字であっても、VS (Variation Selector, 異体字セレクタ) が異なる場合は、異なる文字として扱います。

#### 注※ 9

属性コード「n」,「m」,「f」および「a」は、数値として比較するため、バイトオーダーが関係します。バイトオーダーについては、「2.6 エンディアンオプション機能」および「5.5.4 入力ファイルのエンディアンオプション (BIG | INPBIG | INPLTL オプション)」を参照してください。

- 符号付き固定小数点数「n」: 2 バイト (signed short int), 4 バイト (signed int), 8 バイト (signed long long int)
- 符号なし 2 進数「m」,「a」: 2 バイト (unsigned short int), 4 バイト (unsigned int), 8 バイト (unsigned long long int)
- 内部浮動小数点数「f」: 4 バイト (float), 8 バイト (double)

#### 注※ 10

属性コード「n, m, f」と「a, b」は、どちらも「2 進数」データですが、キーの長さの指定方法が異なります。

- 属性コード「n」,「m」および「f」は、キーの長さを「バイト長」で指定します。
- 属性コード「a」と「b」は、キーの長さ分の「マスクビット」を指定します。

#### + pos

キーの開始バイト位置を 5 桁以内の数字で指定します。

-locatajst パラメタを指定したかどうかによって、レコード先頭の解釈が異なります。

また、テキストファイルの場合は、入力データのエンコードによって指定範囲が異なります。

開始バイト位置 (pos) に指定できる値の範囲を次の表に示します。

5. exsort コマンドのパラメタおよびオペランドの詳細

ファイル編成			-locatajst パラメタ	
			指定あり	指定なし
固定長順編成ファイル			1 ~ 65,535	0 ~ 65,534
可変長順編成ファイル			5 ~ 65,539 ※ <sup>1</sup>	0 ~ 65,534
テキストファイル ※ 2	改行文字「LF」	ASCII SJIS EUC-JP EUC-HJ UTF-8	1 ~ 65,535	0 ~ 65,534
		UCS-2LE UCS-2BE UTF-16LE UTF-16BE	1 ~ 65,533 (2 の倍数 + 1 ※ <sup>3</sup> )	0 ~ 65,532 (2 の倍数 ※ <sup>3</sup> )
		UCS-4LE UCS-4BE UTF-32LE UTF-32BE	1 ~ 65,529 (4 の倍数 + 1 ※ <sup>4</sup> )	0 ~ 65,528 (4 の倍数 ※ <sup>4</sup> )
	改行文字「CRLF」	ASCII SJIS EUC-JP EUC-HJ UTF-8	1 ~ 65,534	0 ~ 65,533
		UCS-2LE UCS-2BE UTF-16LE UTF-16BE	1 ~ 65,531 (2 の倍数 + 1 ※ <sup>3</sup> )	0 ~ 65,530 (2 の倍数 ※ <sup>3</sup> )
		UCS-4LE UCS-4BE UTF-32LE UTF-32BE	1 ~ 65,525 (4 の倍数 + 1 ※ <sup>4</sup> )	0 ~ 65,524 (4 の倍数 ※ <sup>4</sup> )

注※ 1

レコードの前の 4 バイト領域 (RDW : レコードヘッダ) の先頭が 1 となります。

注※ 2

入力レコードの改行文字については、「5.8.7 recsepa オペランド」を参照してください。また、入力データのエンコードについては、「5.8.13 codetype オペランド」を参照してください。

注※ 3

2 の倍数 (-locatajst パラメタを同時に指定している場合は「2 の倍数 + 1」) 以外を指定すると、KBL51205-E メッセージが表示されてエラーになります。

注※ 4

4 の倍数 (-locatajst パラメタを同時に指定している場合は「4 の倍数 + 1」) 以

外を指定すると、KBL1205-E メッセージが表示されてエラーになります。

– len

キーの長さを次の表に示す範囲で指定します。

key-attribute の値	指定できる長さ (単位)
j, c, q, w ※ <sup>1</sup>	<ul style="list-style-type: none"> <li>入力データのエンコードが「ASCII」, 「SJIS」, 「EUC-JP」, 「EUC-HJ」または「UTF-8」の場合 1 ~ 4,096 (バイト)</li> <li>入力データのエンコードが「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」の場合※<sup>2</sup> 2 ~ 8,192 (バイト)</li> <li>入力データのエンコードが「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」の場合※<sup>3</sup> 4 ~ 16,384 (バイト)</li> </ul>
u, v	1 ~ 4,096 (文字数) ※ <sup>6</sup>
m, n	2, 4, または 8 (バイト)
p	1 ~ 10 (バイト)
z, x	1 ~ 18 (桁) ※ <sup>4</sup>
s, t	2 ~ 19 (桁) ※ <sup>4</sup>
f	4 または 8 (バイト)
g	2 ~ 8 (桁) ※ <sup>5</sup>
d	1 ~ 256 (桁) ※ <sup>4</sup>

注※ 1

入力データのエンコードの種別によって、マルチバイト文字の長さは異なります。詳細は「付録 I.2 文字列データの扱い」を参照してください。

注※ 2

入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定した場合、長さには「2 の倍数」を指定してください。

2 の倍数以外を指定すると、KBL1207-E メッセージが表示されてエラーになります。

注※ 3

入力データのエンコードに「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定した場合、長さには「4 の倍数」を指定してください。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

4 の倍数以外を指定すると、KBL51207-E メッセージが表示されてエラーになります。

### 注※ 4

「バイト長」ではなく数値の「桁数」を指定してください。UCS-2、UCS-4、UTF-16 および UTF-32 の数値データの扱いについては、「付録 I.3 数値データの扱い」を参照してください。

### 注※ 5

「バイト長」ではなく日付の「桁数」を指定してください。UCS-2、UCS-4、UTF-16 および UTF-32 の日付データの扱いについては、「付録 I.4 日付データの扱い」を参照してください。

### 注※ 6

「バイト長」ではなく「文字数」を指定してください。UTF-16 のサロゲートペア文字は 1 文字として扱います。また、Unicode の VS (Variation Selector, 異体字セレクタ) は文字数に数えません。マルチバイト文字の扱いについては、「付録 I.2(6) IVS 対応」を参照してください。

### — maskbit

- 属性コード「a」または「b」を指定した場合、マスクビットを 16 進数で指定します。
- 16 進数の A ~ F は、小文字・大文字のどちらでも指定できます。
- キーデータに対して、指定したマスクビットで AND 演算を行った結果でソートします。なお、この AND 演算によってキーデータが更新されることはありません。
- マスクビットの 16 進数は、ビッグエンディアン形式で記述してください。
- すべての桁を「00」とするマスクビット指定はできません。指定した場合は、KBL51207-E エラーとなります。
- 指定できるマスクビットの桁は次のとおりです。

key-attribute に指定する値	指定できる maskbit の桁	備考
a	4, 8 または 16 (桁) ※1	キーの長さは 2, 4 または 8 バイト
b	2 ~ 8,192 (桁) ※2	キーの長さは 1 ~ 4,096 バイト

### 注※ 1

属性コード「a (符号なし 2 進数)」は、2 バイト (unsigned short int)、4 バイト (unsigned int)、8 バイト (unsigned long long int) のキーとして扱います。2, 4 または 8 バイトに対応するマスクビット値を、それぞれ 4, 8 または 16 桁の 16 進数で指定してください。なお、16 進数はビッグエンディアン形式で記

述してください。

(例 1) ra+0-F000 a+0-0FFF

第 1 キーの並び順：降順

第 1 キーの範囲：+ 0 ~ 1 の 2 バイト長

第 1 キーのマスクビット：0xF000

第 2 キーの並び順：昇順

第 2 キーの範囲：+ 0 ~ 1 の 2 バイト長

第 2 キーのマスクビット：0x0FFF

(入力データのエンディアン：リトルエンディアン)

	+0	+1	unsigned short	キー1 F000	キー2 0FFF	並び順
レコード1	22	12	1222	1000	0222	レコード4
レコード2	11	21	2111	2000	0111	レコード1
レコード3	11	11	1111	1000	0111	レコード3
レコード4	22	22	2222	2000	0222	レコード2

(入力データのエンディアン：ビッグエンディアン)

	+0	+1	unsigned short	キー1 F000	キー2 0FFF	並び順
レコード1	12	22	1222	1000	0222	レコード4
レコード2	21	11	2111	2000	0111	レコード1
レコード3	11	11	1111	1000	0111	レコード3
レコード4	22	22	2222	2000	0222	レコード2

注※ 2

属性コード「b (符号なし 2 進数)」は、1 ~ 4,096 バイトの範囲を指定できます。マスクビット値を、2 ~ 8,192 桁の 16 進数で指定してください。なお、16 進数はビッグエンディアン形式で記述してください。

(例 2) rb+0-fff000 a+0-000fff

第 1 キーの並び順：降順

第 1 キーの範囲：+ 0 ~ 2 の 3 バイト長

第 1 キーのマスクビット：0xffff000

第 2 キーの並び順：昇順

第 2 キーの範囲：+ 0 ~ 2 の 3 バイト長

第 2 キーのマスクビット：0x000fff

## 5. exsort コマンドのパラメタおよびオペランドの詳細

	+0	+1	+2	キー1 fff000	キー2 000fff	並び順
レコード1	11	12	22	111000	000222	レコード4
レコード2	22	21	11	222000	000111	レコード1
レコード3	11	11	11	111000	000111	レコード3
レコード4	22	22	22	222000	000222	レコード2

### キーの範囲

キーは、次の範囲内で指定してください。

- 入力ファイルが固定長順編成ファイルのとき、キーはレコード長の範囲
- 入力ファイルが可変長順編成ファイルのとき、キーは最小レコード長の範囲※<sup>1</sup>
- 入力ファイルがテキストファイルのとき、キーは最小レコード長の範囲※<sup>1</sup>, ※<sup>2</sup>

#### 注※1

可変長順編成ファイルまたはテキストファイルで、`-cmdopt` パラメタの `POSNOCHK` オプションを指定している場合は、最大レコード長の範囲となります。なお、`POSNOCHK` オプションでは、最大レコード長を超えた範囲をキーとして指定してもエラーにはなりません、最大レコード長を超えた部分のキー値は必ず `NULL` 値 (`0x00`) が仮定されるので注意してください。

`-cmdopt` パラメタの `POSNOCHK` オプションについては、「5.5.11 キー位置のチェックオプション (`POSNOCHK` オプション)」を参照してください。

#### 注※2

テキストファイルに指定した属性コード「`u`」または「`v`」の場合は、キーの指定範囲が実レコード長を超えていてもエラーにはなりません。実レコード長を超えた部分には空白を仮定してキーを比較します。

### 関連パラメタ・オペランド

- `-locatajst` パラメタ
- `-function` パラメタ
- `-sum` パラメタ
- `-inpfile` パラメタ `codetype` オペランド

### 指定例

#### 【例1】

```
-key△j+0-4△rc+18-5△p+12-8△rz+6-10
```

第1 キー：0 バイト目から長さ4 バイト (文字), 昇順

第2 キー：18 バイト目から長さ5 バイト (文字), 降順



第3 キー : 12 バイト目から長さ 8 バイト (内部 10 進数), 昇順

第4 キー : 6 バイト目から長さ 10 バイト (外部 10 進数), 降順

【例 2】

```
-key△b+0-0f△rb+2-f8△a+4-0ff0△ra+8-0000fff0
```

第1 キー : 0 バイト目から長さ 1 バイト, マスク 0x0f, 昇順

第2 キー : 2 バイト目から長さ 1 バイト, マスク 0xf8, 降順

第3 キー : 4 バイト目から長さ 2 バイト (unsigned short int), マスク 0x0ff0, 昇順

第4 キー : 8 バイト目から長さ 4 バイト (unsigned int), マスク 0x0000fff0, 降順

(凡例)

△ : 半角スペース (空白) を示します。

## 5.7 -sum パラメタ

---

### 意味

集約項目を指定します。

### 形式


-sum [ *field-definition* ]

### 説明

- -sum パラメタを指定する場合は、-key パラメタを必ず指定してください。
- -function copy と組み合わせる指定する場合、入力ファイルはあらかじめ集約キーでソートしておく必要があります。ソートしていない場合は、集約の結果は保証されません。
- -function copy と組み合わせる場合、入力ファイルは一つだけ指定してください。
- 集約項目同士の範囲は重なってはいけません。
- 集約項目の範囲は、キーの範囲と重なってはいけません。集約項目の範囲とキーの範囲と重なった場合は、「KBL51409-E 集約項目とキーが重なっています。」というメッセージを出力して異常終了します。ただし、キー属性「u」または「v」を指定した場合、キーの範囲と集約項目の範囲が重なっていてもエラーにはなりません。重なった範囲は集約結果で更新されて正常終了します。
- 環境ファイル「EXS\_SUMNUMMAX」の「最大個数の倍数」を指定していない場合、指定できる集約項目の個数は 1 ~ 128 個です。集約項目の最大個数は「128 × n 個」の単位で、1,280 個 (128 × 10) まで拡張できます。129 個以上のキーを指定したい場合は、環境ファイル「EXS\_SUMNUMMAX」の「最大個数の倍数」に、「128 × n 個」の n の値を指定してください。
- 環境ファイル「EXS\_SUMNUMMAX」の「合計長の上限の倍数」を指定していない場合、指定できる集約項目の合計長は 4,096 バイトです。集約項目の合計長は「4,096 × m バイト」の単位で、65,536 バイト (4,096 × 16) まで拡張できます。4,097 バイト以上の合計長を指定したい場合は、環境ファイル「EXS\_SUMNUMMAX」の「合計長の上限の倍数」に、「4,096 × m バイト」の m 値を指定してください。
- 集約項目を指定しない場合 (集約項目の個数が 0 個)、同じキーを持つ連続したレコードがあるときは、最初に配置されているレコードだけを抽出します。これを無条件レコード抽出機能といいます。無条件レコード抽出機能を使用した例を次の図に示します。

図 5-4 無条件レコード抽出機能の使用例

支店コード	担当者名	売上数量
03	田中	2
03	田中	7
04	渡辺	5
05	日立	8
06	山田	6
06	山田	10



支店コード	担当者名	売上数量
03	田中	2
04	渡辺	5
05	日立	8
06	山田	6

(凡例) : 支店コードをキーとした場合に抽出されるレコードを示す。

### 指定値

*field-definition* の形式を次に示します。

*field-attribute+pos-len* [ $\Delta$ *field-attribute+pos-len* ... ]

- *field-attribute + pos - len* は必ず続けて指定してください。それぞれの値の間に空白やタブを指定しないでください。
- *field-attribute + pos - len* を複数指定する場合は、1 個以上の空白またはタブで区切って指定します。

*field-definition* に指定するそれぞれの値について次に示します。

#### *field-attribute*

集約項目の属性を指定します。

<i>field-attribute</i> に指定する値	意味
n	集約項目が符号付き固定小数点 2 進数であることを指定します。※
m	集約項目が符号なし 2 進数であることを指定します。※
p	集約項目が内部 10 進数であることを指定します。
z	集約項目が外部 10 進数であることを指定します。
x	集約項目が左符号付き外部 10 進数であることを指定します。
s	集約項目が右分離符号付き外部 10 進数であることを指定します。
t	集約項目が左分離符号付き外部 10 進数であることを指定します。

#### 注※

属性コード「n」および「m」は、数値として比較するため、バイトオーダーが関係します。

バイトオーダーについては、「2.6 エンディアンオプション機能」および

## 5. exsort コマンドのパラメタおよびオペランドの詳細

「5.5.4 入力ファイルのエンディアンオプション (BIG | INPBIG | INPLTL オプション)」を参照してください。

- 符号付き固定小数点数「n」: 2 バイト (signed short int), 4 バイト (signed int), 8 バイト (signed long long int)
- 符号なし 2 進数「m」: 2 バイト (unsigned short int), 4 バイト (unsigned int), 8 バイト (unsigned long long int)

### + pos

集約項目の開始バイト位置を 5 桁以内の数字で指定します。

-locatajst パラメタを指定したかどうかによって、レコード先頭の解釈が異なります。また、テキストファイルの場合は、入力データのエンコードによって指定範囲が異なります。開始バイト位置 (pos) に指定できる値の範囲を次の表に示します。

ファイル編成			-locatajst パラメタ	
			指定あり	指定なし
固定長順編成ファイル			1 ~ 65,535	0 ~ 65,534
可変長順編成ファイル			5 ~ 65,539 <sup>※1</sup>	0 ~ 65,534
テキストファイル <sup>※2</sup>	改行文字「LF」	ASCII SJIS EUC-JP EUC-HJ UTF-8	1 ~ 65,535	0 ~ 65,534
		UCS-2LE UCS-2BE UTF-16LE UTF-16BE	1 ~ 65,533 (2 の倍数 + 1 <sup>※3</sup> )	0 ~ 65,532 (2 の倍数 <sup>※3</sup> )
		UCS-4LE UCS-4BE UTF-32LE UTF-32BE	1 ~ 65,529 (4 の倍数 + 1 <sup>※4</sup> )	0 ~ 65,528 (4 の倍数 <sup>※4</sup> )
	改行文字「CRLF」	ASCII SJIS EUC-JP EUC-HJ UTF-8	1 ~ 65,534	0 ~ 65,533
		UCS-2LE UCS-2BE UTF-16LE UTF-16BE	1 ~ 65,531 (2 の倍数 + 1 <sup>※3</sup> )	0 ~ 65,530 (2 の倍数 <sup>※3</sup> )
		UCS-4LE UCS-4BE UTF-32LE UTF-32BE	1 ~ 65,525 (4 の倍数 + 1 <sup>※4</sup> )	0 ~ 65,524 (4 の倍数 <sup>※4</sup> )

注※1

レコードの前の 4 バイト領域 (RDW : レコードヘッダ) の先頭が 1 となります。

#### 注※ 2

入力レコードの改行文字については、「5.8.7 recsepa オペランド」を参照してください。また、入力データのエンコードについては、「5.8.13 codetype オペランド」を参照してください。

#### 注※ 3

2 の倍数 (-locatajst パラメタを同時に指定している場合は「2 の倍数 + 1」) 以外を指定すると、KBLS1404-E メッセージが表示されてエラーになります。

#### 注※ 4

4 の倍数 (-locatajst パラメタを同時に指定している場合は「4 の倍数 + 1」) 以外を指定すると、KBLS1404-E メッセージが表示されてエラーになります。

#### – len

集約項目の長さを次の表に示す範囲で指定します。

field-attribute の値	指定できる長さ (単位)
m, n	2, 4, または 8 (バイト)
p	1 ~ 10 (バイト)
z, x ※	1 ~ 18 (桁)
s, t ※	2 ~ 19 (桁)

#### 注※

「UCS-2」, 「UCS-4」, 「UTF-16」 および 「UTF-32」 では、10 進数データの 1 桁が 2 または 4 バイトで表現されます。詳細は「付録 I.3 数値データの扱い」を参照してください。

#### 集約項目の範囲

集約項目は、次の表に示す範囲内で指定してください。

入力ファイルの種類	集約項目の範囲
固定長ファイル	レコード長
可変長ファイル	最小レコード長
テキストファイル	最小レコード長

#### 集約結果の範囲

2 進数項目の集約結果の範囲を次の表に示します。

5. exsort コマンドのパラメタおよびオペランドの詳細

表 5-9 固定小数点 2 進数 (n) の範囲

長さ	OVFLW オプション	
	指定なし	指定あり
2	- 32,768 ~ + 32,767	- 9,999 ~ + 9,999
4	- 2,147,483,648 ~ + 2,147,483,647	- 999,999,999 ~ + 999,999,999
8	- 9,223,372,036,854,775,808 ~ + 9,223,372,036,854,775,807	- 999,999,999,999,999,999 ~ + 999,999,999,999,999,999

表 5-10 符号なし 2 進数 (m) の範囲

長さ	OVFLW オプション	
	指定なし	指定あり
2	0 ~ 32,767	0 ~ 9,999
4	0 ~ 2,147,483,647	0 ~ 999,999,999
8	0 ~ 9,223,372,036,854,775,807	0 ~ 999,999,999,999,999,999

集約結果がオーバーフローした場合

集約結果がオーバーフローした場合は、それまで集約した結果を出力したあと、処理中のレコードから新しく集約します。集約結果がオーバーフローした場合の例を次の図に示します。

図 5-5 集約結果がオーバーフローした場合の例

支店コード	担当者名	売上数量		支店コード	担当者名	売上数量
03	田中	2	➔	03	田中	9
03	田中	7		04	渡辺	5
04	渡辺	5		05	日立	8
05	日立	8		06	山田	999
06	山田	999		06	山田	10
06	山田	10				

(凡例)  : 集約結果がオーバーフローしたレコードを示す。

集約結果がオーバーフローした時に exsort コマンドを異常終了させたい場合は、コマンドオプション「OVFERR」を指定してください。最初のオーバーフローを検出した時点で、KBSL1412-E エラーメッセージを出力して異常終了します。

OVFERR オプションについては、「5.5.20 集約結果オーバーフロー検知オプション (OVFERR オプション)」を参照してください。

関連パラメタ・オプション

- -locatajst パラメタ
- -function パラメタ
- -key パラメタ
- -cmdopt パラメタ OVFERR オプション
- -infile パラメタ codetype オペランド

## 5.8 -infile パラメタ

---

### 意味

入力ファイル情報を指定します。

### 形式

```
-infile fileorg=organization-type
  [ file=file-name [ $\Delta$ file-name ... ] ]
  [ reclen=record-length ]
  [ { skiprec=skip-record-number |
      startrec=start-record-number } ]
  [ stopaft=stop-record-number ]
  [ recsepa=record-separator-type]
  [ cellsepa= {TAB | 'cell-separator'}
              [ $\Delta$  {TAB | 'cell-separator'} ... ] ]
  [ quottype=quotation-type]
  [ currency=currency-symbol]
  [ round=option-code]
  [ nullseq=option-code]
  [ codetype=encode]
  [ { reorg=reorganization-format
      [ $\Delta$ reorganization-format ... ]
      | csvreorg=csv-reorganization-format
        [ $\Delta$ csv-reorganization-format ... ] } ]
  [ { {include | omit} =condition-expression
      | {csvinclude | csvomit} =csv-condition-expression } ]
```

(凡例)

$\Delta$  : 半角スペース (空白) を示します。

### 5.8.1 fileorg オペランド

#### 意味

入力ファイルのファイル編成を指定します。

#### 形式

```
fileorg=organization-type
```

#### 注意事項

- このオペランドは省略できません。exsort コマンドの引数、または exsort パラメタファイルのどちらかに必ずこのオペランドを指定してください。exsort コマンドの引数と exsort パラメタファイルの両方に指定した場合は、exsort コマンドの引数の指定が有効になります。
- 複数の入力ファイルを指定する場合は、指定するファイルをすべて同じファイル編成にしてください。

#### 指定値

*organization-type* に指定する値を次の表に示します。



指定値	意味
S	固定長順編成ファイルを指定します。
V	可変長順編成ファイルを指定します。
A	テキストファイルを指定します。
C	CSV ファイルを指定します。

## 5.8.2 file オペランド

### 意味

入力ファイル名または環境変数名を指定します。

### 形式

```
file=file-name [△file-name ... ]
```

### 注意事項

- このオペランドは省略できます。省略した場合は、標準入力ファイルを入力ファイルとします。
- 入力ファイル名を複数指定する場合は、1 個以上の空白またはタブで区切って指定します。
- 同じファイル名のファイルを複数指定できません。
- exsort パラメタファイルに環境変数名を指定する場合は、「% 環境変数名 %」と % で囲んで記述します。
- 空白文字を含んだファイル名を指定する場合は、ファイル名ごとにダブルクォーテーションで囲んで指定してください。ファイル名の指定規則については、「4.3.5 ファイル名およびディレクトリ名の指定形式」を参照してください。
- ファイルは 256 個まで指定できます。
- 選択機能、集約機能およびレコード編集機能を指定していないソート機能を使用する場合に限り、入力ファイルと出力ファイルに同じファイルを指定できます。入力ファイルは、ソートした出力結果で上書きされます。既存ファイルの上書きオプション (-cmdopt OVWRT) を指定してください。OVWRT オプションについては、「5.5.2 ファイルの上書きオプション (OVWRT | INPOVWRT オプション)」を参照してください。
- 選択機能、集約機能またはレコード編集機能を指定したソート機能で、入力ファイルと出力ファイルに同じファイルを指定する場合、入力ファイルの上書きオプション (-cmdopt INPOVWRT) を指定してください。INPOVWRT オプションについては、「5.5.2 ファイルの上書きオプション (OVWRT | INPOVWRT オプション)」を参照してください。

## 5.8.3 reclen オペランド

### 意味

入力レコード長を指定します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

### 形式

`reclen=record-length`

`record-length` は、5 桁以内の数字で指定します。

### 指定値

`record-length` に指定する値を次の表に示します。

入力ファイル編成	レコード長の範囲 (単位：バイト)	説明
固定長順編成ファイル	1 ~ 65,535	固定長順編成ファイルを入力ファイルとする場合は、このオペランドを省略できません。
可変長順編成ファイル	—	<code>reclen</code> オペランドの指定は不要です。指定しても無視されます。ただし、各コマンドで処理できるレコード長の範囲は、1 ~ 65,535 バイトです。

入力ファイル編成	レコード長の範囲 (単位：バイト)	説明
テキストファイル	1 ～ 65,535	<ul style="list-style-type: none"> <li>• 入力データのエンコードの種類によって、最大レコード長が異なります。最大レコード長の詳細については、「5.8.7 recsepa オペランド」を参照してください。また、入力データのエンコードについては、「5.8.13 codetype オペランド」を参照してください。</li> <li>• このオペランドは省略できます。省略した場合は、最大レコード長が設定されます。</li> <li>• 入力ファイルの合計サイズが 1MB 未満の場合にレコード長の指定を省略すると、ソート機能で一時ファイルが作成されることがあります。処理時間を短くしたい場合は、レコード長を指定することを推奨します。</li> <li>• レコード長を指定する場合は、入力ファイル内の最大レコード長よりも大きい値を指定してください。入力ファイル内の最大レコード長よりも小さい値を指定した場合は、KBL51751-E エラーとなります。メッセージが表示されたときの原因と対処方法の詳細は、「付録 K メッセージ」を参照してください。</li> <li>• 入力データのエンコードが「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」の場合、レコード長は 2 の倍数でなければなりません。</li> <li>• 入力データのエンコードが「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」の場合、レコード長は 4 の倍数でなければなりません。</li> </ul>

## 5. exsort コマンドのパラメタおよびオペランドの詳細

入力ファイル編成	レコード長の範囲 (単位：バイト)	説明
CSV ファイル	1 ~ 65,535	<ul style="list-style-type: none"> <li>• 入力データのエンコードの種類によって、最大レコード長が異なります。最大レコード長の詳細については、「5.8.7 recsepa オペランド」を参照してください。また、入力データのエンコードについては、「5.8.13 codetype オペランド」を参照してください。</li> <li>• このオペランドは省略できます。省略した場合は、最大レコード長が設定されます。</li> <li>• 入力ファイルの合計サイズが 1MB 未満の場合にレコード長の指定を省略すると、ソート機能で一時ファイルが作成されることがあります。処理時間を短くしたい場合は、レコード長を指定することを推奨します。</li> <li>• レコード長を指定する場合は、入力ファイル内の最大レコード長よりも大きい値を指定してください。なお、入力ファイル内の最大レコード長よりも小さい値を指定してもエラーにはなりません。</li> <li>• 入力データのエンコードが「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」の場合、レコード長は 2 の倍数でなければなりません。</li> <li>• 入力データのエンコードが「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」の場合、レコード長は 4 の倍数でなければなりません。</li> </ul>

(凡例) - : なし

### 5.8.4 skiprec オペランド

#### 意味

入力レコードを読み飛ばす件数を指定します。

入力ファイルの先頭から、指定されたレコード数までをソート処理またはコピー処理の対象としないで読み飛ばします。

なお、入力ファイルを複数指定した場合は、最初に指定したファイルから読み飛ばします。最初のファイルのレコードがすべて読み飛ばし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読み飛ばしの対象となります。

#### 形式

`skiprec=skip-record-number`

#### 指定値

- `skiprec` オペランドに、入力レコードの件数より大きい数を指定した場合、入力されるレコード件数は 0 件となります。
- 0 を指定した場合、指定なしと見なします。
- マージ (`-function merge`) 機能を使用する際に `skiprec` オペランドを指定しても、無視されます。
- 件数には、1 ~ 10 桁の 10 進数のうち、0 ~ 2147483647 を指定できます。

#### 注意事項

- 入力ファイル編成がテキストファイル、または CSV ファイルで、空行削除オプション (`-cmdopt DELBLANKLINE`) を指定している場合、空行 (データの無い改行文字だけのレコード) は読み飛ばし件数にカウントされません。空行削除オプションについては、「5.5.23 空行削除オプション (DELBLANKLINE オプション)」を参照してください。

#### 指定例

入力ファイルの先頭から 100 件目までのレコードを読み飛ばす場合の指定例を示します。

```
-inpfile△skiprec=100△fileorg=A△file=text.txt
```

(凡例)

△ : 半角スペース (空白) を示します。

## 5.8.5 stopaft オペランド

#### 意味

入力を打ち切るレコード件数を指定します。

入力ファイルの先頭から、指定されたレコード数分までをソート処理またはコピー処理の対象とします。

入力を打ち切るレコード件数は、`-inpfile` パラメタの `include` または `omit` オペランドによる入力レコード選択後、および `skiprec` オペランドによる入力レコードの読み飛ばし後のレコードに対する件数となります。

なお、入力ファイルを複数指定した場合は、最初に指定したファイルから処理します。最初のファイルのレコードがすべて処理対象となった場合、2 番目に指定したファイルが引き続き処理対象となります。

#### 形式

`stopaft=stop-record-number`

#### 指定値

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- stopaft オペランドに、入力レコードの件数より大きい数を指定した場合、すべてのレコードが入力の対象となります。
- 0 を指定した場合、指定なしと見なします。
- マージ (-function merge) 機能を使用する際に stopaft オペランドを指定しても、無視されます。
- 件数には、1 ～ 10 桁の 10 進数のうち、0 ～ 2147483647 を指定できます。

### 注意事項

- 入力ファイル編成がテキストファイル、または CSV ファイルで、空行削除オプション (-cmdopt DELBLANKLINE) を指定している場合、空行（データのない改行文字だけのレコード）は打ち切り件数にカウントされません。空行削除オプションについては、「5.5.23 空行削除オプション (DELBLANKLINE オプション)」を参照してください。

### 指定例

入力レコードを選択し、この後に実行するソートの件数を 100 件までとする場合の例を次に示します。

```
-inpfiler stopaft=100 fileorg=A file=text.txt include=c+0-1 = C'a'
```

また、入力レコードを 100 件読み飛ばし、この後に実行するコピーの件数を 200 件までとする場合の例を次に示します。

```
-inpfiler skiprec=100 stopaft=200 fileorg=A file=text.txt
```

## 5.8.6 startrec オペランド

### 意味

データの入力を開始するレコード番号を指定します。

指定したレコード番号よりも前のレコードは、ソート処理、マージ処理、またはコピー処理の対象となりません。この場合、指定したレコード番号よりも前のレコードは、読み飛ばされます。

入力ファイルを複数指定した場合は、ファイルごとに、このオペランドの指定が適用されます。このオペランドは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、無効となります。skiprec オペランドとこのオペランドは同時に指定できません。どちらか一つだけ指定してください。

### 形式

```
startrec=start-record-number
```

### 指定値

*start-record-number*

- 1 から始まるレコード番号を指定します。例えば、先頭レコードを読み飛ばし、2 番目のレコードから入力を開始したい場合は、「startrec=2」と指定します。

- レコード番号には、1 ～ 10 桁の 10 進数のうち、0 ～ 2147483647 を指定できます。
- 0 を指定した場合、指定なしと見なされます。

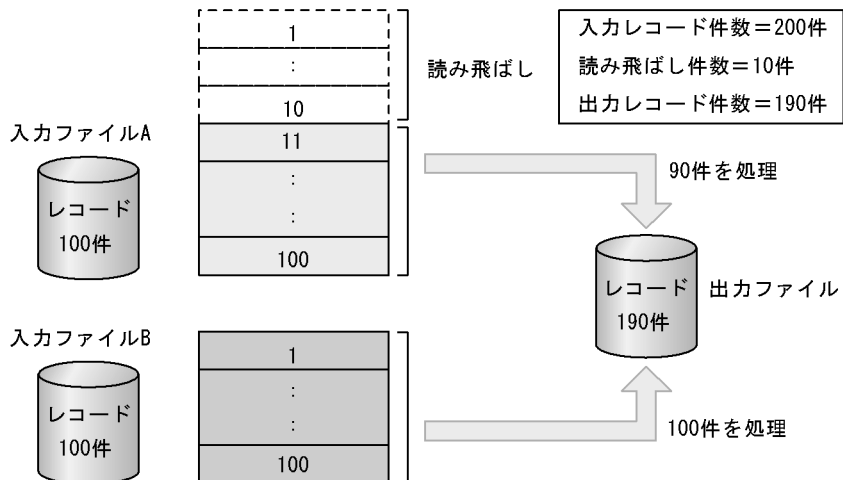
#### 注意事項

- **startrec** オペランドに、入力レコードの件数より大きい値を指定した場合、入力されるレコード件数は 0 件となります。
- **startrec** オペランドと **stopaft** オペランドの両方を指定した場合、最初に入力ファイルごとに **startrec** オペランドで読み飛ばされます。その後、**stopaft** オペランドで入力レコードに対して打ち切り件数がチェックされます。
- 入力ファイル編成がテキストファイル、または CSV ファイルで、空行削除オプション (-cmdopt DELBLANKLINE) を指定している場合、空行（データのない改行文字だけのレコード）を含めない開始レコード番号を指定してください。空行削除オプションについては、「5.5.23 空行削除オプション (DELBLANKLINE オプション)」を参照してください。

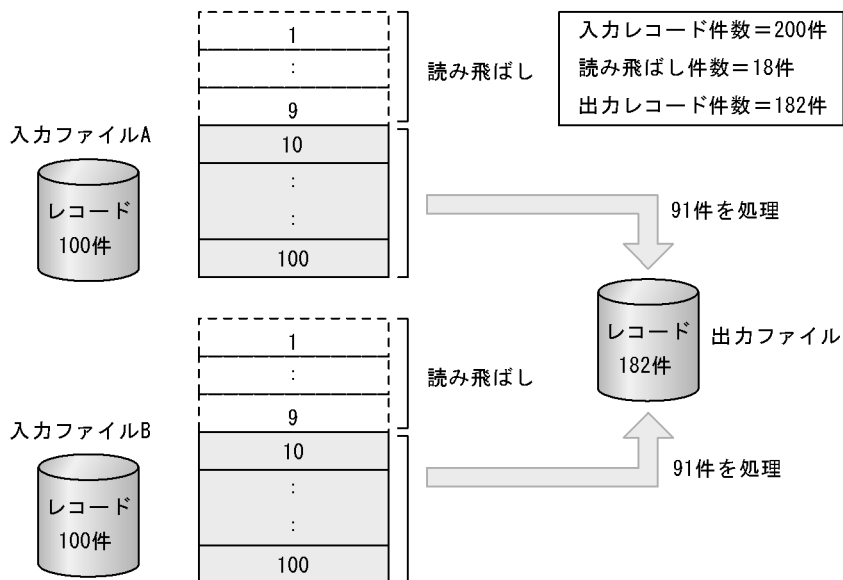
**skiprec** オペランドおよび **startrec** オペランドを指定した場合の処理の概要を次に示します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

【例1】 skiprec=10（先頭ファイルの1~10レコードを読み飛ばす）を指定した場合



【例2】 startrec=10（各ファイルの10レコード目から入力開始）を指定した場合



### 5.8.7 recsepa オペランド

#### 意味

テキストファイルおよび CSV ファイルの入力レコードの改行文字を指定します。このオペランドは、テキストファイルまたは CSV ファイルを指定した場合だけ有効です。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は、KBLS2200-E エラーとなります。



オペランド省略時のデフォルト値を、環境ファイル「EXS\_RECSEPA」で指定できます。

#### 形式

`recsepa=record-separator-type`

#### 指定値

`record-separator-type` に指定する値を次の表に示します。

指定値	意味
CRLF	CRLF を改行とする。
LF	LF を改行とする。
LF CRLF	LF と CRLF の両方を改行として扱う。

環境ファイル「EXS\_RECSEPA」およびこのオペランドの両方を省略した場合、改行文字に LF を仮定します。

#### 注意事項

- 「`recsepa=LF`」を指定している場合に、改行文字が CRLF である入力レコードを処理すると、CR は入力データとして扱われます。例えば、CSV ファイルの入力データを「1,2 改行 (CRLF)」とした場合、セル 1 = 「1」、セル 2 = 「2CR」となります。
- 「`recsepa=CRLF`」を指定している場合に、改行文字が LF である入力レコードを処理すると、LF は改行文字と見なされないで、入力データとして扱われます。CRLF を検出するまで、複数のレコードが 1 レコードと解釈されることになるので注意してください。
- 「`recsepa=LF CRLF`」指定では、入力レコードごとの改行文字を自動的に判定します。入力ファイルの改行文字の種別が不明な場合、または LF と CRLF が混在している入力ファイルの場合に指定してください。
- `codetype` オペランドに指定した入力データのエンコードの種別によって、改行文字の長さや最大レコード長が異なります。`codetype` オペランドについては、「5.8.13 `codetype` オペランド」を参照してください。

エンコード種別	改行文字のコード		テキスト・CSV ファイルの最大レコード長	
	CR	LF	改行文字が LF の場合	改行文字が CRLF の場合
ASCII	0x0d	0x0a	65,535	65,534
SJIS				
EUC-JP				
EUC-HJ				
UTF-8				

## 5. exsort コマンドのパラメタおよびオペランドの詳細

エンコード 種別	改行文字のコード		テキスト・CSV ファイルの最大レコード長	
	CR	LF	改行文字が LF の 場合	改行文字が CRLF の場合
UCS-2LE	0x0d00	0x0a00	65,534	65,532
UTF-16LE				
UCS-2BE	0x000d	0x000a		
UTF-16BE				
UCS-4LE	0x0d000000	0x0a000000	65,532	65,528
UTF-32LE				
UCS-4BE	0x0000000d	0x0000000a		
UTF-32BE				

### 5.8.8 cellsepa オペランド

#### 意味

CSV ファイルの入力レコードのデリミタ（セル区切り文字）を指定します。  
このオペランドは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、KBL52201-E エラーとなります。  
オペランド省略時のデフォルト値を、環境ファイル「EXS\_CELLSEPA」で指定できます。

#### 形式

`cellsepa=cell-separator [△cell-separator … ]`

#### 指定値

`cell-separator` に指定する値を次の表に示します。

指定値	意味
' <i>cell-separator</i> '	セルのデリミタとして、タブを除いた任意の 1 文字（半角英数字※）を、シングルクォーテーションで囲んで指定します。 【指定例】 空白を指定 <code>cellsepa='△'</code> （△は空白 1 文字を表す） タブ、セミコロン、コンマ、空白、コロンの 5 個のデリミタを指定 <code>cellsepa=TAB ';' ',' '△' ':'</code>
TAB	セルのデリミタとして、タブを指定します。 【指定例】 <code>cellsepa=TAB</code>

注※ 半角英数字の文字コードは（0x20 ～ 0x7E）です。

- 最大 5 個のデリミタを指定できます。
- 環境ファイル「EXS\_CELLSEPA」およびこのオペランドの両方を省略した場合、

- デリミタにコンマ (,) を仮定します。
- 引用符 (シングルクォーテーション, ダブルクォーテーション), 改行文字, およびマルチバイト文字 (日本語) はデリミタとして指定できません。
  - `codetype` オペランドに指定した入力データのエンコードで, 入力データ中のデリミタを解析します。入力データのエンコードについては, 「5.8.13 `codetype` オペランド」を参照してください。
  - 入力データのエンコードが「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」の場合, 「UCS-2」の文字コードのデータ形式となります。「付録 H Unicode 操作文字コード一覧 (SORT EE)」の「表 H-3 ASCII 対応予約文字一覧」を参照してください。
  - 入力データのエンコードが「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」の場合, 「UCS-4」の文字コードのデータ形式となります。「付録 H Unicode 操作文字コード一覧 (SORT EE)」の「表 H-3 ASCII 対応予約文字一覧」を参照してください。

#### 注意事項

- `cell-separator` は, 必ずシングルクォーテーション (') で囲んで指定してください。
- 同じ文字は複数指定できません。
- 6 個以上の文字は指定できません。
- 環境ファイル「EXS\_CELLSEPA」に指定する場合は, `cell-separator` をダブルクォーテーション (") で囲みます。コマンドと指定方法が異なるので注意してください。
- `exsort` コマンドに指定する場合は, 指定値全体をダブルクォーテーションで囲んでください。ただし, TAB 指定の場合は, ダブルクォーテーションで囲む必要はありません。

#### 【指定例】

- ```
-inpfile cellsepa="' △ ' ',' TAB"
```
- ダブルクォーテーションで囲む
- ```
-inpfile cellsepa=TAB
```
- ダブルクォーテーションの囲みは不要
- このオペランドを `exsort` パラメタファイルに指定する場合は, 指定値をダブルクォーテーションで囲まないでください。

#### 【指定例】

```
-inpfile cellsepa=' △ ' ',' TAB
```

```
-inpfile cellsepa=TAB
```

## 5.8.9 `quottype` オペランド

### 意味

セルの引用符文字を指定します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

このオペランドは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、KBL52204-E エラーとなります。

オペランド省略時のデフォルト値を、環境ファイル「EXS\_QUOTTYPE」で指定できます。

### 形式

`quottype=quotation-type`

### 指定値

*quotation-type* に指定する値を次の表に示します。

指定値	意味
SQUOTE	シングルクォーテーション (') を引用符とする。
DQUOTE	ダブルクォーテーション (") を引用符とする。

- 環境ファイル「EXS\_QUOTTYPE」とこのオペランドの両方を省略した場合、「quottype=DQUOTE」を仮定します。
- `codetype` オペランドに指定した入力データのエンコードで、入力データ中の引用符を解析します。入力データのエンコードについては、「5.8.13 `codetype` オペランド」を参照してください。
- 入力データのエンコードが「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」の場合、「UCS-2」の文字コードのデータ形式となります。「付録 H Unicode 操作文字コード一覧 (SORT EE)」の「表 H-3 ASCII 対応予約文字一覧」を参照してください。
- 入力データのエンコードが「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」の場合、「UCS-4」の文字コードのデータ形式となります。「付録 H Unicode 操作文字コード一覧 (SORT EE)」の「表 H-3 ASCII 対応予約文字一覧」を参照してください。

## 5.8.10 `currency` オペランド

### 意味

CSV ファイルの固定小数点数の数値データの通貨記号として、任意の 1 文字を指定します。

このオペランドは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、KBL52205-E エラーとなります。

オペランド省略時のデフォルト値を、環境ファイル「EXS\_CURRENCY」で指定できます。

### 形式

`currency=currency-symbol`

### 指定値

*currency-symbol*

- 通貨記号として、任意の 1 文字（半角文字）を指定します。入力データのエンコードの種別によって、指定できる文字コードの範囲が異なります。入力データのエンコードについては、「5.8.13 *codetype* オペランド」を参照してください。

入力データのエンコードの種別	指定できる文字コードの範囲	備考
SJIS	JIS8 単位コード文字	0x20 ~ 0x7E の英数字 0xA1 ~ 0xDF の半角カタカナ
ASCII EUC-JP EUC-HJ UTF-8 UCS-2LE UCS-2BE UCS-4LE UCS-4BE UTF-16LE UTF-16BE UTF-32LE UTF-32BE	ASCII コード文字	0x20 ~ 0x7E の英数字

- 環境ファイル「EXS\_CURRENCY」とこのオペランドの両方を省略した場合、通貨記号に円記号 (¥) を仮定します。
- デリミタ、引用符として指定した文字、改行文字、およびマルチバイト文字（日本語）を通貨記号として指定できません。
- codetype* オペランドに指定した入力データのエンコードで、入力データ中の通貨記号を解析します。入力データのエンコードについては、「5.8.13 *codetype* オペランド」を参照してください。
- 入力データのエンコードが「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」の場合、「UCS-2」の文字コードのデータ形式となります。「付録 H Unicode 操作文字コード一覧 (SORT EE)」の「表 H-3 ASCII 対応予約文字一覧」を参照してください。
- 入力データのエンコードが「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」の場合、「UCS-4」の文字コードのデータ形式となります。「付録 H Unicode 操作文字コード一覧 (SORT EE)」の「表 H-3 ASCII 対応予約文字一覧」を参照してください。

5.8.11 *round* オペランド

## 意味

CSV ファイルのすべての固定小数点数のキー、比較項目、または集約項目の桁の丸め方を指定します。

このオペランドは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外

## 5. exsort コマンドのパラメタおよびオペランドの詳細

を指定した場合は、KBL52206-E エラーとなります。

オペランド省略時のデフォルト値を、環境ファイル「EXS\_ROUND」で指定できます。

### 形式

`round=option-code`

### 指定値

`option-code` に指定する値を次の表に示します。

指定値	意味
0	小数部の右側のあふれ桁を切り捨てます。
1	小数部の右側のあふれ桁を切り上げます。
2	小数部の右側のあふれ桁を四捨五入します。

環境ファイル「EXS\_ROUND」とこのオペランドの両方を省略した場合、「round=0」を仮定します。

### 注意事項

切り上げ、または四捨五入を指定した場合、データによってはオーバーフローしてゼロ値となる場合があります。オーバーフローが発生しない正しい有効桁数を指定してください。

### 指定例

数値データの端数処理の例を次に示します。下線のある数値は、丸めの対象となる範囲を表します。

#### 【例 1】

有効桁を整数部 3 桁、小数部 0 桁に指定した場合の例

指定値 (丸め方法)	入力データ	丸め結果	桁合わせ	説明
0 または省略 (切り捨て)	999. <u>5</u>	999.0	999	0.1 の位を切り捨てます。
1 (切り上げ)	999.0	999.0	999	小数部が 0 なので切り上げません。
	999. <u>1</u>	1000.0	000	小数部が 0 ではないので切り上げます。 オーバーフローし、0 値となります。
2 (四捨五入)	999. <u>4</u>	999.0	999	小数部を切り捨てます。
	999. <u>5</u>	1000.0	000	切り上げます。 オーバーフローし、0 値となります。

#### 【例 2】

有効桁を整数部 2 桁、小数部 1 桁に指定した場合の例

指定値 (丸め方法)	入力データ	丸め結果	桁合わせ	説明
0 または省略 (切り捨て)	99.9 $\underline{5}$	99.90	99.9	0.01 の位を切り捨てます。
1 (切り上げ)	99.9 $\underline{0}$	99.90	99.9	0.01 の位が 0 なので切り上げません。
	99.9 $\underline{1}$	100.00	00.0	0.01 の位が 0 でないので切り上げます。 オーバーフローし、0 値となります。
2 (四捨五入)	99.9 $\underline{4}$	99.90	99.9	0.01 の位を切り捨てます。
	99.9 $\underline{5}$	100.00	00.0	切り上げます。 オーバーフローし、0 値となります。

## 5.8.12 nullseq オペランド

### 意味

CSV ファイルのナル文字列が含まれる入力レコードを、先頭または末尾に配置するオプションです。

このオプションは、CSV ファイルに指定されたソートキーおよびマージキーだけに有効です。CSV ファイル以外を指定した場合は、無効となります。

環境ファイル「EXS\_NULLSEQ」とこのオペランドの両方を省略した場合、オプションコード「0」を仮定します。

### 形式

`nullseq=option-code`

### 指定値

`option-code` に指定する値を次の表に示します。

指定値	意味
0	ナル文字列 (null キー) が含まれるレコードを、末尾に配置します。
1	ナル文字列 (null キー) が含まれるレコードを、先頭に配置します。

### 注意事項

- ソート機能に指定した場合、指定した位置にナル文字列のキーを持つレコードを配置します。
- マージ機能に指定した場合、ナル文字列のキーが指定した位置になければ KBL51212-E エラーとなります。
- コピー機能に指定した場合、有効なりません。
- ナル文字列の扱いについては、「付録 D CSV ファイル内のナル文字列の扱い (SORT EE の機能)」を参照してください。

## 5.8.13 codetype オペランド

### 意味

入力データのエンコード（文字コード）の種類を指定します。  
このオペランドに指定したエンコード（文字コード）で、CSV ファイルの区切り文字（デリミタ）、引用符、文字属性コード「w」、「u」および「v」を解析します。  
このオプションは、テキストファイル、および CSV ファイルだけに有効です。

### 形式

codetype= encode

### 指定値

encode に指定する値を次の表に示します。

encode	意味
ASCII	入力データの文字コードが ASCII（JIS8 コードを含む 0x00 ~ 0xFF の範囲）であることを示します。
SJIS	入力データの文字コードが Shift_JIS であることを示します。
EUC-JP	入力データの文字コードが日本語 EUC であることを示します。
EUC-HJ	入力データの文字コードが EUC-HJ であることを示します。
UTF-8	入力データのエンコードが UTF-8 であることを示します。
UCS-2LE	入力データの文字コードが UCS-2 little endian であることを示します。
UCS-2BE	入力データの文字コードが UCS-2 big endian であることを示します。
UCS-4LE	入力データの文字コードが UCS-4 little endian であることを示します。
UCS-4BE	入力データの文字コードが UCS-4 big endian であることを示します。
UTF-16LE	入力データのエンコードが UTF-16 little endian であることを示します。
UTF-16BE	入力データのエンコードが UTF-16 big endian であることを示します。
UTF-32LE	入力データのエンコードが UTF-32 little endian であることを示します。
UTF-32BE	入力データのエンコードが UTF-32 big endian であることを示します。
UNICODE	入力ファイルの Unicode シグニチャ（BOM）形式を適用します。 SORT EE で使用できる BOM の種類は次のとおりです。 <ul style="list-style-type: none"> <li>• UTF-8…0xEFBBBF（3 バイト）</li> <li>• UTF-16LE…0xFFFE（2 バイト）</li> <li>• UTF-16BE…0xFEFF（2 バイト）</li> <li>• UTF-32LE…0xFFFE0000（4 バイト）</li> <li>• UTF-32BE…0x0000FEFF（4 バイト）</li> </ul>

- このオペランドを省略した場合、環境ファイル「EXS\_CODETYPE」に指定したエンコードが仮定されます。
- このオペランドと環境ファイル「EXS\_CODETYPE」の両方の指定を省略した場合、「SJIS」が仮定されます。
- 「UNICODE」では、入力ファイルが複数個の場合、先頭の入力ファイルの BOM



形式を適用します。

- 「UNICODE」で、最初の入力ファイルに BOM がない場合、次に示すプラットフォームに対応したエンコードを仮定します。

プラットフォーム	エンコード
AIX, HP-UX	UTF-16BE
Linux	UTF-8
Windows	UTF-16LE

#### 注意事項

- 入力ファイル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、このオペランドを指定できません。指定した場合、KBL52209-E メッセージを出力し、エラーになります。
- 入力データと異なるエンコード（文字コード）を指定した場合、デリミタを正しく解析できず、exsort コマンドの実行結果が不当になるおそれがあります。
- 入力ファイルが複数個の場合、入力データのエンコード（文字コード）は統一されていなければなりません。入力ファイルによってエンコード（文字コード）が異なると、改行文字やデリミタを正しく解析できず、exsort コマンドの実行結果が不当になるおそれがあります。
- BOM 付きの入力ファイルを指定する場合、BOM の読み飛ばしを指定する環境変数「EXSORTBOMLNGIN（64 ビット版は「EXSORT64BOMLNGIN」）」または「EXSORTBOMGETPUT（64 ビット版は「EXSORT64BOMGETPUT」）」を指定してください。これらの環境変数の指定がない場合、入力ファイルの BOM を読み飛ばさずに、レコードデータとして処理します。なお、環境変数「EXSORTBOMLNGIN（64 ビット版は「EXSORT64BOMLNGIN」）」および「EXSORTBOMGETPUT（64 ビット版は「EXSORT64BOMGETPUT」）」については「14.1 SORT EE の環境変数の設定」を参照してください。
- 出力ファイルの BOM の有無と種類を指定する場合は、環境変数「EXSORTBOMLNGOUT（64 ビット版は「EXSORT64BOMLNGOUT」）」または「EXSORTBOMGETPUT（64 ビット版は「EXSORT64BOMGETPUT」）」を指定してください。なお、環境変数「EXSORTBOMLNGOUT（64 ビット版は「EXSORT64BOMLNGOUT」）」または「EXSORTBOMGETPUT（64 ビット版は「EXSORT64BOMGETPUT」）」については「14.1 SORT EE の環境変数の設定」を参照してください。

## 5.8.14 reorg オペランド

#### 意味

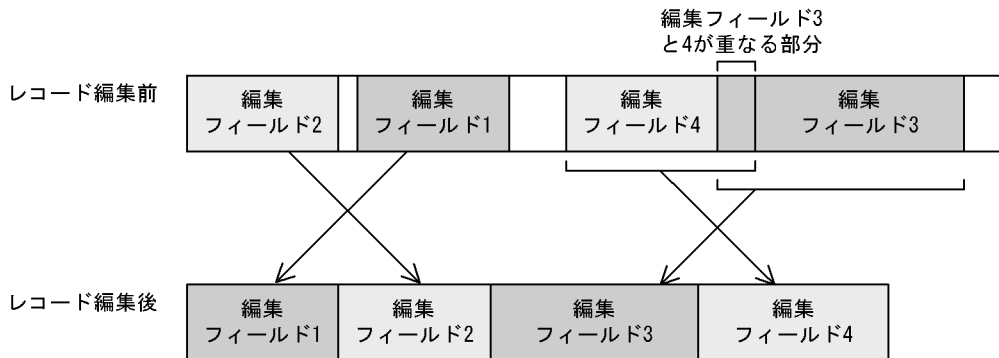
入力レコード編集フォーマットを指定します。

入力レコード中の編集する範囲（以降、編集フィールドと呼びます）を指定した位置に置くことで、レコード形式を再編成する機能です。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

編集フィールドは、指定した順序に従って、順に前の編集フィールドの後方に隣接して置かれます。次の図に示すように、編集フィールドの一部または全部が重なり合う指定もできます。

図 5-6 編集フィールドの指定例



固定長レコードの場合は、編集フィールドがレコード内にすべて含まれている必要があります。

また可変長レコードの場合は、編集フィールドのバイト長を指定しなかった編集フィールドを除き、編集フィールドが実レコード内にすべて含まれている必要があります。

編集フィールドの指定例を次に示します。

### (a) 固定長レコードの場合の例

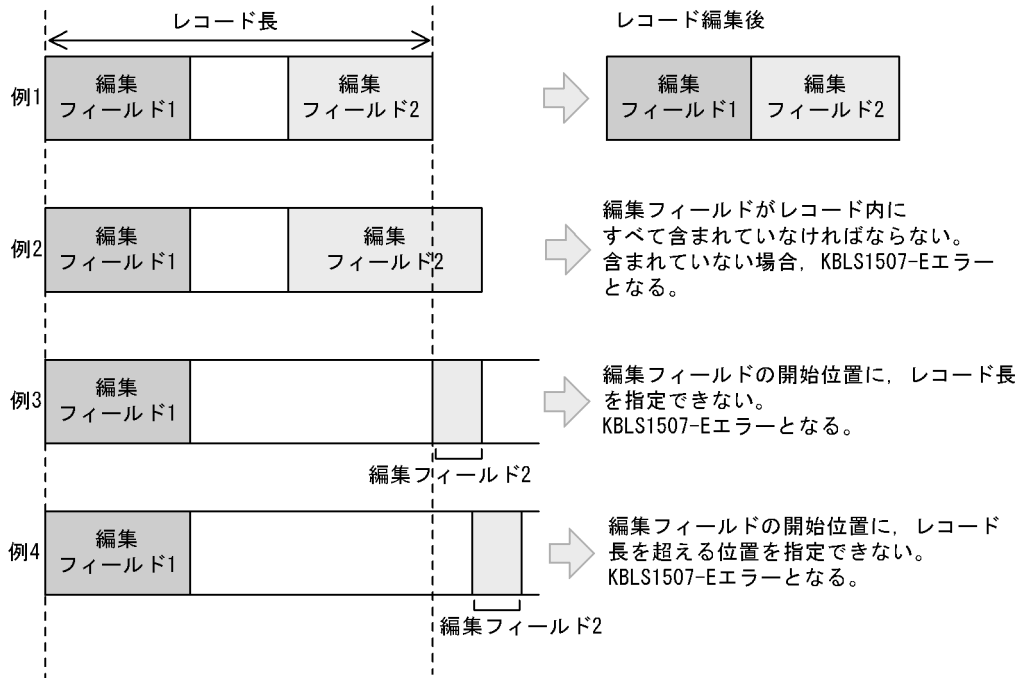
例 1：開始バイト位置 < レコード長，かつ開始バイト位置 + 編集フィールド長 ≤ レコード長

例 2：開始バイト位置 < レコード長，かつ開始バイト位置 + 編集フィールド長 > レコード長

例 3：開始バイト位置 = レコード長

例 4：開始バイト位置 > レコード長

図 5-7 固定長レコードの場合の指定例



## (b) 可変長レコードおよびテキストの場合の例

## (1) 編集フィールド長が指定されている場合

例 1 : 開始バイト位置 < レコード長, かつ 開始バイト位置 + 編集フィールド長 ≤ レコード長

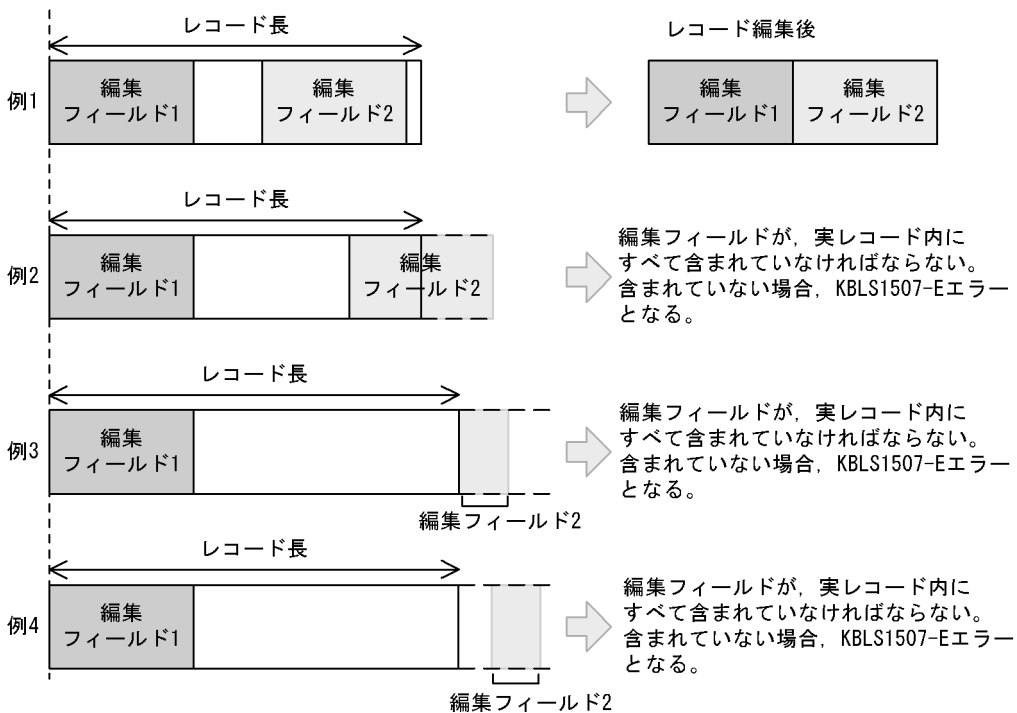
例 2 : 開始バイト位置 < レコード長, かつ 開始バイト位置 + 編集フィールド長 > レコード長

例 3 : 開始バイト位置 = レコード長

例 4 : 開始バイト位置 > レコード長

5. exsort コマンドのパラメタおよびオペランドの詳細

図 5-8 可変長レコードおよびテキストの場合の指定例（編集フィールド長指定あり）



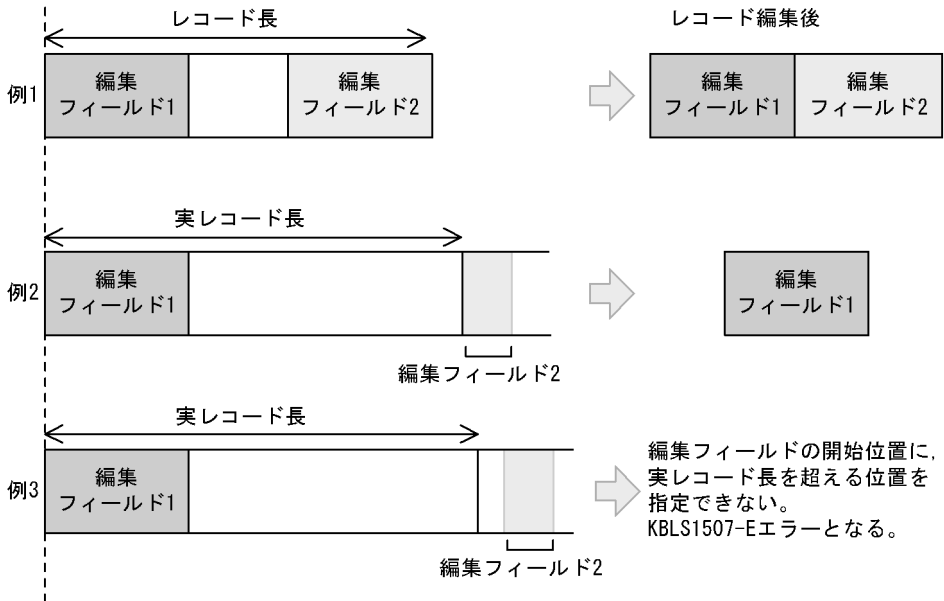
(2) 編集フィールド長が指定されていない場合

例 1 : 開始バイト位置 ≤ レコード長 - 1

例 2 : 開始バイト位置 = レコード長

例 3 : 開始バイト位置 > レコード長

図 5-9 可変長レコードおよびテキストの場合の指定例（編集フィールド長指定なし）



reorg オペランド以外に、中間レコード（ソート、マージ、コピー結果）と、出力レコードに対してレコード編集を指定するパラメタ・オペランドがあります。レコード編集の種類によって、入力レコードと異なった位置にキーや項目を再編成するような場合は注意が必要となります。

ソートキー、マージキー、集約キー、比較項目、および集約項目の位置（範囲）は、「付録 F レコード編集処理で基になるレコード（SORT EE の機能）」に示す規則に従って指定してください。

### 形式

reorg=reorganization-format [ $\Delta$ reorganization-format ...]

指定できるフォーマットの個数は、1～128個です。

### 指定値

reorganization-format の形式を次に示します。

#### reorganization-format オペランド形式 1

[insertpos] + pos [-len] [boundary]

入力レコード中の編集フィールドおよび挿入位置を指定します。

- [insertpos] + pos [-len][boundary] は必ず続けて指定してください。それぞれの値の間に空白やタブを指定しないでください。
- [insertpos] + pos [-len][boundary] を複数指定する場合は、1個以上の空白またはタブで区切って指定します。

#### reorganization-format オペランド形式 2

[insertpos] + insertdata-cnt

## 5. exsort コマンドのパラメタおよびオペランドの詳細

挿入データ形式および挿入位置を指定します。

- `[insertpos] + insertdata - cnt` は必ず続けて指定してください。それぞれの値の間に空白やタブを指定しないでください。
- `[insertpos] + insertdata - cnt` を複数指定する場合は、1 個以上の空白またはタブで区切って指定します。

`reorganization-format` に指定するそれぞれの値について次に示します。

### *insertpos*

- 編集フィールド、および挿入データ（文字列、16 進数、または空白）の挿入開始バイト位置を 5 桁以内の数字で指定します。
- 前の編集フィールドや挿入データと重なる位置は指定できません。
- 前の編集フィールドまたは挿入データとの間に間隔が生じる場合には、空白を埋め込みます。
- `-locatajst` パラメタを指定したかどうかによって、レコード先頭の解釈が異なります。また、テキストファイルの場合は、入力データのエンコードによって指定範囲が異なります。挿入開始バイト位置に指定できる値の範囲を次の表に示します。

ファイル編成			-locatajst パラメタ	
			指定あり	指定なし
固定長順編成ファイル			1 ~ 65,535	0 ~ 65,534
可変長順編成ファイル			1 または 5 ~ 65,539 <sup>※1</sup>	
テキストファイル <sup>※2</sup>	改行文字「LF」	ASCII SJIS EUC-JP EUC-HJ UTF-8	1 ~ 65,535	
		UCS-2LE UCS-2BE UTF-16LE UTF-16BE	1 ~ 65,533 (2 の倍数 + 1 <sup>※3</sup> )	0 ~ 65,532 (2 の倍数 <sup>※3</sup> )
		UCS-4LE UCS-4BE UTF-32LE UTF-32BE	1 ~ 65,529 (4 の倍数 + 1 <sup>※4</sup> )	0 ~ 65,528 (4 の倍数 <sup>※4</sup> )
	改行文字「CRLF」	ASCII SJIS EUC-JP EUC-HJ UTF-8	1 ~ 65,534	0 ~ 65,533
		UCS-2LE UCS-2BE UTF-16LE UTF-16BE	1 ~ 65,531 (2 の倍数 + 1 <sup>※3</sup> )	0 ~ 65,530 (2 の倍数 <sup>※3</sup> )
		UCS-4LE UCS-4BE UTF-32LE UTF-32BE	1 ~ 65,525 (4 の倍数 + 1 <sup>※4</sup> )	0 ~ 65,524 (4 の倍数 <sup>※4</sup> )

## 注※ 1

レコードの前の 4 バイト領域 (RDW : レコードヘッダ) の先頭が 1 となります。

## 注※ 2

入力レコードの改行文字については、「5.8.7 recsepa オペランド」を参照してください。また、入力データのエンコードについては、「5.8.13 codetype オペランド」を参照してください。

## 注※ 3

2 の倍数 (-locatajst パラメタを同時に指定している場合は「2 の倍数 + 1」) 以外を指定すると、KBLS1500-E メッセージが表示されてエラーになります。

## 注※ 4

4 の倍数 (-locatajst パラメタを同時に指定している場合は「4 の倍数 + 1」) 以

## 5. exsort コマンドのパラメタおよびオペランドの詳細

外を指定すると、KBLS1500-E メッセージが表示されてエラーになります。

- 2 番目以降の編集フィールドの挿入開始バイト位置を省略した場合、直前の編集フィールドに隣接して配置します。
- 先頭の編集フィールドの挿入開始バイト位置を省略した場合、挿入開始バイト位置の仮定値を次の表に示します。

ファイル編成	-locatajst パラメタ	
	指定あり	指定なし
固定長順編成ファイル	1	0
可変長順編成ファイル	1 または 5 <sup>※</sup>	0
テキストファイル	1	0

注※

編集フィールドのデータ位置 (+pos) に 1 を指定した場合は 1 を仮定し、1 以外を指定した場合は 5 を仮定します。

### + pos

- 編集フィールドのデータ位置を、5 桁以内の数字で指定します。なお、-locatajst パラメタを指定したかどうかによって、レコード先頭の解釈が異なります。
- 入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定した場合、データ位置には「2 の倍数」を指定してください。2 の倍数以外を指定すると、KBLS1503-E メッセージが表示されてエラーになります。なお、-locatajst パラメタを同時に指定している場合は、「2 の倍数 + 1」を指定してください。
- 入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定した場合、データ位置には「4 の倍数」を指定してください。4 の倍数以外を指定すると、KBLS1503-E メッセージが表示されてエラーになります。なお、-locatajst パラメタを同時に指定している場合は、「4 の倍数 + 1」を指定してください。
- データ位置に指定できる値の範囲は、*insertpos* の挿入開始バイト位置に指定できる値の範囲と同じです。

### - len

- 編集フィールドのバイト長を、5 桁以内の数字で指定します。
- 入力ファイルが可変長順編成ファイル、またはテキストファイルの場合、最後の編集フィールドに限りバイト長を省略できます。省略した場合、指定したデータ位置から実レコードの最後までを編集フィールドとして扱います。
- 編集フィールドのバイト長に指定できる範囲を次の表に示します。



ファイル編成			-locatajst パラメタ	
			指定あり	指定なし
固定長順編成ファイル			1 ~ 65,535	1 ~ 65,535
可変長順編成ファイル			1 ~ 65,539	
テキスト ファイル※ 1	改行文字 「LF」	ASCII SJIS EUC-JP EUC-HJ UTF-8	1 ~ 65,535	
		UCS-2LE UCS-2BE UTF-16LE UTF-16BE	2 ~ 65,534 (2 の倍数※ <sup>2</sup> )	2 ~ 65,534 (2 の倍数※ <sup>2</sup> )
		UCS-4LE UCS-4BE UTF-32LE UTF-32BE	4 ~ 65,532 (4 の倍数※ <sup>3</sup> )	4 ~ 65,532 (4 の倍数※ <sup>3</sup> )
	改行文字 「CRLF」	ASCII SJIS EUC-JP EUC-HJ UTF-8	1 ~ 65,534	1 ~ 65,534
		UCS-2LE UCS-2BE UTF-16LE UTF-16BE	2 ~ 65,532 (2 の倍数※ <sup>2</sup> )	2 ~ 65,532 (2 の倍数※ <sup>2</sup> )
		UCS-4LE UCS-4BE UTF-32LE UTF-32BE	4 ~ 65,528 (4 の倍数※ <sup>3</sup> )	4 ~ 65,528 (4 の倍数※ <sup>3</sup> )

## 注※ 1

入力レコードの改行文字については、「5.8.7 recsepa オペランド」を参照してください。また、入力データのエンコードについては、「5.8.13 codetype オペランド」を参照してください。

## 注※ 2

2 の倍数以外を指定すると、KBLS1505-E メッセージが表示されてエラーになります。

## 注※ 3

4 の倍数以外を指定すると、KBLS1505-E メッセージが表示されてエラーになります。

+ insertdata

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- 挿入するデータの形式を指定します。  
insertdata に指定する値を次の表に示します。

insertdata に指定できる値	データ形式
X	空白
Z	ゼロ
C' 文字列'	文字列
X'16進数'	16進数

- 入力ファイルがテキストファイルの場合、挿入データに改行文字は指定できません。
- 16進数として指定できるのは「0～9」および「A～F」だけです。
- 16進数に指定できる桁数は、次のとおりです。

ファイル編成		X'16進数'の指定桁数
固定長順編成ファイル		偶数桁
可変長順編成ファイル		偶数桁
テキストファイル	ASCII	偶数桁
	SJIS	
	EUC-JP	
	EUC-HJ	
	UTF-8	
	UCS-2LE	4の倍数桁
	UCS-2BE	
	UTF-16LE	
	UTF-16BE	
	UCS-4LE	8の倍数桁
	UCS-4BE	
	UTF-32LE	
	UTF-32BE	

- 入力データのエンコードが「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-16LE または UTF-16BE）」の場合、「UCS-2」の文字コードのデータ形式となります。「付録 H Unicode 操作文字コード一覧（SORT EE）」の「表 H-3 ASCII 対応予約文字一覧」を参照してください。
- 入力データのエンコードが「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-32LE または UTF-32BE）」の場合、「UCS-4」の文字

コードのデータ形式となります。「付録 H Unicode 操作文字コード一覧 (SORT EE)」の「表 H-3 ASCII 対応予約文字一覧」を参照してください。

- 入力データのエンコードが「EUC-HJ」で、マルチバイト文字を挿入する場合は「X'16 進数'」形式で指定してください。
- 文字列データを exsort コマンドに指定する場合、フォーマット全体をダブルクォーテーションで囲んで指定してください。
- exsort パラメタファイルに指定する場合は、ダブルクォーテーションで囲まないでください。

文字列データの指定例を次に示します。

指定先	記述例	備考
exsort コマンド	reorg="0+32-10 +C'ABCD'-1"	ダブルクォーテーションで囲みます。
exsort パラメタファイル	reorg=0+32-10 +C'ABCD'-1	ダブルクォーテーションで囲みません。

#### — cnt

- 挿入データの繰り返し回数を、1 ～ 65,535 の 5 桁以内の数字で指定します。
- この設定は省略できません。ただし、「挿入データのバイト長<sup>※</sup>×繰り返し回数」の値は、65,535 バイト以下にする必要があります。(注※ 16 進数で指定した場合は、2 桁で 1 バイトとなります。)

#### boundary

- 編集フィールドを配置するときの境界調整を指定します。
- レコードの先頭は 8 バイト境界から始まるものとします。
- 境界調整の結果、前の編集フィールドとの間にできる間隔には X'00' が挿入されます。編集フィールドのバイト長を省略した場合は、境界調整は指定できません。

boundary に指定できる値	意味
省略	境界調整をしない。
H	2 バイト境界に調整する。
F	4 バイト境界に調整する。
D	8 バイト境界に調整する。

#### レコード編集後のレコード長の指定

レコード編集後のレコード長を -outfile パラメタの reclen オペランドおよび minlen オペランドに指定します。

各オペランドの形式については、「5.10.3 reclen オペランド」および「5.10.4 minlen オペランド」を参照してください。

レコード編集と reclen オペランドおよび minlen オペランドとの関係を次の表に示

## 5. exsort コマンドのパラメタおよびオペランドの詳細

します。

表 5-11 レコード編集と reclen オペランドおよび minlen オペランドとの関係

入力ファイル編成	条件 (レコード編集後のレコード長と reclen, minlen オペランドとの関係)	レコード編集結果
固定長順	reclen < 編集レコード長	× (KBLS1512-E)
	reclen = 編集レコード長	○
	reclen > 編集レコード長	× (KBLS1512-E)
可変長順	minlen > 編集レコード長	× (KBLS1513-E)
	minlen ≤ 編集レコード長 ≤ reclen	○
	編集レコード長 > reclen	× (KBLS1511-E)
テキスト	minlen > 編集レコード長	× (KBLS1513-E)
	minlen ≤ 編集レコード長 ≤ reclen	○
	編集レコード長 > reclen	× (KBLS1511-E)

### (凡例)

編集レコード長：レコード編集後のレコード長を示します。

○：レコード編集後のレコードをそのまま出力します。

×：丸括弧内に記載しているメッセージが表示され、エラーとなります。

### 指定例

入力レコード形式が可変長レコード (最小レコード長 12 バイト, 最大レコード長 28 バイト) で, `-locatajst` パラメタを指定した場合, および `-locatajst` パラメタを指定しない場合の指定例を示します。

指定例は `exsort` パラメタファイルに指定した場合です。 `exsort` コマンドに指定する場合は, フォーマット全体をダブルクォーテーションで囲んでください。

`-locatajst` パラメタを指定した場合の例 (入力データのエンコードに「SJIS」を指定)

```
-locatajst
-inpfile
fileorg=V
reorg=+1-6 +9-4D +X'FF'-3 21+17
file=INFILE
-outfile
minlen=16
reclen=32
file=OUTFILE
```

- 入力レコード長  
最小レコード長 = 12 バイト, 最大レコード長 = 28 バイト (RDW の長さは含まない)
- 編集フィールド 1  
1 ~ 6 バイトの範囲をレコード先頭に挿入します。レコードの前の 4 バイト

領域 (RDW) の先頭を 1 バイト目とします。

- 編集フィールド 2

9 ~ 12 バイトの範囲を編集フィールド 1 に隣接して挿入します。8 バイト境界調整で生じた間隔には、NULL が挿入されます。

- 挿入データ

X'FF' のデータを編集フィールド 2 に隣接して 3 バイト挿入します。

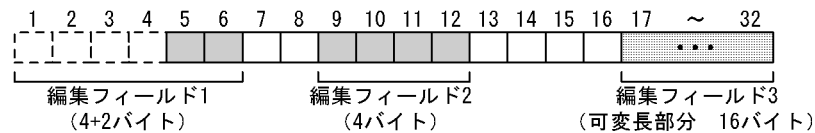
- 編集フィールド 3

可変長部分 (最大 16 バイト) を 21 バイト目に挿入します。挿入データとの間に生じた間隔には、空白が挿入されます。

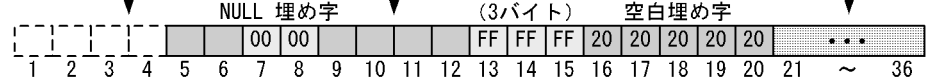
入力レコード長が最小レコード長の場合、可変長部分はありません (0 バイト)。入力レコード長が最大レコード長の場合、可変長部分は 16 バイトです。

編集後のレコード長は、最小レコード長 16 バイト、最大レコード長 32 バイトとなります (RDW の長さ 4 バイトは含まません)。

【入力レコード】



【編集結果】



-locatajst パラメタを指定しない場合の例 (入力データのエンコードに「SJIS」を指定)

```
-infile
fileorg=V
reorg=+0-2 +4-4D +X'FF'-3 16+12
file=INFILE
-outfile
minlen=16
reclen=32
file=OUTFILE
```

- 入力レコード長

最小レコード長 = 12 バイト, 最大レコード長 = 28 バイト

- 編集フィールド 1

0 ~ 2 バイトの範囲をレコード先頭に挿入する。

- 編集フィールド 2

4 ~ 7 バイトの範囲を、編集フィールド 1 に隣接して挿入します。8 バイト境界調整で生じた間隔には、NULL が挿入されます。

- 挿入データ

## 5. exsort コマンドのパラメタおよびオペランドの詳細

X'FF' のデータを、編集フィールド 2 に隣接して 3 バイト挿入します。

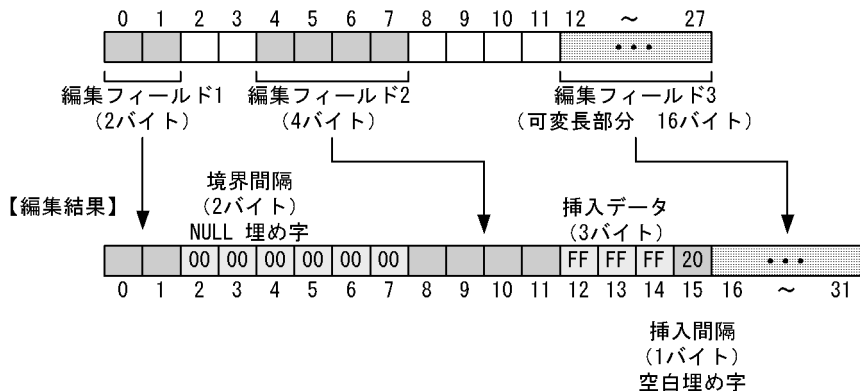
- 編集フィールド 3

可変長部分 (最大 16 バイト) を 16 バイト目に挿入します。挿入データとの間に生じた間隔には、空白が挿入されます。

入力レコード長が最小レコード長の場合は、可変長部分はありません (0 バイト)。入力レコード長が最大レコード長の場合は、可変長部分は 16 バイトです。

編集後のレコード長は、最小レコード長 16 バイト、最大レコード長 32 バイトとなります。

### 【入力レコード】



### 注意事項

- 編集機能を使用する場合、-outfile パラメタの reclen オペランド、および minlen オペランドの指定が必須となる場合があります。出力ファイルのファイル編成と -outfile パラメタの reclen オペランドおよび minlen オペランドの関係を表 5-12 に示します。

表 5-12 出力ファイルのファイル編成と reclen オペランドおよび minlen オペランドとの関係

出力ファイル	reclen オペランド	minlen オペランド
可変長順編成ファイル	◎ (指定しない場合, KBL1659-E エラーとなる)	◎ (指定しない場合, KBL1660-E エラーとなる)
固定長順編成ファイル	◎ (指定しない場合, KBL1659-E エラーとなる)	× (指定している場合, KBL1664-E エラーとなる)
テキストファイル	○ (指定しない場合, 65,535 を仮定する)	○ (指定しない場合, 1 を仮定する)

(凡例)

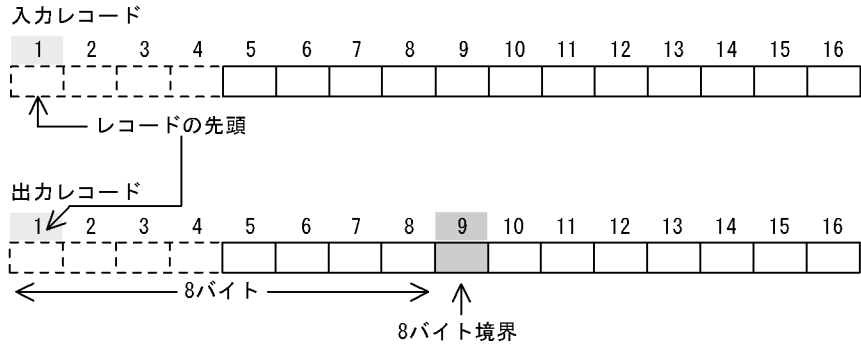
- ◎ : 指定必須です。
- : 指定できます。

× : 指定できません。

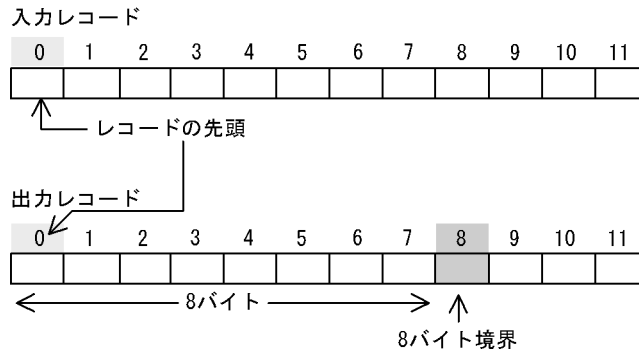
- 可変長レコードの場合、`-locatajst` パラメタを指定したかどうかによって、レコードの先頭位置が異なります。これに伴って、境界調整時の 8 バイト境界の位置も次の例のように変わります。

`-locatajst` パラメタを指定している場合

【例】先頭位置と境界の相違



`-locatajst` パラメタを指定しない場合



- `-locatajst` パラメタを指定し入力ファイルが可変長順編成ファイルの場合、先頭のフォーマットだけは、レコードの前の 4 バイト領域 (RDW) を指定してもエラーになりません。先頭より後のフォーマットにこの領域を指定すると **KBLS1515-E** エラーとなります。レコードの前の 4 バイト領域 (RDW) を指定する場合は、次のどれかの形式で指定してください。

- `1+1-n` ( $n$  は 4 以上)
- `+1-n` ( $n$  は 4 以上)
- `1+1`
- `+1`

ここで示した形式以外を指定すると次に示すエラーになります。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

指定例	出力される エラーメッセージ	エラー原因
2+1-4	KBLS1500-E	開始バイト位置の値が 1 ではありません。
1+2-4	KBLS1503-E	データ位置の値が 1 ではありません。
1+1-3	KBLS1505-E	データ長の値が 4 以上ではありません。

- 挿入データを指定する場合、挿入開始バイト位置にレコードの前の 4 バイト領域 (RDW) を指定できません。指定した場合、KBLS1515-E エラーとなります。
- 挿入開始バイト位置は、フォーマット間で昇順となるように指定してください。昇順となっていない場合、KBLS1501-E エラーとなります。  
指定例を次に示します。  
正しい指定例：10+1-5 20+10-5  
誤った指定例：20+1-5 10+10-5 (最初の挿入位置 (20) と次の挿入位置 (10) が逆転している)
- 入力ファイルがテキストの場合、改行を入力ファイルに合わせて付加します。
- 開始バイト位置や編集フィールド長の指定によっては、入力データと異なる形式 (例えば、漢字が漢字でなくなる、符号付き数字が符号なし数字となるなど) となることがあります。このことを考慮に入れて、開始バイト位置や編集フィールド長の値を設定してください。

### 5.8.15 csvreorg オペランド

#### 意味

CSV ファイルの入力レコード編集フォーマットを指定します。

このオペランドは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、KBLS1522-E エラーとなります。

#### 形式

```
csvreorg=csv-reorganization-format [ $\Delta$ csv-reorganization-format ... ]
```

入力レコード編集フォーマットを指定します。

指定できるフォーマットの個数は、1 ~ 128 個です。

入力レコード中の編集するセル (以降、編集セルと呼びます) を指定した位置に置くことで、レコード形式を再編成する機能です。編集セルは、指定した順序に従って、前の編集セルの後方に隣接して置かれます。

csv-reorganization-format の形式を次に示します。

#### csv-reorganization-format オペランド形式 1

```
[insert_cellpos] + cellpos [-cellnum]
```

入力レコード中のセルと、挿入位置を指定します。

- [insert\_cellpos] + cellpos [- cellnum] は必ず続けて指定してください。そ



それぞれの値の間に空白やタブを指定しないでください。

- `[insert_cellpos] + cellpos [- cellnum]` を複数指定する場合は、1 個以上の空白またはタブで区切って指定します。

#### csv-reorganization-format オペランド形式 2

`[insert_cellpos] + insertdata - cnt`

挿入データ形式と、挿入位置を指定します。

- `[insert_cellpos] + insertdata - cnt` は必ず続けて指定してください。それぞれの値の間に空白やタブを指定しないでください。
- `[insert_cellpos] + insertdata - cnt` を複数指定する場合は、1 個以上の空白またはタブで区切って指定します。

#### 指定値

##### `insert_cellpos`

`insert_cellpos` に指定する値を次の表に示します。レコード内のセルの挿入位置（レコード編集後のセルの位置）を、番号で指定します。

指定値	機能	説明
0	連結	前に指定した編集セルまたは挿入データの内容に連結し、1 つのセルとして配置します。指定した順番に、右側に連結していきます。 <ul style="list-style-type: none"> <li>• 直前に指定された編集セルが連続したセルの場合、最後のセルの内容に連結します。</li> <li>• 最初の編集フォーマットに挿入位置「0」を指定した場合、セル番号「1」からセルまたは挿入データを配置します。</li> <li>• 連結するセルの内容が引用符で囲まれている場合、引用符を除いたデータを連結し、最後にデータ全体を引用符で囲みます。</li> <li>• 連結する挿入データに改行文字やデリミタが指定されている場合、連結したデータを引用符で囲みます。</li> </ul>
1 ~ 16384	挿入	指定した位置に、指定した編集セルまたは挿入データを配置します。 <ul style="list-style-type: none"> <li>• レコードの先頭セル番号を「1」、最大セル番号を「n」とし、1 ~ n までの番号を指定します。最大セル番号は、16,384 番です。</li> <li>• 挿入位置は、フォーマット間で昇順となるように指定します。</li> <li>• 前の編集セル範囲または挿入データとの間に間隔が生じる場合には、ナル文字列のセルが作成されます。</li> </ul>

「0」を指定した連結指定で、連結したデータを引用符で囲む場合の例を次に示します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

### 指定オペランド

csvreorg="1+2-1 0+1-1 +3-1 0+2-1 +C'V,W' -1 0+2-1 +2-1 0+C'xyz' -1"

出力セル1
出力セル2
出力セル3
出力セル4

### 入力レコード

セル1	セル2	セル3	
"ABC"	123	def	←

### 出力レコード

セル1	セル2	セル3	セル4	
"123ABC"	def123	"V,W123"	123xyz	←

(凡例)

NULL : ナル文字列 (空セル)

← : 改行文字

出力セル 1 と 3 の内容が引用符で囲まれて出力されます。

- 出力セル 1 : 入力セル 2 の内容「123」に、入力セル 1 の内容「ABC」を連結し、引用符で囲む。
- 出力セル 2 : 入力セル 3 の内容「def」に、入力セル 2 の内容「123」を連結する。引用符はなし。
- 出力セル 3 : デリミタを含む挿入データ「V,W」に、入力セル 2 の内容「123」を連結し、引用符で囲む。
- 出力セル 4 : 入力セル 2 の内容「123」に、挿入データ「xyz」を連結する。引用符はなし。

### + *cellpos* [ - *cellnum* ]

レコード内の編集セルの位置と個数を指定します。

- レコードの先頭セル番号を「1」、最大セル番号を「n」とし、1～nまでの番号を指定します。最大セル番号は、16,384 番です。
- 連続したセルを指定する場合は、+ *cellpos* - *cellnum* で番号と個数を指定します。例えば、「+ 5 - 3」と指定した場合、5～7 番目の 3 個の連続したセルが対象となります。
- 連続したセルを指定した場合、*insert\_cellpos* で指定された挿入位置に、順番に挿入します。*insert\_cellpos* を省略した場合は、前の編集セルに隣接して挿入します。
- 最後の編集セルに限り、[ - *cellnum* ] を省略できます。省略した場合、+ *cellpos* から実レコードの最後のセルまでを編集セル範囲として扱います。
- [ - *cellnum* ] を省略し、+ *cellpos* で指定した編集セル位置が実レコードの最後のセルを超えた場合、編集セルの挿入は行いません。挿入する編集セルが存在しない場合の例題を次に示します。

入力レコード

セル1	セル2	セル3
"ABC"	123	"def"

 ←

- 「+ce/pos」 指定 (「-ce/num」 省略)

(例1) csvreorg=+2

出力レコード

セル1	セル2
123	"def"

 ←

入力セル「2」から実レコードの最後のセルまでを挿入。

(例2) csvreorg=+4

出力レコード

←

入力セル「4」は存在しないため、挿入セルなし。

- 「+ce/pos-ce/num」 指定

(例3) csvreorg=+2-2

出力レコード

セル1	セル2
123	"def"

 ←

入力セル「2」「3」のセル2個を挿入。

(例4) csvreorg=+3-2

出力レコード

セル1	セル2
"def"	NULL

 ←

入力セル「3」とナル文字列(空セル)のセル2個を挿入。

(例5) csvreorg=+4-2

出力レコード

セル1	セル2
NULL	NULL

 ←

入力セル「4」は存在しないため、ナル文字列(空セル)を2個挿入。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- 「*insert\_cell/pos+cell/pos-cell/num*」 指定

(例6) *csvreorg=3+3-2*

出力レコード

セル1	セル2	セル3	セル4
NULL	NULL	"def"	NULL

←

入力セル「3」とナル文字列（空セル）のセル2個を、出力セル「3」へ挿入。  
出力セル「1」「2」はナル文字列（空セル）を挿入。

- 「*insert\_cell/pos+cell/pos*」 指定（「*-cell/num*」省略）

(例7) *csvreorg=3+4*

出力レコード

セル1	セル2
NULL	NULL

←

入力セル「4」は存在しないため、入力データからの挿入セルはなし。  
出力セル「1」「2」はナル文字列（空セル）を挿入。

(凡例)

NULL : ナル文字列（空セル）

← : 改行文字

### + *insertdata*

*insertdata* に指定する値を次の表に示します。

指定値	データ形式
X	空白
Z	ゼロ
NULL	ナル文字列（空セル）
C' 文字列'	文字列（改行文字は指定できません）
X'16進数'	16進数

- 16進数として指定できるのは「0～9」および「A～F」だけです。
- 16進数に指定できる桁数は、次のとおりです。

入力データのエンコード	X'16進数'の指定桁数
ASCII SJIS EUC-JP EUC-HJ UTF-8	偶数桁

入力データのエンコード	X'16 進数' の指定桁数
UCS-2LE UCS-2BE UTF-16LE UTF-16BE	4 の倍数桁
UCS-4LE UCS-4BE UTF-32LE UTF-32BE	8 の倍数桁

- 挿入データに改行文字やデリミタが指定された場合は、引用符で囲んだ（改行文字とデリミタをエスケープする）形式にして出力します。
- 入力データのエンコードが「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」である場合、「UCS-2」の文字コードのデータ形式となります。「付録 H Unicode 操作文字コード一覧 (SORT EE)」の「表 H-3 ASCII 対応予約文字一覧」を参照してください。
- 入力データのエンコードが「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」である場合、「UCS-4」の文字コードのデータ形式となります。「付録 H Unicode 操作文字コード一覧 (SORT EE)」の「表 H-3 ASCII 対応予約文字一覧」を参照してください。
- 入力データのエンコードが「EUC-HJ」で、マルチバイト文字を挿入する場合は「X'16 進数'」形式で指定してください。
- 改行文字は、「X'16 進数'」形式で挿入指定できます。なお、「C' 文字列'」形式では改行文字は指定できません。次に示す使い方で、挿入データ中に改行文字を指定できます。改行文字のデータ形式については、「付録 H Unicode 操作文字コード一覧 (SORT EE)」の「表 H-3 ASCII 対応予約文字一覧」を参照してください。

入力データのエンコードに「SJIS」を指定した場合の例

```
csvreorg="1+C' ABC' -1 0+X' 0A' -1 0+C' DEF' -1 2+C' 12345' -1"
```

セル1	セル2
ABC←DEF	12345

(凡例) ←: 改行文字

- 文字列または 16 進数データを exsort コマンドに指定する場合、フォーマット全体をダブルクォーテーションで囲んで指定してください。exsort パラメタファイルに指定する場合は、ダブルクォーテーションで囲まないでください。文字列データの指定例を次に示します。

指定先	記述例	備考
exsort コマンド	csvreorg="1+2-1 △ +C'ABCD' -1"	ダブルクォーテーションで囲みます。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

指定先	記述例	備考
exsort パラメタファイル	csvreorg=1+2-1 △ +C'ABCD'-1	ダブルクォーテーションで囲みません。

### — cnt

挿入データの繰り返し回数を、1～65,535の5桁以内の数字で指定します。

- この指定は省略できません。
- 挿入データに「NULL (ナル文字列)」を指定した場合、繰り返し回数は必ず1となります。2以上を指定しても1が仮定されます。
- 「挿入データのバイト長※×繰り返し回数」の値は、65,535バイト以下にする必要があります。

注※ 16進数で指定した場合は、2桁で1バイトです。

### 注意事項

- CSVファイルの編集機能では、`-outfile`、`-catout`、または`-splitout`パラメタの`reclen`オペランド、および`minlen`オペランドの指定は必須ではありません。`reclen`オペランドを省略した場合、CSVファイルの最大レコード長を仮定します。また、`minlen`オペランドを省略した場合、CSVファイルの最小レコード長の「0」を仮定します。CSVファイルの最大レコード長については、「表 3-5 SORT EEのパラメタの最小値および最大値」を参照してください。
- `reclen`および`minlen`オペランドを指定する場合、レコード編集後のレコード長（編集レコード長）は「`minlen ≤ 編集レコード長 ≤ reclen`」である必要があります。

編集レコード長と reclen, minlen との関係	レコード編集結果
<code>minlen &gt; 編集レコード長</code>	× (KBLS1513-E)
<code>minlen ≤ 編集レコード長 ≤ reclen</code>	○
<code>編集レコード長 &gt; reclen</code>	× (KBLS1511-E)

### (凡例)

編集レコード長：レコード編集後のレコード長を示します。

○：レコード編集後のレコードをそのまま出力します。

×：括弧内に記載しているメッセージが表示され、エラーとなります。

- `-outfile`、`-catout`、または`-splitout`パラメタの`cellnum`オペランドを指定している場合、レコード編集後のレコードに対してセル数を調整して出力します。詳細は、「5.10.7 cellnum オペランド」を参照してください。
- 挿入位置は、フォーマット間で昇順となるように指定してください。昇順となっていない場合、KBLS1519-Eエラーとなります。指定例を次に示します。

正しい指定例：10+1-5 20+10-5

誤った指定例：20+1-5 10+10-5

(最初の挿入位置 (20) と次の挿入位置 (10) が逆転している)

- `-outfile`, `-catout`, または `-splitout` パラメタの `recsepa` オペランドに指定した改行文字を付加します。`recsepa` オペランドの指定を省略した場合は, `-infile`, `-catinp`, または `-splitinp` パラメタの `recsepa` オペランドの指定に従います。詳細は, 「5.10.5 `recsepa` オペランド」を参照してください。
- レコード編集では, セルとセルの間のデリミタは必ず 1 文字を配置します。入力レコードの連続したデリミタを 1 つのデリミタとして扱う「連続区切り文字オプション (INPCONSEPA)」を指定している場合でも, デリミタは 1 文字となります。連続しているデリミタは 1 文字にまとめられるので注意してください。
- レコード編集を指定するとデリミタは必ず 1 文字で配置されるので, レコード出力時に連続したデリミタを 1 つのデリミタに置き換える「連続区切り文字オプション (OUTCONSEPA)」と同時に指定した場合, `OUTCONSEPA` オプションは無効となります。
- 配置するデリミタは次の順番で決定されます。なお, 複数のデリミタを指定している場合は, 先頭に指定したデリミタとなります。
  1. `-outfile`, `-catout` または `-splitout` パラメタの `cellsepa` オペランドに指定したデリミタ
  2. 環境ファイルの「`EXS_OUTCELLSEPA`」に指定したデリミタ
  3. `-infile`, `-catinp`, または `-splitinp` パラメタの `cellsepa` オペランドに指定したデリミタ
  4. 環境ファイルの「`EXS_CELLSEPA`」に指定したデリミタ
  5. コンマ

#### 関連パラメタ・オペランド

- `-outfile` パラメタ `recsepa` オペランド
- `-outfile` パラメタ `cellsepa` オペランド
- `-outfile` パラメタ `cellnum` オペランド
- `-outfile` パラメタ `reclen` オペランド
- `-outfile` パラメタ `minlen` オペランド
- `-infile` パラメタ `recsepa` オペランド
- `-infile` パラメタ `cellsepa` オペランド
- `-infile` パラメタ `codetype` オペランド

#### 指定例

指定例は `exsort` パラメタファイルに指定した場合です。`exsort` コマンドに指定する場合は, フォーマット全体をダブルクォーテーションで囲んでください。

**【例 1】(入力データのエンコードに「SJIS」を指定した場合の例)**

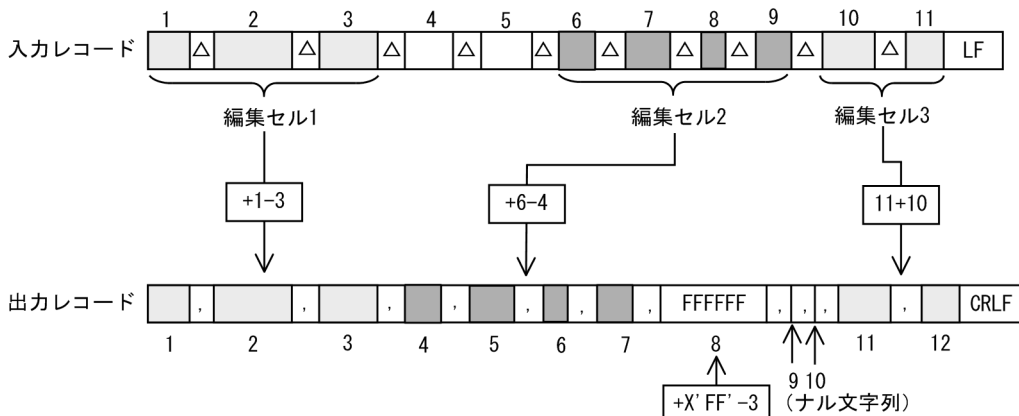
## 5. exsort コマンドのパラメタおよびオペランドの詳細

```

-inpfile
fileorg=C
recsepa=LF
cellsepa='△'
csvreorg=+1-3 +6-4 +X'FF'-3 11+10
file=INFILE
-outfile
recsepa=CRLF
cellsepa=','
file=OUTFILE

```

- 入力レコードの改行文字 (recsepa=LF) : 0x0A
- 入力レコードのデリミタ (cellsepa='△') : 半角空白
- 編集セル 1  
セル 1～3 を, レコード先頭に挿入します。
- 編集セル 2  
セル 6～9 を, 編集セル 1 に隣接して挿入します。
- 挿入データ  
X'FF' (3 バイト) のデータを, 編集セル 2 に隣接して挿入します。
- 編集セル 3  
挿入位置 11 番に, 10 番からレコード末尾までのセルを挿入します。挿入データとの間に生じた間隔 (挿入位置 9 と 10) にはナル文字列が挿入されます。
- 出力レコードの改行文字を, 「CRLF」に変更して出力します。(recsepa=CRLF)
- 出力レコードのデリミタを, コンマに変更して出力します。(cellsepa=',')



### 【例 2】 (入力データのエンコードに「SJIS」を指定した場合の例)

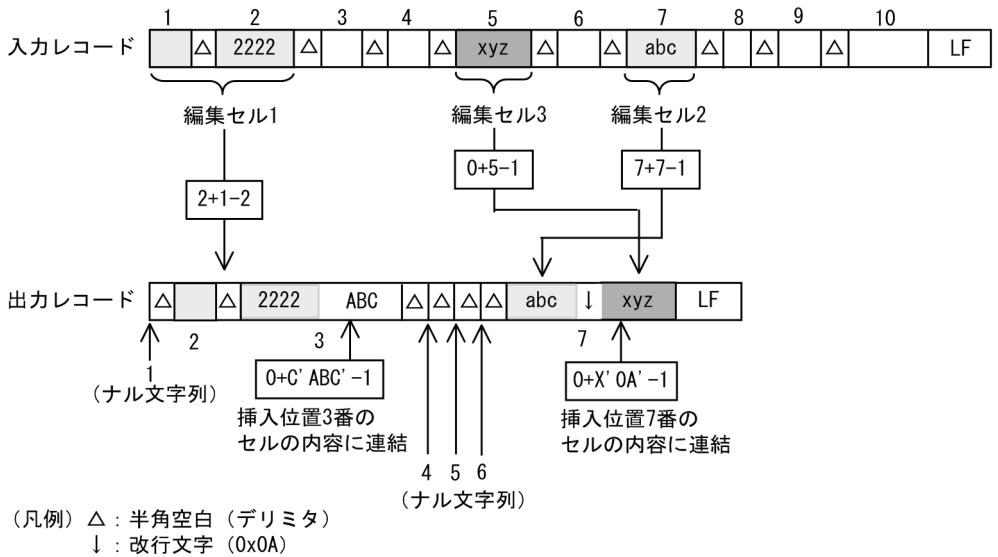
```

-inpfile
fileorg=C
recsepa=LF
cellsepa='△'
csvreorg=+2+1-2 0+C'ABC'-1 7+7-1 0+X'0A'-1 0+5-1
file=INFILE
-outfile
file=OUTFILE
quottype=3

```



- 入力レコードの改行文字 (recsepa=LF) : 0x0A
- 入力レコードのデリミタ (cellsepa='△') : 半角空白
- 編集セル 1 (挿入)  
挿入位置 2 番から, セル 1 と 2 を挿入します。挿入位置との間に生じた間隔 (挿入位置 1) にはナル文字列が挿入されます。
- 挿入データ (連結)  
挿入位置 3 番のセルの内容に, 挿入データ 「C'ABC'」を連結します。セルの内容の右側に連結します。
- 編集セル 2 (挿入)  
挿入位置 7 番から, セル 7 を挿入します。挿入位置との間に生じた間隔 (挿入位置 4 ~ 6) にはナル文字列が挿入されます。
- 挿入データ (連結)  
挿入位置 7 番のセルの内容に, 挿入データ 「X'0A' (改行文字)」を連結します。セルの内容の右側に連結します。
- 編集セル 3 (連結)  
挿入位置 7 番のセルの内容に, セル 5 の内容を連結します。セルの内容の右側に連結します。
- セルを引用符で囲んで出力します。(quottype=3)



## 5.8.16 {include | omit} オペランド

### 意味

入力レコードの選択条件式を指定します。

**include** オペランド :

条件を満たすレコードを選択します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

omit オペランド :

条件を満たさないレコードを選択します。

### 形式

```
{include | omit} ="condition-expression [△ {AND | OR} △  
condition-expression ... ]"
```

レコードを選択するための選択条件式を指定します。

複数の条件式 (*condition-expression*) を丸括弧「()」で囲むことによって、条件式の優先順位を変更することができます。

- コマンドに指定する場合は、選択条件式全体をダブルクォーテーションで囲みます。
- exsort パラメタファイルに指定する場合は、選択条件式をダブルクォーテーションで囲んではなりません。
- 指定できる選択条件式の個数と比較項目の合計長を、環境ファイル「EXS\_SELNUMMAX」で指定できます。なお、比較項目の合計長は、入力データのエンコードによって異なります。環境ファイル「EXS\_SELNUMMAX」については「15.1.1(16) EXS\_SELNUMMAX」を参照してください。
- 使用できる括弧は、ネストレベル 0 ~ 24 です。

### 指定値

*condition-expression* の形式を次に示します。

#### *condition-expression* オペランド形式 1

```
cond-attribute+pos-len △ comp △ cond-value
```

比較項目と条件値を比較する条件式を定義する。

#### *condition-expression* オペランド形式 2

```
cond-attribute+pos-len △ comp △ cond-attribute+pos-len
```

比較項目と比較項目を比較する条件式を定義する。

- *cond-attribute* + *pos* - *len* は間に空白やタブを置かないでつなげて指定します。
- *comp* (比較演算子) の前後は、1 個以上の空白またはタブで区切って指定します。
- {AND|OR} (論理演算子) の前後は、1 個以上の空白またはタブで区切って指定します。

*condition-expression* に指定するそれぞれの値について次に示します。

#### *cond-attribute*

比較項目のデータ形式を指定します。

<i>cond-attribute</i>	意 味
j	比較項目が文字 (JIS8 単位コード <sup>※1</sup> ) であることを指定する。
c	比較項目が文字であることを指定する。 <sup>※2</sup>

cond-attribute	意 味
n	比較項目が符号付き固定小数点 2 進数であることを指定する。※ 8
m	比較項目が符号なし 2 進数であることを指定する。※ 8
p	比較項目が内部 10 進数であることを指定する。
z	比較項目が外部 10 進数であることを指定する。※ 4
x	比較項目が左符号付き外部 10 進数であることを指定する。※ 4
s	比較項目が右分離符号付き外部 10 進数であることを指定する。※ 4
t	比較項目が左分離符号付き外部 10 進数であることを指定する。※ 4
g	比較項目が日付 (英数字) であることを指定する。※ 4
d	比較項目が文字 (左分離符号付き数字) であることを指定する。※ 4
w	比較項目が文字 (ワイド文字※ 3, ※ 5) であることを指定する (テキストファイルだけ指定する)。
u	比較項目が文字 (マルチバイト文字※ 5, ※ 6, ※ 7) であることを指定する (テキストファイルだけ指定できる)。
v	比較項目が文字 (マルチバイト文字※ 5, ※ 6, ※ 7) であることを指定する (Unicode の IVS に対応した属性で、テキストファイルだけ指定できる)。

## 注※ 1

ASCII コード (0x00 ~ 0x7F) を含む, JIS8 単位コード (0x00 ~ 0xFF) を意味します。

入力データに日本語などのマルチバイト文字が含まれている場合, マルチバイト文字ではなく, シングルバイト文字として扱います。

EBCDIK コードの体系に従って比較項目を比較する「EBCDIK コード比較オプション機能」を使用できます。「EBCDIK コード比較オプション機能」については、「5.5.3 EBCDIK コード比較オプション (EBCJ, EBCE オプション)」を参照してください。

## 注※ 2

入力データのエンコードの種別には対応しません。入力データの文字を構成するバイト長を意識しないで, 常に 1 バイト長のシングルバイト文字として扱います。ASCII コードや JIS8 単位コード向けの属性です。

## 注※ 3

入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定した場合, 1 文字 2 バイトのワイド文字として扱います。また, サロゲートペア文字は 2 文字として扱います。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、  
「UTF-32BE」または「UNICODE（入力ファイルの Unicode シグニチャ  
(BOM) の形式が UTF-32LE または UTF-32BE)」を指定した場合、1 文字 4  
バイトのワイド文字として扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参  
照してください。なお、属性コード「w」は固定長順編成ファイルと可変長順  
編成ファイルの入力ファイルに対して指定できません。指定した場合は  
KBLS1303-E メッセージが表示されて、エラーになります。

### 注※ 4

入力データのエンコードで解釈します。

「ASCII」、「SJIS」、「EUC-JP」、「EUC-HJ」または「UTF-8」を指定した場  
合は 1 バイト、「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」  
を指定した場合は 2 バイト、「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または  
「UTF-32BE」を指定した場合は 4 バイトの長さで文字を扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参  
照してください。

### 注※ 5

比較項目範囲の入力データがすべて ASCII 対応コード（英数字）である場  
合は、属性コード「c」を指定した方が処理効率が向上します。属性コード「c」  
は 1 文字 1 バイトで扱いますので、エンコードに対応したバイト長を指定して  
ください。

ASCII 対応コードについては、「付録 H Unicode 操作文字コード一覧 (SORT  
EE)」を参照してください。

1 文字の長さについては、「付録 I.2(1) 文字の長さ」を参照してください。

文字属性コードの特徴については、「付録 I.2(2) 文字属性コードの特徴」を参  
照してください。

### 注※ 6

属性コード「u」と「v」は、入力データのエンコードで解釈します。1 文字の  
長さは可変長として扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参  
照してください。また、マルチバイト文字の扱いについては、「付録 I.2 文字  
列データの扱い」を参照してください。

### 注※ 7

入力データのエンコードに「UTF-8」、「UTF-16LE」、「UTF-16BE」、  
「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE  
(入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE、  
UTF-16BE、UTF-32LE または UTF-32BE)」のどれかを指定した場合、

Unicode の IVS (Ideographic Variation Sequence) を次のとおり解釈します。  
Unicode の IVS については、「付録 I.2(6) IVS 対応」を参照してください。

- 属性コード「u」は、UCS コードが一致する文字であれば、Unicode の IVS の異体字であっても同じ文字として扱います。
- 属性コード「v」は、UCS コードが一致する文字であっても、VS (Variation Selector, 異体字セレクタ) が異なる場合は、異なる文字として扱います。

#### 注※ 8

属性コード「n」、「m」および「a」は、数値として比較するため、バイトオーダーが関係します。バイトオーダーについては、「2.6 エンディアンオプション機能」および「5.5.4 入力ファイルのエンディアンオプション (BIG | INPBIG | INPLTL オプション)」を参照してください。

- 符号付き固定小数点数「n」: 2 バイト (signed short int), 4 バイト (signed int), 8 バイト (signed long long int)
- 符号なし 2 進数「m」、「a」: 2 バイト (unsigned short int), 4 バイト (unsigned int), 8 バイト (unsigned long long int)

#### 比較項目同士の組み合わせ

比較項目同士の比較ができる属性の組み合わせは次のとおりです。

項目 1 の属性	項目 2 の属性													
	j	c	n	m	p	z	x	s	t	g	d	w	u	v
j	○	—	—	—	—	—	—	—	—	—	—	—	—	—
c	—	○	—	—	—	—	—	—	—	—	—	—	—	—
n	—	—	○	—	—	—	—	—	—	—	—	—	—	—
m	—	—	—	○	—	—	—	—	—	—	—	—	—	—
p	—	—	—	—	○	○	○	○	○	—	—	—	—	—
z	—	—	—	—	○	○	○	○	○	—	—	—	—	—
x	—	—	—	—	○	○	○	○	○	—	—	—	—	—
s	—	—	—	—	○	○	○	○	○	—	—	—	—	—
t	—	—	—	—	○	○	○	○	○	—	—	—	—	—
g	—	—	—	—	—	—	—	—	—	○	—	—	—	—
d	—	—	—	—	—	—	—	—	—	—	○	—	—	—
w	—	—	—	—	—	—	—	—	—	—	—	○	—	—
u	—	—	—	—	—	—	—	—	—	—	—	—	○	—
v	—	—	—	—	—	—	—	—	—	—	—	—	—	○

(凡例)

○ : 比較できる組み合わせ。

— : 比較できない組み合わせ。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

### + pos

比較項目の開始バイト位置を 5 桁以内の数字で指定します。

-locatajst パラメタの指定有無によってレコード先頭の解釈が異なります。また、テキストファイルの場合は、入力データのエンコードによって指定範囲が異なります。開始バイト位置 (*pos*) に指定できる値の範囲は次のとおりです。

ファイル編成			-locatajst パラメタ	
			指定あり	指定なし
固定長順編成ファイル			1 ~ 65,535	0 ~ 65,534
可変長順編成ファイル			5 ~ 65,539 <sup>※1</sup>	
テキスト ファイル <sup>※2</sup>	改行文字 「LF」	ASCII SJIS EUC-JP EUC-HJ UTF-8	1 ~ 65,535	
		UCS-2LE UCS-2BE UTF-16LE UTF-16BE	1 ~ 65,533 (2 の倍数 + 1 <sup>※3</sup> )	0 ~ 65,532 (2 の倍数 <sup>※3</sup> )
		UCS-4LE UCS-4BE UTF-32LE UTF-32BE	1 ~ 65,529 (4 の倍数 + 1 <sup>※4</sup> )	0 ~ 65,528 (4 の倍数 <sup>※4</sup> )
	改行文字 「CRLF」	ASCII SJIS EUC-JP EUC-HJ UTF-8	1 ~ 65,534	0 ~ 65,533
		UCS-2LE UCS-2BE UTF-16LE UTF-16BE	1 ~ 65,531 (2 の倍数 + 1 <sup>※3</sup> )	0 ~ 65,530 (2 の倍数 <sup>※3</sup> )
		UCS-4LE UCS-4BE UTF-32LE UTF-32BE	1 ~ 65,525 (4 の倍数 + 1 <sup>※4</sup> )	0 ~ 65,524 (4 の倍数 <sup>※4</sup> )

#### 注※ 1

レコードの前の 4 バイト領域 (RDW : レコードヘッダ) の先頭が 1 となります。

#### 注※ 2

入力レコードの改行文字については、「5.8.7 recsepa オペランド」を参照してください。また、入力データのエンコードについては、「5.8.13 codetype オペランド」を参照してください。

#### 注※ 3

2 の倍数 (-locatajst パラメタを同時に指定している場合は「2 の倍数 + 1」) 以外を指定すると、KBL51306-E メッセージが表示されてエラーになります。

#### 注※ 4

4 の倍数 (-locatajst パラメタを同時に指定している場合は「4 の倍数 + 1」) 以外を指定すると、KBL51306-E メッセージが表示されてエラーになります。

#### – len

比較項目の長さを次の範囲で指定します。

cond-attribute	指定できる長さ (単位)
j, c, w <sup>※1</sup>	<ul style="list-style-type: none"> <li>入力データのエンコードが「ASCII」, 「SJIS」, 「EUC-JP」, 「EUC-HJ」または「UTF-8」の場合 1 ~ 4,096 (バイト)</li> <li>入力データのエンコードが「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」の場合<sup>※2</sup> 2 ~ 8,192 (バイト)</li> <li>入力データのエンコードが「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」の場合<sup>※3</sup> 4 ~ 16,384 (バイト)</li> </ul>
u, v	1 ~ 4,096 (文字数) <sup>※6</sup>
m, n	2, 4, または 8 (バイト)
p	1 ~ 10 (バイト)
z, x	1 ~ 18 (桁) <sup>※4</sup>
s, t	2 ~ 19 (桁) <sup>※4</sup>
g	2 ~ 8 (桁) <sup>※5</sup>
d	1 ~ 256 (桁) <sup>※4</sup>

#### 注※ 1

入力データのエンコードの種別によって、マルチバイト文字の長さは異なります。詳細は「付録 I.2 文字列データの扱い」を参照してください。

#### 注※ 2

入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定した場合、長さには「2 の倍数」を指定してください。

2 の倍数以外を指定すると、KBL51308-E メッセージが表示されてエラーになります。

#### 注※ 3

## 5. exsort コマンドのパラメタおよびオペランドの詳細

入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、  
「UTF-32BE」または「UNICODE（入力ファイルの Unicode シグニチャ  
(BOM) の形式が UTF-32LE または UTF-32BE)」を指定した場合、長さには  
「4 の倍数」を指定してください。

4 の倍数以外を指定すると、KBL51308-E メッセージが表示されてエラーになります。

### 注※ 4

「バイト長」ではなく数値の「桁数」を指定してください。「UCS-2」、  
「UCS-4」、「UTF-16」および「UTF-32」の数値データの扱いについては、「付  
録 I.3 数値データの扱い」を参照してください。

### 注※ 5

「バイト長」ではなく日付の「桁数」を指定してください。「UCS-2」、  
「UCS-4」、「UTF-16」および「UTF-32」の日付データの扱いについて「付録  
I.4 日付データの扱い」を参照してください。

### 注※ 6

「バイト長」ではなく「文字数」を指定してください。UTF-16 のサロゲートペ  
ア文字は 1 文字として扱います。また、Unicode の VS (Variation Selector,  
異体字セレクタ) は文字数に数えません。マルチバイト文字の扱いについては、  
「付録 I.2 文字列データの扱い」を参照してください。

## 比較項目の範囲

比較項目は、次の範囲内で指定してください。

入力ファイル種別	比較項目の範囲
固定長ファイル	レコード長
可変長ファイル	最小レコード長
テキストファイル	最小レコード長

## 比較項目の扱い

比較項目同士の比較で、項目の長さが等しくない場合、短い方の比較項目に埋  
め字を仮定して、長い方の比較項目と同じ長さにします。

文字・日付属性の場合は空白を、数値属性の場合は NULL 値または数字「0」  
を仮定します。

埋め字の位置は、文字・日付属性の場合は下位から、数値属性の場合は上位か  
らとなります。

cond-attribute	埋め字の位置	埋め字データ※
j, c	下位	空白



cond-attribute	埋め字の位置	埋め字データ※
w, u, v	下位	空白
n, m	上位	NULL 値 (0x00)
p	上位	NULL 値 (0x00)
z, x, s, t	上位	数字「0」
g	下位	空白
d	下位	空白

**注※**

入力データのエンコードが「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」である場合は、「UCS-2」の文字コードで埋め字します。

入力データのエンコードが「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」である場合は、「UCS-4」の文字コードで埋め字します。

埋め字の文字コードについては、「表 H-3 ASCII 対応予約文字一覧」を参照してください。

**例 1** (入力データのエンコードは「SJIS」)

## 5. exsort コマンドのパラメタおよびオペランドの詳細

### ・文字属性

0	1	2	3	4	5	6	7	8	9	10	11
a	b	c	d					a	b	d	d

c+0-4 : c+8-2 → abcd : ab△△

右辺の比較項目の下位に、空白を仮定する。

### ・2進数属性

0	1	2	3	4	5	6	7	8	9	10	11
02	1C	A0	FE					15	8B	64	A5

n+0-2 : n+8-4 → 0x0000021C : 0x158B64A5

左辺の比較項目の上位に、NULL値(0x00)を仮定する。

### ・10進数属性

0	1	2	3	4	5	6	7	8	9	10	11
31	32	33	34					12	34	56	7C

z+0-4 : p+8-4 → 0001234 : 1234567

左辺の比較項目の上位に、数字「0」(0x30)を仮定する。

### ・10進数属性

0	1	2	3	4	5	6	7	8	9	10	11
31	32	33	34	35	36			12	3C		

z+0-6 : p+8-2 → 123456 : 000123

右辺の比較項目の上位に、NULL値(0x00)を仮定する。

### ・文字(左分離符号付き数字)属性

0	1	2	3	4	5	6	7	8	9	10	11
0	1	2	4			+	5	6	7		

d+0-3 : d+6-4 → 012△ : +567

左辺の比較項目の下位に、空白を仮定する。

(凡例) \_\_\_\_ : 仮定部分を示す。

△ : 空白を示す。

### 例 2 (入力データのエンコードは「UCS-2LE」)

### ・文字属性

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
42	30	44	30	43	30					42	30	44	30	46	30	48	30
あ	い	う								あ	い	う	え				

w+0-4 : w+10-2 → あい(0x4230 0x4430) : あ△(0x4230 0x2000)

右辺の比較項目の下位に、空白を仮定する。

(凡例) \_\_\_\_ : 仮定部分を示す。

△ : 空白を示す。

### 例 3 (入力データのエンコードは「UCS-4LE」)

## ・文字属性

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
42	30	00	00	44	30	00	00	42	30	00	00	44	30	00	00	46	30	00	00
あ				い				あ				い				う			

w+0-8 : w+12-4 → あい(0x42300000 0x44300000) : い△(0x44300000 0x20000000)

右辺の比較項目の下位に、空白を仮定する。

(凡例) \_\_\_\_ : 仮定部分を示す。

△ : 空白を示す。

## comp

比較演算子を指定します。

comp	意味
=	左辺=右辺
!=	左辺≠右辺
>	左辺>右辺
>=	左辺≧右辺
<	左辺<右辺
<=	左辺≦右辺

## cond-value

比較の条件となる値（条件値）を指定します。

cond-value	意味	入力データのエンコード	条件値の桁		備考
			最小	最大	
C' 文字列 '	文字条件値	—	1	4,096	—
S' 文字列 '※4	文字条件値	—	1	4,096	文字条件値を環境変数で指定する場合に使用します。
N'10 進数 '※4	10 進数条件値※1 (n, m, p, z, x, s, t)	—	1	19	符号なしの場合は 18 桁です。比較項目が符号なし 2 進数 (m) の場合は、符号は付けてはなりません。
	10 進数条件値※1 (d)	—	1	256	符号を指定する場合、符号を含めて 256 桁までです。

5. exsort コマンドのパラメタおよびオペランドの詳細

cond-value	意味	入力データのエンコード	条件値の桁		備考
			最小	最大	
X'16 進数'※ <sup>4</sup>	16 進数条件値※ <sup>2</sup>	ASCII SJIS EUC-JP EUC-HJ※ <sup>5</sup> UTF-8	2	8,192	指定できるのは偶数桁のデータだけです。 2 桁で 1 バイトを表します。
		UCS-2LE UCS-2BE UTF-16LE UTF-16BE	4	16,384	
		UCS-4LE UCS-4BE UTF-32LE UTF-32BE	8	32,768	
D' 日付'※ <sup>4</sup>	日付条件値※ <sup>3</sup>	—	2	8	—

(凡例)

— : なし

注※ 1

10 進数条件値として指定できるのは「0～9」、「+」および「-」だけです。符号は先頭に付けます。ただし、比較項目が符号なし 2 進数 (m) の場合は、符号を付けてはなりません。

2 進数属性に対する条件値として指定できる範囲を、表 5-13 と表 5-14 に示します。

表 5-13 固定小数点 2 進数 (n) に対する 10 進数条件値の指定範囲

長さ	最小値	最大値
2	-32,768	32,767
4	-2,147,483,648	2,147,483,647
8	-999,999,999,999,999,999	999,999,999,999,999,999

表 5-14 符号なし 2 進数 (m) に対する 10 進数条件値の指定範囲

長さ	最小値	最大値
2	0	65,535
4	0	4,294,967,295
8	0	999,999,999,999,999,999

注※ 2

16 進数条件値として指定できるのは「0～9」および「A～F」だけです。

注※ 3

先頭の 1 文字目に指定できるのは空白または「0～9」の文字列です。

2文字目に指定できるのは「0～9」の文字列です。

1文字の空白を指定した場合、数字の「00」を設定します。

1文字の「0～9」を指定した場合、先頭の1文字目には数字の「0」を設定します。

注※4

SORT EE 08-50以降、これらの cond-value の指定値は、環境変数を使って exsort パラメタファイルに記述できます。環境変数名で指定する場合は「%環境変数名%」と%で囲んで記述します。詳細は「4.4.5 環境変数」を参照してください。

注※5

入力データのエンコードが「EUC-HJ」の場合、マルチバイト文字の条件値は「X'16進数」形式で指定してください。

### 条件値の属性

指定できる条件値の属性は、比較項目の属性によって異なります。

比較項目の形式	条件値			
	文字	10進	日付	16進
j	○	—	—	○
c	○	—	—	○
n	—	○	—	○
m	—	○	—	○
p	—	○	—	○
z	—	○	—	○
x	—	○	—	○
s	—	○	—	○
t	—	○	—	○
g	—	—	○	—
d	—	○	—	○
w	○	—	—	○
u	○	—	—	○
v	○	—	—	○

(凡例)

○：指定できます。

—：指定できません。

### ワイド文字およびマルチバイト文字に対する条件値の指定のしかた

属性コード「w」、「u」および「v」に対する条件値の指定のしかたについて説明します。

- 入力データのエンコードのエンディアンを合わせて指定してください。例えば、全角ひらがな「あ (U+3042) い (U+3044)」を「UCS-2LE」または「UTF-16LE」で指定する場合は、「X'42304430'」、「UCS-2BE」または

## 5. exsort コマンドのパラメタおよびオペランドの詳細

「UTF-16BE」で指定する場合は「X'30423044'」と指定します。

- Unicode の入力データに対して属性コード「u」を指定した場合、UCS コードだけを比較します。Unicode の IVS (Ideographic Variation Sequence) の異体字は区別しません。例えば、「葛 (U+845B;U+E0102)」の異体字を文字条件値に指定した場合、VS (Variation Selector, 異体字セレクタ) の「U+E0102」は無視し、UCS コードの「U+845B」で比較します。「include=u+0-1 = C' 葛 '」の指定では、「葛 (U+845B;VS なし)」、「葛 (U+845B;U+E0102)」、「葛 (U+845B;U+E0103)」はすべて同じ文字として選択されます。
- Unicode の入力データに対して属性コード「v」を指定した場合、Unicode の IVS (Ideographic Variation Sequence) の異体字を区別します。例えば、「葛 (U+845B;U+E0102)」の異体字を文字条件値に指定した場合、UCS コードと VS の両方で比較します。「include=v+0-1 = C' 葛 '」の指定では、「葛 (U+845B;U+E0102)」を選択し、「葛 (U+845B;VS なし)」と「葛 (U+845B;U+E0103)」は選択されません。
- エディタなどの制約によって、Unicode のサロゲートペア文字、異体字、または Unicode 以外の第三水準・第四水準・外字などの文字を文字条件値「C' 文字列 '」に直接記述できない場合は、16 進数条件値「X'16 進数 '」で文字コードを指定してください。
- Unicode のサロゲートペア文字、および IVS についての詳細は、「付録 I.2(6) IVS 対応」を参照してください。

### 条件値の扱い

条件値の長さが指定した比較項目より短い場合、不足分の埋め字を仮定し、比較項目と同じ長さの条件値と見なして比較します。

文字・日付条件値の場合は埋め字として空白が仮定されます。また、10 進・16 進数条件値の場合は埋め字として数字「0」が仮定されます。

条件値の長さが指定した比較項目より長い場合、桁あふれ部分を切り捨て、比較項目と同じ長さの条件値と見なして比較項目と同じ長さにします。

10 進数条件値の埋め字・切り捨て位置は上位からとなります。10 進数条件値以外は、下位からとなります。

条件値	埋め字・切り捨て位置	埋め字データ
文字条件値	下位	空白
10 進数条件値	上位	数字「0」
16 進数条件値	下位	数字「00」
日付条件値	下位	空白

## 例 1 比較項目の長さ &gt; 条件値の長さ

条件値の種類	選択条件式の指定		桁そろえ後の条件値	解説
	比較項目	条件値		
文字条件値	c+0-4	C'ab'	C'ab_△△'	入力データのエンコードに「SJIS」を指定した場合 2文字分の空白 (0x20 0x20) を下位に仮定した条件値と見なします。
	w+0-8	C'ab'	C'ab_△△△'	入力データのエンコードに「UCS-2LE」を指定した場合 2文字分の空白 (0x2000 0x2000) を下位に仮定した条件値と見なします。
10進数条件値	z+0-4	N'12'	N'0012'	2桁分の数字「00」を上位に仮定した条件値と見なします。
16進数条件値	j+0-4	X'3132'	X'31320000'	入力データのエンコードに「SJIS」を指定した場合 2桁分の数字「0000」を下位に仮定した条件値と見なします。
	w+0-8	X'31003200'	X'3100320000000000'	入力データのエンコードに「UCS-2LE」を指定した場合 2桁分の数字「00000000」を下位に仮定した条件値と見なします。
日付条件値	g+0-4	D'12'	D'12_△△△'	入力データのエンコードに「UCS-2LE」を指定した場合 2桁分の空白 (0x2000 0x2000) を下位に仮定した条件値と見なします。

(凡例)

\_ : 埋め字を仮定する部分です。

△ : 空白を示します。

## 例 2 比較項目の長さ &lt; 条件値の長さ

## 5. exsort コマンドのパラメタおよびオペランドの詳細

条件値の種類	選択条件式の指定		桁そろえ後の条件値	解説
	比較項目	条件値		
文字条件値	c+0-4	C'abcdef'	C'abcd'	入力データのエンコードに「SJIS」を指定した場合 下位の桁あふれ部分を切り捨てます。
	w+0-16	C'abcdef'	C'abcd'	入力データのエンコードに「UCS-4LE」を指定した場合 下位の桁あふれ部分を切り捨てます。
10進数条件値	z+0-4	N' <u>123456</u> '	N'3456'	上位の桁あふれ部分を切り捨てます。
16進数条件値	j+0-2	X' <u>313233</u> '	X'3132'	入力データのエンコードに「SJIS」を指定した場合 下位の桁あふれ部分を切り捨てます。
	w+0-8	X'31000000320000 <u>00</u> <u>33000000</u> '	X'3100000032000000'	入力データのエンコードに「UCS-4LE」を指定した場合 下位の桁あふれ部分を切り捨てます。
日付条件値	g+0-5	D' <u>07/10/03</u> '	D'07/10'	入力データのエンコードに「UCS-4LE」を指定した場合 下位の桁あふれ部分を切り捨てます。

(凡例)

\_\_ : 切り捨て部分です。

### { AND | OR }

複数の条件式を指定する場合、論理演算子「AND」（論理積）または「OR」（論理和）を指定して条件式を結合します。「AND」と「OR」が混在した場合は、「AND」を判定後、「OR」を判定します。

また、条件式を括弧で囲むことによって、論理演算の順序を変えることができます。括弧の入れ子は 24 レベルまで指定できます。条件式と括弧の間に空白またはタブを置くことができます。論理演算子（AND または OR）と括弧の間には、空白またはタブが必要です。

指定例は exsort パラメタファイルに指定した場合です。exsort コマンドに指定する場合は、選択条件式全体をダブルクォーテーションで囲んでください。

### 例 1

次のうちどちらかの条件を満たしたレコードが選択されます。

- 「z+10-2」は 30 以上かつ 「z+20-4」は 1000
- 「z+10-2」は 10 以下かつ 「z+20-4」は 2000



```
include=z+10-2 >= N'30' AND z+20-4 = N'1000'
OR
z+10-2 <= N'10' AND z+20-4 = N'2000'
```

## 例 2

次の条件をすべて満たしたレコードが選択されます。

- 「z+10-2」は 30 以上または「z+10-2」は 10 以下
- 「z+20-4」は 1000 または「z+20-4」は 2000

### 括弧指定あり

1 回のコマンドで選択することができます。

```
include=(z+10-2 >= N'30' OR z+10-2 <= N'10')
AND
(z+20-4 = N'1000' OR z+20-4 = N'2000')
```

### 括弧指定なし

括弧を指定しない場合は 2 回のコマンドに分けて選択します。

(1 回目)

```
include=z+10-2 >= N'30' OR z+10-2 <= N'10'
```

(2 回目)

```
include=z+20-4 = N'1000' OR z+20-4 = N'2000'
```

## 5.8.17 {csvinclude | csvomit} オペランド

### 意味

CSV ファイルの入力レコードの選択条件式を指定します。

このオペランドは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、KBLS1322-E エラーとなります。

csvinclude オペランド :

条件を満たすレコードを選択します。

csvomit オペランド :

条件を満たさないレコードを選択します。

### 形式

```
{csvinclude | csvomit} ="csv-condition-expression [△ {AND | OR}
△csv-condition-expression ... ]"
```

レコードを選択するための選択条件式を指定します。

複数の条件式 (*csv-condition-expression*) を丸括弧「()」で囲むことによって、条件式の優先順位を変更できます。

- このオペランドは、CSV ファイルを指定した場合だけ有効です。
- このオペランドは、{include | omit} オペランドと同時に指定できません。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- コマンドに指定する場合は、選択条件式全体をダブルクォーテーションで囲みます。
- exsort パラメタファイルに指定する場合は、選択条件式をダブルクォーテーションで囲まないでください。
- 指定できる選択条件式の個数と比較項目の合計長を、環境ファイル「EXS\_CSVSELNUMMAX」で指定できます。なお、比較項目の合計長は、入力データのエンコードによって異なります。環境ファイル「EXS\_CSVSELNUMMAX」については「15.1.1(19) EXS\_CSVSELNUMMAX」を参照してください。
- 使用できる括弧は、ネストレベル 0 ~ 24 です。
- 指定したセルに対応するデータが入力されていない場合、ナル文字列として処理します。ナル文字列の扱いについては、「付録 D CSV ファイル内のナル文字列の扱い (SORT EE の機能)」を参照してください。
- 入力セルデータの桁調整については、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。

### 指定値

csv-condition-expression の形式を次に示します。

#### csv-condition-expression オペランド形式 1

```
cond-attribute+cellpos- {len | p[, s] | d, dateform |
datedigit[, timedigit]} ΔcompΔcond-value
```

比較項目 (セル) と条件値を比較する条件式を定義します。

#### csv-condition-expression オペランド形式 2

```
cond-attribute+cellpos- {len | p[, s] | d, dateform |
datedigit[, timedigit]} ΔcompΔcond-attribute+cellpos-
{len | p[, s] | d, dateform | datedigit[, timedigit]}
```

比較項目 (セル) と比較項目 (セル) を比較する条件式を定義します。

csv-condition-expression に指定するそれぞれの値について次に示します。

#### cond-attribute

比較項目 (セル) のデータ形式を指定します。

cond-attribute の指定値	意 味
j	比較項目が文字 (JIS8 単位コード※1) であることを指定します。
c	比較項目が文字であることを指定します。※2
d	比較項目が文字 (左分離符号付き数字) であることを指定します。※4
z	比較項目が固定小数点数であることを指定します。※4
g	比較項目が日付 (英数字) であることを指定します。※4
h	比較項目が日付時刻 (数字) であることを指定します。※6
w	比較項目が文字 (ワイド文字※3, ※5) であることを指定します。

<i>cond-attribute</i> の指定値	意 味
u	比較項目が文字 (マルチバイト文字※5, ※7, ※8) であることを指定します。
v	比較項目が文字 (マルチバイト文字※5, ※7, ※8) であることを指定します (Unicode の IVS に対応した属性です)。

## 注※ 1

ASCII コード (0x00 ~ 0x7F) を含む, JIS8 単位コード (0x00 ~ 0xFF) を意味します。

入力データに日本語などのマルチバイト文字が含まれている場合, マルチバイト文字ではなく, シングルバイト文字として扱われます。

EBCDIK コードの体系に従って比較項目を比較する EBCDIK コード比較オプション機能を使用できます。EBCDIK コード比較オプション機能については, 「5.5.3 EBCDIK コード比較オプション (EBCJ, EBCE オプション)」を参照してください。

## 注※ 2

入力データのエンコードの種別には対応しません。入力データの文字を構成するバイト長を意識しないで, 常に 1 バイト長のシングルバイト文字として扱います。ASCII コードや JIS8 単位コード向けの属性です。

## 注※ 3

入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定した場合, 1 文字 2 バイトのワイド文字として扱います。また, サロゲートペア文字は 2 文字として扱います。

入力データのエンコードに「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定した場合, 1 文字 4 バイトのワイド文字として扱います。

文字のエンコードの指定方法については, 「5.8.13 codetype オペランド」を参照してください。

## 注※ 4

入力データのエンコードで解釈します。

「ASCII」, 「SJIS」, 「EUC-JP」, 「EUC-HJ」または「UTF-8」を指定した場合は 1 バイト, 「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」を指定した場合は 2 バイト, 「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」を指定した場合は 4 バイトの長さで文字を扱います。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。

### 注※ 5

比較項目範囲の入力データがすべて ASCII 対応コード（英数字）である場合は、属性コード「c」を指定した方が処理効率が向上します。属性コード「c」は 1 文字 1 バイトで扱いますので、エンコードに対応したバイト長を指定してください。

ASCII 対応コードについては、「付録 H Unicode 操作文字コード一覧 (SORT EE)」を参照してください。

1 文字の長さについては、「付録 I.2(1) 文字の長さ」を参照してください。

文字属性コードの特徴については、「付録 I.2(2) 文字属性コードの特徴」を参照してください。

### 注※ 6

比較項目範囲の入力データが次の条件すべてに該当する場合は、属性コード「c」を指定した方が処理効率は向上します。

- 桁数がそろっている。
- 時刻が 24 時間制で表記されている。
- 日付、時刻またがりのデータがない。

属性コード「c」は 1 文字 1 バイトで扱いますので、エンコードに対応したバイト長を指定してください。

1 文字の長さについては、「付録 I.2(1) 文字の長さ」を参照してください。

文字属性コードの特徴については、「付録 I.2(2) 文字属性コードの特徴」を参照してください。

### 注※ 7

属性コード「u」と「v」は、入力データのエンコードで解釈します。1 文字の長さは可変長として扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。また、マルチバイト文字の扱いについては、「付録 I.2 文字列データの扱い」を参照してください。

### 注※ 8

入力データのエンコードに「UTF-8」、「UTF-16LE」、「UTF-16BE」、「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-16LE、UTF-16BE、UTF-32LE または UTF-32BE）」のどれかを指定した場合、Unicode の IVS（Ideographic Variation Sequence）を次のとおり解釈します。Unicode の IVS については、「付録 I.2(6) IVS 対応」を参照してください。

- 属性コード「u」は、UCS コードが一致する文字であれば、Unicode の IVS

の異体字であっても同じ文字として扱います。

- 属性コード「v」は、UCS コードが一致する文字であっても、VS (Variation Selector, 異体字セレクタ) が異なる場合は、異なる文字として扱います。

#### + cellpos

レコード内の比較対象セル (列) の位置を、番号で指定します。

レコードの先頭セル番号を「1」、最大セル番号を「n」とし、1～n までの番号を指定します。最大セル番号は、16,384 番です。

#### – len

cond-attribute に「j」、「c」、「w」、「u」、「v」または「d」を指定した場合に、文字データの最大バイト長を 10 進数で指定します。

- 属性「d」は最大桁数を指定します。
- 比較項目の長さを次の範囲で指定します。

cond-attribute	指定できる長さ (単位)
j, c, w ※1	<ul style="list-style-type: none"> <li>• 入力データのエンコードが「ASCII」、「SJIS」、「EUC-JP」、「EUC-HJ」または「UTF-8」の場合 1～4,096 (バイト)</li> <li>• 入力データのエンコードが「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」の場合※2 2～8,192 (バイト)</li> <li>• 入力データのエンコードが「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」の場合※3 4～16,384 (バイト)</li> </ul>
u, v	1～4,096 (文字数) ※5
d	1～256 (桁) ※4

#### 注※1

入力データのエンコードの種別によって、マルチバイト文字の長さは異なります。詳細は「付録 I.2 文字列データの扱い」を参照してください。

#### 注※2

入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定した場合、長さには「2 の倍数」を指定してください。

2 の倍数以外を指定すると、KBLS1308-E メッセージが表示されてエラーになります。

#### 注※3

入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、

## 5. exsort コマンドのパラメタおよびオペランドの詳細

「UTF-32BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-32LE または UTF-32BE）」を指定した場合、長さには「4 の倍数」を指定してください。

4 の倍数以外を指定すると、KBL51308-E メッセージが表示されてエラーになります。

### 注※ 4

「バイト長」ではなく数値の「桁数」を指定してください。「UCS-2」、  
「UCS-4」、**「UTF-16」** および **「UTF-32」** の数値データの扱いについては、  
「付録 I.3 数値データの扱い」を参照してください。

### 注※ 5

「バイト長」ではなく「文字数」を指定してください。UTF-16 のサロゲート  
ペア文字は 1 文字として扱います。また、Unicode の VS (Variation  
Selector, 異体字セレクタ) は文字数に数えません。マルチバイト文字の扱い  
については、「付録 I.2 文字列データの扱い」を参照してください。

- 実際に入力したデータ長に過不足がある場合は調整を行います。詳細は、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- 入力データを引用符で囲んでいる場合は、引用符を除外した内部データ形式で比較項目を比較します。なお、セルデータが変更されることはありません。
- 入力データの先頭が引用符の場合だけ、対の引用符で囲んだ範囲がエスケープされます。詳細は、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- 入力データの先頭および末尾の引用符は、有効バイト長に含まれません。また、エスケープされた引用符は、2 文字で 1 文字となります。
- マルチバイト文字（日本語）の途中で最大長を超えた場合、超えた部分のマルチバイト文字以降を切り捨て、代わりに 1 バイトの空白で埋め字をします。なお、切り捨ては比較項目属性「c」、「u」および「v」の場合だけです。比較項目属性「j」は日本語として扱われません。詳細は、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。

### — p[. s]

cond-attribute に「z」を指定した場合に、数値データの有効桁を 10 進数で指定します。

- p (precision : 精度) は、数値データの総桁数を表します。p から s を引いた桁が整数部の有効桁となります。
- p の指定範囲は、1 ～ 38 桁です。
- s (scale : 位) は、数値データの小数部（小数点以下）の有効桁を表します。
- s の指定範囲は、0 ～ p 桁です ( $0 \leq s \leq p$ )。p = s を指定した場合は、整数部の有効桁は 0（整数部なし）となります。

- `s` を省略した場合は、小数部の有効桁は 0 (小数部なし) を仮定します。データ中に小数点を含まない場合は、`s` の指定は必要ありません。
- データ中には、0 ~ 9 の数字以外に、データを囲んでいる引用符、符号、通貨記号、および桁修飾文字 (コンマ、小数点、空白) を含むことができ、これらの文字を除いた内部データ形式で比較項目を比較します。また、それぞれの除外文字は位置規則があります。違反するデータを検出した場合は、データ形式が不正と見なされ「KBL51320-E 比較項目にエラーデータを検出しました。」というメッセージが表示されて、エラーになります。メッセージが表示されたときの原因と対処方法の詳細は、「付録 K メッセージ」を参照してください。
- データ形式が不正な場合でもエラーにしたくないときは、コマンドオプション「`-cmdopt KEYNOCHK`」を指定してください。
- 除外文字の位置規則については、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- データ中の符号、通貨記号、桁修飾文字 (コンマ、小数点、空白)、およびデータを囲んでいる引用符は、有効桁に含まれません。有効桁の対象は、0 ~ 9 の数字だけです。
- データ中に小数点がある場合は、小数点位置を合わせて入力します。データ中に小数点がない場合は、データの末尾に小数点を仮定して入力します。詳細は、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- 実際に入力したデータの桁に過不足がある場合は桁調整を行います。詳細は、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- すべて空白のデータは、0 (ゼロ) として入力します。
- 環境ファイル「`EXS_ROUND`」または `-inpfile`、`-catinp`、`-splitinp` パラメタの `round` オペランドで指定した丸め方式 (四捨五入、切り捨て、切り上げのどれか) に従って、整数部または小数部の丸め処理がされます。`EXS_ROUND` および `round` オペランドの両方の指定を省略している場合、切り捨ての処理がされます。詳細は、「5.8.11 `round` オペランド」を参照してください。
- 引用符、符号、通貨記号、桁修飾文字の除外、および丸め処理によって、入力セルデータが変更されることはありません。

#### — d, *dateform*

`cond-attribute` に「g」を指定した場合に、日付データの有効桁と書式番号を指定します。

- `d` (`digit` : 桁) は、日付データの有効桁を表します。
- `d` の指定範囲は、2 ~ 8 桁です。なお、書式 3 ~ 4 の 2 桁年号に対して 7 桁以上を指定した場合、6 桁を仮定します。
- 指定可能な範囲より大きな桁を指定した場合、指定可能な最大桁を仮定します。例えば、書式 3 の日付データに対して 7 桁以上を指定した場合、6 桁を

## 5. exsort コマンドのパラメタおよびオペランドの詳細

仮定します。

- データの左端から指定された書式に適用します。有効桁以降のデータは無視します。
- 実際に入力したデータの桁に過不足がある場合は桁調整を行います。詳細は、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- データ中には、データを囲んでいる引用符と年月日を区切る桁修飾文字を含むことができ、これらの文字を除いた内部データ形式で比較項目を比較します。日付データの形式については、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- データの先頭と末尾を囲んでいる引用符、およびデータ中の桁修飾文字は、有効桁に含まれません。
- 入力データの年号部分が数字の 0～9 以外を指定した場合、データ形式が不正と見なされ「KBL51320-E 比較項目にエラーデータを検出しました。」というメッセージが表示されてエラーになります。メッセージが表示されたときの原因と対処方法の詳細は、「付録 K メッセージ」を参照してください。
- データ形式が不正な場合でもエラーにしたいときは、コマンドオプション「-cmdopt DATENOCHK」を指定してください。なお、年号以外のデータ(月、日)は、形式チェックしません。
- 引用符、および桁修飾文字の除外によって、入力セルデータが変更されることはありません。
- 桁修飾文字は、「ハイフン (-)」と「スラント (/)」の 2 種類です。
- 環境変数「EXSORTGOPTION (64 ビット版は「EXSORT64GOPTION)」」を指定すると、「年月日 {T | 空白} 時分秒」形式の入力データを扱えます。また、桁修飾文字は、「ハイフン」、「スラント」および「ピリオド (.)」の 3 種類に変更されます。環境変数については、「14.1.1(29) EXSORTGOPTION」または「14.1.2(29) EXSORT64GOPTION」を参照してください。
- dateform は日付データの書式を表します。dateform の指定値を次に示します。

dateform の指定値	データ形式	書式	d (桁)
1	前置 4 桁年号	YYYY/MM/DD YYYY-MM-DD YYYY.MM.DD ※ YYYYMMDD YYYY	2～8
2	後置 4 桁年号	MM/DD/YYYY MM-DD-YYYY MM.DD.YYYY ※ MMDDYYYY	2～8



dateform の指定値	データ形式	書式	d (桁)
3	前置 2 桁年号	YY/MM/DD YY-MM-DD YY.MM.DD ※ YYMMDD YY	2 ~ 6
4	後置 2 桁年号	MM/DD/YY MM-DD-YY MM.DD.YY ※ MMDDYY	2 ~ 6

(凡例)

YYYY : 西暦 4 桁年号

YY : 西暦下 2 桁年号

MM : 月

DD : 日

桁修飾文字で年月日を区切っている場合は、MM および DD 部分のデータは 1 桁でもかまいません。

(例)

「YYYY-MM-DD」の場合 : 2008-7-7

「MM/DD/YY」の場合 : 1/1/07

注※

環境変数「EXSORTGOPTION (64 ビット版は「EXSORT64GOPTION)」」の指定がある場合、ピリオドを桁修飾文字として扱います。

#### — datedigit[, timedigit]

cond-attribute に「h」を指定した場合に、日付および時刻データの有効桁を 10 進数で指定します。

- datedigit は、日付データの有効桁を表します。
- timedigit は、時刻データの有効桁を表します。
- datedigit に指定できる桁は 0, 4, 6 および 8 桁、timedigit に指定できる桁は 0, 2, 4, 6 および 9 桁のどれかです。それ以外を指定した場合、KBLS1326-E メッセージが表示されてエラーになります。また、datedigit と timedigit 両方に 0 を指定した場合もエラーになります。
- 入力データを項目の区切り文字で区切り、それぞれのデータの左端から指定された桁数の書式に適用します。
- 入力したデータの桁に過不足がある場合は、項目によって桁調整が実施されるか、またはエラーとなります。端数処理の詳細については、「付録 J 日付時刻データの形式 (SORT EE の機能)」を参照してください。
- データ中には、データの先頭と末尾を囲んでいる引用符、項目区切り文字

## 5. exsort コマンドのパラメタおよびオペランドの詳細

(**T** または空白), および桁修飾文字 (スラント, ハイフン, コロン, およびピリオド) を含むことができ, これらの文字を除いた内部データ形式で比較項目を比較します。日付時刻データの形式については, 「3.2 キー属性と形式」を参照してください。

- データの先頭と末尾を囲んでいる引用符, 項目区切り文字 (**T** または空白), 桁修飾文字 (スラント, ハイフン, コロン, およびピリオド), および 12 時間制表記は, 有効桁に含まれません。
- 引用符, 項目の区切り文字, および桁修飾文字を取り除くことで, 入力セルデータが変更されることはありません。
- 入力データが数字の 0 ~ 9 以外を指定した場合, データ形式が不正と見なされ **KBLS1320-E** メッセージが表示されてエラーになります。また, 年の桁数が 4 桁未満, 入力データが指定可能な範囲を超えた場合もエラーとなります。メッセージが表示されたときの原因と対処方法の詳細は, 「付録 **K** メッセージ」を参照してください。データ形式が不正な場合でもエラーにしたいくないときは, コマンドオプション「**-cmdopt DATETIMENOCK**」を指定してください。
- 指定した桁数に応じた日付時刻キーの書式を次に示します。

項目	桁数	書式
日付	4	YYYY
	6	YYYYMM YYYY-MM YYYY/MM YYYY.MM
	8	YYYYMMDD YYYY-MM-DD YYYY/MM/DD YYYY.MM.DD
時刻	2	hh
	4	hhmm hh:mm hh.mm
	6	hhmmss hh:mm:ss hh:mm.ss hh.mm.ss
	9	hhmmssfff hh:mm:ss:fff hh:mm:ss.fff hh.mm.ss.fff

項目	桁数	書式
12 時間制表記	2	AM am PM pm
	4	A.M. a.m. P.M. p.m.

書式の説明を次に示します。

	書式	説明
日付	YYYY	西暦年 4 桁
	MM	月※
	DD	日※
時刻	hh	時※
	mm	分※
	ss	秒※
	fff	秒の小数部 (ミリ秒) ※
12 時間制表記	AM am A.M. a.m. PM pm P.M. p.m.	時 (hh) を 12 時間制で表記した場合に指定
項目区切り文字 桁修飾文字	/	日付の桁修飾文字
	-	日付の桁修飾文字
	:	時刻の桁修飾文字
	.	日付, 時刻の桁修飾文字
	空白	日付項目と時刻項目の区切り文字 時刻項目と 12 時間制表記の区切り文字
	T	日付項目と時刻項目の区切り文字

**注※**

桁修飾文字 (スラント, ハイフン, コロン, およびピリオド) で年月日時分秒を区切っている場合は, 月日時分秒の各データは 1 桁でもかまいません。

**`comp`**

比較演算子を指定します。`comp` の指定値を次に示します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

comp の指定値	意味
=	左辺=右辺
!=	左辺≠右辺
>	左辺>右辺
>=	左辺≥右辺
<	左辺<右辺
<=	左辺≤右辺

### cond-value

比較の条件となる値（条件値）を指定します。cond-value の指定値を次に示します。

cond-value の指定値	比較項目のデータ形式	入力データのエンコード	条件値の桁		備考
			最小	最大	
C' 文字列 '	j (文字) c (文字) w (ワイド文字) u (マルチバイト文字) v (マルチバイト文字)	—	1	4,096	—
S' 文字列 ' ※8, ※9	j (文字) c (文字) w (ワイド文字) u (マルチバイト文字) v (マルチバイト文字)	—	1	4,096	文字条件値を環境変数で指定する場合に使用します。
N' 固定小数点数 ' ※1, ※8	z (固定小数点)	—	1	38	小数点と符号は、桁に含みません。
N'10 進数 ' ※2, ※8	d (左分離符号付き数字)	—	1	256	符号を指定する場合、符号を含めて 256 桁までです。
D' 日付 ' ※3, ※8	g (日付)	—	2	8	指定できるのは数字だけです。桁修飾文字（スラント、ハイフンおよびピリオド）は指定できません。
D' マスク日付 ' ※6, ※7, ※8, ※9	g (日付)	—	2	8	指定できるのは数字とマスク文字（アスタリスク）だけです。桁修飾文字（スラント、ハイフンおよびピリオド）は指定できません。桁数は比較項目の桁数と一致させてください。

cond-value の指定値	比較項目のデータ 形式	入力 データ のエン コード	条件値の桁		備考
			最小	最大	
H' 日付 T 時刻 ※5, ※8	h (日付時刻)	—	日付 : 0 時刻 : 0	日付 : 8 時刻 : 9	指定できるのは数字だけです。 時刻の前には項目区切り文字の 「T」を指定します。 桁数は比較項目の桁数と一致させ てください。
H' マスク日 付 T マスク 時刻※5, ※ 8, ※9	h (日付時刻)	—	日付 : 0 時刻 : 0	日付 : 8 時刻 : 9	指定できるのは数字とマスク文字 (アスタリスク) だけです。時刻の 前には項目区切り文字の「T」を 指定します。桁数は比較項目の桁 数と一致させてください。
X'16 進数' ※4, ※8	j (文字) c (文字) w (ワイド文字) u (マルチバイト 文字) v (マルチバイト 文字)	ASCII SJIS EUC-J P EUC- HJ※ 10 UTF-8	2	8,192	指定できるのは偶数桁のデータだ けです。 2 桁で 1 バイトを表します。
		UCS-2 LE UCS-2 BE UTF-1 6LE UTF-1 6BE	4	16,384	
		UCS-4 LE UCS-4 BE UTF-3 2LE UTF-3 2BE	8	32,768	
NULL	すべてのデータ形 式	—	—	—	ナル文字列 (空のセル) を意味し ます。

(凡例)

— : なし

注※1

固定小数点数 (z) の条件値として指定できるのは、数字 (0 ~ 9)、小数点 (.), および符号 (+, -) だけです。符号を付ける場合は先頭に付けます。小数点以下と整数部の桁を合わせて 38 桁まで指定できます。

注※2

左分離符号付き数字 (d) の条件値として指定できるのは、数字 (0 ~ 9) および符号 (+, -) だけです。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

符号を付ける場合は先頭に付けます。符号を含めて 256 桁まで指定できます。

### 注※ 3

日付 (g) の条件値として指定できるのは、数字 (0 ~ 9) だけです。桁修飾文字 (スラント、ハイフンおよびピリオド) は指定できません。

比較項目の書式に合わせて条件値を指定してください。次に「2012年2月24日」の指定例を示します。

【例 1】前置 4 桁年号 (書式=1)

```
csvinclude=g+1-8,1 = D'20120224'
```

【例 2】後置 4 桁年号 (書式=2)

```
csvinclude=g+1-8,2 = D'02242012'
```

【例 3】前置 2 桁年号 (書式=3)

```
csvinclude=g+1-6,3 = D'120224'
```

【例 4】後置 2 桁年号 (書式=4)

```
csvinclude=g+1-6,4 = D'022412'
```

なお、指定した条件値が日付として正しくない場合でも、エラーにならないで正常に扱われます。

例えば、前置 4 桁年号の比較項目に対して、後置 4 桁年号の条件値「D'02242012」を指定した場合、条件値は「0224年20月12日」と解釈され、正しくレコードを選択できなくなるので注意してください。

### 注※ 4

16 進数の条件値として指定できるのは、数字 (0 ~ 9) および英字 (A ~ F) だけです。

### 注※ 5

- 日付時刻 (h) の条件値として指定できるのは、数字 (0 ~ 9) だけです。マスク日付時刻 (h) の条件値として指定できるのは、数字 (0 ~ 9) とマスク文字 (アスタリスク) だけです。桁修飾文字 (スラント、ハイフン、コロンおよびピリオド) は指定できません。
- 日付に指定できる桁は 0, 4, 6 または 8 桁、時刻に指定できる桁は 0, 2, 4, 6 または 9 桁のどれかです。
- 日付だけ、または時刻だけを指定できます。時刻の先頭には必ず「T」を指定してください。
- 日付時刻条件値の桁数は、比較項目の `datedigit` および `timedigit` と同じ桁を指定します。なお、区切り文字「T」は桁に含みません。
- 次に示す条件値を指定した場合、KBLS1314-E メッセージが表示されてエラーとなります。
  - 日付時刻条件値に、数字 (0 ~ 9) および区切り文字の「T」以外を指定した場合。
  - マスク日付時刻条件値に、数字 (0 ~ 9)、マスク文字 (アスタリスク) および区切り文字の「T」以外を指定した場合。
  - 日付時刻条件値またはマスク日付時刻条件値の桁数が、比較項目の `datedigit` および `timedigit` で指定した桁数と異なる場合。
  - 日付時刻条件値またはマスク日付時刻条件値に、0 月または 0 日を指定した場合。
- 日付条件値の指定例を示します。

【例 1】日付「2013年7月25日」を選択する。

```
csvinclude=h+1-8,0 = H'20130725'
```

【例 2】時刻「20時16分53秒498」を選択する。

```
csvinclude=h+1-0,9 = H'T201653498'
```

【例 3】日時「2013年7月25日20時16分53秒498」以降を選択する。

```
csvinclude=h+1-8,9 >= H'20130725T201653498'
```

【例 4】日付「2013年7月」、時刻「20時16分」を選択する。

csvinclude=h+1-6,4 = H'201307T2016'

- マスク日付時刻条件値の指定例を示します。

【例 1】日付「24 日」を選択する。

csvinclude=h+1-8,0 = H'\*\*\*\*\*24'

【例 2】時刻「0 分」を選択する。

csvinclude=h+1-0,4 = H'T\*\*00'

【例 3】日付「31 日」の時刻「59 分」を選択する。

csvinclude=h+1-8,4 = H'\*\*\*\*\*31T\*\*59'

- 日付時刻条件値の指定可能な範囲は、使用する SORT EE のコマンド、およびコマンド実行システムによって異なります。指定可能な範囲の詳細については、「付録 J 日付時刻データの形式 (SORT EE の機能)」を参照してください。日付時刻条件値に範囲外の値を指定した場合は、KBL1314-E メッセージが表示されてエラーになります。
- 日付時刻として正しくない条件値の場合も、扱える値の範囲内 (0 月および 0 日を除く) であればエラーにしないで、日付、時刻またがりとして扱われます。例えば「H'20139999T999999」を指定した場合、条件値は「2021 年 06 月 11 日 04 時 40 分 39 秒」と解釈されます。
- 日付時刻条件値は実行システムのローカルタイムとして扱われます。
- 指定したマスク日付時刻条件値の、日付、時刻のまたがり調整されません。入力した日付時刻データは、日付、時刻またがり調整されます。このため、マスク日付条件値に存在しない日時を指定した場合、意図したレコードが選択されないおそれがあるので注意してください。例えば、マスク日付時刻条件値に時刻「60 分」と指定した場合、入力データ「20 時 60 分 01 秒」は「21 時 00 分 01 秒」と扱われ選択されません。入力データ「20 時 60 分 01 秒」を選択する場合は、マスク日付時刻条件値の時刻は「0 分」と指定してください。

#### 注※ 6

日付 (g) の条件値として指定できるのは、数字 (0 ~ 9) とマスク文字 (アスタリスク) だけです。桁修飾文字 (スラント、ハイフンおよびピリオド) は指定できません。比較項目の書式に合わせて条件値を指定してください。

「24 日」のレコードを選択する指定例を次に示します。

年と月の部分にマスク文字 (アスタリスク) を指定し、日に「24」を指定します。

【例 1】前置 4 桁年号 (書式 =1)

csvinclude=g+1-8,1 = D'\*\*\*\*\*24'

【例 2】後置 4 桁年号 (書式 =2)

csvinclude=g+1-8,2 = D'\*\*24\*\*\*\*'

【例 3】前置 2 桁年号 (書式 =3)

csvinclude=g+1-6,3 = D'\*\*\*\*\*24'

【例 4】後置 2 桁年号 (書式 =4)

csvinclude=g+1-6,4 = D'\*\*24\*\*'

マスク文字は、年月日のどの位置にも指定できます。例えば、前置 4 桁年号の比較項目に対して「D'\*\*\*\*\*1\*」と指定した場合、年月に関係なく 10 ~ 19 日のレコードが該当します。

「D'200\*01\*\*」指定した場合は、2000 ~ 2009 年の 1 月のレコードが該当します。

なお、指定した条件値が日付として正しくない場合でも、エラーにならないで正常に扱われます。例えば、前置 4 桁年号の比較項目に対して、条件値「D'\*\*\*24\*\*\*」を指定した場合、条件値は「\*\*\*2 年 4\* 月 \*\* 日」と解釈され、意図したレコードが選択されないおそれがあるので注意してください。

#### 注※ 7

前置 2 桁年号および後置 2 桁年号で、年にマスク文字を指定した場合、基準年は適用されない

## 5. exsort コマンドのパラメタおよびオペランドの詳細

ので注意してください。

### 注※ 8

exsort パラメタファイルへ次に示す条件値を指定する場合、環境変数を使って記述することができます。環境変数名で指定する場合は「% 環境変数名 %」と環境変数名を % で囲んで記述します。

【SORT EE 08-30 以降】

「H' 日付 T 時刻」

【SORT EE 08-50 以降】

「S' 文字列」、 「N' 固定小数点数」、 「X'16 進数」、 「D' 日付」、 「D' マスク日付」、 「H' マスク日付 T マスク時刻」

詳細は「4.4.5 環境変数」を参照してください。

### 注※ 9

「S' 文字列」、 「D' マスク日付」 および 「H' マスク日付 T マスク時刻」 は、 SORT EE 08-50 以降で有効な条件値です。

### 注※ 10

入力データのエンコードが「EUC-HJ」の場合、マルチバイト文字の条件値は「X'16 進数」形式で指定してください。

### ワイド文字およびマルチバイト文字に対する条件値の指定のしかた

属性コード「w」、 「u」 および 「v」 に対する条件値の指定のしかたについて説明します。

- 入力データのエンコードのエンディアンを合わせて指定してください。例えば、全角ひらがな「あ (U+3042) い (U+3044)」を「UCS-2LE」または「UTF-16LE」で指定する場合は「X'42304430」、 「UCS-2BE」または「UTF-16BE」で指定する場合は「X'30423044」と指定します。
- Unicode の入力データに対して属性コード「u」を指定した場合、UCS コードだけを比較します。Unicode の IVS (Ideographic Variation Sequence) の異体字は区別しません。

例えば、「葛 (U+845B;U+E0102)」の異体字を文字条件値に指定した場合、VS (Variation Selector, 異体字セレクタ) の「U+E0102」は無視し、UCS コードの「U+845B」で比較します。「include=u+0-1 = C' 葛」の指定では、

「葛 (U+845B;VS なし)」、 「葛 (U+845B;U+E0102)」、

「葛 (U+845B;U+E0103)」はすべて同じ文字として選択されます。

- Unicode の入力データに対して属性コード「v」を指定した場合、Unicode の IVS (Ideographic Variation Sequence) の異体字を区別します。

例えば、「葛 (U+845B;U+E0102)」の異体字を文字条件値に指定した場合、

UCS コードと VS の両方で比較します。「include=v+0-1 = C' 葛」の指定では、

「葛 (U+845B;U+E0102)」を選択し、「葛 (U+845B;VS なし)」と



「葛 (U+845B;U+E0103)」は選択されません。

- エディタなどの制約によって、Unicode のサロゲートペア文字、異体字、または Unicode 以外の第三水準・第四水準・外字などの文字を文字条件値「C' 文字列'」に直接記述できない場合は、16 進数条件値「X'16 進数'」で文字コードを指定してください。
- Unicode のサロゲートペア文字、および IVS についての詳細は、「付録 I.2(6) IVS 対応」を参照してください。

#### 条件値の扱い

cond-attribute が「h (日付時刻)」、または cond-attribute が「g (日付)」で、条件値に「D' マスク日付'」を指定した場合は、条件値と比較項目は同じ長さ (桁数) で指定してください。桁数が一致しない場合は、KBLS1314-E エラーとなります。cond-attribute および条件値が上記以外で、かつ条件値と比較項目の長さ (桁数) が一致しない場合、比較項目の桁数に合わせて条件値の桁数はそろえられます。

#### • 条件値の桁が指定した比較項目より短い場合

不足部分に埋め字をして比較項目と同じ長さにします。

文字条件値の場合は埋め字としてスペースが挿入されます。また、固定小数点数条件値、10 進数条件値、日付条件値、および 16 進数条件値の場合は、埋め字として数値「0」が挿入されます。

cond-value	形式	埋め字の位置	埋め字データ
C' 文字列 '	—	下位	空白
N' 固定小数点数 '	整数部	上位	数字「0」
	小数部	下位	
N'10 進数 '	—	上位	数字「0」
D' 日付 '	—	下位	数字「0」
X'16 進数 '	—	下位	数字「00」

(凡例)

— : 該当せず。

#### • 条件値の桁が指定した比較項目より長い場合

桁あふれ部分を切り捨てて比較項目と同じ長さにします。

cond-value	形式	切り捨て位置
C' 文字列 '	—	下位 <sup>※1</sup>
N' 固定小数点数 '	整数部	上位
	小数部	下位 <sup>※2</sup>
N'10 進数 '	—	上位
D' 日付 '	—	下位
X'16 進数 '	—	下位

## 5. exsort コマンドのパラメタおよびオペランドの詳細

(凡例)

— : 該当せず。

### 注※ 1

マルチバイト文字（日本語）の途中で桁あふれ部分を切り捨てた場合、切り捨てた部分のマルチバイト文字以降を切り捨て、代わりに空白またはゼロ値で埋め字をします。

なお、埋め字は比較項目属性「c」の場合だけです。比較項目属性「j」は日本語として扱われません。詳細は「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。

### 注※ 2

切り捨てた小数部について、桁の丸めは行わず切り捨てとなります。

環境ファイル (EXS\_ROUND), または round オプションで指定された方式は適用されません。

条件値の桁そろえの例を次に示します。なお、exsort パラメタファイルに指定する場合は、選択条件式をダブルクォーテーションで囲まないでください。

#### 【例 1】 比較項目の桁 > 条件値の桁

条件値の種類	選択条件式の指定		桁そろえ後の条件値	解説
	比較項目	条件値		
文字条件値	w+1-4	C'ab'	C'ab_△△'	2 文字分の空白を下位に仮定した条件値と見なします。
固定小数点数条件値	z+2-10,3	N'1234.567'	N' <u>000</u> 1234.567'	3 桁分の数字「0」を整数部の上位に仮定した条件値と見なします。
	z+2-10,3	N'1234567.8'	N'1234567. <u>800</u> '	2 桁分の数字「0」を小数部の下位に仮定した条件値と見なします。
	z+2-10,3	N'+123.45'	N'+ <u>0000</u> 123.45 <u>0</u> '	4 桁分の数字「0」を整数部の上位に、1 桁分の数字「0」を小数部の下位に仮定した条件値と見なします。
10 進数条件値	d+2-10	N'+12345'	N'+ <u>0000</u> 12345'	4 桁分の数字「0」を上位に仮定した条件値と見なします。条件値は符号を含めて 10 桁です。
日付条件値 ※	g+3-8,1	D'201113'	D'201113 <u>00</u> '	2 桁分の数字「0」を下位に仮定した条件値と見なします。
16 進数条件値	c+1-8	X'61006200'	X'61006200 <u>00000000</u> '	入力データのエンコードに「UCS-2LE」を指定した場合 2 文字分の数字「0000」を下位に仮定した条件値と見なします。

(凡例)

— : 埋め字を仮定する部分です。

△ : 空白を示します。

## 【例 2】比較項目の桁&lt;条件値の桁

条件値の種類	選択条件式の指定		桁そろえ後の条件値	解説
	比較項目	条件値		
文字条件値	c+1-4	C'abcdef'	C'abcd'	下位の桁あふれ部分を切り捨てます。
固定小数点 数条件値	z+2-10,3	N'123456789.123'	N'-3456789.123'	整数部の上位の桁あふれ部分を切り捨てます。
	z+2-10,3	N'1234567.890123'	N'1234567.890'	小数部の下位の桁あふれ部分を切り捨てます。
	z+2-10,3	N'+123456789.012345'	N'+3456789.012'	整数部の上位の桁あふれ部分と、小数部の下位の桁あふれ部分を切り捨てます。
10進数条件値	d+2-10	N'+123456789012345'	N'+789012345'	上位の桁あふれ部分を切り捨てます。条件値は符号を含めて 10 桁です。
日付条件値 ※	g+3-8,1	D'2011113011'	D'20111130'	下位の桁あふれ部分を切り捨てます。
16進数条件値	w+1-8	X'610062006300640065006600'	X'6100620063006400'	入力データのエンコードに「UCS-2LE」を指定した場合 下位の桁あふれ部分を切り捨てます。

(凡例)

— : 切り捨て部分です。

注※

日付条件値は、桁そろえの結果、日付として正しくないデータとなった場合でもエラーにしないで正常に扱われます。例えば、「11110103」や「20111133」などの存在しない日付、または計算機やほかのアプリケーションで表現ができないとされている年号でも、その条件値で比較します。

## 比較項目（セル）の扱い

cond-attribute が日付時刻 (h) の比較項目（セル）同士の比較では、比較項目の長さ（桁数）を一致させてください。

cond-attribute が日付時刻 (h) 以外の比較項目（セル）同士の比較で、比較項目の長さ（バイト長／桁数）が等しくない場合、長い方の比較項目の長さを比較長として仮定します。

実際に入力したデータの長さが、仮定した比較長よりも不足している場合、文字属性は空白を、数値属性、日付属性および日付時刻属性は数字「0」を埋め字します。埋め字によって、実際の入力セルデータおよび出力セルデータは変更されません。

5. exsort コマンドのパラメタおよびオペランドの詳細

cond-attribute の指定値	形式	埋め字の位置	埋め字データ
j, c	—	下位	空白
w, u, v	—	下位	空白※3
z※1	整数部	上位	数字「0」
	小数部	下位	
d	—	上位	数字「0」
h※4	日付	桁修飾文字あり	月・日ごとの上位
		桁修飾文字なし	下位
	時刻	桁修飾文字あり	時・分・秒ごとの上位 ミリ秒は下位
		桁修飾文字なし	下位
g※2	桁修飾文字あり	年・月・日ごとの上位	数字「0」
	桁修飾文字なし	下位	

(凡例)

— : 該当せず。

注※1

固定小数点数 (z) 同士の比較の場合、次の優先順位で桁を合わせます。

1. 整数部の桁 (p-s) の大きい方に、有効桁 (p) と位 (s) を合わせる。

【例 1】

左辺の比較項目 : 有効桁 (p) =10, 位 (s) =3 (整数部 7 桁, 小数部 3 桁)

右辺の比較項目 : 有効桁 (p) =8, 位 (s) =5 (整数部 3 桁, 小数部 5 桁)

→左辺の比較項目 (p=10, s=3) に合わせる。

【例 2】

左辺の比較項目 : 有効桁 (p) =6, 位 (s) =4 (整数部 2 桁, 小数部 4 桁)

右辺の比較項目 : 有効桁 (p) =5, 位 (s) =1 (整数部 4 桁, 小数部 1 桁)

→右辺の比較項目 (p=5, s=1) に合わせる。

2. 整数部の桁 (p-s) が同じ場合、位 (s) の大きい方に合わせる。

【例 3】

左辺の比較項目 : 有効桁 (p) =13, 位 (s) =8 (整数部 5 桁, 小数部 8 桁)

右辺の比較項目 : 有効桁 (p) =10, 位 (s) =5 (整数部 5 桁, 小数部 5 桁)

→左辺の比較項目 (p=13, s=8) に合わせる。

注※2

日付 (g) 同士の比較で 2 桁年号と 4 桁年号を比較する場合、2 桁年号を 4 桁年号に補正して比較します。

基準年よりも小さい場合は 1900 を、基準年以上の場合は 2000 を 2 桁年号に加

算した値に補正します。なお、基準年に 0 を指定した場合、常に 1900 を加算した値にしか補正されないので注意してください。

2桁年号 (yy)	4桁年号補正	例
yy < 基準年	20yy	基準年 = 70 : 69 → 2069 年
yy ≥ 基準年	19yy	基準年 = 70 : 70 → 1970 年 基準年 = 0 : 69 → 1969 年, 70 → 1970 年

#### 注※ 3

入力データのエンコードが「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」の場合は、「UCS-2」の文字コードで埋め字します。

入力データのエンコードが「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」の場合は、「UCS-4」の文字コードで埋め字します。

「UCS-2」および「UCS-4」の文字コードについては、「表 H-3 ASCII 対応予約文字一覧」を参照してください。

#### 注※ 4

日付時刻 (h) は、入力データの桁または埋め字の結果によって、データ形式不正と判断される場合があります。日付時刻データの端数処理については、「付録 J 日付時刻データの形式 (SORT EE の機能)」を参照してください。

#### 比較項目同士の組み合わせ

比較項目同士の比較ができる属性の組み合わせは次のとおりです。

項目 1 の属性	項目 2 の属性								
	j	c	z	d	g	h	w	u	v
j	○	—	—	—	—	—	—	—	—
c	—	○	—	—	—	—	—	—	—
z	—	—	○	—	—	—	—	—	—
d	—	—	—	○	—	—	—	—	—
g	—	—	—	—	○	—	—	—	—
h	—	—	—	—	—	○	—	—	—
w	—	—	—	—	—	—	○	—	—
u	—	—	—	—	—	—	—	○	—
v	—	—	—	—	—	—	—	—	○

(凡例)

○ : 比較できる組み合わせ

— : 比較できない組み合わせ

{AND | OR}

複数の条件式を指定する場合、論理演算子「AND」(論理積) または「OR」(論理

## 5. exsort コマンドのパラメタおよびオペランドの詳細

和)を指定して条件式を結合します。

- 「AND」と「OR」が混在した場合は、「AND」を判定後、「OR」を判定します。
- 条件式を括弧で囲むことによって、論理演算の順序を変えることができます。
- 括弧の入れ子は 24 レベルまで指定できます。条件式と括弧の間に空白またはタブを置くことができます。
- 論理演算子 (AND または OR) と括弧の間には、空白またはタブが必要です。
- exsort コマンドに指定する場合は、選択条件式全体をダブルクォーテーションで囲んでください。
- exsort パラメタファイルに指定する場合は、選択条件式をダブルクォーテーションで囲まないでください。

## 5.9 -reorg パラメタ

---

### 意味

中間レコード編集フォーマットを指定します。編集フィールドの詳細は、「5.10.10 reorg オペランド」を参照してください。

### 形式

`-reorg△reorganization-format [△reorganization-format ... ]`

(凡例)

△：半角スペース（空白）を示します。

## 5.10 -outfile パラメタ

### 意味

出力ファイル情報を指定します。

### 形式

```
-outfile [ fileorg=organization-type ]
[ file=file-name [△file-name ... ] ]
[ reclen=record-length ]
[ minlen=minimum-record-length ]
[ recsepa=record-separator-type]
[ { cellsepa= {TAB | 'cell-separator'}
  | cellsepa= {TAB | 'cell-separator-in_1'}
              △ {TAB | 'cell-separator-out_1'}
              [△ {TAB | 'cell-separator-in_2'}
                △ {TAB | 'cell-separator-out_2'} ...] } ]
[ cellnum= {ADJUST | cell-number} ]
[ quottype=option-code]
[ padding= {X | Z | C'padding-character' | X'hex-digit'} ]
[ { reorg=reorganization-format
    [△reorganization-format ... ]
  | csvreorg=csv-reorganization-format
    [△csv-reorganization-format ... ] } ]
[ { {include | omit} =condition-expression
  | {csvinclude | csvomit} =csv-condition-expression} ]
```

(凡例)

△ : 半角スペース (空白) を示します。

### 複数ファイル出力機能

出力ファイル別に、出力レコードの選択や、レコード編集ができます。この場合、出力ファイル対応に、`-outfile` パラメタを指定します。

このパラメタは最大 64 個指定できます。また、`-outfile` パラメタ 1 個につき、最大 4 個ファイル名を指定することができます。

処理終了時に出力する完了メッセージは、`-outfile` パラメタを指定した数だけ出力されます。

次に示す例では、`-outfile` パラメタを二つ指定して、2 種類の出力先を定義しています。

#### 【exsortパラメタファイル例】

```
-outfile ...1.
file=OK_FILE
include=z+10-3 >= N'80'
-outputfile ...2.
file=NG_FILE
omit=z+10-3 >= N'80'
```

1. : 出力先ファイル名 「OK\_FILE」

10 バイト目から 3 バイト (外部 10 進) が 80 以上のレコードを出力します (条件式に該当するレコードを出力します)。



## 2. : 出力先ファイル名 「NG\_FILE」

10 バイト目から 3 バイト（外部 10 進）が 80 未満のレコードを出力します（条件式に該当しないレコードを出力します）。

## 5.10.1 fileorg オペランド

## 意味

出力ファイルのファイル編成を指定します。

## 形式

`fileorg=organization-type`

- CSV ファイル以外は、入力ファイルのファイル編成と異なるファイル編成を指定できます。
- この指定は省略できます。指定を省略した場合は、入力ファイルと同じファイル編成を設定します。

## 指定値

*organization-type* に指定する値を次の表に示します。

指定値	意味
S	固定長順編成ファイルを指定します。
V	可変長順編成ファイルを指定します。
A	テキストファイルを指定します。
C	CSV ファイルを指定します。

指定できる組み合わせを次の表に示します。

入力ファイル編成	出力ファイル編成			
	S	V	A	C
S	○	○	○	—
V	○	○	○	—
A	○	○	○	—
C	—	—	—	○

(凡例) ○ : 指定できます。

— : 指定できません。

## 注意事項

- 入力ファイルと異なるファイル編成をこのオペランドに指定する場合、同時にレコード編集機能（`reorg` オペランドまたは `-reorg` パラメタ）を指定できません。出力ファイルのファイル編成の変更と、レコード編集機能は背反です。どちらか一方しか指定できません。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- 固定長順編成ファイルの場合は、`-outfile`、`-catout`、または `-splitout` パラメタの `reclen` オペランドに指定した出力レコード長にそろえて出力します。出力レコード長のそろえ方については、「5.10.3 `reclen` オペランド」の「表 5-17 固定長順編成ファイルの出力レコードのそろえ方」を参照してください。
- 可変長順編成ファイル、テキストファイル、または CSV ファイルの場合は、`-outfile`、`-catout`、または `-splitout` パラメタの `reclen` および `minlen` オペランドに指定した出力レコード長にそろえて出力します。出力レコード長のそろえ方については、「5.10.3 `reclen` オペランド」の「表 5-18 固定長順編成ファイル以外の出力レコードのそろえ方」を参照してください。また、`minlen` オペランドについては、「5.10.4 `minlen` オペランド」を参照してください。
- 入力ファイルと出力ファイルのファイル編成が異なる場合、`-outfile`、`-catout`、または `-splitout` パラメタの `reclen` および `minlen` オペランドが必要となる組み合わせがあります。組み合わせの詳細は、「5.10.3 `reclen` オペランド」の「表 5-16 ファイル編成と `reclen`、`minlen` オペランドの関係」を参照してください。
- 入力ファイルと出力ファイルのファイル編成が同じ場合、レコード形式は変更しません。出力レコード長だけそろえます。入力ファイルと出力ファイルのファイル編成が異なる場合は、レコード形式を変更して出力します。ファイル編成の組み合わせを次の表に示します。

表 5-15 ファイル編成の組み合わせ

ファイル編成		レコードヘッダ※ 3	出力レコード長の そろえ方	改行文字
入力	出力			
S	S※1	—	△	—
	V	付加	○	—
	A※2	—	△	付加※4
V	S	削除	○	—
	V※1	—	△	—
	A※2	削除	△	付加※4
A	S	—	○	削除
	V	付加	○	削除
	A※1	—	△	変更※5
C	C※1	—	△	変更※5

(凡例)

— : レコード出力時の変更なし。

○ : `-outfile`、`-catout`、または `-splitout` パラメタの `reclen` と `minlen` オペランドに指定したレコード長にそろえます。

△ : `-outfile`、`-catout`、または `-splitout` パラメタの `reclen` または `minlen` オペ

ランドの指定がある場合だけレコード長にそろえます。

注※ 1

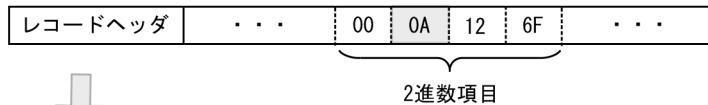
入力ファイルと出力ファイルのファイル編成が同じ場合、レコード形式は変更されず、レコード長だけそろえます。

注※ 2

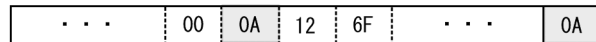
固定長順または可変長順編成ファイルをテキストファイルに変換する場合、レコードデータ中に改行文字が含まれていても、その改行文字は削除されません。この場合、出力ファイルのレコード件数が実際に処理したレコード件数よりも多くなるので注意してください。

【例】

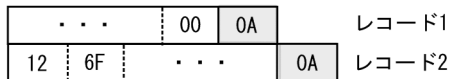
入力ファイルのファイル編成：可変長順 (V)



出力ファイルのファイル編成：テキスト (A)



次の処理の入力ファイルにする：テキスト (A)



注※ 3

`-cmdopt` パラメタの `OUTBIG` または `OUTLTL` オプションを指定している場合、レコードヘッダに格納されているレコード長を、指定したバイトオーダーに変更して出力します。

注※ 4

出力ファイルの改行文字を指定している場合、指定された改行文字を付加して出力します。出力ファイルの改行文字の指定がない場合は、`LF` を付加して出力します。出力ファイルの改行文字の詳細については、「5.10.5 `recsepa` オペランド」を参照してください。

注※ 5

出力ファイルの改行文字を指定している場合、指定された改行文字に変更して出力します。出力ファイルの改行文字の指定がない場合は、入力時の改行文字を変更しないで出力します。出力ファイルの改行文字の詳細については、「5.10.5 `recsepa` オペランド」を参照してください。

## 5.10.2 file オペランド

### 意味

出力ファイル名または環境変数名を指定します。

### 形式

```
file=file-name [△file-name … ]
```

### 注意事項

- このオペランドは省略できます。省略した場合は、標準出力ファイルを出力ファイルとします。ただし、`-outfile` パラメタを複数指定した場合は省略できません。
- 出力ファイル名を複数指定した場合、それぞれのファイルには同じレコードが出力されます。
- 出力ファイル名を複数指定する場合は、1 個以上の空白またはタブで区切って指定します。
- 同じファイル名のファイルを複数指定できません（ほかの `-outfile` パラメタの `file` オペランドに指定したファイル名と重複して指定することはできません。出力ファイル名は一意でなければなりません）。
- `exsort` パラメタファイルに環境変数名を指定する場合は、「% 環境変数名 %」と % で囲んで記述します。
- 空白文字を含んだファイル名を指定する場合は、ファイル名ごとにダブルクォーテーションで囲んで指定してください。ファイル名の指定規則については、「4.4.6 ファイル名およびディレクトリ名」を参照してください。
- `-outfile` パラメタ 1 個につき、最大 4 個ファイル名を指定することができます。
- 選択機能、集約機能およびレコード編集機能を指定していないソート機能を使用する場合に限り、入力ファイルと出力ファイルに同じファイルを指定できます。入力ファイルは、ソートした出力結果で上書きされます。既存ファイルの上書きオプション (`-cmdopt OVWRT`) を指定してください。OVWRT オプションについては、「5.5.2 ファイルの上書きオプション (OVWRT | INPOVWRT オプション)」を参照してください。
- 選択機能、集約機能、またはレコード編集機能を指定したソート機能で、入力ファイルと出力ファイルに同じファイルを指定する場合、入力ファイルの上書きオプション (`-cmdopt INPOVWRT`) を指定してください。INPOVWRT オプションについては、「5.5.2 ファイルの上書きオプション (OVWRT | INPOVWRT オプション)」を参照してください。

## 5.10.3 reclen オペランド

### 意味

出力レコード長を指定します。可変長順、テキスト、または CSV ファイルの場合は最大レコード長を指定します。

### 形式

```
reclen=record-length
```

- レコード編集 (`-reorg` パラメタ, `reorg` オペランド) の指定があり, かつ出力ファイルのファイル編成が固定長順または可変長順の場合, このオペランドは必須です (入力レコードと出力レコードのレコード長が変わらなくても必要です)。出力ファイルのファイル編成と `reclen`, `minlen` オペランドの関係を次の表に示します。

出力ファイルのファイル編成	オペランド	
	<code>reclen</code>	<code>minlen</code> ※
固定長順編成ファイル	○	×
可変長順編成ファイル	○	○
テキストファイル	△	△
CSV ファイル	△	△

(凡例)

○ : 必須

× : 指定できない

△ : 指定できる (レコード編集後のレコード長を指定)

注※

`minlen` オペランドの詳細については, 「5.10.4 `minlen` オペランド」を参照してください。

- 入力ファイルと出力ファイルのファイル編成が異なり, かつ出力ファイルのファイル編成が固定長順または可変長順の場合, `reclen` オペランドは必須です (入力レコードと出力レコード長が変わらなくても必要です)。入力ファイル, 出力ファイルのファイル編成と `reclen`, `minlen` オペランドの関係を次の表に示します。

表 5-16 ファイル編成と `reclen`, `minlen` オペランドの関係

入力ファイルのファイル編成	出力ファイルのファイル編成	オペランド	
		<code>reclen</code>	<code>minlen</code> ※
固定長順編成	固定長順編成	△	×
	可変長順編成	○	○
	テキスト	△	△
可変長順編成	固定長順編成	○	×
	可変長順編成	△	△
	テキスト	△	△
テキスト	固定長順編成	○	×
	可変長順編成	○	○
	テキスト	△	△
CSV	CSV	△	△

## 5. exsort コマンドのパラメタおよびオペランドの詳細

(凡例)

- : 必須
- × : 指定できない
- △ : 指定できる (指定したレコード長にそろえて出力します)

注※

minlen オペランドの詳細については、「5.10.4 minlen オペランド」を参照してください。

指定値

*record-length*

- 出力レコード長を、5桁以内の数字で指定します。

出力ファイル編成	指定できる範囲
固定長順編成ファイル	1 ~ 65,535
可変長順編成ファイル	1 ~ 65,535
テキストファイル※	1 ~ 65,535
CSV ファイル※	0 ~ 65,535

- 可変長順編成ファイルの場合、reclen オペランドの指定値を最大レコード長にします。

注※

- codetype オペランドに指定した入力データのエンコードの種別によって、改行文字の長さと最大レコード長が異なります。codetype オペランドについては、「5.8.13 codetype オペランド」を参照してください。

エンコード種別	改行文字のコード		テキスト・CSV ファイルの最大レコード長	
	CR	LF	改行文字が LF の場合	改行文字が CRLF の場合
ASCII	0x0d	0x0a	65,535	65,534
SJIS				
EUC-JP				
EUC-HJ				
UTF-8				
UCS-2LE	0x0d00	0x0a00	65,534	65,532
UTF-16LE				
UCS-2BE	0x000d	0x000a		
UTF-16BE				

エンコード種別	改行文字のコード		テキスト・CSV ファイルの最大レコード長	
	CR	LF	改行文字が LF の場合	改行文字が CRLF の場合
UCS-4LE	0x0d000000	0x0a000000	65,532	65,528
UTF-32LE				
UCS-4BE	0x0000000d	0x0000000a		
UTF-32BE				

- 入力データのエンコードが「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」の場合、レコード長は 2 の倍数でなければなりません。
- 入力データのエンコードが「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」の場合、レコード長は 4 の倍数でなければなりません。

### 注意事項

#### (a) レコード編集機能を指定している場合の注意事項

- 複数のレコード編集機能を指定した場合は、最後に指定したレコード編集後のレコード長を検証します。
- レコード編集後のレコード長が、このオペランドで指定した出力レコード長と不一致（または超えている）場合はエラーとなります。レコード編集と `reclen` オペランドの関係については、「5.8.14 `reorg` オペランド」の「表 5-11 レコード編集と `reclen` オペランドおよび `minlen` オペランドとの関係」および「表 5-12 出力ファイルのファイル編成と `reclen` オペランドおよび `minlen` オペランドとの関係」を参照してください。

#### (b) レコード編集機能を指定していない場合の注意事項

- このオペランドに指定したレコード長で、出力レコード長をそろえます。
- 固定長順編成ファイルの出力レコードのそろえ方を次の表に示します。

表 5-17 固定長順編成ファイルの出力レコードのそろえ方

reclen オペランド		レコード長のそろえ方
指定あり	実レコード長 < reclen	<ul style="list-style-type: none"> <li>• 出力レコード長に足りない部分を埋めます。</li> <li>• 埋め込みデータは <code>padding</code> オペランドで指定できます。</li> <li>• <code>padding</code> オペランドを省略した場合、ゼロ値で埋めます。</li> </ul>
	実レコード長 = reclen	実レコード長で出力します。
	実レコード長 > reclen	<ul style="list-style-type: none"> <li>• 出力レコード長を超えた部分を切り捨てます。</li> <li>• マルチバイト文字（日本語）の途中で超えた場合、文字の途中バイトから切り捨てます。途中バイトから切り捨てた文字は日本語として正しく表示できなくなるので注意してください。</li> </ul>
指定なし		実レコード長で出力します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- 固定長順編成ファイル以外（可変長順編成ファイル、テキストファイル、または CSV ファイル）の出力レコードのそろえ方を次の表に示します。minlen オペランドの指定方法については、「5.10.4 minlen オペランド」を参照してください。

表 5-18 固定長順編成ファイル以外の出力レコードのそろえ方

reclen, minlen オペランド		レコード長のそろえ方
指定あり	実レコード長 < minlen	<ul style="list-style-type: none"> <li>出力レコード長に足りない部分を埋めます。</li> <li>埋め込みデータは padding オペランドで指定できます。</li> <li>padding オペランドを省略した場合、次の値で埋めます。 テキスト, CSV ファイル: 空白 上記以外: ゼロ値</li> </ul>
	minlen ≤ 実レコード長 ≤ reclen	実レコード長で出力します。
	実レコード長 > reclen	<ul style="list-style-type: none"> <li>出力レコード長を超えた部分を切り捨てます。</li> <li>マルチバイト文字（日本語）の途中で超えた場合、文字の途中バイトから切り捨てます。途中バイトから切り捨てた文字は日本語として正しく表示できなくなるので注意してください。</li> </ul>
指定なし		実レコード長で出力します。

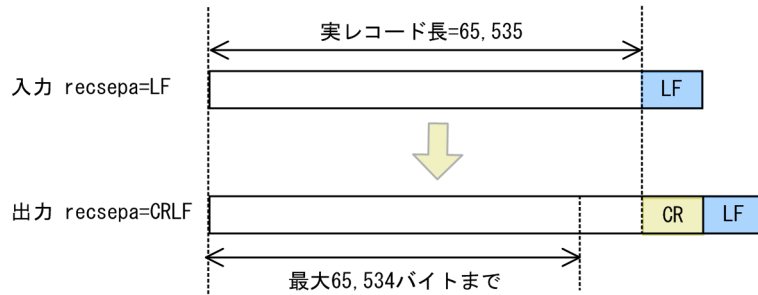
- 出力ファイルのレコード長は、改行文字を除いた長さとなります。出力レコードの改行文字の種類は、-outfile, -catout, または -splitout パラメタの recsepa オペランドの指定で決まります。詳細は、「5.10.5 recsepa オペランド」を参照してください。
- テキストファイルおよび CSV ファイルで出力レコードの改行文字を変更する場合、指定できる最大レコード長も変更されるので注意してください。例えば入力データのエンコードが「SJIS」の場合、改行文字「LF」で、かつレコード長が 65,535 バイトの入力レコードのとき、改行文字を「CRLF」に変更すると、最大レコード長の 65,534 バイトを超えて KBLS1630-E エラーとなります。

【例】

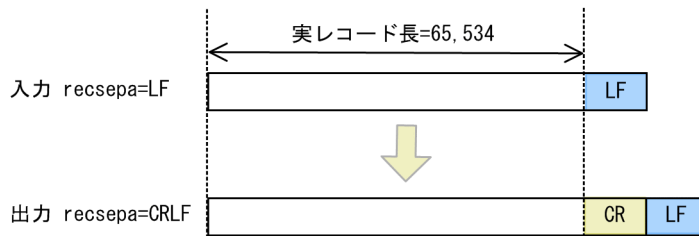


入力データのエンコード「SJIS」

●KBLS1630-Eエラーケース



●正常ケース



## 5.10.4 minlen オペランド

### 意味

可変長順, テキスト, または CSV ファイルの最小レコード長を指定します。

### 形式

`minlen=minimum-record-length`

- レコード編集 (`-reorg` パラメタ, `reorg` オペランド) の指定があり, かつ出力ファイルのファイル編成が可変長順の場合, このオペランドの指定は必須です (入力レコードと出力レコードのレコード長が変わらなくても必要です)。同様に, `reclen` オペランドの指定も必須となります。必須指定の詳細については, 「5.10.3 `reclen` オペランド」の「形式」を参照してください。
- 入力ファイルと出力ファイルのファイル編成が異なり, かつ出力ファイルのファイル編成が可変長順の場合, このオペランドの指定は必須です (入力レコードと出力レコード長が変わらなくても必要です)。同様に, `reclen` オペランドの指定も必須となります。必須指定の詳細については, 「5.10.3 `reclen` オペランド」の「形式」を参照してください。
- 出力ファイルのファイル編成が, 固定長順編成ファイルの場合は, 指定できません。指定した場合, **KBLS1664-E** エラーとなります。

### 指定値

## 5. exsort コマンドのパラメタおよびオペランドの詳細

### *minimum-record-length*

- 出力レコード長を、5桁以内の数字で指定します。

出力ファイル編成	指定できる範囲
固定長順編成ファイル	1 ~ 65,535
可変長順編成ファイル	1 ~ 65,535
テキストファイル※	1 ~ 65,535
CSV ファイル※	0 ~ 65,535

#### 注※

- 入力データのエンコードが「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」の場合、レコード長は2の倍数でなければなりません。
- 入力データのエンコードが「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」の場合、レコード長は4の倍数でなければなりません。
- 可変長順編成ファイルの場合、minlen オペランドの指定値を最小レコード長に設定します。

#### 注意事項

##### (a) レコード編集機能を指定している場合の注意事項

- 複数のレコード編集機能を指定した場合は、最後に指定したレコード編集後のレコード長を検証します。
- レコード編集後の最小レコード長が、このオペランドで指定した出力レコード長より小さい場合はエラーとなります。レコード編集と minlen オペランドの関係については、「5.8.14 reorg オペランド」の「表 5-11 レコード編集と reclen オペランドおよび minlen オペランドとの関係」および「表 15-12 出力ファイルのファイル編成と reclen オペランドおよび minlen オペランドとの関係」を参照してください。

##### (b) レコード編集機能を指定していない場合の注意事項

- このオペランドに指定したレコード長で、出力レコード長をそろえます。
- 固定長順編成ファイルの場合は、このオペランドに指定した出力レコード長にそろえて出力します。出力レコード長のそろえ方については、「5.10.3 reclen オペランド」の「表 5-17 固定長順編成ファイルの出力レコードのそろえ方」を参照してください。
- 可変長順編成ファイル、テキストファイル、または CSV ファイルの場合は、このオペランドに指定した出力レコード長にそろえて出力します。出力レコード長のそろえ方については、「5.10.3 reclen オペランド」の「表 5-17 固定長順編成ファイルの出力レコードのそろえ方」を参照してください。
- 出力ファイルの最小レコード長は、改行文字を除いた長さとなります。出力レコードの改行文字の種類は、-outfile、-catout、または -splitout の recsepa オペランドの指定で決まります。詳細は、「5.10.5 recsepa オペランド」を参照してください。

## 5.10.5 recsepa オペランド

### 意味

テキストファイルおよび CSV ファイルの出力レコードの改行文字を指定します。このオペランドは、出力ファイルのファイル編成にテキストファイルまたは CSV ファイルを指定した場合だけ有効です。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は、KBL52200-E エラーとなります。オペランド省略時のデフォルト値を、環境ファイル「EXS\_OUTRECSEPA」で指定できます。

### 形式

`recsepa=record-separator-type`

### 指定値

`record-separator-type` に指定する値を次の表に示します。

指定値	意味
CRLF	CRLF を改行とします。
LF	LF を改行とします。

### 注意事項

#### (a) 入力ファイルが固定長順または可変長順編成ファイルの場合

固定長順または可変長順の入力ファイルを、テキストファイルに変更して出力（ファイル編成変更）する場合の注意事項を次に示します。

- 環境ファイル「EXS\_OUTRECSEPA」およびこのオペランドを省略した場合、LF を付加します。

出力ファイルの recsepa 指定 (環境ファイル EXS_OUTRECSEPA)	出力レコードの改行文字
指定なし	LF
LF	LF
CRLF	CRLF

#### (b) 入力ファイルがテキストまたは CSV ファイルの場合

- 「recsepa=CRLF」または「recsepa=LF」を指定した場合、入力レコードの改行文字の種類に関係なく、このオペランドに指定した改行文字で出力します。改行文字を変更したい場合に指定してください。
- 最終レコードに改行文字がなく、eof で終わっている入力ファイルを指定した場合、改行文字がない入力レコードには改行文字が付加されて出力されます。
- 環境ファイル「EXS\_OUTRECSEPA」およびこのオペランドを省略した場合、入力ファイルに指定した改行文字※<sup>1</sup>で出力します。入力レコードの改行文字は変更され

## 5. exsort コマンドのパラメタおよびオペランドの詳細

ません。

出力ファイルの recsepa 指定 (環境ファイル EXS_OUTRECSEPA)	入力ファイルの recsepa 指定 (環境ファイル EXS_RECSEPA)	出力レコードの改行文字
指定なし	指定なし (LF を仮定)	LF
	LF	LF
	CRLF	CRLF
	LF CRLF	入力レコードの改行文字※ 2
LF	—	LF
CRLF	—	CRLF

### 注※ 1

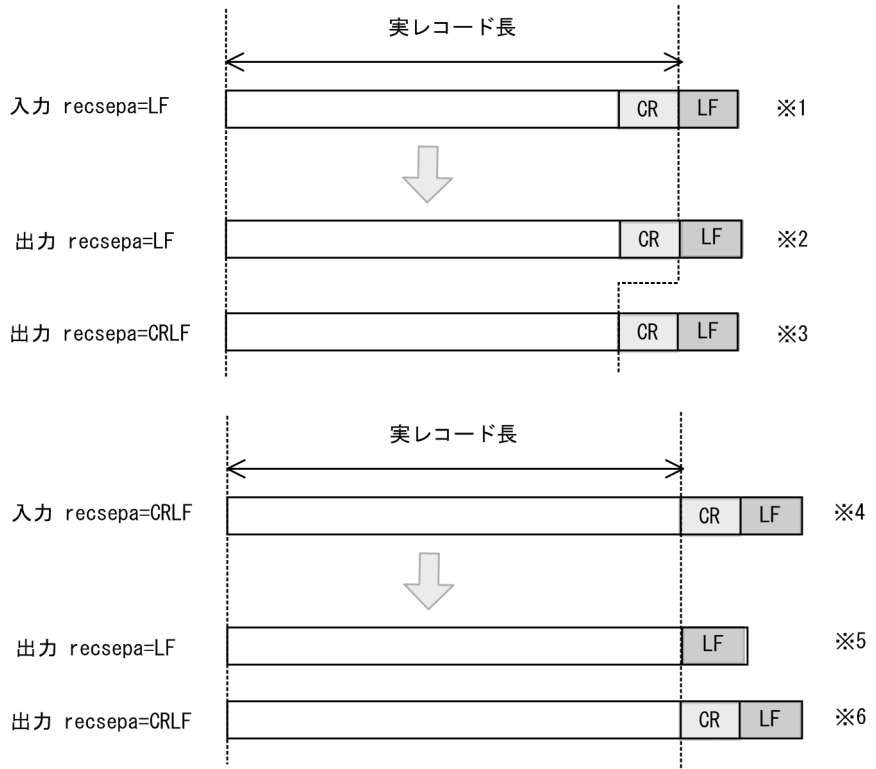
-inpfiler, -catinp, または -spllitinp の recsepa オペランド, または環境ファイル「EXS\_RECSEPA」に指定した入力ファイルの改行文字。

### 注※ 2

入力レコードごとの改行文字を変更しないで出力します。CRLF と LF が混在している入力ファイルの場合, 出力レコードも CRLF と LF が混在するので注意してください。また, 最終レコードに改行がなく, eof で終わっている入力ファイルを指定した場合, 改行がない入力レコードには, 「LF」が付加されて出力されます。

- 実際の入力レコードの改行文字が「CRLF」の場合, recsepa オペランドに指定する改行文字の種別や出力レコード長によって, 「CR」がデータとして残ってしまう場合があります。ただし, 入力レコードの改行文字を自動判定する「recsepa=LF CRLF」を指定している場合は「CR」がデータとして残ることはありません。出力レコード長の変更と改行文字の関係を次の図に示します。

## ●出力レコード長 (reclen) 指定なし



## 注※ 1

入力ファイルの改行文字を「`recsepa=LF`」と指定した場合、「`CR`」は入力データとして扱います。

## 注※ 2

出力レコード長 (`reclen`) の指定がない場合、レコード末尾に「`LF`」を付加します。「`CR`」はデータとしてレコードの末尾に残るので注意してください。

## 注※ 3

出力レコード長 (`reclen`) の指定がない場合、レコード末尾の「`CR`」を改行文字として扱い、「`LF`」を付加します。

## 注※ 4

入力ファイルの改行文字を「`recsepa=CRLF`」と指定した場合、「`CR`」は改行文字として扱います。

## 注※ 5

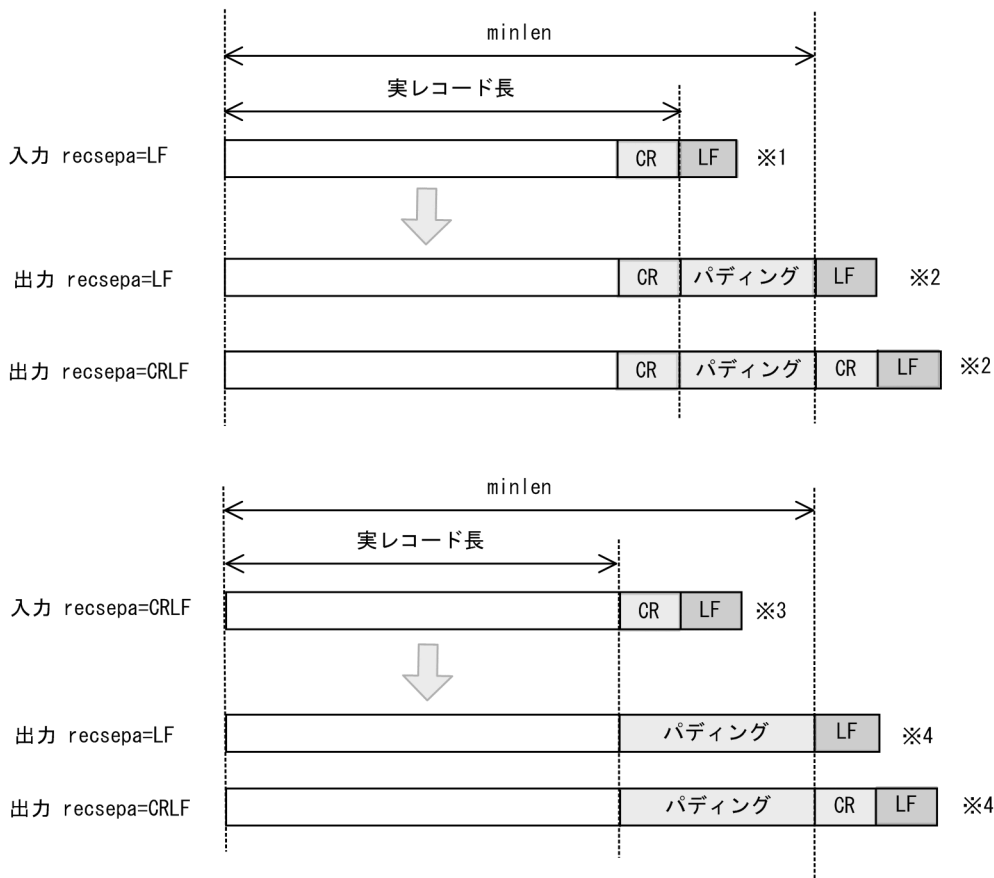
出力レコード長 (`reclen`) の指定がない場合、レコード末尾に「`LF`」を付加します。

## 注※ 6

出力レコード長 (`reclen`) の指定がない場合、レコード末尾に「`CRLF`」を付加します。

5. exsort コマンドのパラメタおよびオペランドの詳細

●実レコード長<最小出力レコード長 (minlen)



注※ 1

入力ファイルの改行文字を「recsepa=LF」と指定した場合、「CR」は入力データとして扱います。

注※ 2

実レコード長が minlen よりも小さい場合、「CR」以降をパディングし、改行文字を付加します。「CR」はデータとしてレコードの途中に残るため注意してください。

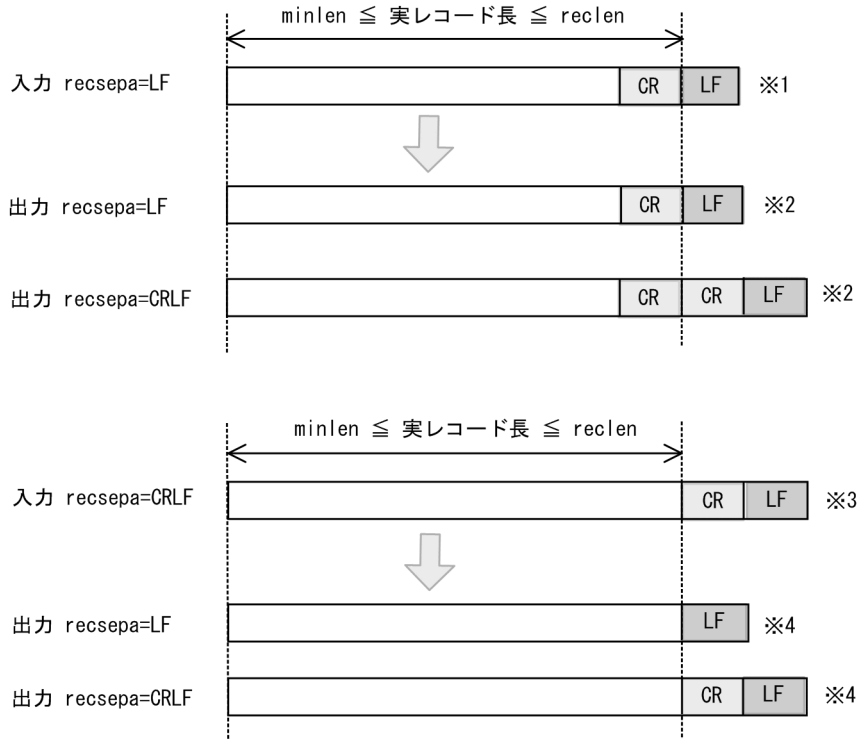
注※ 3

入力ファイルの改行文字を「recsepa=CRLF」と指定した場合、「CR」は改行文字として扱います。

注※ 4

実レコード長が minlen よりも小さい場合、入力データ以降をパディングし、改行文字を付加します。

## ●最小出力レコード長 (minlen) ≤ 実レコード長 ≤ 最大出力レコード長 (reclen)



## 注※ 1

入力ファイルの改行文字を「recsepa=LF」と指定した場合、「CR」は入力データとして扱います。

## 注※ 2

$\text{minlen} \leq \text{実レコード長} \leq \text{reclen}$  である場合、レコード末尾に「LF」を付加します。「CR」はデータとしてレコードの末尾に残るため注意してください。

## 注※ 3

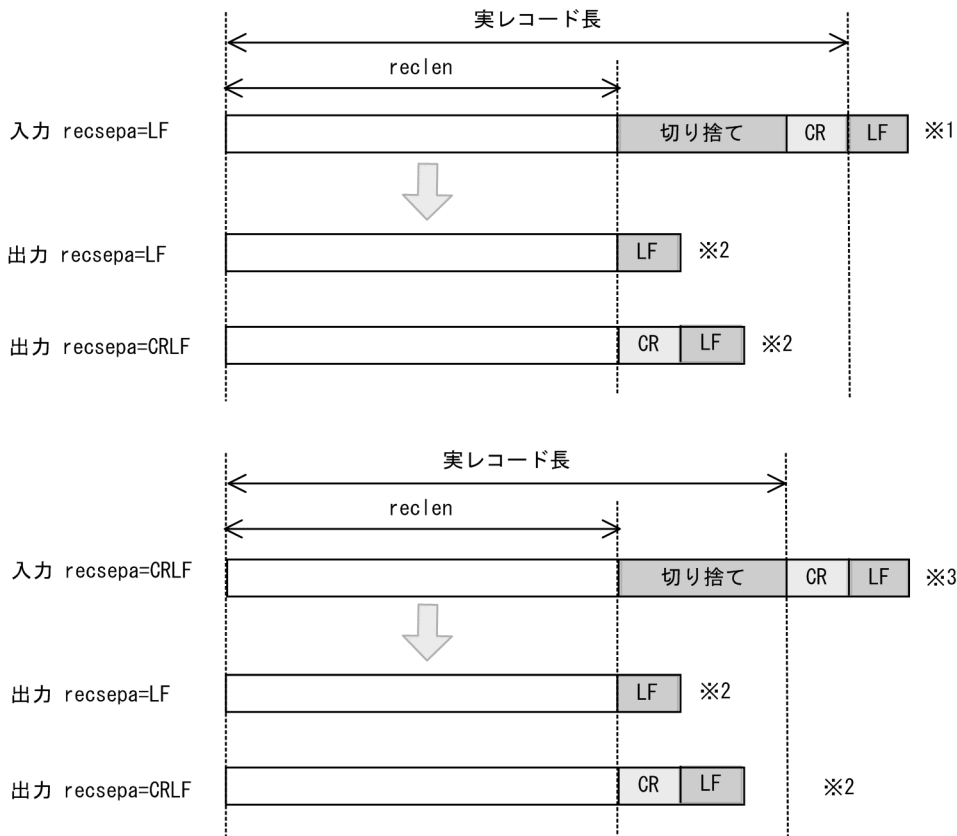
入力ファイルの改行文字を「recsepa=CRLF」と指定した場合、「CR」は改行文字として扱います。

## 注※ 4

$\text{minlen} \leq \text{実レコード長} \leq \text{reclen}$  である場合、レコード末尾に改行文字を付加します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

### ● 実レコード長 > 最大出力レコード長 (reclen)



#### 注※ 1

入力ファイルの改行文字を「recsepa=LF」と指定した場合、「CR」は入力データとして扱います。

#### 注※ 2

実レコード長が reclen より大きい場合、出力レコード長以降を切り捨て、レコード末尾に改行文字を付加します。

#### 注※ 3

入力ファイルの改行文字を「recsepa=CRLF」と指定した場合、「CR」は改行文字として扱います。

## 5.10.6 cellsepa オペランド

### 意味

CSV ファイルの出力レコードのデリミタ (セル区切り文字) を指定します。

入力レコードのデリミタを、このオペランドに指定したデリミタに変更して出力します。指定がない場合は、デリミタは変更しません。



このオペランドは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、無効となります。

## 形式

### オペランド形式 1

`cellsepa=cell-separator`

- オペランド形式 1 は、デリミタを 1 個だけ指定します。
- 入力レコードのデリミタを複数指定している場合、このオペランドで指定したデリミタにすべてのデリミタを変更して出力します。

### オペランド形式 2

`cellsepa=cell-separator-in_1△cell-separator-out_1  
[△cell-separator-in_2△cell-separator-out_2 … ]`

- オペランド形式 2 は、入力のデリミタごとに出力のデリミタを指定します。左に入力のデリミタ「`cell-separator-in`」、右に出力のデリミタ「`cell-separator-out`」を対にして 2 個ずつ指定します。
- 2 個 1 対で、最大 5 対のデリミタを指定できます。
- 入力のデリミタに指定されていない文字を指定した場合、エラーとなります。入力のデリミタは、環境ファイルの「`EXS_CELLSEPA`」または `-inpile`、`-catinp`、`-splitinp` パラメタの `cellsepa` オペランドで指定します。
- 3 個以上の奇数個を指定した場合、および 11 個以上指定した場合、指定が対になっていないのでエラーとなります。

## 指定値

`cell-separator` に指定する値を次の表に示します。

指定値	意味
<code>'cell-separator'</code>	セルのデリミタとして、タブを除いた任意の 1 文字（半角英数字※）を、シングルクォーテーションで囲んで指定します。 【指定例】 空白を指定： <code>cellsepa='△'</code> （△は空白 1 文字を表す） コンマとセミコロンの 2 個のデリミタを、それぞれ空白とコロンに変更する指定： <code>cellsepa=', '△ ' '; ' :'</code>
TAB	セルのデリミタとして、タブを指定します。 【指定例】 <code>cellsepa=TAB</code>

注※ 半角英数字の文字コードは「0x20～0x7E」です。

- 引用符（シングルクォーテーション、ダブルクォーテーション）、改行文字、およびマルチバイト文字（日本語）はデリミタとして指定できません。
- 入力データのエンコードが「`UCS-2LE`」、`UCS-2BE`、`UTF-16LE` または「`UTF-16BE`」の場合、「`UCS-2`」の文字コードのデータ形式となります。「表 H-3 ASCII 対応予約文字一覧」を参照してください。
- 入力データのエンコードが「`UCS-4LE`」、`UCS-4BE`、`UTF-32LE` または

## 5. exsort コマンドのパラメタおよびオペランドの詳細

「UTF-32BE」の場合、「UCS-4」の文字コードのデータ形式となります。「表 H-3 ASCII 対応予約文字一覧」を参照してください。

- `cell-separator` は、必ずシングルクォーテーション (') で囲んで指定します。

### 注意事項

- 環境ファイル「EXS\_OUTCELLSEPA」に初期値を指定できます。
- 環境ファイル「EXS\_OUTCELLSEPA」に指定する場合は、`cell-separator` をダブルクォーテーション (") で囲みます。コマンドと指定方法が異なるので注意してください。
- `cell-separator` 指定を `exsort` コマンドに指定する場合は、指定値全体をダブルクォーテーションで囲んでください。TAB 指定の場合は、ダブルクォーテーションで囲む必要はありません。

#### 【例】

```
-outfile cellsepa="' Δ '": ダブルクォーテーションで囲む  
-outfile cellsepa=TAB: ダブルクォーテーションの囲みは不要
```

- このオペランドを `exsort` パラメタファイルに指定する場合は、指定値をダブルクォーテーションで囲まないでください。

#### 【例】

```
-outfile cellsepa=' Δ '  
-outfile cellsepa=TAB
```

- 連続デリミタを 1 つにして出力する場合は、コマンドオプション「OUTCONSEPA」を指定してください。OUTCONSEPA オプションについては、「5.5.18 連続区切り文字オプション (INPCONSEPA, OUTCONSEPA オプション)」を参照してください。

### 関連パラメタ・オペランド

- `-infile` パラメタ `cellsepa` オペランド
- `-cmdopt` パラメタ `OUTCONSEPA` オプション

### 指定例

#### 【例 1】

exsort コマンドの指定内容

```
-infile cellsepa="' Δ ' TAB" -outfile cellsepa="','"
```

#### 入出力デリミタの関係

項番	入力のデリミタ (2 種類)	出力のデリミタ
1	Δ (半角空白)	, (コンマ)
2	→ (タブ)	

入力レコード

0001	△	0002	→	0003	△	0004	→	0005	LF
------	---	------	---	------	---	------	---	------	----

出力レコード

0001	,	0002	,	0003	,	0004	,	0005	LF
------	---	------	---	------	---	------	---	------	----

【例 2】

exsort コマンドの指定内容

```
-infile cellsepa="',' '△' TAB ';' ':'"
-outfile cellsepa="',' ' _ ' '△' '*' TAB '-' '/' ':' '%'"
```

入出力デリミタの関係

項番	入力のデリミタ (5種類)	出力のデリミタ
1	, (コンマ)	_ (アンダーバー)
2	△ (半角空白)	* (アスタリスク)
3	→ (タブ)	- (ハイフン)
4	; (セミコロン)	/ (スラッシュ)
5	: (コロン)	% (パーセント)

入力レコード

0001	△	0002	→	0003	;	0004	,	0005	:	0006	LF
------	---	------	---	------	---	------	---	------	---	------	----

出力レコード

0001	*	0002	-	0003	/	0004	_	0005	%	0006	LF
------	---	------	---	------	---	------	---	------	---	------	----

## 5.10.7 cellnum オペランド

意味

レコード出力時のセル数調整を指定します。

このオペランドは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、KBLS2202-E エラーとなります。

形式

```
cellnum= { ADJUST | cell-number }
```

指定値

このオペランドを指定した場合、すべての出力レコードのセル数を一致させて出力します。

このオペランドを省略した場合、入力レコード (または編集後のレコード) のセル数を変更せずに出力します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

指定値	意味
ADJUST	入力時またはレコード編集後の最大セル数に合わせて調整します。 セル数が不足している場合は、デリミタを付加して不足数分のナル文字列 (null セル) を作成し、セル数を一致させて出力します。
<i>cell-number</i>	<i>cell-number</i> で指定したセル数に合わせて調整します。 指定できる範囲は、1 ~ 16,384 です。 <ul style="list-style-type: none"><li>• <i>cell-number</i> で指定したセル数が実際のセル数よりも少ない場合、<i>cell-number</i> を超えた部分のセルを削除して出力します。</li><li>• <i>cell-number</i> で指定したセル数が実際のセル数よりも多い場合、デリミタを付加して <i>cell-number</i> に足りない分のナル文字列 (空のセル) を作成して出力します。</li></ul>

### 注意事項

- 「cellnum=ADJUST」は、ソート機能だけに有効です。マージおよびコピー機能に指定しても無効です (セル数は変更されず、入力時のセル数のまま出力されま
- す)。
- ナル文字列が作成されると、付加されたデリミタの分だけレコード長が増加しますので注意してください。
- ナル文字列を作成する場合、付加するデリミタは次の順番で決定されます。なお、複数のデリミタを指定している場合は、先頭に指定したデリミタとなります。

1. -outfile, -catout, または -splitout パラメタの cellsepa オペランドに指定したデリミタ
2. 環境ファイルの「EXS\_OUTCELLSEPA」に指定したデリミタ
3. -infile, -catinp, または -splitinp パラメタの cellsepa オペランドに指定したデリミタ
4. 環境ファイルの「EXS\_CELLSEPA」に指定したデリミタ
5. コンマ

### 関連パラメタ・オペランド

- -outfile パラメタ cellsepa オペランド
- -infile パラメタ cellsepa オペランド

## 5.10.8 quottype オペランド

### 意味

レコード出力時のセルの引用符を指定します。

入力レコードのセルの引用符を削除、付加、または変更して出力します。このオペランドの指定がない場合は、引用符の削除、付加、および変更は行いません。

このオペランドは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、KBLS2204-E エラーとなります。

### 形式

quottype=option-code

### 指定値

*option-code* に指定する値を次の表に示します。

option-code の指定値	出力レコードの 引用符の変更	セルデータ	レコード編集 (csvreorg) の挿 入データ
0 (省略)	変更なし	入力のセルデータをそのまま出力します。	指定された挿入データをそのまま出力します。
1	削除	入力のセルデータに引用符がある場合、引用符を削除して出力します。	指定された挿入データをそのまま出力します。
2	付加	入力のセルデータが引用符で囲まれていない場合、引用符で囲んで出力します。 <ul style="list-style-type: none"> <li>付加する引用符は、入力の <b>quottype</b> に指定した引用符<sup>※1</sup>です。</li> <li>ナル文字列についても引用符で囲んで出力します。</li> </ul>	指定された挿入データを、引用符で囲んで出力します。 <ul style="list-style-type: none"> <li>付加する引用符は、入力の <b>quottype</b> に指定した引用符<sup>※1</sup>です。</li> <li>ナル文字列についても引用符で囲んで出力します。</li> </ul>
3	付加	入力のセルデータが引用符で囲まれていない場合、引用符で囲んで出力します。 <ul style="list-style-type: none"> <li>付加する引用符は、入力の <b>quottype</b> に指定した引用符<sup>※1</sup>です。</li> <li>ナル文字列は引用符で囲まらずに出力します。</li> </ul>	指定された挿入データを、引用符で囲んで出力します。 <ul style="list-style-type: none"> <li>付加する引用符は、入力の <b>quottype</b> に指定した引用符<sup>※1</sup>です。</li> <li>ナル文字列は引用符で囲まらずに出力します。</li> </ul>
4	付加・変更	入力のセルデータが引用符で囲まれていない場合は、引用符で囲んで出力します。 入力のセルデータが引用符で囲まれている場合は、引用符を変更して出力します。 <ul style="list-style-type: none"> <li>付加する引用符は、入力の <b>quottype</b> に指定していない引用符<sup>※2</sup>です。</li> <li>ナル文字列についても引用符で囲んで出力します。</li> </ul>	指定された挿入データを、引用符で囲んで出力します。 <ul style="list-style-type: none"> <li>付加する引用符は、入力の <b>quottype</b> に指定していない引用符<sup>※2</sup>です。</li> <li>ナル文字列についても引用符で囲んで出力します。</li> </ul>
5	付加・変更	入力のセルデータが引用符で囲まれていない場合、引用符で囲んで出力します。 入力のセルデータが引用符で囲まれている場合は、引用符を変更して出力します。 <ul style="list-style-type: none"> <li>付加する引用符は、入力の <b>quottype</b> に指定していない引用符<sup>※2</sup>です。</li> <li>ナル文字列は引用符で囲まらずに出力します。</li> </ul>	指定された挿入データを、引用符で囲んで出力します。 <ul style="list-style-type: none"> <li>付加する引用符は、入力の <b>quottype</b> に指定していない引用符<sup>※2</sup>です。</li> <li>ナル文字列は引用符で囲まらずに出力します。</li> </ul>

注※1

環境ファイルの「EXS\_QUOTTYPE」または `-inpf`、`-catinp`、`-splitinp` パラメタの **quottype** オペランドに指定した引用符が付加されます。引用符の指定を省略した場合は、ダブルクォー

## 5. exsort コマンドのパラメタおよびオペランドの詳細

テーションが仮定されます。

入力の引用符 (quottype 指定値)	入力の引用符	出力の引用符
省略	ダブルクォーテーション (")	ダブルクォーテーション (")
SQUOTE	シングルクォーテーション (')	シングルクォーテーション (')
DQUOTE	ダブルクォーテーション (")	ダブルクォーテーション (")

### 注※2

環境ファイルの「EXS\_QUOTTYPE」または `-inpfiler`、`-catinpf`、`-splitinpf` パラメタの `quottype` オペランドに指定していない引用符が付加されます。

入力の引用符 (quottype 指定値)	入力の引用符	出力の引用符
省略	ダブルクォーテーション (")	シングルクォーテーション (')
SQUOTE	シングルクォーテーション (')	ダブルクォーテーション (")
DQUOTE	ダブルクォーテーション (")	シングルクォーテーション (')

### 注意事項

- オプションコード「0」または「1」を指定すると引用符が付加されません。セルデータや挿入データ中に、改行文字やデリミタが含まれる場合、出力レコード件数や出力セル数が変わるので注意してください。
- オプションコード「0」以外は、引用符の削除・付加によって、出力レコードのレコード長が増減する場合がありますので注意してください。
- 入力データのエンコードが「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」の場合、「UCS-2」の文字コードのデータ形式となります。「表 H-3 ASCII 対応予約文字一覧」を参照してください。また、入力データのエンコードについては、「5.8.13 codetype オペランド」を参照してください。
- 入力データのエンコードが「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」の場合、「UCS-4」の文字コードのデータ形式となります。「表 H-3 ASCII 対応予約文字一覧」を参照してください。また、入力データのエンコードについては、「5.8.13 codetype オペランド」を参照してください。

### 関連パラメタ・オペランド

- `-inpfiler` パラメタ `quottype` オペランド
- `-inpfiler` パラメタ `csvreorg` オペランド
- `-csvreorg` パラメタ
- `-outfile` パラメタ `csvreorg` オペランド

## 5.10.9 padding オペランド

### 意味

出力レコード長をそろえるためのパディング文字を指定します。

このオペランドは、`-outfile`、`-catout`、および `-splitout` パラメタの `reclen` または `minlen` オペランドを指定した場合だけ有効です。

- 固定長順編成ファイルの場合は、`-outfile`、`-catout`、または `-splitout` パラメタの `reclen` オペランドに指定したレコード長にそろえて出力します。実レコード長が、`reclen` オペランドに指定した出力レコード長よりも小さい場合、このオペランドに指定したパディング文字で足りない部分を埋めます。
- 可変長順編成ファイル、テキストファイル、または CSV ファイルの場合は、`-outfile`、`-catout`、または `-splitout` パラメタの `reclen` および `minlen` オペランドに指定したレコード長にそろえて出力します。実レコード長が、`minlen` オペランドに指定した出力レコード長よりも小さい場合、このオペランドに指定したパディング文字で足りない部分を埋めます。
- 環境ファイル「`EXS_PADDING`」およびこのオペランドの両方を省略した場合、出力ファイルのファイル編成がテキストまたは CSV ファイルであればパディング文字に空白を、テキストおよび CSV ファイル以外であればゼロ値を仮定します。

出力ファイルのファイル編成	オペランド省略時のパディング文字
S	ゼロ値
V	ゼロ値
A	空白
C	空白

#### 形式

```
padding= { X | Z | C'padding-character' | X'16進数' }
```

#### 指定値

指定値	意味
X	パディング文字として、空白を指定します。
Z	パディング文字として、ゼロ値を指定します。
C'padding-character'	パディング文字として、任意の 1 文字（半角英数字）を指定します。 【指定例】padding=C'*'
X'16進数'	パディング文字として、任意の 1 バイトのデータを指定します。 【指定例】padding=X'ff'

- `padding-character` には、改行文字およびマルチバイト文字（日本語）は指定できません。
- 16 進数として指定できるのは「0～9」および「A～F」だけです。
- `padding-character` または 16 進数指定でこのオペランドを `exsort` コマンドに指定する場合、指定値全体をダブルクォーテーションで囲んで指定してください。また、`padding-character` に引用符（シングルクォーテーションまたはダブルクォー

## 5. exsort コマンドのパラメタおよびオペランドの詳細

テーション) を指定する場合は、padding-character を「¥」でエスケープしてください。なお、padding-character および 16 進数を指定しない場合は、ダブルクォーテーションで囲むことは不要です。

### 【例】

```
-outfile padding="C'*'" : ダブルクォーテーションで囲む
-outfile padding="X'ff'" : ダブルクォーテーションで囲む
-outfile padding="C'¥'" : パディング文字 (引用符) を「¥」でエスケープ
-outfile padding=X : ダブルクォーテーションの囲み不要
-outfile padding=Z : ダブルクォーテーションの囲み不要
```

- exsort パラメタファイルに指定する場合は、ダブルクォーテーションで囲まないでください。また、padding-character に引用符 (シングルクォーテーションまたはダブルクォーテーション) を指定する場合は、「¥」でのエスケープは不要です。

### 【例】

```
-outfile padding=C'*'
-outfile padding=X'ff'
-outfile padding=C''
-outfile padding=X
-outfile padding=Z
```

- 入力データのエンコードが「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」の場合、「UCS-2」の文字コードのデータ形式となります。「表 H-3 ASCII 対応予約文字一覧」を参照してください。また、入力データのエンコードについては、「5.8.13 codetype オペランド」を参照してください。
- 入力データのエンコードが「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」の場合、「UCS-4」の文字コードのデータ形式となります。「表 H-3 ASCII 対応予約文字一覧」を参照してください。また、入力データのエンコードについては、「5.8.13 codetype オペランド」を参照してください。

## 5.10.10 reorg オペランド

### 意味

出力レコード編集フォーマットを指定します。編集フォーマットの詳細は、「5.8.14 reorg オペランド」を参照してください。

### 形式

```
reorg=reorganization-format [ $\Delta$ reorganization-format ...]
```

指定できるフォーマットの個数は、1 ~ 128 個です。

## 5.10.11 csvreorg オペランド

### 意味

CSV ファイルの出力レコード編集フォーマットを指定します。



このオペランドは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、KBLS1522-E エラーとなります。

編集フォーマットの詳細は、「5.8.15 csvreorg オペランド」を参照してください。

#### 形式

```
csvreorg=csv-reorganization-format [ $\Delta$ csv-reorganization-format
... ]
```

指定できるフォーマットの個数は、1 ～ 128 個です。

### 5.10.12 {include | omit} オペランド

#### 意味

出力レコードの選択条件式を指定します。

ここで指定した選択条件式で選択されたレコードは、指定した複数の出力先ファイルへ振り分けて出力することができます。

詳細は、「5.8.16 {include | omit} オペランド」を参照してください。

#### 形式

```
{include | omit} ="condition-expression [ $\Delta$  {AND | OR}  $\Delta$ 
condition-expression ... ]"
```

### 5.10.13 {csvinclude | csvomit} オペランド

#### 意味

CSV ファイルの出力レコードの選択条件式を指定します。

このオペランドは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、KBLS1322-E エラーとなります。

ここで指定した選択条件式で選択されたレコードは、指定した複数の出力先ファイルへ振り分けて出力できます。

詳細は、「5.8.17 {csvinclude | csvomit} オペランド」を参照してください。

#### 形式

```
{csvinclude | csvomit} ="csv-condition-expression [ $\Delta$  {AND | OR}
 $\Delta$ csv-condition-expression ... ]"
```

## 5.11 -sorttmpdir パラメタ

---

### 意味

ソート処理で使用するソート用一時ファイルの割り当てディレクトリ名を指定します。

### 形式

```
-sorttmpdir temporary-directory1
```

- このパラメタは、ソート機能 (-function sort) の指定がある場合だけ有効です。ソート機能の指定がない場合は、指定しても無視されます。
- 空白文字を含んだディレクトリ名を指定する場合は、ディレクトリ名をダブルクォーテーションで囲んで指定してください。ディレクトリ名の指定規則については、「4.3.5 ファイル名およびディレクトリ名の指定形式」を参照してください。
- 指定するディレクトリには、exsort コマンドの実行ユーザの読み込み、書き込み権限が必要です。さらに、Windows の場合は、削除権限も必要となります。
- 一時ファイルが作成される手順を次に示します。

#### 1. ディレクトリ名の仮定

次に示す優先順位でディレクトリ名を仮定します。

1. -sorttmpdir パラメタに指定されたディレクトリ
2. 環境変数「EXSORTSORTTMPDIR (64 ビット版は「EXSORT64SORTTMPDIR)」に設定されたディレクトリ
3. Windows の場合、SORT の環境ファイルの temp\_dir1 オプションに設定されたディレクトリ※1
- UNIX の場合、SORT の環境変数「SORTOPT」の -T オプションに指定されたディレクトリ※2
4. エラー (KBLS1674+E 一時ファイルのディレクトリが指定されていません)

#### 2. 一時ファイルの作成

1. で仮定したディレクトリに対して、一時ファイルを作成します。

ただし、UNIX の場合、1. で仮定したディレクトリとは別のディレクトリに一時ファイルを作成する場合があります。UNIX の作成条件を次に示します。

- 環境変数「TMPDIR」の指定がある場合、1. で仮定したディレクトリではなく、環境変数「TMPDIR」で設定されているディレクトリを優先します。
- 環境変数「TMPDIR」で設定されたディレクトリ、または 1. で仮定したディレクトリに書き込み権限がない場合、HP-UX は「/var/tmp」に、AIX または Linux は「/tmp」に一時ファイルを作成します。

#### 注※1

環境ファイルの設定方法については、「15.2 SORT の環境ファイルの設定 (Windows の場合)」を参照してください。

注※ 2

環境変数の設定方法については、「14.2 SORT の環境変数の設定 (UNIX の場合)」を参照してください。

## 5.12 -mergetmpdir パラメタ

### 意味

ソート処理で使用するマージ用一時ファイルの割り当てディレクトリ名を指定します。

### 形式

```
-mergetmpdir temporary-directory2
```

- このパラメタは、ソート機能 (-function sort) の指定がある場合だけ有効です。ソート機能の指定がない場合は、指定しても無視されます。
- 空白文字を含んだディレクトリ名を指定する場合は、ディレクトリ名をダブルクォーテーションで囲んで指定してください。ディレクトリ名の指定規則については、「4.3.5 ファイル名およびディレクトリ名の指定形式」を参照してください。
- 指定するディレクトリには、exsort コマンドの実行ユーザの読み込み、書き込み権限が必要です。さらに、Windows の場合は、削除権限も必要となります。
- 一時ファイルが作成される手順を次に示します。
  1. ディレクトリ名の仮定
 

次に示す優先順位でディレクトリ名を仮定します。

    1. 環境変数「EXSORTMERGETMPDIR (64 ビット版は「EXSORT64MERGETMPDIR)」」に設定されたディレクトリ
    2. Windows の場合、SORT の環境ファイルの temp\_dir3 オプションに設定されたディレクトリ※<sup>1</sup>
    3. UNIX の場合、SORT の環境変数「SORTOPT」の -T2 オプションに指定されたディレクトリ※<sup>2</sup>
    4. -sorttmpdir パラメタで決定したソート用一時ファイルのディレクトリ (詳細は、「5.11 -sorttmpdir パラメタ」を参照してください)
  2. 一時ファイルの作成
 

1. で仮定したディレクトリに対して、一時ファイルを作成します。

ただし、UNIX の場合、1. で仮定したディレクトリとは別のディレクトリに一時ファイルを作成する場合があります。UNIX の作成条件を次に示します。

    - 環境変数「TMPDIR」の指定がある場合、1. で仮定したディレクトリではなく、環境変数「TMPDIR」で設定されているディレクトリを優先します。
    - 環境変数「TMPDIR」で設定されたディレクトリ、または 1. で仮定したディレクトリに書き込み権限がない場合、HP-UX は「/var/tmp」に、AIX または Linux は「/tmp」に一時ファイルを作成します。

### 注※ 1

環境ファイルの設定方法については、「15.2 SORT の環境ファイルの設定 (Windows の場合)」を参照してください。

注※2

環境変数の設定方法については、「14.2 SORT の環境変数の設定 (UNIX の場合)」を参照してください。

## 5.13 -worksize パラメタ

### 意味

ソート処理で使用するワークバッファのサイズを指定します。

### 形式

```
-worksize {  buffer-size [unit ]
             | auto [ , max-buffer-size [unit ] ]
             | incore [ , max-buffer-size [unit ] ] }
```

- このパラメタは、ソート機能 (-function sort) の指定がある場合だけ有効です。ソート機能の指定がない場合は、指定しても無視されます。
- 環境変数「EXSORT\_HI\_WORKSIZE (64 ビット版は「EXSORT64\_HI\_WORKSIZE)」」が指定された場合、-worksize パラメタおよび環境変数「EXSORTWORKSIZE (64 ビット版は「EXSORT64WORKSIZE)」」に指定された値よりも優先されます。次の優先順位でワークバッファサイズが設定されます。
  1. 環境変数「EXSORT\_HI\_WORKSIZE (64 ビット版は「EXSORT64\_HI\_WORKSIZE)」」に設定された値
  2. -worksize パラメタで設定された値
  3. 環境変数「EXSORTWORKSIZE (64 ビット版は「EXSORT64WORKSIZE)」」に設定された値
  4. Windows の場合、SORT の環境ファイル<sup>※1</sup>の work\_size オプションに設定された値
  - UNIX の場合、SORT の環境変数<sup>※2</sup>「SORTOPT」の -y オプションに設定された値
  5. auto オペランドを設定 (省略値)
- auto 指定時、incore 指定時、または -worksize 省略時、確保したワークサイズが自動で見積もりした値より小さい場合、KBLS1007-I が出力されます。ワークサイズが 1MB 確保できない場合は、KBLS1720-E が出力されエラーとなります。
- 標準入力ファイルに対して、auto オペランドまたは incore オペランドを指定した場合、ワークバッファサイズは、1MB に設定されます。

#### 注※1

SORT の環境ファイルの設定方法については、「15.2 SORT の環境ファイルの設定 (Windows の場合)」を参照してください。

#### 注※2

SORT の環境変数の設定方法については、「14.2 SORT の環境変数の設定 (UNIX の場合)」を参照してください。

### 指定値

**buffer-size**

- 32 ビット版  
ソート処理で使用するワークバッファサイズを 1 ～ 4 桁の 10 進数で指定します。  
指定できる範囲は、1 ～ 2,000 です。
- 64 ビット版  
ソート処理で使用するワークバッファサイズを 1 ～ 7 桁の 10 進数で指定します。  
指定できる範囲は、1 ～ 2,048,000 です。

**unit**

ソート処理で使用するワークバッファサイズの単位を指定します。

**M**：単位を MB にします。

**G**：単位を GB にします。

単位を省略した場合は、**M(MB)** が設定されます。

**G(GB)** は、64 ビット版だけ指定できます。

ワークバッファサイズの指定範囲を次の表に示します。

SORT EE の種別	unit 指定	指定範囲
32 ビット版	省略	1 ～ 2,000
	M(MB)	1M ～ 2,000M
	G(GB)	指定不可
64 ビット版	省略	1 ～ 2,048,000
	M(MB)	1M ～ 2,048,000M
	G(GB)	1G ～ 2,000G

**auto**

ソート処理で使用するワークバッファサイズを自動で見積もります (アウトコアソート対応)。ワークバッファサイズの見積もり式については、「17.1.1(2) アウトコアソート」の「図 17-3 一時ファイルを作成する (アウトコアソート) 場合のワークバッファサイズ見積もり式」を参照してください。

**incore**

ソート処理で使用するワークバッファサイズを自動で見積もります (インコアソート対応)。ワークバッファサイズの見積もり式については、「17.1.1(1) インコアソート」の「図 17-2 一時ファイルを作成しない (インコアソート) 場合のワークバッファサイズ見積もり式」を参照してください。

**max-buffer-size**

ソート処理で使用するワークバッファサイズの最大値を指定します。

- 32 ビット版  
ソート処理で使用するワークバッファサイズを 1 ～ 4 桁の 10 進数で指定しま

## 5. exsort コマンドのパラメタおよびオペランドの詳細

す。

指定できる範囲は、1～2,000 です。

*max-buffer-size* 省略時は、「2000(MB)」を仮定します。

- 64 ビット版

ソート処理で使用するワークバッファサイズを1～7桁の10進数で指定します。

指定できる範囲は、1～2,048,000 です。

*max-buffer-size* 省略時は、「128(GB)」を仮定します。

### ワークバッファの確保サイズ

- *buffer-size* 指定

指定されたサイズで確保します。

- auto 指定

*max-buffer-size* の指定がない場合は、見積もりサイズで確保します。

*max-buffer-size* の指定がある場合は、見積もりサイズが *max-buffer-size* 指定値を超えた場合、*max-buffer-size* 指定値を仮定します。

- *incore* 指定

*max-buffer-size* の指定がない場合 (32 ビット版) :

見積もりサイズで確保します。

*max-buffer-size* の指定がない場合 (64 ビット版) :

見積もりサイズが 128GB (*max-buffer-size* 省略時の仮定値) を超えた場合、auto (*max-buffer-size* 指定なし) を仮定してワークバッファサイズを見積もり直します。

*max-buffer-size* の指定がある場合 :

見積もりサイズが *max-buffer-size* 指定値を超えた場合、auto

(*max-buffer-size* 指定なし) を仮定してワークバッファサイズを見積もり直します。

なお、システム環境によっては、auto および *incore* 指定で見積もったサイズを割り当てられない場合があります。この場合は、見積もりサイズよりも少ないサイズでソート処理を実行します。このときのソート完了を示す KBLS1007-I メッセージには、見積もりサイズおよび実際に割り当てられたサイズが出力されます。

### 注意事項

- *incore* オペランドのワークバッファサイズの最大値「*max-buffer-size*」が小さ過ぎると、アウトコアソートとなり、期待する性能が得られないおそれがありますので、最大値の設定時は注意が必要です。
- *incore* オペランドのワークバッファサイズの最大値「*max-buffer-size*」が大き過ぎると、システムのメモリが不足し、システムに影響を与えるおそれがありますので、最大値の設定時は注意が必要です。
- ワークバッファサイズの見積もりについては、「17.1 ソート機能の処理時間を短くするためには」を参照してください。



- `buffer-size` を指定した場合、ソート完了を示す **KBLS1007-I** メッセージには、ソート処理方式や確保したワークバッファサイズなどの情報「`worksize=...`」は出力されません。
- `buffer-size` で指定したワークバッファサイズを割り当てられない場合は、**KBLS1720-E** が出力されエラーとなります。

## 5.14 -dateopt パラメタ

---

### 意味

日付項目の基準年を指定します。

### 形式

`-dateopt base-year`

- 最小年として処理する値として、西暦下 2 桁の数値を、1 ~ 2 桁の 10 進数で指定します。
- 基準年はキー比較時に、指定した値を年度の先頭として位置づけます。基準年よりも小さい値は、100 を加えた値を仮定して処理します。
- このパラメタは、キーまたは比較項目に日付属性 (g) を指定した場合だけ有効となります。
- 0 を指定した場合は、文字項目で処理した結果と同じになります。
- 指定を省略した場合は、次の優先順位で基準年が設定されます。
  1. 環境変数「EXSORTDATEOPT (64 ビット版は「EXSORT64DATEOPT)」」に設定された値
  2. Windows の場合、SORT の環境ファイルの `date_opt` オプションに設定された値  
※1  
UNIX の場合、SORT の環境変数「SORTOPT」の `-g` オプションに指定された値  
※2
  3. 70 を設定 (省略値)

### 注※1

環境ファイルの設定方法については、「15.2 SORT の環境ファイルの設定 (Windows の場合)」を参照してください。

### 注※2

環境変数の設定方法については、「14.2 SORT の環境変数の設定 (UNIX の場合)」を参照してください。

## 5.15 -altseq パラメタ

### 意味

JIS8 単位コード体系以外の順序でソートキー、マージキー、または集約キーを比較したい場合に、文字コードの大小関係を定義します。

また、このパラメタ以外に、altseq 変換ファイルで文字コードの大小関係を定義する方法があります。

altseq 変換ファイルについては、「付録 G altseq 変換ファイル (SORT EE の機能)」を参照してください。

### 形式

`-altseq code=xyy1 [△…△xyy256]`

- `-key` パラメタのキー属性に「q」と指定したキーの比較順序が、このパラメタで定義された順序に変換されます。キー属性「q」の指定がない場合は、このパラメタは無効です。
- `-altseq` パラメタを指定しないでキー属性「q」を指定した場合、altseq 変換ファイルが設定されていればこのファイルの設定値に従って比較して、altseq 変換ファイルが設定されていない場合は、JIS8 単位コード体系で比較します。

### 指定値

#### xyy

変換対象の文字コード `xx` と、変換後の順序 `yy` を `xyy` の対で指定します。

指定のない文字コードについては、JIS8 単位コード体系順と同じとなります。

`xyy` は、複数指定できますが `xx` は重複して指定できません。`yy` は重複して指定できます。

`xyy` は最大 256 対指定できます。

文字コード (対) を複数指定する場合は、1 個以上の空白またはタブで区切ってください。

`xx`: 変換対象の文字コード。JIS8 単位コード中の順序を変更したい文字 (コード) を 16 進数の 2 桁 (00 ~ FF) で指定します。

`yy`: `xx` の変更後の順序。`xx` をどのような順序で扱うのか、16 進数の 2 桁 (00 ~ FF) で指定します。

### 注意事項

- `-altseq` パラメタは、比較項目および集約項目には有効なりません。
- altseq 変換ファイルと `-altseq` パラメタを両方指定した場合、altseq 変換ファイルの設定値のうち、`-altseq` パラメタで指定した値で置き換えます。`-altseq` パラメタで指定しなかった値は、altseq 変換ファイルの値が有効となります。

### 指定例

数字 0 ~ 9(0x30 ~ 0x39) を英字より大きい 0xF0 ~ 0xF9 の順序とする場合の指定

## 5. exsort コマンドのパラメタおよびオペランドの詳細

例を示します。

```
-altseq△code=30F0△31F1△32F2△33F3△34F4△35F5△36F6△37F7△38F8△39F9
```

(凡例)

△：半角スペース（空白）を示します。

## 5.16 -csvkey パラメタ

### 意味

CSV ファイルをソートまたはマージする場合のソートキー、マージキー、または集約キーを指定します。このパラメタは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、KBLS1213-E エラーとなります。また、`-key` パラメタと同時に指定できません。

### 形式

`-csvkey csv-key-definition`

- このパラメタに指定したセルに対応するデータが入力されなかった場合、ナル文字列として処理します。ナル文字列の扱いについては、「付録 D CSV ファイル内のナル文字列の扱い (SORT EE の機能)」を参照してください。
- 指定できるキーの個数と合計長を、環境ファイル「EXS\_CSVKEYNUMMAX」で指定できます。なお、キーの合計長は、入力データのエンコードによって異なります。環境ファイル「EXS\_CSVKEYNUMMAX」については「15.1.1(17) EXS\_CSVKEYNUMMAX」を参照してください。
- キーを複数指定した場合、キーの指定順によってソートまたはマージの優先順位が決まります。先に指定したキーほど優先順位が高く、後に指定したキーほど優先順位が低くなります。
- 入力セルデータの桁調整については、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- ソート機能では、重複レコードの入力順序保存機能 (`-cmdopt DUP`) を指定すると、重複キーのレコードを入力順に並べて出力できます。重複レコードの入力順序保存機能については、「5.5.1 重複レコードの入力順序保存機能 (DUP | NODUP オプション)」を参照してください。
- マージおよびコピー機能では、重複キーのレコードは入力順に並べて出力されません。

### 指定値

`csv-key-definition` の形式を次に示します。

#### csv-key-definition 1 形式

```
[r] key-attribute + cellpos -len [Δ[r] key-attribute +
cellpos -len ... ]
```

- 文字データのセルを定義します。
- `key-attribute` には、「j」、「c」、「d」、「w」、「u」、「v」または「q」を指定できます。指定できるキーの属性については、`key-attribute` に指定する値を示す表を参照してください。
- `[r]key-attribute + cellpos -len` は必ず続けて指定してください。それぞれの値の間に空白やタブを指定しないでください。
- `[r]key-attribute + cellpos -len` を複数指定する場合は、1 個以上の空白また

## 5. exsort コマンドのパラメタおよびオペランドの詳細

はタブで区切って指定します。

### csv-key-definition 2 形式

```
[r] key-attribute +cellpos -p [,s] [Δ[r] key-attribute +
cellpos -p [,s] … ]
```

- 数値データ（固定小数点数）のセルを定義します。
- *key-attribute* には、「z」だけを指定できます。
- データ中に、引用符による囲み、符号、通貨記号、桁修飾文字（コンマ、小数点、空白）が含まれる場合、これらの文字を除いた内部データ形式でキーを比較します。なお、セルデータが変更されることはありません。
- 数値データ（固定小数点数）のセルを定義する場合、環境ファイル「EXS\_ROUND」または `-inpf`、`-catinp`、`-splitnp` パラメタの `round` オペランドで、丸め方式（四捨五入、切り捨て、切り上げのどれか）を指定します。この場合、整数部または小数部は指定に従って処理されます。`EXS_ROUND` および `round` オペランドの両方の指定を省略している場合、切り捨てられて処理されます。詳細は、「5.8.11 round オペランド」を参照してください。
- `[r]key-attribute + cellpos - p[,s]` は必ず続けて指定してください。それぞれの値の間に空白やタブを指定しないでください。
- `[r]key-attribute + cellpos - p[,s]` を複数指定する場合は、1 個以上の空白またはタブで区切って指定します。

### csv-key-definition 3 形式

```
[r] key-attribute +cellpos -d,dateform [Δ[r] key-attribute
+cellpos -d,dateform … ]
```

- 日付データのセルを定義します。
- *key-attribute* には、「g」だけを指定できます。
- `[r]key-attribute + cellnum` は必ず続けて指定してください。それぞれの値の間に空白やタブを指定しないでください。
- `[r]key-attribute + cellnum` を複数指定する場合は、1 個以上の空白またはタブで区切って指定します。

### csv-key-definition 4 形式

```
[r] key-attribute +cellpos -datedigit[, timedigit] [Δ[r]
key-attribute +cellpos -datedigit[, timedigit]… ]
```

- 日付時刻データのセルを定義します。
- *key-attribute* には、「h」だけを指定できます。
- `[r]key-attribute + cellpos - datedigit [, timedigit]` は必ず続けて指定してください。それぞれの値の間に空白やタブを指定しないでください。
- `[r]key-attribute + cellpos - datedigit[, timedigit]` を複数指定する場合は、1 個以上の空白またはタブで区切って指定します。

*csv-key-definition* に指定するそれぞれの値について次に示します。

[r]

キーのソート順序を指定します。

r オプション	意味
指定なし	キーの昇順でソートまたはマージします。
指定あり	キーの降順でソートまたはマージします。

- 降順でソートまたはマージする場合は、「rc」のようにつなげて指定します。
- 昇順か降順かは、キーごとに指定します。キーを複数指定して、昇順と降順を混在させたソートまたはマージもできます。

**key-attribute**

キーの属性を指定します。

key-attribute に指定する値	意味	指定できる csv-key-definition の形式
j	キーが文字 (JIS8 単位コード※ <sup>1</sup> ) であることを指定します。	形式 1
c	キーが文字であることを指定します。※ <sup>3</sup>	形式 1
q	キーが文字 (文字順序変更機能※ <sup>2</sup> で指定した文字) であることを指定します。※ <sup>3</sup>	形式 1
d	キーが文字 (左分離符号付き数字) であることを指定します。※ <sup>5</sup>	形式 1
z	キーが数値 (固定小数点数) であることを指定します。※ <sup>5</sup>	形式 2
g	キーが日付であることを指定します。※ <sup>5</sup>	形式 3
h	キーが日付時刻 (数字) であることを指定します。※ <sup>7</sup>	形式 4
w	キーが文字 (ワイド文字※ <sup>4</sup> , ※ <sup>6</sup> ) であることを指定します。	形式 1
u	キーが文字 (マルチバイト文字※ <sup>6</sup> , ※ <sup>8</sup> , ※ <sup>9</sup> ) であることを指定します。	形式 1
v	キーが文字 (マルチバイト文字※ <sup>6</sup> , ※ <sup>8</sup> , ※ <sup>9</sup> ) であることを指定します (Unicode の IVS に対応した属性です)。	形式 1

## 注※ 1

ASCII コード (0x00 ~ 0x7F) を含む, JIS8 単位コード (0x00 ~ 0xFF) を意味します。

入力データに日本語などのマルチバイト文字が含まれている場合, マルチバイ

## 5. exsort コマンドのパラメタおよびオペランドの詳細

ト文字として扱わず、シングルバイト文字として扱います。

EBCDIK コードの体系に従ってキーを比較する「EBCDIK コード比較オプション機能」を使用できます。EBCDIK コード比較オプション機能については、「5.5.3 EBCDIK コード比較オプション (EBCJ, EBCE オプション)」を参照してください。

### 注※ 2

文字順序変更機能については、「5.15 -altseq パラメタ」を参照してください。

### 注※ 3

入力データのエンコードの種別には対応しません。入力データの文字を構成するバイト長を意識しないで、常に 1 バイト長のシングルバイト文字として扱います。ASCII コードおよび JIS8 単位コード向けの属性です。

### 注※ 4

入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定した場合、1 文字 2 バイトのワイド文字として扱います。また、サロゲートペア文字は 2 文字として扱います。

入力データのエンコードに「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定した場合、1 文字 4 バイトのワイド文字として扱います。文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。

### 注※ 5

入力データのエンコードで解釈します。

「ASCII」, 「SJIS」, 「EUC-JP」, 「EUC-HJ」または「UTF-8」を指定した場合は 1 バイト, 「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」を指定した場合は 2 バイト, 「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」を指定した場合は 4 バイトの長さで文字を扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。

### 注※ 6

キー範囲の入力データがすべて ASCII 対応コード (英数字) である場合は、属性コード「c」を指定した方が処理効率が向上します。属性コード「c」は 1 文字 1 バイトで扱いますので、エンコードに対応したバイト長を指定してください。

ASCII 対応コードについては、「付録 H Unicode 操作文字コード一覧 (SORT



EE)」を参照してください。

1 文字の長さについては、「付録 I.2(1) 文字の長さ」を参照してください。

文字属性コードの特徴については、「付録 I.2(2) 文字属性コードの特徴」を参照してください。

#### 注※ 7

キー範囲の入力データが次の条件すべてに該当する場合は、属性コード「c」を指定した方が処理効率は向上します。

- 桁数がそろっている。
- 時刻が 24 時間制で表記されている。
- 日付、時刻またがりのデータがない。

属性コード「c」は 1 文字 1 バイトで扱いますので、エンコードに対応したバイト長を指定してください。

1 文字の長さについては、「付録 I.2(1) 文字の長さ」を参照してください。

文字属性コードの特徴については、「付録 I.2(2) 文字属性コードの特徴」を参照してください。

#### 注※ 8

属性コード「u」と「v」は、入力データのエンコードで解釈します。1 文字の長さは可変長として扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。また、マルチバイト文字の扱いについては、「付録 I.2 文字列データの扱い」を参照してください。

#### 注※ 9

入力データのエンコードに「UTF-8」、「UTF-16LE」、「UTF-16BE」、「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE, UTF-16BE, UTF-32LE または UTF-32BE)」のどれかを指定した場合、Unicode の IVS (Ideographic Variation Sequence) を次のとおり解釈します。Unicode の IVS については、「付録 I.2(6) IVS 対応」を参照してください。

- 属性コード「u」は、UCS コードが一致する文字であれば、Unicode の IVS の異体字であっても同じ文字として扱います。
- 属性コード「v」は、UCS コードが一致する文字であっても、VS (Variation Selector, 異体字セレクタ) が異なる場合は、異なる文字として扱います。

#### + cellpos

レコード内の処理対象セル (列) の位置を、番号で指定します。

- レコードの先頭セル番号を「1」、最大セル番号を「n」とし、1～nの番号を指定します。最大セル番号は、16384 番となります。

#### – len

## 5. exsort コマンドのパラメタおよびオペランドの詳細

key-attribute に「j」、「c」、「q」、「w」、「u」、「v」または「d」を指定した場合に、文字データの最大バイト長を、10進数で指定します。

- 属性「d」は最大桁数を指定します。
- キーの長さを次の表に示す範囲で指定します。

key-attribute に指定する値	指定できる長さ（単位）
j, c, q, w ※1	<ul style="list-style-type: none"> <li>• 入力データのエンコードが「ASCII」、「SJIS」、「EUC-JP」、「EUC-HJ」または「UTF-8」の場合 1～4,096（バイト）</li> <li>• 入力データのエンコードが「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」の場合※2 2～8,192（バイト）</li> <li>• 入力データのエンコードが「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」の場合※3 4～16,384（バイト）</li> </ul>
u, v	1～4,096（文字数）※5
d	1～256（桁）※4

### 注※1

入力データのエンコードの種別によって、マルチバイト文字の長さは異なります。詳細は「付録 I.2 文字列データの扱い」を参照してください。

### 注※2

入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-16LE または UTF-16BE）」を指定した場合、長さには「2の倍数」を指定してください。

2の倍数以外を指定すると、KBLS1207-E メッセージが表示されてエラーになります。

### 注※3

入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-32LE または UTF-32BE）」を指定した場合、長さには「4の倍数」を指定してください。

4の倍数以外を指定すると、KBLS1207-E メッセージが表示されてエラーになります。

### 注※4

「バイト長」ではなく数値の「桁数」を指定してください。「UCS-2」、「UCS-4」、「UTF-16」および「UTF-32」の数値データの扱いについては、

「付録 I.3 数値データの扱い」を参照してください。

#### 注※ 5

「バイト長」ではなく「文字数」を指定してください。UTF-16 のサロゲートペア文字は 1 文字として扱います。また、Unicode の VS (Variation Selector, 異体字セレクタ) は文字数に数えません。マルチバイト文字の扱いについては、「付録 I.2 文字列データの扱い」を参照してください。

- 実際に入力したデータ長に過不足がある場合は調整を行います。詳細は、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- 入力データを引用符で囲んでいる場合は、引用符を除いた内部データ形式でキーを比較します。なお、セルデータが変更されることはありません。
- 入力データの先頭が引用符の場合だけ、対の引用符で囲んだ範囲がエスケープされます。詳細は、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- 入力データの先頭および末尾の引用符は、有効バイト長に含まれません。また、エスケープされた引用符は、2 文字で 1 文字となります。
- マルチバイト文字 (日本語) の途中で最大長を超えた場合、超えた部分のマルチバイト文字以降を切り捨て、代わりに 1 バイトの空白で埋め字をします。なお、切り捨てはキー属性「c」および「w」の場合だけです。キー属性「j」、「q」、および「d」は日本語として扱われません。詳細は、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。

#### – p[, s]

key-attribute に「z」を指定した場合に、数値データの有効桁を 10 進数で指定します。

- **p** (precision : 精度) は、数値データの総桁数を表します。**p** から **s** を引いた桁が整数部の有効桁となります。
- **p** に指定できる桁の範囲は、1 ~ 38 桁です。
- **s** (scale : 位) は、数値データの小数部 (小数点以下) の有効桁を表します。
- **s** に指定できる桁の範囲は、0 ~ **p** 桁です (**p** 以下の桁を指定  $0 \leq s \leq p$ )。  $p = s$  を指定した場合は、整数部の有効桁は 0 (整数部なし) となります。
- **s** を省略した場合は、小数部の有効桁は 0 (小数部なし) を仮定します。データ中に小数点を含まない場合は、**s** の指定は必要ありません。
- データ中には、0 ~ 9 の数字以外に、データを囲んでいる引用符、符号、通貨記号、および桁修飾文字 (コンマ、小数点、空白) を含むことができ、これらの文字を除いた内部データ形式でキーを比較します。また、それぞれの除外文字は位置規則があります。違反するデータを検出した場合は、データ形式が不正と見なされ「KBLS1211-E キーにエラーデータを検出しました。」というメッセージが表示されて、エラーになります。メッセージが表示されたときの原因と対処方法の詳細は、「付録 K メッセージ」を参照してください。データ形式が不正な場合でもエラーにしたいときは、コマンド

## 5. exsort コマンドのパラメタおよびオペランドの詳細

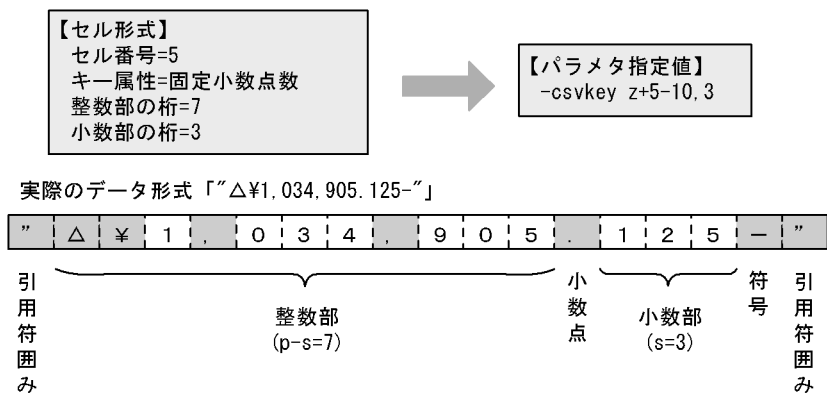
オプション「`-cmdopt KEYNOCHK`」を指定してください。

除外文字の位置規則については、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。

- データ中の符号、通貨記号、桁修飾文字 (コンマ、小数点、空白)、およびデータを囲んでいる引用符は、有効桁に含まれません。有効桁の対象は、0 ~ 9 の数字だけです。
- データ中に小数点がある場合は、小数点位置を合わせて入力します。データ中に小数点がない場合は、データの末尾に小数点を仮定して入力します。詳細は、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- 実際に入力したデータの桁に過不足がある場合は桁調整を行います。詳細は、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- すべて空白のデータは、0 (ゼロ) として入力します。
- 環境ファイル「`EXS_ROUND`」または `-inpfile`、`-catinp`、`-splitinp` パラメタの `round` オペランドで、指定した丸め方式 (四捨五入、切り捨て、切り上げのどれか) に従って、整数部または小数部を丸め処理します。`EXS_ROUND` および `round` オペランドの両方の指定を省略している場合、切り捨てられます。詳細は、「5.8.11 round オペランド」を参照してください。
- 引用符、符号、通貨記号、桁修飾文字の除外、および丸め処理によって、入力セルデータが変更されることはありません。

数値データの指定例を次の図に示します。

図 5-10 数値データの指定例



引用符囲み、符号、通貨記号、桁修飾文字 (コンマ、小数点、空白) を除いた桁数を指定する。

(凡例)

- ” : 引用符を示す。
- △ : 空白を示す。
- ¥ : 通貨記号を示す。
- ,
- .
- .

- *d*, *dateform*

key-attribute に「g」を指定した場合に、日付データの有効桁、および書式番号を指定します。

- *d* (digit : 桁) は、日付データの有効桁を表します。
- *d* に指定できる桁の範囲は、2 ~ 8 桁です。なお、書式 3 ~ 4 の 2 桁年号に対して 7 桁以上を指定した場合、6 桁を仮定します。
- データの左端から指定された書式に適用します。有効桁以降のデータは無視されます。
- 実際に入力したデータの桁に過不足がある場合は桁調整を行います。詳細は、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- データ中には、データを囲んでいる引用符と年月日を区切る桁修飾文字を含むことができます。その場合、これらの文字を除いた内部データ形式でキーを比較します。日付データの形式については、「付録 B CSV ファイルのデータの扱い (SORT EE の機能)」を参照してください。
- データの先頭と末尾を囲んでいる引用符、およびデータ中の桁修飾文字は、有効桁に含まれません。
- 入力データの年号部分が数字の 0 ~ 9 以外を指定した場合、データ形式が不正と見なされ「KBL1211-E キーにエラーデータを検出しました。」というメッセージが表示されてエラーになります。メッセージが表示されたときの原因と対処方法の詳細は、「付録 K メッセージ」を参照してください。データ形式が不正な場合でもエラーにしたいときは、コマンドオプション「-cmdopt DATENOCHK」を指定してください。なお、年号以外のデータ (月、日) は、形式チェックしません。
- 引用符、および桁修飾文字の除外によって、入力セルデータが変更されることはありません。
- 桁修飾文字は、「ハイフン (-)」と「スラント (/)」の 2 種類です。
- 環境変数「EXSORTGOPTION (64 ビット版は「EXSORT64GOPTION)」」を指定すると、「年月日 {T | 空白} 時分秒」形式の入力データを扱えます。また、桁修飾文字は、「ハイフン」、「スラント」および「ピリオド (.)」の 3 種類に変更されます。環境変数については、「14.1.1(29) EXSORTGOPTION」または「14.1.2(29) EXSORT64GOPTION」を参照してください。
- *dateform* は日付データの書式を表します。日付データの書式を次に示します。

<i>dateform</i> (書式)	データ形式	書式	<i>d</i> (桁)
1	前置 4 桁年号	YYYY/MM/DD YYYY-MM-DD YYYY.MM.DD ※ YYYYMMDD YYYY	2 ~ 8

5. exsort コマンドのパラメタおよびオペランドの詳細

dateform (書式)	データ形式	書式	d (桁)
2	後置 4 桁年号	MM/DD/YYYY MM-DD-YYYY MM.DD.YYYY※ MMDDYYYY	2 ~ 8
3	前置 2 桁年号	YY/MM/DD YY-MM-DD YY.MM.DD※ YYMMDD YY	2 ~ 6
4	後置 2 桁年号	MM/DD/YY MM-DD-YY MM.DD.YY※ MMDDYY	2 ~ 6

(凡例)

YYYY : 西暦 4 桁年号

YY : 西暦下 2 桁年号

MM : 月

DD : 日

桁修飾文字で年月日を区切っている場合は、MM および DD 部分のデータは 1 桁でもかまいません。

(例)

「YYYY-MM-DD 」 …… 2008-7-7

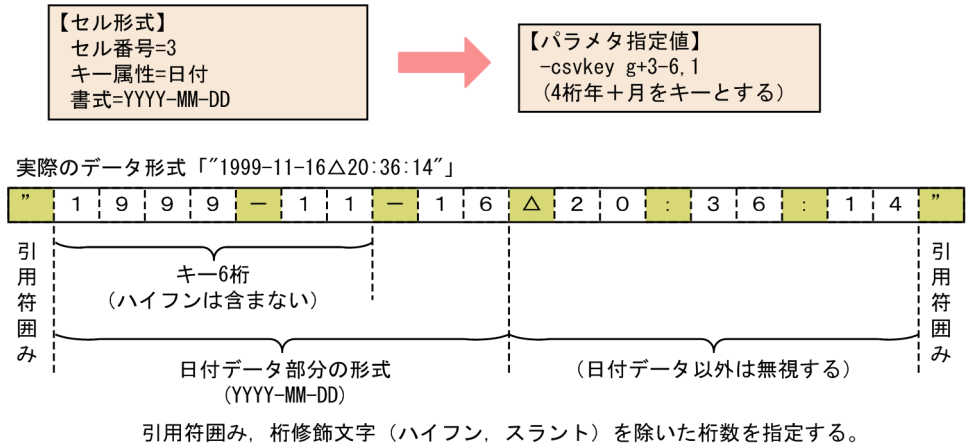
「MM/DD/YY 」 …… 1/1/07

注※

環境変数「EXSORTGOPTION (64 ビット版は「EXSORT64GOPTION)」の指定がある場合、ピリオドを桁修飾文字として扱います。

日付データの指定例を次の図に示します。

図 5-11 日付データの指定例



(凡例)

” : 引用符囲みを示す。

△ : 空白を示す。

- *datedigit* [, *timedigit*]

key-attribute に「h」を指定した場合に、日付および時刻データの有効桁を 10 進数で指定します。

- *datedigit* は、日付データの有効桁を表します。
- *timedigit* は、時刻データの有効桁を表します。
- *datedigit* に指定できる桁は 0, 4, 6 および 8 桁, *timedigit* に指定できる桁は 0, 2, 4, 6 および 9 桁のどれかです。それ以外を指定した場合は、KBL51217-E メッセージが表示されてエラーになります。また、*datedigit* と *timedigit* 両方に 0 を指定した場合もエラーになります。
- 入力データを項目の区切り文字で区切り、それぞれのデータの左端から指定された桁数の書式に適用します。
- 入力したデータの桁に過不足がある場合は、項目によって桁調整が実施されるか、またはエラーとなります。端数処理の詳細については、「付録 J 日付時刻データの形式 (SORT EE の機能)」を参照してください。
- データ中には、データの先頭と末尾を囲んでいる引用符、項目区切り文字 ('T' または空白)、および桁修飾文字 (スラント、ハイフン、コロン、およびピリオド) を含められ、これらの文字を除いた内部データ形式でキーを比較します。日付時刻データの形式については、「3.2 キー属性と形式」を参照してください。
- データの先頭と末尾を囲んでいる引用符、項目区切り文字 ('T' または空白)、桁修飾文字 (スラント、ハイフン、コロン、およびピリオド)、および 12 時間制表記は、有効桁に含まれません。
- 引用符、項目の区切り文字、および桁修飾文字を取り除くことによって、入力セルデータが変更されません。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- 入力データが数字の 0～9 以外を指定した場合、データ形式が不正と見なされ、KBL51211-E メッセージが表示されてエラーになります。また、年の桁数が 4 桁未満、または入力データが指定可能な範囲を超えた場合もエラーとなります。メッセージが表示されたときの原因と対処方法の詳細は、「付録 K メッセージ」を参照してください。データ形式が不正な場合でもエラーにしたいときは、コマンドオプション「-cmdopt DATETIMENCHK」を指定してください。
- 日付時刻として正しくないデータの場合も、扱える値の範囲内であればエラーにしないで日付、時刻またがりデータとして扱います。
- 指定した桁数に応じた日付時刻キーの書式を次に示します。

項目	桁数	書式
日付	4	YYYY
	6	YYYYMM YYYY-MM YYYY/MM YYYY.MM
	8	YYYYMMDD YYYY-MM-DD YYYY/MM/DD YYYY.MM.DD
時刻	2	hh
	4	hhmm hh:mm hh.mm
	6	hhmmss hh:mm:ss hh:mm.ss hh.mm.ss
	9	hhmmssfff hh:mm:ss:fff hh:mm:ss.fff hh.mm.ss.fff
12 時間制表記	2	AM am PM pm
	4	A.M. a.m. P.M. p.m.

書式の説明を次に示します。



書式		説明
日付	YYYY	西暦年 4 桁
	MM	月※
	DD	日※
時刻	hh	時※
	mm	分※
	ss	秒※
	fff	秒の小数部 (ミリ秒) ※
12 時間制表記	AM am A.M. a.m. PM pm P.M. p.m.	時 (hh) を 12 時間制で表記した場合に指定
項目区切り文字 桁修飾文字	/	日付の桁修飾文字
	-	日付の桁修飾文字
	:	時刻の桁修飾文字
	.	日付, 時刻の桁修飾文字
	空白	日付項目と時刻項目の区切り文字 時刻項目と 12 時間制表記の区切り文字
	T	日付項目と時刻項目の区切り文字

## 注※

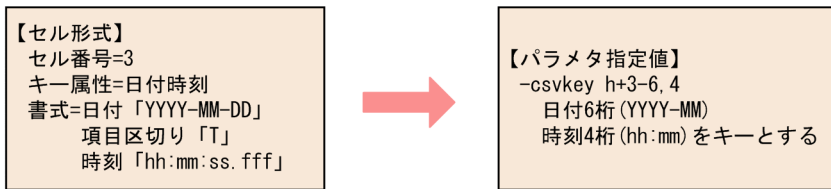
桁修飾文字 (スラント, ハイフン, コロン, およびピリオド) で年月日時分秒を区切っている場合は, 月日時分秒の各データは 1 桁でもかまいません。

(例) 日付「YYYY-MM-DD」, 項目区切り「T」, 時刻「hh:mm:ss.fff」の場合  
2013-7-27T13:10:80.095

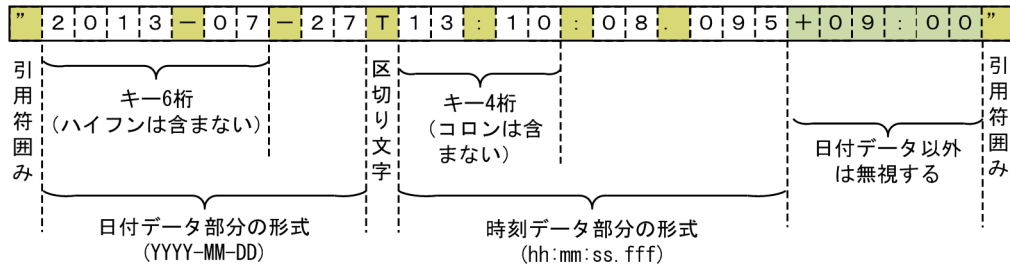
日付時刻データの指定例を次の図に示します。

5. exsort コマンドのパラメタおよびオペランドの詳細

図 5-12 日付時刻データの指定例



実際のデータ形式 「"2013-07-27T13:10:08.095+09:00"」



引用符囲み、項目区切り文字（'T'）、および桁修飾文字（ハイフン、スラント、コロン、およびピリオド）を除いた桁数を指定する。

（凡例）

- " : 引用符囲みを示す。
- T : 区切り文字'T'を示す。

## 5.17 -csvsum パラメタ

### 意味

CSV ファイルの集約項目を指定します。

このパラメタは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、KBL51213-E エラーとなります。

### 形式

```
-csvsum [△csv-field-definition [△csv-field-definition ... ] ]
```

- **-csvsum** パラメタを指定する場合は、**-csvkey** パラメタを必ず指定してください。
- **-function copy** と組み合わせて指定する場合、入力ファイルはあらかじめ集約キーでソートしておく必要があります。ソートしていない場合は、集約の結果は保証されません。
- **-function copy** と組み合わせる場合、入力ファイルは 1 つだけ指定してください。
- このパラメタに指定したセルに対応するデータが入力されなかった場合、ナル文字列として処理します。ナル文字列の扱いについては、「付録 D CSV ファイル内のナル文字列の扱い (SORT EE の機能)」を参照してください。
- **-csvkey** パラメタに指定したセル番号を、**-csvsum** パラメタに指定しないでください。
- 環境ファイル「EXS\_CSVSUMNUMMAX」の「最大個数の倍数」を指定していない場合、指定できる集約項目の個数は 0 ～ 128 個です。集約項目の最大個数は「128 × n 個」の単位で、1,280 個 (128 × 10) まで拡張できます。129 個以上の集約項目を指定したい場合は、環境ファイル「EXS\_CSVSUMNUMMAX」の「最大個数の倍数」に、「128 × n 個」の n の値を指定してください。
- 環境ファイル「EXS\_CSVSUMNUMMAX」の「合計長の上限の倍数」を指定していない場合、指定できる集約項目の合計長は 4,096 バイトです。集約項目の合計長は「4,096 × m バイト」の単位で、65,536 バイト (4,096 × 16) まで拡張できます。4,097 バイト以上の合計長を指定したい場合は、環境ファイル「EXS\_CSVSUMNUMMAX」の「合計長の上限の倍数」に、「4,096 × m バイト」の m 値を指定してください。
- 集約項目を指定しない場合 (集約項目の個数が 0 個)、同じキーを持つ連続したレコードがあるときは、最初に配置されているレコードだけを抽出します。これを無条件レコード抽出機能といいます。

### csv-field-definition の形式

```
field-attribute+cellpos-p[, s] [△field-attribute+cellpos-p[, s] ... ]
```

- 数値データ (固定小数点数) のセルを定義します。
- *field-attribute* には、「z」だけを指定できます。
- *field-attribute + cellpos - p*[, *s*] は必ず続けて指定してください。それぞれの値の間に空白やタブを指定しないでください。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- *field-attribute + cellpos - p*[, *s*] を複数指定する場合は、1 個以上の空白またはタブで区切って指定します。

### 指定値

#### *field-attribute*

集約項目の属性を指定します。

*z* : 集約項目が固定小数点数であることを指定します。

#### + *cellpos*

レコード内の処理対象セル (列) の位置を、番号で指定します。

レコードの先頭セル番号を「1」、最大セル番号を「*n*」とし、1 ~ *n* までの番号を指定します。最大セル番号は、16,384 番となります。

#### - *p*[, *s*]

固定小数点データの有効桁を 10 進数で指定します。

- *p* (precision : 精度) は、固定小数点データの総桁数を表します。*p* から *s* を引いた桁が整数部の有効桁となります。
- *p* に指定できる桁の範囲は、1 ~ 38 桁です。
- *s* (scale : 位) は、固定小数点データの小数部 (小数点以下) の有効桁を表します。
- *s* に指定できる桁の範囲は、0 ~ *p* 桁です ( $0 \leq s \leq p$ )。
- *s* を省略した場合は、0 (小数部なし) を仮定します。
- データ中の引用符囲み、符号、通貨記号、桁修飾文字 (コンマ、小数点、空白) は、有効桁に含みません。

### 注意事項

- すべて空白文字の集約項目は、0 (ゼロ) として扱います。ただし、集約した範囲 (オーバーフローが発生する前まで) のすべての集約項目が空白だった場合は、集約結果も空白で出力します。

#### 【例】

-csvsum z+2-4 を指定した場合 (2 セル目を有効桁 4 桁 (小数以下 0 桁) で集約項目として指定)

(入力データ)		(集約結果)		
集約キー	集約項目	集約キー	集約項目	
A	△△△△	A	0	0+0=0
A	0	B	10000	入力データ桁あふれ (オーバーフロー)
B	10000	B	△△△△	オーバーフロー後、すべて空白
B	△△△△	C	△△△△	すべて空白
C	△△△△			
C	△△△△			

#### (凡例)

△ : 空白

- ナル文字列（空のセル）の集約項目は、0（ゼロ）として扱います。ただし、集約した範囲（オーバーフローが発生する前まで）のすべての集約項目がナル文字列だった場合は、集約結果もナル文字列で出力します。詳細は、「付録 D CSV ファイル内のナル文字列の扱い（SORT EE の機能）」を参照してください。
- ナル文字列を集約対象としたりたくない場合は、コマンドオプション「SUMNONUL」を指定してください。SUMNONUL オプションを指定すると、ナル文字列は集約されずに 1 個のレコードで出力します。SUMNONUL オプションについては、「5.5.19 ナル文字列の集約オプション（SUMNULKEY, SUMNONULL オプション）」を参照してください。
- ナル文字列（空のセル）の集約キーは同値として集約します。ナル文字列を同値として扱いたくない場合は、コマンドオプション「SUMKEYYNUL」を指定してください。SUMKEYYNUL オプションを指定すると、ナル文字列キーの集約項目は集約されず、1 個のレコードで出力されます。SUMKEYYNUL オプションについては、「5.5.19 ナル文字列の集約オプション（SUMNULKEY, SUMNONULL オプション）」を参照してください。
- 入力データの桁あふれはオーバーフローと判定されます。例えば、有効桁数が 5 桁、小数部 2 桁と指定した場合、最小値は「-999.99」、最大値は「+999.99」です。この指定に対して、入力データが「1000」や「-999.999」など、整数部または小数部が桁あふれしていればオーバーフローとなります。
- 入力データが桁あふれしていなければ、丸め処理をします。丸め処理の結果、整数部が桁あふれした場合はオーバーフローとなります。丸め方法は、環境ファイルの「EXS\_ROUND」または `-inpf`、`-catinp`、`-splitinp` パラメタの `round` オペランドに指定した方法を適用します。丸め方法の詳細については、「5.8.11 round オペランド」を参照してください。
- 指定された有効桁よりも大きい桁の入力データを検出した場合、KBLS1012-I インフォメーションメッセージを出力して正常終了します。KBLS1012-I が出力された場合は、`-csvsum` に指定した有効桁を見直してください。
- 入力データの丸め処理でオーバーフローを検出した場合、KBLS1013-I インフォメーションメッセージを出力して正常終了します。KBLS1013-I が出力された場合は、`-csvsum` に指定した有効桁と、`round` オペランドに指定した丸め方式を見直してください。
- 集約した結果のセルの書式は、次に示す書式に合わせて出力します。

[通貨記号] [符号] 数字 <sup>*</sup> [小数点] [数字] [符号]
---

注※ 整数部の有効桁によっては、桁区切りのコンマが含まれます。

書式	説明
有効桁	指定された有効桁に合わせてます。 <ul style="list-style-type: none"> <li>• 小数部の桁を指定している場合は、小数点も付加されます。小数部の有効桁が指定されていない場合（s=0）は、整数部だけです。</li> <li>• 整数部の有効桁が指定されていない場合（p・s=0）は、整数部は必ず「0.」とします。</li> </ul>

## 5. exsort コマンドのパラメタおよびオペランドの詳細

書式	説明
通貨記号	集約した範囲（オーバーフローが発生する前まで）のすべての集約項目に通貨記号が含まれている場合、通貨記号を付加します。通貨記号は、左端に付加されます。
符号位置	集約した範囲（オーバーフローが発生する前まで）の集約項目のうち、最初の符号付きの位置に合わせます。 <ul style="list-style-type: none"> <li>最初の符号付きの符号位置が左側の場合、符号の位置は整数部の左側です。</li> <li>最初の符号付きの符号位置が右側の場合、符号の位置は小数部の右側です。</li> </ul>
符号	集約した範囲（オーバーフローが発生する前まで）の集約項目のうち、「+」の符号が含まれている集約項目が 1 個でもあれば、正值の集約結果に「+」を付加します。「+」の符号を含む集約項目が 1 個もなければ、符号は付加しません。また、集約結果が負値の場合は、「-」を付加します。
桁区切り	集約した範囲（オーバーフローが発生する前まで）の集約項目のうち、桁区切り（コンマ）が含まれている集約項目が 1 個でもあれば、整数部に桁区切りを付加します。桁区切りは、3 桁ごとにコンマを付加します。桁区切り（コンマ）を含む集約項目が 1 個もなければ、桁区切りは付加しません。

- 集約した結果を引用符で囲んだ形式で出力したい場合は、`-outfile`、`-catout` および `-splitout` パラメタの `quottype` オペランドで引用符を付加する指定をしてください。`quottype` オペランドについては、「5.10.8 `quottype` オペランド」を参照してください。
- 同じキーを持つ連続したレコードのうち、最初に配置されているレコードを基に集約レコードを作成して出力します。`SUMNONULL` オプションの指定がなく、かつ最初に配置されているレコードに集約項目のセルがない場合、集約項目のセルが付加されます。また、元のレコードの最後のセルと集約項目のセルの間に間隔が生じる場合には、ナル文字列のセルが作成されます。

### 【例 1】`SUMNONULL` オプション指定なし

集約キー : `c+1-5` (セル 1)

集約項目 : `z+4-4` (セル 4)

(入力レコード)		(出力レコード)	
レコード1	BYH92, 120318, 840, 5000	BYH92, 120318, 840, 7300	レコード1※1
レコード2	BYH92, 110211, 250, 2300	JST04, 120915, 400, 6000	レコード2※2
レコード3	JST04, 120915, 400	NFE67, 110227, 7800	レコード3※3
レコード4	JST04, 111204, 100, 6000		
レコード5	NFE67, 110227		
レコード6	NFE67, 120116, 050, 3000		
レコード7	NFE67, 110826, 800, 4800		

(凡例)

— (下線) : 追加セル

注※1

入力レコード1の4番目のセルを、入力レコード1と2の集約項目の加算結果 (7300) に置き換える。

注※2

入力レコード3と4の集約項目の加算結果 (6000) を、入力レコード3の4番目のセルとして追加する。

注※3

入力レコード5から7の集約項目の加算結果 (7800) を、入力レコード5の4番目のセルとして追加する。3番目のセルはナル文字列とする。

#### 【例2】SUMNONULL オプション指定あり

集約キー : c+1-5 (セル 1)

集約項目 : z+4-4 (セル 4)

(入力レコード)		(出力レコード)	
レコード1	BYH92, 120318, 840, 5000	BYH92, 120318, 840, 7300	レコード1
レコード2	BYH92, 110211, 250, 2300	JST04, 120915, 400	レコード2
レコード3※	JST04, 120915, 400	JST04, 111204, 100, 6000	レコード3
レコード4	JST04, 111204, 100, 6000	NFE67, 110227	レコード4
レコード5※	NFE67, 110227	NFE67, 120116, 050, 7800	レコード5
レコード6	NFE67, 120116, 050, 3000		
レコード7	NFE67, 110826, 800, 4800		

(凡例)

— (下線) : 追加セル

注※

入力レコード3と5の集約項目はナル文字列 (空セル) のため加算対象外。

- 入力レコードの連続区切り文字オプション (INPCONSEPA) だけが指定されている場合、作成されるナル文字列は、引用符で囲んだ形式となります。出力レコードの連続区切り文字オプション (OUTCONSEPA) の指定がある、または INPCONSEPA と OUTCONSEPA の両方の指定がない場合は、ナル文字列を引用符で囲みません。

#### 関連パラメタ・オペランド

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- `-csvkey` パラメタ
- `-cmdopt` パラメタ `SUMNULKEY`, `SUMNONUL` オペランド
- `-cmdopt` パラメタ `INPCONSEPA`, `OUTCONSEPA` オペランド
- `-inpfileround` パラメタ `round` オペランド
- `-outfile` パラメタ `quottype` オペランド



## 5.18 -csvreorg パラメタ

---

### 意味

CSV ファイルの中間レコード編集フォーマットを指定します。

本パラメタは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、KBL51522-E エラーとなります。

編集フォーマットの詳細は、「5.8.15 csvreorg オペランド」を参照してください。

### 形式

```
-csvreorg△csv-reorganization-format [△  
csv-reorganization-format …]
```

指定できるフォーマットの個数は、1 ～ 128 個です。

## 5.19 -iosize パラメタ

### 意味

**exsort** コマンドで処理する入力ファイル、出力ファイル、ソート用一時ファイル、マージ用一時ファイル、およびファイル分割用一時ファイルに対する 1 回の I/O サイズを指定します。

### 形式 1

```
-iosize I/Oサイズ
```

このパラメタで指定した I/O サイズは、入力ファイルからの読み込み、出力ファイルへの書き出し、ソート用/マージ用一時ファイル、およびファイル分割用一時ファイルへの書き出しと読み込みに対して適用されます。

### 形式 2

```
-iosize [inpiosize=I/Oサイズ]
        [outiosize=I/Oサイズ]
        [sorttmpiosize=I/Oサイズ]
        [mergetmpiosize=I/Oサイズ]
        [splittmpiosize=I/Oサイズ]※
```

**inpiosize = I/O サイズ**

このオペランドで指定した I/O サイズは、入力ファイルからの読み込みに対して適用されます。

**outiosize = I/O サイズ**

このオペランドで指定した I/O サイズは、出力ファイルへの書き出しに対して適用されます。

**sorttmpiosize = I/O サイズ**

このオペランドで指定した I/O サイズは、ソート用一時ファイルへの書き出しに対して適用されます。

**mergetmpiosize = I/O サイズ**

このオペランドで指定した I/O サイズは、マージ用一時ファイルへの書き出しに対して適用されます。

**splittmpiosize = I/O サイズ**

このオペランドで指定した I/O サイズは、ファイル分割用一時ファイルへの書き出しと読み込みに対して適用されます。

オペランドを指定する場合は、1 個以上の空白で区切って指定します。パラメタファイルの 1 行に複数のオペランドを指定できます。

形式 1 の I/O サイズと同時に指定することはできません。同時に指定した場合はエラーとなり、KBL51105-E が出力されます。

- **exsort** コマンドが発行する 1 回の I/O サイズ (単位 : KB) を 10 進数で指定します。

I/O サイズの指定できる範囲を次の表に示します。

表 5-19 ファイルの種別と指定したパラメタに対するデフォルト I/O サイズ

ファイルの種別		運用 I/O	デフォルト I/O サイズ (単位: KB)	指定したパラメタ					
				-iosize	-iosize inpio size	-iosize outio size	-iosize sortmpio size	-iosize mergempi size	-iosize splittmpio size
入力ファイル	標準入力	RD	8	×	×	—	—	—	—
	上記以外		1,024	○※	○※	—	—	—	—
出力ファイル	標準出力	WR	8	×	—	×	—	—	—
	上記以外		1,024	○	—	○	—	—	—
ソート用一時ファイル		WR	1,024	○	—	—	○	—	—
マージ用一時ファイル		WR	1,024	○	—	—	—	○	—
ファイル分割用一時ファイル		RD, WR	1,024	○	—	—	—	—	○

凡例

RD : 読み込み WR : 書き出し

○ : 有効 × : 無効 (デフォルト I/O サイズが適用される) — : 該当しない

注※

使用する機能によって指定できる I/O サイズが異なります。I/O サイズの範囲については、次の表を参照してください。

5. exsort コマンドのパラメタおよびオペランドの詳細

表 5-20 使用する機能とファイル編成の種類に対して指定できる I/O サイズの範囲

使用する機能	ファイル編成の種類	-iosize または -iosize inpiosize に指定できる I/O サイズの範囲 (単位: KB)
ソート/コピー	<ul style="list-style-type: none"> <li>固定長順</li> <li>可変長順</li> <li>テキスト</li> <li>CSV</li> </ul>	8 ~ 32,768
マージ	<ul style="list-style-type: none"> <li>固定長順</li> <li>可変長順</li> </ul>	64 ~ 32,768
	<ul style="list-style-type: none"> <li>テキスト</li> <li>CSV</li> </ul>	8 ~ 32,768

- マージ機能 (-function merge 指定時) で固定長順ファイルまたは可変長順編成ファイルに対して指定できる範囲は、64 ~ 32,768 (KB) です。
- 標準入力ファイル、標準出力ファイルの場合、I/O サイズを変更できません。このパラメタを指定しても無視されます。標準入力ファイル、標準出力ファイルの I/O サイズは 8 (KB) に仮定されます。
- このパラメタを省略した場合、次の表に示す環境変数を指定しているときは、環境変数に指定した値が設定されます。環境変数を指定していない場合、1,024KB が設定されます。

表 5-21 環境変数の指定

モード	環境変数	対象 I/O-buff
32 ビット版	EXSORTIOSIZE	入力ファイル 出力ファイル 一時ファイル (ソート/マージ)
	EXSORTINPIOSIZE	入力ファイル
	EXSORTOUTIOSIZE	出力ファイル
	EXSORTSORTTMPIOSIZE	一時ファイル (ソート)
	EXSORTMERGETMPIOSIZE	一時ファイル (マージ)
	EXSORTSPLITTMPIOSIZE	一時ファイル (ファイル分割)

モード	環境変数	対象 I/O-buff
64 ビット版	EXSORT64IOSIZE	入力ファイル 出力ファイル 一時ファイル (ソート/マージ)
	EXSORT64INPIOSIZE	入力ファイル
	EXSORT64OUTIOSIZE	出力ファイル
	EXSORT64SORTTMPPIOSIZE	一時ファイル (ソート)
	EXSORT64MERGETMPPIOSIZE	一時ファイル (マージ)
	EXSORT64SPLITTMPPIOSIZE	一時ファイル (ファイル分割)

### 注意事項

このパラメタで指定した値で I/O 用バッファ領域を確保します。I/O 用バッファ領域は、ファイルごとに確保する領域のため、使用メモリの再見積もりが必要となります。

### 指定例

- I/O サイズに 1MB (1,024KB) を指定する。

(例) `-iosize 1024`

- 入力ファイルの I/O サイズに 128KB, 出力ファイルの I/O サイズに 32KB, ソート用一時ファイルの I/O サイズに 2,048KB, マージ用一時ファイルの I/O サイズに 1,024KB, ファイル分割用一時ファイルの I/O サイズに 512KB を指定する。

(例) `-iosize inpiosize=128 outiosize=32 sorttmpiosize=2048  
mergetmpiosize=1024 splittmpiosize=512`

## 5.20 -splitfile パラメタ

---

### 意味

均等分割情報を指定します。

- 指定された分割ファイル数を基に、均等なデータ量（レコード件数またはファイルサイズ）に分割して出力します。
- 分割ファイルの大きさは、入力ファイルのレコード件数またはファイルサイズを基に算出します。

分割ファイルのレコード件数 = 総レコード件数 ÷ 分割ファイル数

分割ファイルのファイルサイズ = 総ファイルサイズ ÷ 分割ファイル数

### 形式

```
-splitfile  
  △fileno=split-file-number  
  △method= {reccnt | filesize}
```

- -splitfile パラメタは、-splitdata パラメタおよび -splitkey パラメタと同時に指定できません。どれか一つだけ指定してください。

### 5.20.1 fileno オペランド

#### 意味

出力（分割）するファイル数を指定します。

#### 形式

```
fileno=split-file-number
```

#### 指定値

*split-file-number*

- 出力ファイル数を、1～5桁の10進数で指定します。
- 指定できる範囲は、2～99,999です。

### 5.20.2 method オペランド

#### 意味

分割基準を指定します。

#### 形式

```
method= {reccnt | filesize}
```

#### 指定値

reccnt

入力ファイルのレコード件数と、出力ファイル数を基に算出した平均レコード数でファイルを分割します（レコード数が均等となるように分割します）。

**filesize**

入力ファイルのファイルサイズと、出力ファイル数を基に算出した平均ファイルサイズでファイルを分割します（ファイルサイズが均等となるように分割します）。なお、Windows の場合、ファイルサイズは「ディスク上のサイズ」ではなく、UNIX と同様に、ファイルに格納されているレコードの合計サイズとなります。

**注意事項**

- 次のどちらかに該当する場合、分割できる最小単位で分割します。詳細については、指定例の 3., 4., および 5. を参照してください。該当する場合、指定したファイル数よりも少ないファイル数が作成されます。
  - **reccnt** 指定時、入力ファイルのレコード数が **fileno** オペランドで指定した値よりも少ない。
  - **filesize** 指定時、分割後のファイルサイズが入力ファイルのレコード長より小さい。
- レコードの途中で分割することはありません。
- **filesize** 指定時、レコードの途中が分割位置となったレコードは、次の分割ファイルに格納します。詳細については、指定例の 6. を参照してください。

**指定例**

1. レコード件数を基に、10 個のファイルに分割します。  
 入力ファイル  
     レコード件数 : 500 件  
 パラメタ指定例  
     **-splitfile fileno=10 method=reccnt**  
 分割後のファイル  
     ファイル数 : 10 個  
     各ファイルのレコード件数 : 50 件
2. ファイルサイズを基に、10 個のファイルに分割します。  
 入力ファイル  
     ファイルサイズ : 100MB  
 パラメタ指定例  
     **-splitfile fileno=10 method=filesize**  
 分割後のファイル  
     ファイル数 : 10 個  
     各ファイルのサイズ : 10MB
3. レコード件数を基に、20 個のファイルに分割します。  
 入力ファイル  
     レコード件数 : 5 件  
 パラメタ指定例  
     **-splitfile fileno=20 method=reccnt**  
 分割後のファイル

## 5. exsort コマンドのパラメタおよびオペランドの詳細

ファイル数 : 5 個

各ファイルのレコード件数 : 1 件

4. ファイルサイズを基に, 50 個のファイルに分割します。

入力ファイル

ファイルサイズ : 10KB

レコード長 : 1KB

パラメタ指定例

```
-splitfile fileno=50 method=filesize
```

分割後のファイル

ファイル数 : 10 個

各ファイルのサイズ : 1KB

5. ファイルサイズを基に, 5 個のファイルに分割します。

入力ファイル

ファイルサイズ : 10KB

レコード件数 : 6 件

レコード長 : 1 件目…1KB

レコード長 : 2 件目…1KB

レコード長 : 3 件目…1KB

レコード長 : 4 件目…1KB

レコード長 : 5 件目…3KB

レコード長 : 6 件目…3KB

パラメタ指定例

```
-splitfile fileno=5 method=filesize
```

分割後のファイル

ファイル数 : 4 個

1 個目のファイルのサイズ : 2KB

2 個目のファイルのサイズ : 2KB

3 個目のファイルのサイズ : 3KB

4 個目のファイルのサイズ : 3KB

6. ファイルサイズを基に, 2 個のファイルに分割します。

入力ファイル

ファイルサイズ : 10KB

レコード長 : 2KB

パラメタ指定例

```
-splitfile fileno=2 method=filesize
```

分割後のファイル

ファイル数 : 2 個

1 個目のファイルのサイズ : 4KB

2 個目のファイルのサイズ : 6KB

### 関連パラメタ



## 5. exsort コマンドのパラメタおよびオペランドの詳細

- -splitdata
- -splitkey
- -splitinp
- -splitout
- -splittmpdir

## 5.21 -splitdata パラメタ

### 意味

定数分割情報を指定します。

- 指定されたデータ量（レコード件数またはファイルサイズ）でファイルを分割して出力します。データ量は、`recno` オペランドまたは `spno` オペランドで指定します。
- 分割した出力ファイルの数は不定となります。  
 分割ファイル数 = 総レコード件数 ÷ `recno` オペランドで指定したレコード件数  
 分割ファイル数 = 総ファイルサイズ ÷ `spno` オペランドで指定したファイルサイズ
- ファイルを分割するときに、前のレコードと同じデータであれば、分割しないで前のレコードと同じファイルに出力できます。ファイル分割させたくないデータを「非分割データ」といい、`nosplitdata` オペランドまたは `nosplitcsvdata` オペランドで非分割データの位置とサイズを指定します。

### 形式

```
-splitdata
  △ {recno=split-record-number | spno=split-file-size [unit] }
  [△ {nosplitdata=no-split-data-definition
    [△no-split-data-definition...]
    | nosplitcsvdata=no-split-csv-data-definition
    [△no-split-csv-data-definition...]} ]
```

- `-splitdata` パラメタは、`-splitfile` パラメタおよび `-splitkey` パラメタと同時に指定できません。どれか一つだけ指定してください。

### 5.21.1 recno オペランド

#### 意味

出力ファイルに出力する最大レコード件数を指定します。

#### 形式

```
recno=split-record-number
```

#### 指定値

*split-record-number*

- 出力ファイルの最大レコード件数を、1 ~ 10 桁の 10 進数で指定します。
- 指定できる範囲は、1 ~ 1,073,741,824 です。

#### 注意事項

- 入力ファイルのレコード件数が分割レコード数より少ない場合、出力ファイルは一つです。
- `nosplitdata` オペランドまたは `nosplitcsvdata` オペランドを同時に指定した場合、分割したファイルのレコード件数は、`recno` オペランドで指定した値を超えることがあります。

## 5.21.2 spno オペランド

### 意味

出力ファイルの最大出力ファイルサイズを指定します。

### 形式

`spno=split-file-size [unit]`

### 指定値

#### *split-file-size*

- 出力ファイルの最大ファイルサイズを、1～7桁の10進数で指定します。
- 指定できる範囲は、1～2,048,000です。

#### *unit*

*split-file-size*の単位を指定します。出力ファイルサイズの指定範囲を次に示します。なお、単位を省略した場合は、Mが設定されます。

unit 指定	指定範囲
省略	1～2,048,000
M	1M～2,048,000M
G	1G～2,000G

### (凡例)

M：単位をMBにします

G：単位をGBにします

### 注意事項

- 入力ファイルサイズが `spno` オペランドで指定したサイズより小さい場合、出力ファイルは一つです。
- 一つのレコードが、二つの出力ファイルにわたって出力されることはありません。
- `nosplitdata` オペランドまたは `nosplitcsvdata` オペランドを同時に指定した場合、分割したファイルのサイズは、`spno` オペランドで指定した値を超えることがあります。

## 5.21.3 nosplitdata オペランド

### 意味

非分割データを指定します。

### 形式

`nosplitdata=no-split-data-definition`  
`[△no-split-data-definition…]`

### 指定値

*no-split-data-definition* オペランド形式 1

5. exsort コマンドのパラメタおよびオペランドの詳細

*data-attribute + pos - len*

*no-split-data-definision* オペランド形式 2

*data-attribute + pos - maskbit*

*no-split-data-definition* に指定するそれぞれの値について次に示します。

**data-attribute**

非分割データの属性を指定します。

<i>data-attribute</i> に指定する値	意味	<i>no-split-data-definision</i> 形式
j	データが文字 (JIS8 単位コード※ <sup>1</sup> ) であることを指定します。	形式 1
c	データが文字 (英数字, 漢字) であることを指定します。※ <sup>3</sup>	形式 1
n	データが符号付き小数点 2 進数であることを指定します。※ <sup>9</sup> , ※ <sup>10</sup>	形式 1
m	データが符号なし 2 進数 (0 から始まる整数) であることを指定します。※ <sup>9</sup> , ※ <sup>10</sup>	形式 1
a	データが符号なし 2 進数 (0 から始まる整数) であることを指定します※ <sup>9</sup> , ※ <sup>10</sup> (マスクビットで指定するビット単位のキーです)。	形式 2
b	データが符号なし 2 進数 (バイトデータ) であることを指定します※ <sup>10</sup> (マスクビットで指定するビット単位のキーです)。	形式 2
p	データが内部 10 進数であることを指定します。	形式 1
z	データが外部 10 進数であることを指定します。※ <sup>5</sup>	形式 1
x	データが左符号付き外部 10 進数であることを指定します。※ <sup>5</sup>	形式 1
s	データが右分離符号付き外部 10 進数であることを指定します。※ <sup>5</sup>	形式 1
t	データが左分離符号付き外部 10 進数であることを指定します。※ <sup>5</sup>	形式 1
f	データが内部浮動小数点であることを指定します。※ <sup>9</sup> , ※ <sup>10</sup>	形式 1
g	データが日付 (英数字) であることを指定します。※ <sup>5</sup>	形式 1

<i>data-attribute</i> に指定する値	意味	<i>no-split-data-definision</i> 形式
q	データが文字（文字順序変更機能 <sup>※2</sup> で指定した文字）であることを指定します。 ※3	形式 1
d	データが文字（左分離符号付き数字）であることを指定します。 <sup>※5</sup>	形式 1
w	データが文字（ワイド文字 <sup>※4</sup> , <sup>※6</sup> ）であることを指定します（テキストファイルだけ指定できます）。	形式 1
u	データが文字（マルチバイト文字 <sup>※6</sup> , <sup>※7</sup> , <sup>※8</sup> ）であることを指定します（テキストファイルだけ指定できます）。	形式 1
v	データが文字（マルチバイト文字 <sup>※6</sup> , <sup>※7</sup> , <sup>※8</sup> ）であることを指定します（Unicode の IVS に対応した属性で、テキストファイルだけ指定できます）。	形式 1

## 注※ 1

ASCII コード (0x00 ~ 0x7F) を含む、JIS8 単位コード (0x00 ~ 0xFF) を意味します。

入力データに日本語などのマルチバイト文字が含まれている場合、マルチバイト文字として扱わず、シングルバイト文字として扱います。

EBCDIK コードの体系に従ってキーを比較する EBCDIK コード比較オプション機能を使用できます。EBCDIK コード比較オプション機能については、「5.5.3 EBCDIK コード比較オプション (EBCJ, EBCE オプション)」を参照してください。

## 注※ 2

文字順序変更機能については、「5.15 -altseq パラメタ」を参照してください。

## 注※ 3

入力データのエンコードの種別には対応しません。入力データの文字を構成するバイト長を意識しないで、常に 1 バイト長のシングルバイト文字として扱います。ASCII コードや JIS8 単位コード向けの属性です。

## 注※ 4

入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定した場合、1 文字 2 バイトのワイド文字として扱います。また、サロゲートペア文字は 2 文字として扱います。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

入力データのエンコードに「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定した場合、1文字4バイトのワイド文字として扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。

なお、属性コード「w」は固定長順編成ファイルと可変長順編成ファイルの入力ファイルに対して指定できません。指定した場合は KBL52052-E メッセージが表示されて、エラーになります。

### 注※ 5

入力データのエンコードで解釈します。

「ASCII」, 「SJIS」, 「EUC-JP」, 「EUC-HJ」または「UTF-8」を指定した場合は1バイト、

「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」を指定した場合は2バイト、「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」を指定した場合は4バイトの長さで文字を扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。

### 注※ 6

非分割データ範囲の入力データがすべて ASCII 対応コード (英数字) である場合は、属性コード「c」を指定した方が処理効率が向上します。属性コード「c」は1文字1バイトで扱いますので、エンコードに対応したバイト長を指定してください。

ASCII 対応コードについては、「付録 H Unicode 操作文字コード一覧 (SORT EE)」を参照してください。

1文字の長さについては、「付録 I.2(1) 文字の長さ」を参照してください。

文字属性コードの特徴については、「付録 I.2(2) 文字属性コードの特徴」を参照してください。

### 注※ 7

属性コード「u」と「v」は、入力データのエンコードで解釈します。1文字の長さは可変長として扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。また、マルチバイト文字の扱いについては、「付録 I.2 文字列データの扱い」を参照してください。

### 注※ 8

入力データのエンコードに「UTF-8」, 「UTF-16LE」, 「UTF-16BE」, 「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」または「UNICODE

(入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE, UTF-16BE, UTF-32LE または UTF-32BE) のどれかを指定した場合、Unicode の IVS (Ideographic Variation Sequence) を次のとおり解釈します。Unicode の IVS については、「付録 I.2(6) IVS 対応」を参照してください。

- 属性コード「u」は、UCS コードが一致する文字であれば、Unicode の IVS の異体字であっても同じ文字として扱います。
- 属性コード「v」は、UCS コードが一致する文字であっても、VS (Variation Selector, 異体字セレクタ) が異なる場合は、異なる文字として扱います。

#### 注※ 9

属性コード「n」、「m」、「f」および「a」は、数値として比較するため、バイトオーダーが関係します。バイトオーダーについては、「2.6 エンディアンオプション機能」および「5.5.4 入力ファイルのエンディアンオプション (BIG | INPBIG | INPLTL オプション)」を参照してください。

- 符号付き固定小数点数「n」: 2 バイト (signed short int), 4 バイト (signed int), 8 バイト (signed long long int)
- 符号なし 2 進数「m」、「a」: 2 バイト (unsigned short int), 4 バイト (unsigned int), 8 バイト (unsigned long long int)
- 内部浮動小数点数「f」: 4 バイト (float), 8 バイト (double)

#### 注※ 10

属性コード「n, m, f」と「a, b」は、どちらも「2 進数」データですが、キーの長さの指定方法が異なります。

- 属性コード「n」、「m」および「f」は、キーの長さを「バイト長」で指定します。
- 属性コード「a」と「b」は、キーの長さ分の「マスクビット」を指定します。

#### + pos

非分割データの開始バイト位置を 5 桁以内の数字で指定します。-locatajst パラメタの指定有無によって、レコード先頭の解釈が異なります。また、テキストファイルの場合は、入力データのエンコードによって指定範囲が異なります。開始バイト位置 (pos) に指定できる値の範囲を次の表に示します。

ファイル編成	-locatajst パラメタ	
	指定あり	指定なし
固定長順編成ファイル	1 ~ 65,535	0 ~ 65,534
可変長順編成ファイル	5 ~ 65,539※1	0 ~ 65,534

5. exsort コマンドのパラメタおよびオペランドの詳細

ファイル編成			-locatajst パラメタ	
			指定あり	指定なし
テキスト ファイル※2	改行文字 「LF」	ASCII SJIS EUC-JP EUC-HJ UTF-8	1 ~ 65,535	0 ~ 65,534
		UCS-2LE UCS-2BE UTF-16LE UTF-16BE	1 ~ 65,533 (2 の倍数 + 1 ※3)	0 ~ 65,532 (2 の倍数※3)
		UCS-4LE UCS-4BE UTF-32LE UTF-32BE	1 ~ 65,529 (4 の倍数 + 1 ※4)	0 ~ 65,528 (4 の倍数※4)
	改行文字 「CRLF」	ASCII SJIS EUC-JP EUC-HJ UTF-8	1 ~ 65,534	0 ~ 65,533
		UCS-2LE UCS-2BE UTF-16LE UTF-16BE	1 ~ 65,531 (2 の倍数 + 1 ※3)	0 ~ 65,530 (2 の倍数※3)
		UCS-4LE UCS-4BE UTF-32LE UTF-32BE	1 ~ 65,525 (4 の倍数 + 1 ※4)	0 ~ 65,524 (4 の倍数※4)

注※1

レコードの前の4バイト領域 (RDW: レコードヘッダ) の先頭が1となります。

注※2

入力レコードの改行文字については、「5.8.7 recsepa オペランド」を参照してください。また、入力データのエンコードについては、「5.8.13 codetype オペランド」を参照してください。

注※3

2の倍数 (-locatajst パラメタを同時に指定している場合は「2の倍数+1」) 以外を指定すると、KBL2054-E メッセージが表示されてエラーになります。

注※4

4の倍数 (-locatajst パラメタを同時に指定している場合は「4の倍数+1」) 以外を指定すると、KBL2054-E メッセージが表示されてエラーになります。

- len



非分割データの長さを次の表に示す範囲で指定します。

data-attribute	指定できる長さ (単位)
j, c, q, w <sup>※1</sup>	<ul style="list-style-type: none"> <li>入力データのエンコードが「ASCII」, 「SJIS」, 「EUC-JP」, 「EUC-HJ」または「UTF-8」の場合 1 ~ 4,096 (バイト)</li> <li>入力データのエンコードが「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」の場合<sup>※2</sup> 2 ~ 8,192 (バイト)</li> <li>入力データのエンコードが「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」の場合<sup>※3</sup> 4 ~ 16,384 (バイト)</li> </ul>
u, v	1 ~ 4,096 (文字数) <sup>※6</sup>
m, n	2, 4 または 8 (バイト)
p	1 ~ 10 (バイト)
z, x	1 ~ 18 (桁) <sup>※4</sup>
s, t	2 ~ 19 (桁) <sup>※4</sup>
f	4 または 8 (バイト)
g	2 ~ 8 (桁) <sup>※5</sup>
d	1 ~ 256 (桁) <sup>※4</sup>

#### 注※1

入力データのエンコードの種別によって、マルチバイト文字の長さは異なります。詳細は「付録 I.2 文字列データの扱い」を参照してください。

#### 注※2

入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定した場合、長さには「2 の倍数」を指定してください。

2 の倍数以外を指定すると、KBLS2056-E メッセージが表示されてエラーになります。

#### 注※3

入力データのエンコードに「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定した場合、長さには「4 の倍数」を指定してください。

4 の倍数以外を指定すると、KBLS2056-E メッセージが表示されてエラーになります。

注※ 4

「バイト長」ではなく数値の「桁数」を指定してください。「UCS-2」, 「UCS-4」, 「UTF-16」および「UTF-32」の数値データの扱いについては、「付録 I.3 数値データの扱い」を参照してください。

注※ 5

「バイト長」ではなく日付の「桁数」を指定してください。「UCS-2」, 「UCS-4」, 「UTF-16」および「UTF-32」の日付データの扱いについて「付録 I.4 日付データの扱い」を参照してください。

注※ 6

「バイト長」ではなく「文字数」を指定してください。UTF-16 のサロゲートペア文字は 1 文字として扱います。また、Unicode の VS (Variation Selector, 異体字セレクタ) は文字数に数えません。マルチバイト文字の扱いについては、「付録 I.2 文字列データの扱い」を参照してください。

— *maskbit*

- 属性コード「a」または「b」を指定した場合、マスクビットを 16 進数で指定します。
- 16 進数の A ~ F は、小文字・大文字のどちらでも指定できます。
- キーデータに対して、指定したマスクビットで AND 演算を行った結果でソートします。なお、この AND 演算によってキーデータが更新されることはありません。
- すべての桁を「00」とするマスクビット指定はできません。指定した場合は、KBLS1201-E エラーとなります。
- 指定できるマスクビットの桁は次のとおりです。

<i>key-attribute</i> に指定する値	指定できる <i>maskbit</i> の桁	備考
a	4, 8 または 16 (桁) ※1	キーの長さは 2, 4 または 8 バイト
b	2 ~ 8,192 (桁) ※2	キーの長さは 1 ~ 4,096 バイト

注※ 1

属性コード「a (符号なし 2 進数)」は、2 バイト (unsigned short int), 4 バイト (unsigned int), 8 バイト (unsigned long long int) のキーとして扱います。2, 4 または 8 バイトに対応するマスクビット値を、それぞれ 4, 8 または 16 桁の 16 進数で指定してください。なお、16 進数はビッグエンディアン形式で記述してください。

注※ 2

属性コード「b (符号なし 2 進数)」は、1 ~ 4,096 バイトの範囲を指定できま

す。マスクビット値を、2～8,192桁の16進数で指定してください。なお、16進数はビッグエンディアン形式で記述してください。

#### 非分割データの範囲

非分割データは、次の範囲で指定してください。

- ・入力ファイルが固定長順編成ファイルのとき、レコード長の範囲
- ・入力ファイルが可変長順編成ファイルのとき、最小レコード長の範囲※1
- ・入力ファイルがテキストファイルのとき、最小レコード長の範囲※1、※2

##### 注※1

可変長順編成ファイルで、`-cmdopt` パラメタの `POSNOCHK` オプションを指定している場合は、最大レコード長の範囲となります。

なお、`POSNOCHK` オプションでは、最大レコード長を超えた範囲を非分割データとして指定してもエラーにはなりませんが、最大レコード長を超えた部分の非分割データの値は必ず `NULL` 値 (`0x00`) が仮定されるので注意してください。

`-cmdopt` パラメタの `POSNOCHK` オプションについては、「5.5.11 キー位置のチェックオプション (`POSNOCHK` オプション)」を参照してください。

##### 注※2

テキストファイルに指定した属性コード「`u`」または「`v`」の場合は、キーの指定範囲が実レコード長を超えていてもエラーにはなりません。実レコード長を超えた部分には空白を仮定してキーを比較します。

#### 注意事項

- ・指定できる個数は1～64個です。
- ・次の例のように、属性によってはデータが異なっても同じ値として扱います。  
データ:「`+0`」と「`-0`」  
属性 `t`: 同じ値として扱います。  
属性 `c`: 同じ値として扱いません。
- ・非分割データを複数指定した場合、先に指定した非分割データほど優先順位が高く、あとに指定した非分割データほど優先順位が低くなります。
- ・非分割データの範囲は、ほかの非分割データと重なってもかまいません。
- ・レコード編集によって、入力レコードと異なった位置に非分割データを再編成するような場合は注意が必要となります。非分割データの範囲は、最後に実行したレコード編集後のレコードを基に指定してください。非分割データの位置 (範囲) は、「付録 F レコード編集処理で基になるレコード (`SORT EE` の機能)」に示す規則に従って指定してください。

## 5.21.4 nosplitcsvdata オペランド

#### 意味

非分割 CSV データを指定します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

### 形式

```
nosplitcsvdata=no-split-csv-data-definition  
[△no-split-csv-data-definition…]
```

### 指定値

#### no-split-csv-data-definition オペランド形式 1

```
data-attribute+cellpos-len [△data-attribute+cellpos-len  
… ]
```

- 文字データのセルを定義します。
- data-attribute には、「j」、「c」、「d」、「w」、「u」、「v」または「q」を指定できます。

#### no-split-csv-data-definition オペランド形式 2

```
data-attribute+cellpos-p [, s] [△data-attribute+cellpos-  
p[, s] … ]
```

- 数値データ（固定小数点数）のセルを定義します。
- data-attribute には、「z」だけを指定できます。
- データ中に、引用符による囲み、符号、通貨記号、桁修飾文字（コンマ、小数点、空白）が含まれる場合、これらの文字を除いたデータで比較します。
- 環境ファイルの「EXS\_ROUND」または `-inpfile`、`-catinp`、`-splitinp` パラメタの `round` オペランドで、丸め方式（四捨五入、切り捨て、または切り上げ）を指定します。この場合、小数部は指定に従って処理されます。  
EXS\_ROUND および `round` オペランドの両方の指定を省略している場合、切り捨てられて処理されます。詳細は、「5.8.11 round オペランド」を参照してください。

#### no-split-csv-data-definition オペランド形式 3

```
data-attribute+cellpos-d, dateform [△data-attribute+  
cellpos-d, dateform … ]
```

- 日付データのセルを定義します。
- data-attribute には、「g」だけを指定できます。

#### no-split-csv-data-definition オペランド形式 4

```
data-attribute+cellpos-datedigit[, timedigit]  
[△data-attribute+cellpos-datedigit[, timedigit] … ]
```

- 日付時刻データのセルを定義します。
- data-attribute には、「h」だけを指定できます。

no-split-csv-data-definition に指定するそれぞれの値について次に示します。

#### data-attribute

非分割 CSV データの属性を指定します。

data-attribute に指定する 値	意味	指定できる形式
j	データが文字 (JIS8 単位コード※ <sup>1</sup> ) であることを指定します。	形式 1
c	データが文字であることを指定します。※ <sup>3</sup>	形式 1
d	データが文字 (左分離符号付き数字) であることを指定します。※ <sup>5</sup>	形式 1
q	データが文字 (文字順序変更機能※ <sup>2</sup> で指定した文字) であることを指定します。※ <sup>3</sup>	形式 1
z	データが数値 (固定小数点数) であることを指定します。※ <sup>5</sup>	形式 2
g	データが日付 (英数字) であることを指定します。※ <sup>5</sup>	形式 3
h	データが日付時刻 (数字) であることを指定します。※ <sup>7</sup>	形式 4
w	データが文字 (ワイド文字※ <sup>4</sup> , ※ <sup>6</sup> ) であることを指定します。	形式 1
u	データが文字 (マルチバイト文字※ <sup>6</sup> , ※ <sup>8</sup> , ※ <sup>9</sup> ) であることを指定します。	形式 1
v	データが文字 (マルチバイト文字※ <sup>6</sup> , ※ <sup>8</sup> , ※ <sup>9</sup> ) であることを指定します (Unicode の IVS に対応した属性です)。	形式 1

## 注※ 1

ASCII コード (0x00 ~ 0x7F) を含む, JIS8 単位コード (0x00 ~ 0xFF) を意味します。

入力データに日本語などのマルチバイト文字が含まれている場合, マルチバイト文字として扱わず, シングルバイト文字として扱います。

EBCDIK コードの体系に従ってキーを比較する EBCDIK コード比較オプション機能を使用できます。「EBCDIK コード比較オプション機能」については, 「5.5.3 EBCDIK コード比較オプション (EBCJ, EBCE オプション)」を参照してください。

## 注※ 2

文字順序変更機能については, 「5.15 -altseq パラメタ」を参照してください。

## 注※ 3

入力データのエンコードの種別には対応しません。入力データの文字を構成するバイト長を意識しないで, 常に 1 バイト長のシングルバイト文字として扱います。ASCII コードや JIS8 単位コード向けの属性です。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

### 注※ 4

入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定した場合、1 文字 2 バイトのワイド文字として扱います。またサロゲートペア文字は 2 文字として扱います。

入力データのエンコードに「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定した場合、1 文字 4 バイトのワイド文字として扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。

### 注※ 5

入力データのエンコードで解釈します。

「ASCII」, 「SJIS」, 「EUC-JP」, 「EUC-HJ」または「UTF-8」を指定した場合は 1 バイト、「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」を指定した場合は 2 バイト、「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」を指定した場合は 4 バイトの長さで文字を扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。

### 注※ 6

非分割 CSV データ範囲の入力データがすべて ASCII 対応コード (英数字) である場合は、属性コード「c」を指定した方が処理効率が向上します。属性コード「c」は 1 文字 1 バイトで扱いますので、エンコードに対応したバイト長を指定してください。

ASCII 対応コードについては、「付録 H Unicode 操作文字コード一覧 (SORT EE)」を参照してください。

1 文字の長さについては、「付録 I.2(1) 文字の長さ」を参照してください。

文字属性コードの特徴については、「付録 I.2(2) 文字属性コードの特徴」を参照してください。

### 注※ 7

非分割 CSV データ範囲の入力データが次の条件すべてに該当する場合は、属性コード「c」を指定した方が処理効率は向上します。

- ・桁数がそろっている。
- ・時刻が 24 時間制で表記されている。
- ・日付、時刻またがりのデータがない。

属性コード「c」は 1 文字 1 バイトで扱いますので、エンコードに対応したバイト長を指定してください。

1 文字の長さについては、「付録 I.2(1) 文字の長さ」を参照してください。文字属性コードの特徴については、「付録 I.2(2) 文字属性コードの特徴」を参照してください。

#### 注※ 8

属性コード「u」と「v」は、入力データのエンコードで解釈します。1 文字の長さは可変長として扱います。

文字のエンコードの指定方法については、「5.8.13 codetype オペランド」を参照してください。また、マルチバイト文字の扱いについては、「付録 I.2 文字列データの扱い」を参照してください。

#### 注※ 9

入力データのエンコードに「UTF-8」、「UTF-16LE」、「UTF-16BE」、「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-16LE, UTF-16BE, UTF-32LE または UTF-32BE）」のどれかを指定した場合、Unicode の IVS（Ideographic Variation Sequence）を次のとおり解釈します。Unicode の IVS については、「付録 I.2(6) IVS 対応」を参照してください。

- 属性コード「u」は、UCS コードが一致する文字であれば、Unicode の IVS の異体字であっても同じ文字として扱います。
- 属性コード「v」は、UCS コードが一致する文字であっても、VS（Variation Selector, 異体字セレクタ）が異なる場合は、異なる文字として扱います。

#### + *cellpos*

非分割 CSV データとするセル番号を数字で指定します。

指定できる範囲は 1 ～ 16384 です。

指定方法については、「5.16 -csvkey パラメタ」の「+*cellpos*」を参照してください。

#### - *len*

data-attribute に「j」、「c」、「q」、「w」、「u」、「v」または「d」を指定した場合に、非分割 CSV データの最大長を 10 進数で指定します。

- 属性「d」は最大桁数を指定します。
- キーの長さを次の表に示す範囲で指定します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

data-attribute に指定する値	指定できる長さ (単位)
j, c, q, w ※1	<ul style="list-style-type: none"> <li>• 入力データのエンコードが「ASCII」, 「SJIS」, 「EUC-JP」, 「EUC-HJ」または「UTF-8」の場合 1 ~ 4,096 (バイト)</li> <li>• 入力データのエンコードが「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」の場合※2 2 ~ 8,192 (バイト)</li> <li>• 入力データのエンコードが「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」の場合※3 4 ~ 16,384 (バイト)</li> </ul>
u, v	1 ~ 4,096 (文字数) ※5
d	1 ~ 256 (桁) ※4

### 注※1

入力データのエンコードの種別によって、マルチバイト文字の長さは異なります。詳細は「付録 I.2 文字列データの扱い」を参照してください。

### 注※2

入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定した場合、長さには「2 の倍数」を指定してください。

2 の倍数以外を指定すると、KBL52056-E メッセージが表示されてエラーになります。

### 注※3

入力データのエンコードに「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-16BE)」を指定した場合、長さには「4 の倍数」を指定してください。

4 の倍数以外を指定すると、KBL52056-E メッセージが表示されてエラーになります。

### 注※4

「バイト長」ではなく数値の「桁数」を指定してください。「UCS-2」, 「UCS-4」, 「UTF-16」および「UTF-32」の数値データの扱いについては、「付録 I.3 数値データの扱い」を参照してください。

### 注※5

「バイト長」ではなく「文字数」を指定してください。UTF-16 のサロゲート



ペア文字は 1 文字として扱います。また、Unicode の VS (Variation Selector, 異体字セレクタ) は文字数に数えません。マルチバイト文字の扱いについては、「付録 I.2 文字列データの扱い」を参照してください。

#### – *p* [ *s* ]

`data-attribute` に「*z*」を指定した場合に、非分割 CSV データの有効桁を 10 進数で指定します。

指定方法については、「5.16 `-csvkey` パラメタ」の「*-p* [ *s* ]」を参照してください。

#### – *d*, *dateform*

`data-attribute` に「*g*」を指定した場合に、非分割 CSV データの有効桁および書式番号を指定します。

指定方法については、「5.16 `-csvkey` パラメタ」の「*-d*, *dateform*」を参照してください。

#### – *datedigit* [ *timedigit* ]

`data-attribute` に「*h*」を指定した場合に、非分割 CSV データの有効桁および書式番号を指定します。指定方法については、「5.16 `-csvkey` パラメタ」の「*- datedigit* [ *timedigit* ]」を参照してください。

### 注意事項

- 指定できる個数は、1 ~ 64 個です。
- 次の例のように、属性によってはデータが異なっても同じ値として扱います。  
データ : 「001」 と 「1」  
属性 *z* : 同じ値として扱います。  
属性 *c* : 同じ値として扱いません。
- 非分割 CSV データを複数指定した場合、先に指定した非分割 CSV データほど優先順位が高く、あとに指定した非分割 CSV データほど優先順位が低くなります。
- 非分割 CSV データの範囲は、ほかの非分割 CSV データと重なってもかまいません。
- `nosplitcsvdata` オペランドは、CSV ファイルを指定した場合だけ有効です。CSV ファイル以外を指定した場合は、KBL52061-E エラーとなります。
- レコード編集によって、入力レコードと異なった位置に非分割 CSV データを再編成するような場合は注意が必要となります。非分割 CSV データの範囲は、最後に実行したレコード編集後のレコードを基に指定してください。非分割 CSV データの位置は、「付録 F レコード編集処理で基になるレコード (SORT EE の機能)」に示す規則に従って指定してください。

### 指定例

- 分割後のファイルに格納されるレコード件数が最大で 10 件となるように分割します。  
入力ファイル

## 5. exsort コマンドのパラメタおよびオペランドの詳細

レコード件数 : 50 件

パラメタ指定例

`-splitdata recno=10`

分割後のファイル

ファイル数 : 5 個

各ファイルのレコード件数 : 10 件

2. 分割後のファイルに格納されるレコード件数が最大で 3 件となるように分割します。ただし、レコードの先頭から 4 バイトの値（文字属性）が同じであれば分割しません。

レコード件数 : 6 件

ファイルの内容 : 値はすべて文字

st-a	0000000011111110	no1-record
st-a	0000000011111101	no2-record
st-a	0000000011111011	no3-record
st-a	0000000011110111	no4-record
st-b	0000000011101111	no5-record
st-b	0000000011011111	no6-record

パラメタ指定例

`-splitdata recno=3 nosplitdata=c+0-4`

分割後のファイル

ファイル数 : 2 個

ファイルの内容 : 1 個目のファイル (レコード数は 4 件)

st-a	0000000011111110	no1-record
st-a	0000000011111101	no2-record
st-a	0000000011111011	no3-record
st-a	0000000011110111	no4-record

ファイルの内容 : 2 個目のファイル (レコード数は 2 件)

st-b	0000000011101111	no5-record
st-b	0000000011011111	no6-record

### 関連パラメタ

- `-splitfile`
- `-splitkey`
- `-splitinp`
- `-splitout`

## 5.22 -splitkey パラメタ

### 意味

キー分割情報を指定します。

- 指定されたキー（分割キー）でファイルを分割して出力します。分割キーの値が前のレコードと異なる場合にファイルを分割します。前のレコードと同じであれば、前のレコードと同じファイルに出力します。
- 分割キーは `key` オペランドまたは `csvkey` オペランドで指定します。
- 出力する分割ファイルの数は不定となります。
- 分割ファイルの最大データ量（レコード件数またはファイルサイズ）を指定できます。分割キーの値が前のレコードと同じであっても、最大データ量を超える場合はファイルを分割します。最大データ量は `recno` オペランドまたは `spno` オペランドで指定します。

### 形式

```
-splitkey
  [△ {key=split-key-definition [△split-key-definition...]
    | csvkey=split-csv-key-definition
    [△split-csv-key-definition...]}]
  [△ {recno=split-record-number
    | spno=split-file-size [unit]}]
```

- `-splitkey` パラメタは、`-splitfile` パラメタおよび `-splitdata` パラメタと同時に指定できません。どれか一つだけ指定してください。

### 5.22.1 key オペランド

#### 意味

分割キーを指定します。

#### 形式

```
key=split-key-definition [△split-key-definition...]
```

#### 指定値

*split-key-definition*

指定形式については、「5.6 `-key` パラメタ」を参照してください。

#### 注意事項

- 分割するファイルは、分割キーで昇順または降順にソートされている必要があります。ソートされていない場合、**KBLS2030-E** エラーとなります。ソートされていないファイルを分割したい場合、強制ファイル分割機能オプション (`-cmdopt FSPLIT`) を指定することでエラーにならず、ファイル分割できます。詳細については、「5.5.16 強制ファイル分割機能 (FSPLIT オプション)」を参照してください。
- 指定できる個数は 1 ～ 64 個です。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- 次の例のように、属性によってはデータが異なっても同じ値として扱います。  
データ : 「+0」 と 「-0」  
属性 t : 同じ値として扱います。  
属性 c : 同じ値として扱いません。
- 分割キーを複数指定した場合、先に指定したキーほど優先順位が高く、あとに指定したキーほど優先順位が低くなります。
- 分割キーの範囲は、ほかの分割キーの範囲と重なってもかまいません。
- 次の条件にすべて該当する場合は、`-key` パラメタで指定したキーを分割キーとして設定します。
  - ソート機能またはマージ機能を使用する。
  - ファイル分割機能を使用する。
  - 入力ファイルが CSV ファイルでない。
  - `-splitfile`, `-splitdata`, および `-splitkey` パラメタのすべてのパラメタを指定しない。または、`-splitkey` パラメタで `key` オペランドを指定しない。
- レコード編集によって、入力レコードと異なった位置に分割キーを再編成するような場合は注意が必要となります。分割キーの範囲は、最後に実行したレコード編集後のレコードを基に指定してください。分割キーの位置 (範囲) は、「付録 F レコード編集処理で基になるレコード (SORT EE の機能)」に示す規則に従って指定してください。

### 5.22.2 csvkey オペランド

#### 意味

分割 CSV キーを指定します。

#### 形式

`csvkey=split-csv-key-definition` [`△split-csv-key-definition...`]

#### 指定値

*split-csv-key-definition*

指定形式は、「5.16 `-csvkey` パラメタ」を参照してください。

#### 注意事項

- 分割するファイルは、分割キーで昇順または降順にソートされている必要があります。ソートされていない場合、**KBLS2030-E** エラーとなります。ソートされていないファイルを分割したい場合は、強制ファイル分割機能オプション (`-cmdopt FSPLIT`) を指定してください。詳細については、「5.5.16 強制ファイル分割機能 (FSPLIT オプション)」を参照してください。
- 指定できる個数は 1 ~ 64 個です。
- 次の例のように、属性によってはデータが異なっても同じ値として扱います。  
データ : 「001」 と 「1」  
属性 z : 同じ値として扱います。

- 属性 *c* : 同じ値として扱いません。
- 分割 CSV キーを複数指定した場合、先に指定したキーほど優先順位が高く、あとに指定したキーほど優先順位が低くなります。
  - 分割 CSV キーの範囲は、ほかの分割 CSV キーの範囲と重なってもかまいません。
  - 次の条件にすべて該当する場合は、`-csvkey` パラメタで指定したキーを、分割 CSV キーとして設定します。
    - ソート機能またはマージ機能を使用する。
    - ファイル分割機能を使用する。
    - 入力ファイルが CSV ファイルである。
    - `-splitfile`, `-splitdata`, および `-splitkey` パラメタのすべてのパラメタを指定しない。または、`-splitkey` パラメタで `csvkey` オペランドを指定しない。
  - レコード編集によって、入力レコードと異なった位置に分割 CSV キーを再編成するような場合は注意が必要となります。分割 CSV キーの範囲は、最後に実行したレコード編集後のレコードを基に指定してください。分割 CSV キーの位置（範囲）は、「付録 F レコード編集処理で基になるレコード (SORT EE の機能)」に示す規則に従って指定してください。

### 5.22.3 recno オペランド

#### 意味

出力ファイルに出力する最大レコード件数を指定します。詳細については、「5.21 `-splitdata` パラメタ」の「5.21.1 `recno` オペランド」を参照してください。

#### 形式

`recno=split-record-number`

### 5.22.4 spno オペランド

#### 意味

出力ファイルの最大出力ファイルサイズを指定します。詳細については、「5.21 `-splitdata` パラメタ」の「5.21.2 `spno` オペランド」を参照してください。

#### 形式

`spno=split-file-size [unit]`

#### 指定例

1. レコードの先頭から 4 バイトの値（文字属性）が異なれば、ファイルを分割します。  
 入力ファイル  
 レコード件数 : 6 件  
 ファイルの内容 : 値はすべて文字

5. exsort コマンドのパラメタおよびオペランドの詳細

st-a	0000000011111110	no1-record
st-a	0000000011111101	no2-record
st-a	0000000011111011	no3-record
st-b	0000000011110111	no4-record
st-b	0000000011101111	no5-record
st-c	0000000011011111	no6-record

パラメタ指定例

-splitdata key=c+0-4

分割後のファイル

ファイル数 : 3 個

ファイルの内容 : 1 個目のファイル (レコード数は 3 件)

st-a	0000000011111110	no1-record
st-a	0000000011111101	no2-record
st-a	0000000011111011	no3-record

ファイルの内容 : 2 個目のファイル (レコード数は 2 件)

st-b	0000000011110111	no4-record
st-b	0000000011101111	no5-record

ファイルの内容 : 3 個目のファイル (レコード数は 1 件)

st-c	0000000011011111	no6-record
------	------------------	------------

- レコードの先頭から 4 バイトの値 (文字属性) が異なれば、ファイルを分割します。ただし、分割後のファイルに格納されるレコード件数の最大は 2 件とします。

入力ファイル

レコード件数 : 6 件

ファイルの内容 : 値はすべて文字

st-a	0000000011111110	no1-record
st-a	0000000011111101	no2-record
st-a	0000000011111011	no3-record
st-b	0000000011110111	no4-record
st-b	0000000011101111	no5-record
st-c	0000000011011111	no6-record

パラメタ指定例

-splitdata key=c+0-4 recno=2

分割後のファイル

ファイル数 : 3 個

ファイルの内容 : 1 個目のファイル (レコード数は 2 件)

st-a	0000000011111110	no1-record
st-a	0000000011111101	no2-record

ファイルの内容 : 2 個目のファイル (レコード数は 1 件)

st-a	0000000011111011	no3-record
------	------------------	------------

ファイルの内容 : 3 個目のファイル (レコード数は 2 件)

st-b	0000000011110111	no4-record
st-b	0000000011101111	no5-record

ファイルの内容 : 4 個目のファイル (レコード数は 1 件)

st-c	0000000011011111	no6-record
------	------------------	------------

#### 関連パラメタ

- -splitfile
- -splitdata
- -splitinp
- -splitout

## 5.23 -splitinp パラメタ

### 意味

ファイル分割機能の入力ファイル情報を指定します。

### 形式

```
-splitinp
  △fileorg=organization-type
  △file=file-name [△file-name...]
  [△reclen=record-length]
  [△ {skiprec=skip-record-number
    | startrec=start-record-number} ]
  [△stopaft=stop-record-number]
  [△recsepa=record-separator-type]
  [△cellsepa= {TAB | 'cell-separator'}
    [△ {TAB | 'cell-separator'} ... ] ]
  [△quottype=quotation-type]
  [△currency=currency-symbol]
  [△round=option-code]
  [△nullseq=option-code]
  [△codetype=encode]
  [△ { reorg=reorganization-format
    [△reorganization-format ... ]
    | csvreorg=csv-reorganization-format
    [△csv-reorganization-format ... ] } ]
  [△ { {include | omit} =condition-expression
    | {csvinclude | csvomit} =csv-condition-expression} ]
```

- 出力ファイル情報は、`-splitout` パラメタで指定してください。
- `-splitinp` パラメタは、`-inpfile` パラメタおよび `-catinp` パラメタと同時に指定できません。どれか一つだけ指定してください。
- コピー機能とファイル分割機能を使用する場合、`-splitfile`、`-splitdata`、または `-splitkey` パラメタのどれか一つを指定する必要があります。ソート機能またはマージ機能と、ファイル分割機能を使用する場合に、`-splitfile`、`-splitdata`、および `-splitkey` パラメタのすべてを指定しないときは、`-key` パラメタまたは `-csvkey` パラメタで指定したキーを分割キーとして設定します。

### 5.23.1 fileorg オペランド

#### 意味

入力ファイルのファイル編成を指定します。詳細については、「5.8.1 fileorg オペランド」を参照してください。

#### 形式

```
fileorg=organization-type
```



## 5.23.2 file オペランド

### 意味

入力ファイル名または環境変数名を指定します。詳細については、「5.8 -inpfile パラメタ」の「5.8.2 file オペランド」を参照してください。

### 形式

`file=file-name` [`△file-name...`]

### 注意事項

- file オペランドは省略できません。

## 5.23.3 reclen オペランド

### 意味

入力レコード長を指定します。詳細については、「5.8 -inpfile パラメタ」の「5.8.3 reclen オペランド」を参照してください。

### 形式

`reclen=record-length`

## 5.23.4 skiprec オペランド

### 意味

入力レコードを読み飛ばす件数を指定します。詳細については、「5.8 -inpfile パラメタ」の「5.8.4 skiprec オペランド」を参照してください。

### 形式

`skiprec=skip-record-number`

## 5.23.5 stopaft オペランド

### 意味

入力を打ち切るレコード件数を指定します。詳細については、「5.8 -inpfile パラメタ」の「5.8.5 stopaft オペランド」を参照してください。

### 形式

`stopaft=stop-record-number`

## 5.23.6 startrec オペランド

### 意味

データの入力を開始するレコード番号を指定します。詳細については、「5.8 -inpfile パラメタ」の「5.8.6 startrec オペランド」を参照してください。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

### 形式

`startrec=start-record-number`

## 5.23.7 recsepa オペランド

### 意味

テキストファイルおよび CSV ファイルの入力レコードの改行文字を指定します。詳細については、「5.8 `-inpf` パラメタ」の「5.8.7 `recsepa` オペランド」を参照してください。

### 形式

`recsepa=record-separator-type`

## 5.23.8 cellsepa オペランド

### 意味

CSV ファイルの入力レコードのデリミタ（セル区切り文字）を指定します。詳細については、「5.8 `-inpf` パラメタ」の「5.8.8 `cellsepa` オペランド」を参照してください。

### 形式

`cellsepa=cell-separator [△cell-separator ... ]`

## 5.23.9 quottype オペランド

### 意味

セルの引用符文字を指定します。詳細については、「5.8 `-inpf` パラメタ」の「5.8.9 `quottype` オペランド」を参照してください。

### 形式

`quottype=quotation-type`

## 5.23.10 currency オペランド

### 意味

CSV ファイルの固定小数点数の数値データの通貨記号として、任意の 1 文字を指定します。詳細については、「5.8 `-inpf` パラメタ」の「5.8.10 `currency` オペランド」を参照してください。

### 形式

`currency=currency-symbol`

## 5.23.11 round オペランド

### 意味

CSV ファイルのすべての固定小数点数のキー、比較項目、または集約項目の桁の丸め方式を指定します。詳細については、「5.8 `-inpf` パラメタ」の「5.8.11 `round` オペランド」を参照してください。

**形式**

`round=option-code`

## 5.23.12 nullseq オペランド

**意味**

CSV ファイルのナル文字列が含まれる入力レコードを、先頭または末尾に配置するオプションです。詳細については、「5.8 `-inpf` パラメタ」の「5.8.12 `nullseq` オペランド」を参照してください。

**形式**

`nullseq=option-code`

## 5.23.13 codetype オペランド

**意味**

入力データの文字コードの種類を指定します。詳細については、「5.8 `-inpf` パラメタ」の「5.8.13 `codetype` オペランド」を参照してください。

**形式**

`codetype=encode`

## 5.23.14 reorg オペランド

**意味**

入力レコード編集フォーマットを指定します。詳細については、「5.8 `-inpf` パラメタ」の「5.8.14 `reorg` オペランド」を参照してください。

**形式**

`reorg=reorganization-format` [ $\Delta$ `reorganization-format`...]

## 5.23.15 csvreorg オペランド

**意味**

CSV ファイルの入力レコード編集フォーマットを指定します。詳細については、「5.8 `-inpf` パラメタ」の「5.8.15 `csvreorg` オペランド」を参照してください。

**形式**

`csvreorg=csv-reorganization-format` [ $\Delta$ `csv-reorganization-format` ... ]

## 5.23.16 {include | omit} オペランド

### 意味

入力レコードの選択条件式を指定します。

**include** オペランド:

条件を満たすレコードを選択します。

**omit** オペランド:

条件を満たさないレコードを選択します。

詳細については、「5.8 **-infile** パラメタ」の「5.8.16 {include | omit} オペランド」を参照してください。

### 形式

```
{include | omit} ="condition-expression" [△ {AND | OR} △  
condition-expression ... ] "
```

## 5.23.17 {csvinclude | csvomit} オペランド

### 意味

CSV ファイルの入力レコードの選択条件式を指定します。詳細については、「5.8 **-infile** パラメタ」の「5.8.17 {csvinclude | csvomit} オペランド」を参照してください。

### 形式

```
{csvinclude | csvomit} ="csv-condition-expression" [△ {AND | OR}  
△csv-condition-expression ... ] "
```

### 関連パラメタ

- **-splitfile**
- **-splitdata**
- **-splitkey**
- **-splitout**
- **-splittmpdir**

## 5.24 -splitout パラメタ

### 意味

ファイル分割機能の出力ファイル情報を指定します。

### 形式

```
-splitout
  [Δfileorg=organization-type]
  [Δ {suffix | prefix} = {output-generic-name
                        | output-directory-name} ]
  [Δreclen=record-length]
  [Δminlen=minimum-record-length]
  [Δrecsepa=record-separator-type]
  [Δ { cellsepa= {TAB | 'cell-separator'}
      | cellsepa= {TAB | 'cell-separator-in_1'}
                  Δ {TAB | 'cell-separator-out_1'}
                  [Δ {TAB | 'cell-separator-in_2'}
                    Δ {TAB | 'cell-separator-out_2'} ...] } ]
  [Δcellnum= {ADJUST | cell-number} ]
  [Δquottype=option-code]
  [Δpadding= {X | Z | C'padding-character' | X'hex-digit'} ]
  [Δ { reorg=reorganization-format
      [Δreorganization-format ... ]
      | csvreorg=csv-reorganization-format
        Δcsv-reorganization-format ... } ]
  [Δ { {include | omit} =condition-expression
      | {csvinclude | csvomit} =csv-condition-expression} ]
[-splitout]
```

- 入力ファイル情報は、`-splitinp` パラメタで指定してください。
- `-splitout` パラメタの指定を省略した場合、`prefix` オペランドにカレントディレクトリを指定したものと仮定します。
- `-splitout` パラメタは、`-outfile` パラメタおよび `-catout` パラメタと同時に指定できません。どれか一つだけ指定してください。
- 分割後の出力ファイル数が多い場合、システムの制限によってコマンドがエラーとなることがあります。ソート機能またはコピー機能使用時は、`-cmdopt` パラメタの `SEQOPNCLS` オプションを指定することで、このエラーを回避できることがあります。
- 分割処理終了時、終了メッセージに続いて、`KBLS1010-I` メッセージを分割したファイル数分、出力します。
- `KBLS1010-I` メッセージの出力を抑止する場合は、環境変数名「`EXSORTSUPPRESSSPLITMSG`」または「`EXSORT64SUPPRESSSPLITMSG`」に `YES` を指定してください。
- 出力レコード件数が 0 件のファイルを作成しない場合は、環境変数名「`EXSORTNOCREATEZEROFIELD`」または「`EXSORT64NOCREATEZEROFIELD`」に `YES` を指定してください。この場合、`KBLS1010-I` メッセージは出力しません。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- `-splitout` パラメタは複数指定できます。ただし、出力ファイル名の総称および出力ファイル格納ディレクトリ名に、ほかの `-splitout` パラメタで指定した名称と同じものは指定できません。また、出力ファイル名の総称および出力ファイル格納ディレクトリ名を省略した場合に名称を仮定しますが、この仮定した名称についても、同じものは指定できません。

### 注意事項

- `-cmdopt` パラメタの `OVWRT` オペランドと、環境変数名「`EXSORTNOCREATEZEROFILE`」または「`EXSORT64NOCREATEZEROFILE`」は、同時に指定できません。同時に指定した場合は `KBLS1107-E` メッセージでエラーとなります。
- 次に示す理由によって、ファイル分割後にファイル連結を行う場合、ファイル分割前に不要なファイルを削除してください。
  - 連結するファイル名には、総称名の指定が可能です。総称名に該当するファイルは、不要なファイルも含めてすべて連結の対象となります。そのため、不要なファイルを連結しないようにするには、不要なファイルを削除する必要があります。
  - 連結するファイル名には、ディレクトリの指定が可能です。ディレクトリの場合も不要なファイルを連結する可能性があります。

## 5.24.1 fileorg オペランド

### 意味

出力ファイルのファイル編成を指定します。詳細については、「5.10.1 fileorg オペランド」を参照してください。

### 形式

`fileorg=organization-type`

## 5.24.2 {suffix | prefix} オペランド

### 意味

`suffix` オペランド:

分割後の出力ファイル名の末尾、または出力ファイルを格納するディレクトリ名を指定します。

`prefix` オペランド:

分割後の出力ファイル名の先頭、または出力ファイルを格納するディレクトリ名を指定します。

### 形式

`{suffix|prefix}={output-generic-name | output-directory-name}`

### 指定値

**output-generic-name**

- 分割後の出力ファイル名の総称（先頭または末尾）を指定します。
- 出力ファイル名の総称（先頭または末尾）には、分割後の出力ファイルを格納するディレクトリ名とつなげて指定できます。
- UNIX の場合、**prefix** オペランドには、先頭がピリオド (.) で始まるファイル名を指定することはできません。

**output-directory-name**

- 分割後の出力ファイルを格納するディレクトリ名を指定します。
- 指定値の末尾はパス区切り記号<sup>※</sup>を指定してください。パス区切り記号の指定がない場合は、出力ファイル名の先頭として扱います。
- 分割後の出力ファイル名の先頭または末尾は、入力ファイル名となります。

## 注※

UNIX の場合はスラッシュ (/), Windows の場合は円記号 (¥) がパス区切り記号です。

**出力ファイル名の名称規則**

1. 分割キー以外による分割、または分割キーによる分割で環境変数「EXSORTSPLITNUMBONE (64 ビット版は「EXSORT64SPLITNUMBONE)」=YES 指定時  
suffix オペランド指定時

追番\_入力ファイル名または出力ファイル名の末尾

prefix オペランド指定時

入力ファイル名または出力ファイル名の先頭\_追番

2. 分割キーによる分割で、環境変数「EXSORTSPLITNUMBONE (64 ビット版は「EXSORT64SPLITNUMBONE)」指定なし  
suffix オペランド指定時

追番1\_追番2\_・・・\_追番n [recno追番\_] 入力ファイル名または出力ファイル名の末尾

prefix オペランド指定時

入力ファイル名または出力ファイル名の先頭\_追番1\_追番2\_・・・\_追番n [\_recno追番]

- 追番は、1 から分割したファイル数までの値を、8 桁<sup>※</sup>の固定長の数字で示します。値は分割ごとに +1 します。

## 注※

環境変数「EXSORTFILESPLITNUMB (64 ビット版は「EXSORT64FILESPLITNUMB)」で追番の桁数を指定できます。この環境変数を省略した時は、8 桁を仮定します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

- 追番 1, 追番 2, …追番 n は, 分割キーで分割した場合の追番となります。n は指定された分割キーまたは分割 CSV キーの数となります。
- recno 追番は, `-splitkey` パラメタの `recno` オペランドまたは `spno` オペランドを指定した場合の追番となります。
- 出力ファイル名長がシステムの上限を超過する場合は, KBL2021-E エラーとなります。

### 指定例

1. カレントディレクトリへ, ファイル名「出力ファイル名の先頭\_追番」のファイルを三つ出力。

```
<パラメタ>
-splitinp file=splitIN ...
-splitout※ prefix=splitOUT
-splitfile fileno=3
:
<カレントディレクトリ>
/sortee/exework1
<出力ファイル>
/sortee/exework1/splitOUT_00000001
/sortee/exework1/splitOUT_00000002
/sortee/exework1/splitOUT_00000003
```

### 注※

`-splitout` パラメタはファイル名の先頭を指定してください。

2. カレントディレクトリへ, ファイル名「追番\_出力ファイル名の末尾」のファイルを三つ出力。

```
<パラメタ>
-splitinp file=splitIN ...
-splitout※ suffix=splitOUT
-splitfile fileno=3
:
<カレントディレクトリ>
/sortee/exework1
<出力ファイル>
/sortee/exework1/00000001_splitOUT
/sortee/exework1/00000002_splitOUT
/sortee/exework1/00000003_splitOUT
```

### 注※

`-splitout` パラメタはファイル名の末尾を指定してください。

3. カレントディレクトリへ, ファイル名「入力ファイル名\_追番」のファイルを三つ出力。

```
<パラメタ>※
-splitinp file=splitIN ...
-splitfile fileno=3
:
<カレントディレクトリ>
/sortee/exework1
<出力ファイル>
/sortee/exework1/splitIN_00000001
/sortee/exework1/splitIN_00000002
/sortee/exework1/splitIN_00000003
```



## 注※

-splitout パラメタは省略してください。

4. 指定された出力ファイル格納ディレクトリへ、ファイル名「追番\_出力ファイル名の末尾」のファイルを二つ出力。

```
<パラメタ>
-splitinp file=splitIN ...※1
-splitout※2 suffix=/sortee/workout2/splitA
-splitdata recno=600
:
<カレントディレクトリ>
/sortee/exework1
<出力ファイル>
/sortee/workout2/00000001_splitA※3
/sortee/workout2/00000002_splitA
```

## 注※ 1

入力レコード件数は 1,000 件です。

## 注※ 2

-splitout パラメタはファイル名の末尾をディレクトリ名と一緒に指定してください。

## 注※ 3

レコード数 600 件と、400 件の分割ファイルを作成します。

5. 指定された出力ファイル格納ディレクトリへ、ファイル名「追番\_入力ファイル名」のファイルを二つ出力。

```
<パラメタ>
-splitinp file=splitIN ...※1
-splitout※2 suffix=/sortee/workout1/
-splitdata spno=1G
:
<カレントディレクトリ>
/sortee/exework1
<出力ファイル>
/sortee/workout2/00000001_splitIN※3
/sortee/workout2/00000002_splitIN
```

## 注※ 1

入力ファイルサイズは 2GB です。

## 注※ 2

-splitout パラメタはディレクトリ名を指定してください。

## 注※ 3

ファイルサイズ 1GB の分割ファイルを二つ出力します。

6. 指定された出力ファイル格納ディレクトリへ、ファイル名「追番 1\_追番

## 5. exsort コマンドのパラメタおよびオペランドの詳細

2\_recno 追番\_入力ファイル名」のファイルを六つ出力。

```
<パラメタ>
-splitinp file=splitIN ...
-splitkey key=c+0-4 j+8-4 recno=100
-splitout※1 suffix=/sortee/workout2/
<カレントディレクトリ>
/sortee/exework1
<出力ファイル>
/sortee/workout2/00000001_00000001_00000001_splitIN
/sortee/workout2/00000001_00000001_00000002_splitIN※2
/sortee/workout2/00000002_00000001_00000001_splitIN※3
/sortee/workout2/00000003_00000001_00000001_splitIN
/sortee/workout2/00000003_00000002_00000001_splitIN※4
/sortee/workout2/00000003_00000002_00000002_splitIN
```

### 注※ 1

-splitout パラメタはディレクトリ名を指定してください。

### 注※ 2

recno よる recno 追番更新。

### 注※ 3

第 1 分割キーによる追番 1 更新。

### 注※ 4

第 2 分割キーによる追番 2 更新。

### 注意事項

- suffix および prefix オペランドを省略した場合、出力ファイルディレクトリにカレントディレクトリを指定したものと仮定します。ファイル名「入力ファイル名\_追番」の分割ファイルを、カレントディレクトリへ出力します。
- 末尾にパス区切り記号<sup>※</sup>が指定された場合は出力ファイル格納ディレクトリ名、パス区切り記号以外は出力ファイル名の総称指定として扱います。したがって、出力ファイル名の総称の末尾にパス区切り記号は指定できません。
- 出力ファイル格納ディレクトリ名が存在しなかった場合、KBLS2021-E メッセージでエラーとなります。
- 追番の値が上限（8桁の場合は 99,999,999）に達した場合、KBLS2020-I メッセージを出力し、それ以降の分割は行いません。残りのレコードは最後の追番（8桁の場合は「99999999」）のファイルにすべて出力します。
- 出力ファイル名にファイル拡張子は付加されません。

### 注※

UNIX の場合はスラッシュ (/)、Windows の場合は円記号 (¥) がパス区切り記号です。

### 5.24.3 reclen オペランド

#### 意味

出力レコード長（レコード編集後のレコード長）を指定します。可変長レコードの場合は最大レコード長を指定します。詳細については、「5.10 -outfile パラメタ」の「5.10.3 reclen オペランド」を参照してください。

#### 形式

`reclen=record-length`

### 5.24.4 minlen オペランド

#### 意味

出力レコード長（レコード編集後の最小レコード長）を指定します。詳細については、「5.10 -outfile パラメタ」の「5.10.4 minlen オペランド」を参照してください。

#### 形式

`minlen=minimum-record-length`

### 5.24.5 recsepa オペランド

#### 意味

テキストファイルおよび CSV ファイルの出力レコードの改行文字を指定します。詳細については、「5.10 -outfile パラメタ」の「5.10.5 recsepa オペランド」を参照してください。

#### 形式

`recsepa=record-separator-type`

### 5.24.6 cellsepa オペランド

#### 意味

CSV ファイルの出力レコードのデリミタ（セル区切り文字）を指定します。詳細については、「5.10 -outfile パラメタ」の「5.10.6 cellsepa オペランド」を参照してください。

#### 形式

##### オペランド形式 1

`cellsepa=cell-separator`

##### オペランド形式 2

`cellsepa=cell-separator-in_1△cell-separator-out_1  
[△cell-separator-in_2△cell-separator-out_2 ...]`

## 5.24.7 cellnum オペランド

### 意味

レコード出力時のセル数調整を指定します。詳細については、「5.10 `-outfile` パラメタ」の「5.10.7 `cellnum` オペランド」を参照してください。

### 形式

```
cellnum= { ADJUST | cell-number }
```

## 5.24.8 quottype オペランド

### 意味

レコード出力時のセルの引用符を指定します。詳細については、「5.10 `-outfile` パラメタ」の「5.10.8 `quottype` オペランド」を参照してください。

### 形式

```
quottype=option-code
```

## 5.24.9 padding オペランド

### 意味

出力レコード長をそろえるためのパディング文字を指定します。詳細については、「5.10 `-outfile` パラメタ」の「5.10.9 `padding` オペランド」を参照してください。

### 形式

```
padding= { X | Z | C'padding-character' | X'16進数' }
```

## 5.24.10 reorg オペランド

### 意味

出力レコード編集フォーマットを指定します。詳細については、「5.10 `-outfile` パラメタ」の「5.10.10 `reorg` オペランド」を参照してください。

### 形式

```
reorg=reorganization-format [ $\Delta$ reorganization-format...]
```

## 5.24.11 csvreorg オペランド

### 意味

CSV ファイルの出力レコード編集フォーマットを指定します。詳細については、「5.10 `-outfile` パラメタ」の「5.10.11 `csvreorg` オペランド」を参照してください。

### 形式

```
csvreorg=csv-reorganization-format [ $\Delta$ csv-reorganization-format
```

… ]

## 5.24.12 {include | omit} オペランド

### 意味

出力レコードの選択条件式を指定します。詳細については、「5.10 -outfile パラメタ」の「5.10.12 {include | omit} オペランド」を参照してください。

include オペランド:

条件を満たすレコードを選択します。

omit オペランド:

条件を満たさないレコードを選択します。

### 形式

```
{include | omit} ="condition-expression [△ {AND | OR} △
condition-expression … ] "
```

## 5.24.13 {csvinclude | csvomit} オペランド

### 意味

CSV ファイルの出力レコードの選択条件式を指定します。詳細については、「5.10 -outfile パラメタ」の「5.10.13 {csvinclude | csvomit} オペランド」を参照してください。

### 形式

```
{csvinclude | csvomit} ="csv-condition-expression [△ {AND | OR}
△csv-condition-expression … ] "
```

### 関連パラメタ

- -splitfile
- -splitdata
- -splitkey
- -splitinp
- -splittmpdir

## 5.25 -splittmpdir パラメタ

---

### 意味

ファイル分割用一時ファイルのディレクトリを指定します。

### 形式

`-splittmpdir split-temporary-directory`

- 均等分割情報 (`-splitfile` パラメタ指定) でファイル分割する場合、分割途中の情報を一時ファイルに出力します。この一時ファイルを作成するディレクトリを指定します。
- 空白文字を含んだディレクトリ名を指定する場合は、ディレクトリ名をダブルクォーテーションで囲んで指定してください。
- 指定するディレクトリには、`exsort` コマンドの実行ユーザの読み込み、書き込み権限が必要です。Windows の場合は削除権限も必要です。
- ソート機能の指定があり、かつ `-splittmpdir` パラメタを省略した場合、次の優先順位でディレクトリが設定されます。
  1. 環境変数「`EXSORTSPLITTMPDIR` (64 ビット版は「`EXSORT64SPLITTMPDIR`)」に設定されたディレクトリ
  2. `-sorttmpdir` パラメタで決定したソート用一時ファイルのディレクトリ
- マージ機能またはコピー機能の指定があり、`-splittmpdir` パラメタを省略した場合は、環境変数「`EXSORTSPLITTMPDIR` (64 ビット版は「`EXSORT64SPLITTMPDIR`)」に設定されたディレクトリが設定されます。

### 関連パラメタ

- `-splitfile`
- `-splitdata`
- `-splitkey`
- `-splitinp`
- `-splitout`

## 5.26 -catinp パラメタ

### 意味

ファイル連結機能の入力ファイル情報を指定します。

### 形式

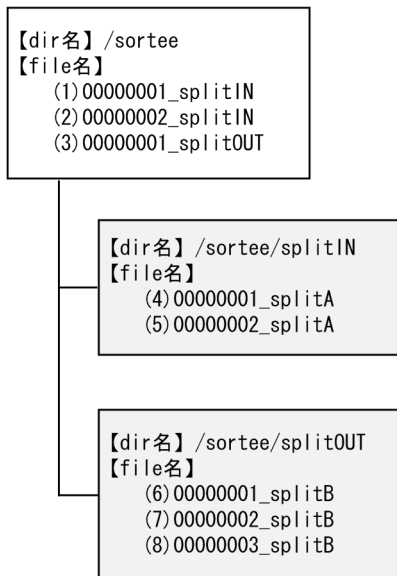
```
-catinp
  Δfileorg=organization-type
  Δ {file=file-name [Δfile-name...]
    | {suffix | prefix}
    = {input-generic-name | input-directory-name}
      [Δ {input-generic-name | input-directory-name} ...]
      [Δnameseq=input-order-code] }
  [Δreclen=record-length]
  [Δ {skiprec=skip-record-number
    | startrec=start-record-number} ]
  [Δstopaft=stop-record-number]
  [Δrecsepa=record-separator-type]
  [Δcellsepa= {TAB | 'cell-separator'}
    [Δ {TAB | 'cell-separator'} ... ] ]
  [Δquottype=quotation-type]
  [Δcurrency=currency-symbol]
  [Δround=option-code]
  [Δnullseq=option-code]
  [Δcodetype=encode]
  [Δ { reorg=reorganization-format
    [Δreorganization-format ... ]
    | csvreorg=csv-reorganization-format
      [Δcsv-reorganization-format ... ] } ]
  [Δ { {include | omit} =condition-expression
    | {csvinclude | csvomit} =csv-condition-expression} ]
```

- 出力ファイル情報は、`-catout` パラメタで指定してください。
- `-catinp` パラメタは、`-inpf` パラメタおよび `-splitinp` パラメタと同時に指定できません。どれか一つだけ指定してください。
- 連結する入力ファイル数が多い場合、システムの制限によってコマンドがエラーとなることがあります。ソート機能またはコピー機能使用時は、`-cmdopt` パラメタの `SEQOPNCLS` オプションを指定することで、このエラーを回避することができます。
- 連結処理終了時、終了メッセージにつづき、`KBLS1011-I` メッセージを入力したファイル数分、出力します。
- `KBLS1011-I` メッセージの出力を抑止する場合は、環境変数名「`EXSORTSUPPRESSCATMSG`」または「`EXSORT64SUPPRESSCATMSG`」に `YES` を指定してください。

### 指定例

## 5. exsort コマンドのパラメタおよびオペランドの詳細

<入力ファイル一覧>



1. ディレクトリ '/sortee/splitIN/' に格納されているファイル名「00000001\_splitA」を入力とする。

```
<パラメタ>
file=/sortee/splitIN/00000001_splitA
<対象入力ファイル>
(4) 00000001_splitA
```

2. ディレクトリ '/sortee/' に格納されている末尾が 'splitIN' のファイルを入力とする。  
入力ファイルの末尾が指定されたため、ディレクトリ '/sortee/splitIN/' は対象外となり、また末尾の異なる (3) も対象外となります。

```
<パラメタ>
suffix=/sortee/splitIN
<対象入力ファイル>
(1) 00000001_splitIN
(2) 00000002_splitIN
```

3. ディレクトリ '/sortee/splitIN/' に格納されているファイルを入力とする。  
パラメタに prefix=/sortee/splitIN/ と指定しても同じ結果となります。

```
<パラメタ>
suffix=/sortee/splitIN/
<対象入力ファイル>
(4) 00000001_splitA
(5) 00000002_splitA
```

4. ディレクトリ '/sortee/' に格納されている先頭が '00000001' のファイルを入力とする。



```

<パラメタ>
prefix=/sortee/00000001
<対象入力ファイル>
(1) 00000001_splitIN
(3) 00000001_splitOUT

```

## 5.26.1 fileorg オペランド

### 意味

入力ファイルのファイル編成を指定します。詳細については、「5.8.1 fileorg オペランド」を参照してください。

### 形式

fileorg=organization-type

## 5.26.2 file オペランド

### 意味

入力ファイル名または環境変数名を指定します。詳細については、「5.8 -inpfile パラメタ」の「5.8.2 file オペランド」を参照してください。

### 形式

file=file-name [ $\Delta$ file-name…]

### 注意事項

- file オペランドは、suffix オペランドおよび prefix オペランドと同時に指定できません。どれか一つだけ指定してください。
- 環境変数名「EXSORTCATINPNOERR」または「EXSORT64CATINPNOERR」に YES が指定されていた場合、指定した入力ファイルの一部が存在しない場合でも、連結処理は続行します。存在しなかった入力ファイル名は、KBLS2100-I メッセージで確認してください。ただし、指定したすべてのファイルが存在しなかった場合は、KBLS2100-I メッセージは出力しないで、指定した先頭のファイルに対する KBLS1635-E メッセージだけを出力します。
- 環境変数名「EXSORTCATINPNOERR」または「EXSORT64CATINPNOERR」に YES が指定されている以外の場合は、指定した入力ファイルが一つでも存在しないときは、KBLS1635-E メッセージを出力し、処理は終了します。

## 5.26.3 {suffix | prefix} オペランド

### 意味

suffix オペランド：

連結する入力ファイルを格納するディレクトリ名、または連結する入力ファイル名の末尾を指定します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

**prefix** オペランド :

連結する入力ファイルを格納するディレクトリ名, または連結する入力ファイル名の先頭を指定します。

形式

```
{suffix | prefix} = {input-generic-name | input-directory-name}
[△ {input-generic-name | input-directory-name} ...]
```

指定値

*input-generic-name*

- 入力ファイル名の総称 (先頭または末尾) を指定します。
- 入力ファイル名の総称 (先頭または末尾) には, 入力ファイルが格納されているディレクトリ名とつなげて指定できます。
- UNIX の場合, **prefix** オペランドには, 先頭がピリオド (.) で始まるファイル名を指定できません。

*input-directory-name*

- 入力ファイルを格納するディレクトリ名を指定します。
- 指定値の末尾はパス区切り記号<sup>※</sup>を指定してください。パス区切り記号の指定がない場合は, 入力ファイル名の総称として扱います。
- 指定したディレクトリ下にある入力ファイルをすべて含みます。ただし, サブディレクトリ下のファイルは入力ファイルに含みません。

注<sup>※</sup>

UNIX の場合はスラッシュ (/), Windows の場合は円記号 (¥) がパス区切り記号です。

注意事項

- **suffix** または **prefix** オペランドは, **file** オペランドと同時に指定できません。どれか一つだけ指定してください。
- **suffix** または **prefix** オペランドで指定された入力ファイル格納ディレクトリ, または入力ファイルの総称 (先頭または末尾) に該当するファイルが一つも存在しなかった場合は, **KBLS1635-E** メッセージでエラー終了します。
- 入力ファイル名の総称, または入力ファイル格納ディレクトリ名を複数指定した場合, 指定された順にファイル名を取得します。UNIX の場合, 取得した入力ファイル名を, **exsort** コマンド実行環境に設定されているロケールに従って昇順にソートし入力します。Windows の場合, **nameseq** オペランドで指定した入力順に, ファイルを入力します。なお, **KBLS1011-I** メッセージは, 処理した入力ファイル順に出力します。
- UNIX の場合, ピリオド (.) で始まる名前のファイルは, ファイル連結の対象になりません。
- Windows の場合, 隠しファイルはファイル連結の対象になりません。
- **suffix** または **prefix** オペランドを指定した場合, 次の例に示すように解釈し動作します。なお, 例中の「\*」はすべての文字列が対象となる意味です。

**prefix=A 指定時**

「A\*」と解釈し、該当ファイルを検索します。例えば、「A」や「ABC」が該当します。

**suffix=A 指定時**

「\*A」と解釈し、該当ファイルを検索します。例えば、「A」や「CBA」が該当します。

**prefix=.A 指定時**

「\*.A」と解釈し、該当ファイルを検索します。例えば、「.A」や「.ABC」が該当します。なお、UNIX の場合は、先頭がピリオド (.) から始まるファイル名を指定することはできません。指定した場合は KBL52101-E エラーとなります。

**suffix=.A 指定時**

「\*.A」と解釈し、該当ファイルを検索します。例えば、「A.A」や「B.A」が該当します。なお、UNIX の場合は、「.A」など「.」から始まる名前のファイルは対象となりません。

**prefix=A. 指定時**

「A.\*」と解釈し、該当ファイルを検索します。例えば、「A.」や「A.BC」が該当します。

**suffix=A. 指定時**

「\*A.」と解釈し、該当ファイルを検索します。例えば、「A.」や「CA.」が該当します。

## 5.26.4 nameseq オペランド (Windows の場合)

**意味**

suffix または prefix オペランドを指定した場合の入力ファイルの入力順を、0 から 3 までの数字で指定します。このオペランドを省略した場合、0 を設定します。

**形式**

`nameseq=input-order-code`

**指定値**

*input-order-code*

0

ファイル名のソートは行わないで、該当するファイル名を獲得した順番に入力します。この場合、ファイルシステムが異なると、獲得順も異なる場合があります。ファイルの入力順を決める必要がない場合に指定します。

1

exsort コマンド実行環境に設定されているロケールに従ってファイル名を昇順にソートして入力します。この場合、OS が異なると、ソート結果が異なること

## 5. exsort コマンドのパラメタおよびオペランドの詳細

があります。次のように、カタカナとひらがなを区別したくない場合などに指定します。

表 5-22 exsort コマンド実行環境に設定されているロケールに従ってファイル名を昇順にソートする例

ソート前	ソート後
集計_あいち_2011_03_#1	集計_あいち_2011_03_#1
集計_とうきょう_2011_03_#2	集計_アイチ_2011_03_#2
集計_やまなし_2011_03_#2	集計_トウキョウ_2011_03_#1
集計_アイチ_2011_03_#2	集計_とうきょう_2011_03_#2
集計_トウキョウ_2011_03_#1	集計_ヤマナシ_2011_03_#1
集計_ヤマナシ_2011_03_#1	集計_やまなし_2011_03_#2

### 2

Shift\_JIS の文字コードの値で先頭から 1 バイトずつ比較し、ファイル名を昇順にソートし、入力します。次のように、英字の大文字と小文字を区別したい場合などに指定します。

表 5-23 Shift\_JIS の文字コードの値で先頭から 1 バイトずつ比較してファイル名を昇順にソートする例

ソート前	ソート後
精算_TOKYO_A-G 区	精算_TOKYO_A-G 区
精算_TOKYO_a-g 市	精算_TOKYO_H-Q 区
精算_TOKYO_H-Q 区	精算_TOKYO_R-Z 区
精算_TOKYO_h-q 市	精算_TOKYO_a-g 市
精算_TOKYO_R-Z 区	精算_TOKYO_h-q 市
精算_TOKYO_r-z 市	精算_TOKYO_r-z 市

### 3

数字は、数値として扱い、ファイル名を昇順にソートし、入力します。例えば、「007」と「7」は同値として扱います。この場合、OS が異なると、ソート結果が異なることがあります。このソートは、Windows 2000 では指定できません。指定した場合、KBLS2106-E エラーとなります。次のように数字の桁数が異なるものが混在していても、年月日順や項番順にしたい場合などに指定します。

表 5-24 数字を数値として扱いファイル名を昇順にソートする例

ソート前	ソート後
決算_2011年_06月_東京_no15	決算_2011年_6月_東京_no02

ソート前	ソート後
決算_2011年_06月_東京_no3	決算_2011年_06月_東京_no3
決算_2011年_12月_東京_no1	決算_2011年_6月_東京_no004
決算_2011年_6月_東京_no004	決算_2011年_06月_東京_no15
決算_2011年_6月_東京_no02	決算_2011年_12月_東京_no1

### 5.26.5 reclen オペランド

#### 意味

入力レコード長を指定します。詳細については、「5.8 -inpfile パラメタ」の「5.8.3 reclen オペランド」を参照してください。

#### 形式

`reclen=record-length`

### 5.26.6 skiprec オペランド

#### 意味

入力レコードを読み飛ばす件数を指定します。詳細については、「5.8 -inpfile パラメタ」の「5.8.4 skiprec オペランド」を参照してください。

#### 形式

`skiprec=skip-record-number`

### 5.26.7 stopaft オペランド

#### 意味

入力を打ち切るレコード件数を指定します。詳細については、「5.8 -inpfile パラメタ」の「5.8.5 stopaft オペランド」を参照してください。

#### 形式

`stopaft=stop-record-number`

### 5.26.8 startrec オペランド

#### 意味

データの入力を開始するレコード番号を指定します。詳細については、「5.8 -inpfile パラメタ」の「5.8.6 startrec オペランド」を参照してください。

#### 形式

`startrec=start-record-number`

## 5.26.9 recsepa オペランド

### 意味

テキストファイルおよび CSV ファイルの入力レコードの改行文字を指定します。詳細については、「5.8 `-inpfile` パラメタ」の「5.8.7 `recsepa` オペランド」を参照してください。

### 形式

`recsepa=record-separator-type`

## 5.26.10 cellsepa オペランド

### 意味

CSV ファイルの入力レコードのデリミタ（セル区切り文字）を指定します。詳細については、「5.8 `-inpfile` パラメタ」の「5.8.8 `cellsepa` オペランド」を参照してください。

### 形式

`cellsepa=cell-separator` [`△cell-separator …`]

## 5.26.11 quottype オペランド

### 意味

セルの引用符文字を指定します。詳細については、「5.8 `-inpfile` パラメタ」の「5.8.9 `quottype` オペランド」を参照してください。

### 形式

`quottype=quotation-type`

## 5.26.12 currency オペランド

### 意味

CSV ファイルの固定小数点数の数値データの通貨記号として、任意の 1 文字を指定します。詳細については、「5.8 `-inpfile` パラメタ」の「5.8.10 `currency` オペランド」を参照してください。

### 形式

`currency=currency-symbol`

## 5.26.13 round オペランド

### 意味

CSV ファイルのすべての固定小数点数のキー、比較項目、または集約項目の桁の丸め方を指定します。詳細については、「5.8 `-inpfile` パラメタ」の「5.8.11 `round` オペランド」を参照してください。

形式

`round=option-code`

## 5.26.14 nullseq オペランド

意味

CSV ファイルのナル文字列が含まれる入力レコードを、先頭または末尾に配置するオプションです。詳細については、「5.8 `-inpfiler` パラメタ」の「5.8.12 `nullseq` オペランド」を参照してください。

形式

`nullseq=option-code`

## 5.26.15 codetype オペランド

意味

入力データの文字コードの種類を指定します。詳細については、「5.8 `-inpfiler` パラメタ」の「5.8.13 `codetype` オペランド」を参照してください。

形式

`codetype=encode`

## 5.26.16 reorg オペランド

意味

入力レコード編集フォーマットを指定します。詳細については、「5.8 `-inpfiler` パラメタ」の「5.8.14 `reorg` オペランド」を参照してください。

形式

`reorg=reorganization-format [△reorganization-format…]`

## 5.26.17 csvreorg オペランド

意味

CSV ファイルの入力レコード編集フォーマットを指定します。詳細については、「5.8 `-inpfiler` パラメタ」の「5.8.15 `csvreorg` オペランド」を参照してください。

形式

`csvreorg=csv-reorganization-format [△csv-reorganization-format … ]`

## 5.26.18 {include | omit} オペランド

意味

入力レコードの選択条件式を指定します。

## 5. exsort コマンドのパラメタおよびオペランドの詳細

**include** オペランド :

条件を満たすレコードを選択します。

**omit** オペランド :

条件を満たさないレコードを選択します。

詳細については、「5.8 `-infile` パラメタ」の「5.8.16 `{include | omit}` オペランド」を参照してください。

**形式**

```
{include | omit} ="condition-expression [△ {AND | OR} △  
condition-expression ... ] "
```

### 5.26.19 `{csvinclude | csvomit}` オペランド

**意味**

CSV ファイルの入力レコードの選択条件式を指定します。詳細については、「5.8 `-infile` パラメタ」の「5.8.17 `{csvinclude | csvomit}` オペランド」を参照してください。

**形式**

```
{csvinclude | csvomit} ="csv-condition-expression [△ {AND | OR}  
△csv-condition-expression ... ] "
```

**関連パラメタ**

- `-catout`



## 5.27 -catout パラメタ

### 意味

ファイル連結機能の出力ファイル情報を指定します。

### 形式

```
-catout
  [△fileorg=organization-type]
  [△file=file-name [△file-name...]]
  [△reclen=record-length]
  [△minlen=minimum-record-length]
  [△recsepa=record-separator-type]
  [△ { cellsepa= {TAB | 'cell-separator'}
    | cellsepa= {TAB | 'cell-separator-in_1'}
      Δ {TAB | 'cell-separator-out_1'}
        [△ {TAB | 'cell-separator-in_2'}
          Δ {TAB | 'cell-separator-out_2'} ... ] } ]
  [△cellnum= {ADJUST | cell-number} ]
  [△quottype=option-code]
  [△padding= {X | Z | C'padding-character' | X'hex-digit'} ]
  [△ { reorg=reorganization-format
    [△reorganization-format ... ]
    | csvreorg=csv-reorganization-format
      [△csv-reorganization-format ... ] } ]
  [△ { {include | omit} =condition-expression
    | {csvinclude | csvomit} =csv-condition-expression} ]
[△-catout ...]
```

- 入力ファイル情報は、`-catout` パラメタで指定してください。
- `-catout` パラメタは、`-outfile` パラメタおよび `-splitout` パラメタと同時に指定できません。どれか一つだけ指定してください。
- 処理終了時に出力する終了メッセージは、`-catout` パラメタを指定した数だけ出力されます。
- `-catout` パラメタは複数指定できます。ただし、出力ファイル名に、ほかの `-catout` パラメタで指定した名称と同じものは指定できません。

### 5.27.1 fileorg オペランド

#### 意味

出力ファイルのファイル編成を指定します。詳細については、「5.10.1 fileorg オペランド」を参照してください。

#### 形式

```
fileorg=organization-type
```

### 5.27.2 file オペランド

#### 意味

## 5. exsort コマンドのパラメタおよびオペランドの詳細

出力ファイル名または環境変数名を指定します。詳細については、「5.10 `-outfile` パラメタ」の「5.10.2 `file` オペランド」を参照してください。

### 形式

`file=file-name` [ $\Delta$ `file-name`…]

### 注意事項

- このパラメタの指定を省略した場合、標準出力ファイルへ出力します。

## 5.27.3 `reclen` オペランド

### 意味

出力レコード長（レコード編集後のレコード長）を指定します。可変長レコードの場合は最大レコード長を指定します。詳細については、「5.10 `-outfile` パラメタ」の「5.10.3 `reclen` オペランド」を参照してください。

### 形式

`reclen=record-length`

## 5.27.4 `minlen` オペランド

### 意味

出力レコード長（レコード編集後の最小レコード長）を指定します。詳細については、「5.10 `-outfile` パラメタ」の「5.10.4 `minlen` オペランド」を参照してください。

### 形式

`minlen=minimum-record-length`

## 5.27.5 `recsepa` オペランド

### 意味

テキストファイルおよび CSV ファイルの出力レコードの改行文字を指定します。詳細については、「5.10 `-outfile` パラメタ」の「5.10.5 `recsepa` オペランド」を参照してください。

### 形式

`recsepa=record-separator-type`

## 5.27.6 `cellsepa` オペランド

### 意味

CSV ファイルの出力レコードのデリミタ（セル区切り文字）を指定します。詳細については、「5.10 `-outfile` パラメタ」の「5.10.6 `cellsepa` オペランド」を参照してください。

**形式****オペランド形式 1**

```
cellsepa=cell-separator
```

**オペランド形式 2**

```
cellsepa=cell-separator-in_1△cell-separator-out_1  
[△cell-separator-in_2△cell-separator-out_2 … ]
```

## 5.27.7 cellnum オペランド

**意味**

レコード出力時のセル数調整を指定します。詳細については、「5.10 -outfile パラメタ」の「5.10.7 cellnum オペランド」を参照してください。

**形式**

```
cellnum= { ADJUST | cell-number }
```

## 5.27.8 quottype オペランド

**意味**

レコード出力時のセルの引用符を指定します。詳細については、「5.10 -outfile パラメタ」の「5.10.8 quottype オペランド」を参照してください。

**形式**

```
quottype=option-code
```

## 5.27.9 padding オペランド

**意味**

出力レコード長をそろえるためのパディング文字を指定します。詳細については、「5.10 -outfile パラメタ」の「5.10.9 padding オペランド」を参照してください。

**形式**

```
padding= { X | Z | C'padding-character' | X'16進数' }
```

## 5.27.10 reorg オペランド

**意味**

出力レコード編集フォーマットを指定します。詳細については、「5.10 -outfile パラメタ」の「5.10.10 reorg オペランド」を参照してください。

**形式**

```
reorg=reorganization-format [△reorganization-format…]
```

## 5.27.11 csvreorg オペランド

### 意味

CSV ファイルの出力レコード編集フォーマットを指定します。詳細については、「5.10 -outfile パラメタ」の「5.10.11 csvreorg オペランド」を参照してください。

### 形式

```
csvreorg=csv-reorganization-format [△csv-reorganization-format  
... ]
```

## 5.27.12 {include | omit} オペランド

### 意味

出力レコードの選択条件式を指定します。

**include** オペランド:

条件を満たすレコードを選択します。

**omit** オペランド:

条件を満たさないレコードを選択します。

詳細については、「5.10 -outfile パラメタ」の「5.10.12 {include | omit} オペランド」を参照してください。

### 形式

```
{include | omit} = "condition-expression [△ {AND | OR} △  
condition-expression ... ] "
```

## 5.27.13 {csvinclude | csvomit} オペランド

### 意味

CSV ファイルの出力レコードの選択条件式を指定します。詳細については、「5.10 -outfile パラメタ」の「5.10.13 {csvinclude | csvomit} オペランド」を参照してください。

### 形式

```
{csvinclude | csvomit} = "csv-condition-expression [△ {AND | OR}  
△csv-condition-expression ... ] "
```

### 関連パラメタ

- -catinp

## 5.28 -cpyinp パラメタ

### 意味

簡易コピー機能で使用する入力ファイル情報を指定します。

### 形式

```
-cpyinp fileorg=organization-type
      [ file-name [ $\Delta$ file-name ... ] ]
```

### 注意事項

- `-function copy` を必ず指定しなければなりません。また、`-function sort`、および `-function merge` と同時に指定することはできません。
- `-p` パラメタと同時に指定できません。指定した場合は、KBSL1106-E エラーとなります。
- `-inpfile`、`-catinp`、および `-splitinp` パラメタと同時に指定できません。指定した場合は、KBSL1106-E エラーとなります。
- 出力ファイル情報は、`-cpyout` パラメタで指定してください。

### 5.28.1 fileorg オペランド

#### 意味

入力ファイルのファイル編成を指定します。詳細については、「5.8.1 fileorg オペランド」を参照してください。

#### 形式

```
fileorg=organization-type
```

#### 指定値

`organization-type` に指定する値を次の表に示します。

指定値	意味
S	可変長順編成ファイル、テキストファイル、および CSV ファイル以外のファイルを指定します。
V	可変長順編成ファイルを指定します。
A	テキストファイルを指定します。
C	CSV ファイルを指定します。

#### 注意事項

- このオペランドは省略できません。
- 複数の入力ファイルを指定する場合は、指定するファイルをすべて同じファイル編成にする必要があります。
- 可変長順ファイルを指定した場合は、ファイル形式がチェックされます。
- テキストファイルおよび CSV ファイルを指定した場合は、Unicode シグニチャ

## 5. exsort コマンドのパラメタおよびオペランドの詳細

(BOM) の読み飛ばしと付加機能が有効となります。Unicode シグニチャ関連オプションについては、「2.35 Unicode シグニチャ (BOM) 関連オプション (SORT EE の機能)」を参照してください。

- 可変長順ファイル、テキストファイルおよび CSV ファイル以外をコピーする場合は、fileorg オペランドに「S」を指定してください。

### 5.28.2 入力ファイル名

#### 意味

入力ファイル名、または環境変数名を指定します。

#### 形式

`file-name` [ $\Delta$ `file-name` ... ]

#### 注意事項

- `-inpf` パラメタのように、オペランド名「file=」は不要です。直接、入力ファイル名を指定します。
- このオペランドは省略できます。省略した場合は、標準入力ファイルを入力ファイルとします。
- 入力ファイル名を複数指定する場合は、1 個以上の空白またはタブで区切って指定します。
- 同じファイル名のファイルを複数指定できません。
- 空白文字を含んだファイル名を指定する場合は、ファイル名ごとにダブルクォーテーションで囲んで指定してください。ファイル名の指定規則については、「4.3.5 ファイル名およびディレクトリ名の指定形式」を参照してください。
- 指定できるファイル数に制限はありません。
- 出力ファイル名と同じファイル名は指定できません。
- 通常のコピー機能では、指定されたすべての入力ファイルを一度にオープンしますが、簡易コピー機能では、入力するファイルごとにオープンします。SORT EE, SORT, COBOL, および ISAM 間での入力ファイルの共有・排他機能は、ファイルがオープンされている間だけ有効であるため、簡易コピーを実行中であっても、オープンされていない入力ファイルの内容は保証できないため注意してください。

## 5.29 -cpyout パラメタ

---

### 意味

簡易コピー機能で使用する出力ファイル情報を指定します。

### 形式

```
-cpyout [ file-name [△file-name … ] ]
```

### 注意事項

- `-p` パラメタと同時に指定することはできません。指定した場合は、KBLS1106-E エラーとなります。
- `-outfile`, `-catout`, および `-splitout` パラメタと同時に指定することはできません。指定した場合は、KBLS1106-E エラーとなります。
- 入力ファイル情報は、`-cpyinp` パラメタで指定してください。
- `-cpyout` パラメタは、1 個だけ指定できます。複数指定した場合は、KBLS1101-E エラーとなります。

### 5.29.1 出力ファイル名

#### 意味

出力ファイル名、または環境変数名を指定します。

#### 形式

```
file-name [△file-name … ]
```

#### 注意事項

- `-outfile` パラメタのように、オペランド名「file=」は不要です。直接、出力ファイル名を指定します。
- このオペランドは省略できます。省略した場合は、標準出力ファイルを出力ファイルとします。
- 出力ファイル名を複数指定する場合は、1 個以上の空白またはタブで区切って指定します。
- 最大 4 個のファイル名を指定できます。
- 同じファイル名のファイルを複数指定できません。
- 入力ファイル名と同じファイル名は指定できません。
- 出力ファイル名を複数指定した場合、それぞれのファイルには同じレコードが出力されます。
- 空白文字を含んだファイル名を指定する場合は、ファイル名ごとにダブルクォーテーションで囲んで指定してください。ファイル名の指定規則については、「4.4.6 ファイル名およびディレクトリ名」を参照してください。





# 6

## SORT EE の使用例

この章では、SORT EE の使用例について説明します。また、altseq 変換ファイルの作成例について説明します。

---

6.1 サンプルファイルの格納ディレクトリ

---

6.2 ソート機能の使用例

---

6.3 マージ機能の使用例

---

6.4 コピー機能の使用例

---

6.5 altseq 変換ファイルの作成例

---

## 6.1 サンプルファイルの格納ディレクトリ

---

6.2 節以降の使用例で使用するサンプルファイルの格納ディレクトリを次に示します。

<サンプルファイルの格納ディレクトリ名>

Windows : *SORT EE* インストールディレクトリ¥Samples (32 ビット版または 64 ビット版) ※

UNIX : /opt/HIEXSORT/samples (32 ビット版)

        /opt/HIEXSORT64/samples (64 ビット版)

注※

Windows 版 SORT EE のデフォルトインストールディレクトリを次に示します。なお、「C:」はシステムがインストールされているドライブ名を示します。

- 32 ビット版 : C:¥Program Files¥Hitachi¥ExSORT
- 64 ビット版 : C:¥Program Files¥Hitachi¥ExSORT64

## 6.2 ソート機能の使用例

ソート機能の使用例を次に示します。

### 6.2.1 ソート

サンプルファイル名 : exsort\_sample1\_sort.txt

-function sort	(1)
-cmdopt DUP	(2)
-key g+4-8 rj+10-3	(3)
-inpfiler	
fileorg=A	(4)
file=INFILE	(5)
reclen=64	(6)
-outfile	
fileorg=A	(7)
file=OUTFILE	(8)
-sorttmpdir sorttemp	(9)
-worksize 10	(10)
-dateopt 75	(11)

- ソート機能を使用する (1)
- キーの比較は、JIS8 単位コードの体系に従う
- 重複レコードの入力順序保存機能を使用 (2)
- 第 1 キーでのソートは昇順、キーの属性は日付項目 (3)
- 第 1 キーの開始バイト位置は「4」バイト目、バイト長は「8」バイト (3)
- 第 2 キーでのソートは降順、キーの属性は文字 (英数字) (3)
- 第 2 キーの開始バイト位置は「10」バイト目、バイト長は「3」バイト (3)
- 入力ファイルおよび出力ファイルは、テキストファイル (4), (7)
- ソートする入力ファイルは「INFILE」 (5)
- 入力ファイルのレコード長は「64」バイト (6)
- ソート結果の出力ファイルは「OUTFILE」 (8)
- ソート用一時ファイルディレクトリは、カレントディレクトリ下の「sorttemp」 (9)
- ワークバッファのサイズは「10」MB (10)
- 日付項目の基準年は「75」 (11)

### 6.2.2 選択+ソート

サンプルファイル名 : exsort\_sample5\_sortsel.txt

## 6. SORT EE の使用例

<code>-function sort</code>	(1)
<code>-cmdopt OVWRT</code>	(2)
<code>-key c+0-4</code>	(3)
<code>-inpfile</code>	
<code>fileorg=S</code>	(4)
<code>file=INFILE1 INFILE2 INFILE3</code>	(5)
<code>reclen=64</code>	(6)
<code>omit=j+0-1 = C'0'</code>	(7)
<code>-outfile</code>	
<code>fileorg=S</code>	(8)
<code>file=OUTFILE</code>	(9)
<code>-sorttmpdir sorttemp</code>	(10)
<code>-worksize 10</code>	(11)

- ソート機能を使用する (1)
- キーの比較は JIS8 単位コードの体系に従う
- 出力ファイルがすでに存在する場合、上書き出力される (2)
- キーの属性は文字 (3)
- キーの開始バイト位置は「0」バイト目、バイト長は「4」バイト (3)
- 入力ファイルおよび出力ファイルは、固定長順編成ファイル (4), (8)
- 入力ファイルは「INFILE1」, 「INFILE2」, および「INFILE3」 (5)
- 入力ファイルのレコード長は「64」バイト (6)
- 選択条件を満たさないレコードだけを入力する (7)
- 比較項目の属性は文字 (7)
- 比較項目の開始バイト位置は「0」バイト目、バイト長は「1」バイト (7)
- 比較演算子は「=」 (7)
- 条件値は文字「0」 (7)
- 出力ファイルは「OUTFILE」 (9)
- ソート用一時ファイルディレクトリは、カレントディレクトリ下の「sorttemp」 (10)
- ワークバッファのサイズは「10」MB (11)

### 6.2.3 選択＋ソート＋集約

サンプルファイル名 : exsort\_sample7\_sortselsum.txt

<code>-function sort</code>	(1)
<code>-cmdopt OVWRT</code>	(2)
<code>-key c+4-2</code>	(3)
<code>-sum z+8-4</code>	(4)

-inpfile	
fileorg=S	(5)
file=INFILE1 INFILE2 INFILE3	(6)
reclen=50	(7)
include=z+16-4 > N'999'	(8)
-outfile	
fileorg=S	(9)
file=OUTFILE	(10)
-sorttmpdir sorttemp	(11)
-mergetmpdir mergetemp	(12)
-worksize 10	(13)

- ソート機能を使用する (1)
- キーの比較は JIS8 単位コードの体系に従う
- 出力ファイルがすでに存在する場合、上書き出力される (2)
- キーの属性は文字 (3)
- キーの開始バイト位置は「4」バイト目、バイト長は「2」バイト (3)
- 集約項目の属性は外部 10 進数 (4)
- 集約項目の開始バイト位置は「8」バイト目、バイト長は「4」バイト (4)
- 入力ファイルおよび出力ファイルは、固定長順編成ファイル (5), (9)
- 入力ファイルは「INFILE1」, 「INFILE2」, および「INFILE3」 (6)
- 入力ファイルのレコード長は「50」バイト (7)
- 選択条件を満たすレコードだけを入力する (8)
- 比較項目の属性は外部 10 進数 (8)
- 比較項目の開始バイト位置は「16」バイト目、バイト長は「4」バイト (8)
- 比較演算子は「>」 (8)
- 条件値は数値「999」 (8)
- 出力ファイルは「OUTFILE」 (10)
- ソート用一時ファイルディレクトリは、カレントディレクトリ下の「sorttemp」 (11)
- マージ用一時ファイルディレクトリは、カレントディレクトリ下の「mergetemp」 (12)
- ワークバッファのサイズは「10」MB (13)

## 6.2.4 ソート + altseq

サンプルファイル名 : exsort\_sample12\_altseq.txt

-function sort	(1)
-key q+16-8	(2)
-inpfile	
fileorg=S	(3)

## 6. SORT EE の使用例

file=INFILE1 INFILE2 INFILE3	(4)
reclen=128	(5)
-outfile	
fileorg=S	(6)
file=OUTFILE	(7)
-sorttmpdir sorttemp	(8)
-mergetmpdir mergetemp	(9)
-worksize 10	(10)
-altseq code=30F0 31F1 32F2 33F3 34F4 35F5 36F6 37F7 38F8 39F9	(11)

- ソート機能を使用する (1)
- キーの比較は (11) で指定した文字については (11) の順序に従い、これ以外の文字については JIS8 単位コードの体系に従う (2)
- キーの属性は文字 ((11) の文字順序変更機能で指定した文字) (2)
- キーの開始バイト位置は「16」バイト目、バイト長は「8」バイト (2)
- 入力ファイルおよび出力ファイルは、固定長順編成ファイル (3), (6)
- ソートする入力ファイルは「INFILE1」, 「INFILE2」, および「INFILE3」 (4)
- 入力ファイルのレコード長は「128」バイト (5)
- ソート結果の出力ファイルは「OUTFILE」 (7)
- ソート用一時ファイルディレクトリは、カレントディレクトリ下の「sorttemp」 (8)
- マージ用一時ファイルディレクトリは、カレントディレクトリ下の「mergetemp」 (9)
- ワークバッファのサイズは「10」MB (10)
- キーの比較の順序を変更する (11)

### 6.2.5 ソート+ csv ファイル

サンプルファイル名 : exsort\_sample13\_csv.txt

-function sort	(1)
-csvkey z+3-8,2 rc+2-10 g+1-8,1	(2)
-infile	
fileorg=C	(3)
file=INFILE	(4)
reclen=50	(5)
-outfile	
fileorg=C	(6)
file=OUTFILE	(7)
-sorttmpdir sorttemp	(8)

-worksize 10	(9)
--------------	-----

- ソート機能を使用する (1)
- CSV キーの比較は JIS8 単位コードの体系に従う (2)
- 第 1 キーでのソートは昇順, キーの属性は固定小数点数 (2)
- 第 1 キーのセル番号は「3」, 総桁数は「8」桁, 小数点以下の有効桁は「2」桁 (2)
- 第 2 キーでのソートは降順, キーの属性は文字 (英数字) (2)
- 第 2 キーのセル番号は「2」, 有効バイト長は「10」バイト (2)
- 第 3 キーでのソートは昇順, キーの属性は日付 (2)
- 第 3 キーのセル番号は「1」, 有効桁は「8」桁, 日付の書式は「1 (前置 4 桁年号)」 (2)
- 入力ファイルおよび出力ファイルは, CSV ファイル (3), (6)
- ソートする入力ファイルは「INFILE」 (4)
- 入力ファイルのレコード長は「50」バイト (5)
- ソート結果の出力ファイルは「OUTFILE」 (7)
- ソート用一時ファイルディレクトリは, カレントディレクトリ下の「sorttemp」 (8)
- ワークバッファのサイズは「10」MB (9)

## 6.2.6 ソート+均等分割

サンプルファイル名 : exsort\_sample14\_sortsplitfile.txt

-function sort	(1)
-cmdopt DUP	(2)
-key g+4-8 rj+10-3	(3)
-splitinp	
fileorg=A	(4)
file=INFILE	(5)
reclen=64	(6)
-splitout	
fileorg=A	(7)
prefix=OUTFILE	(8)
-sorttmpdir sorttemp	(9)
-splittmpdir splittemp	(10)
-splitfile	
fileno=10	(11)
method=recct	(12)
-worksize 10	(13)
-dateopt 75	(14)

## 6. SORT EE の使用例

- ソート機能を使用する (1)
- キーの比較は、JIS8 単位コードの体系に従う
- 重複レコードの入力順序保存機能を使用 (2)
- 第 1 キーでのソートは昇順、キーの属性は日付項目 (3)
- 第 1 キーの開始バイト位置は「4」バイト目、バイト長は「8」バイト (3)
- 第 2 キーでのソートは降順、キーの属性は文字 (英数字) (3)
- 第 2 キーの開始バイト位置は「10」バイト目、バイト長は「3」バイト (3)
- 入力ファイルおよび出力ファイルは、テキストファイル (4), (7)
- ソートする入力ファイルは「INFILE」 (5)
- 入力ファイルのレコード長は「64」バイト (6)
- ソート結果の出力ファイルは「OUTFILE」 (8)
- ソート用一時ファイルディレクトリは、カレントディレクトリ下の「sorttemp」 (9)
- ファイル分割用一時ファイルディレクトリは、カレントディレクトリ下の「splittemp」 (10)
- 分割ファイル数は「10」 (11)
- 分割基準はレコード件数 (12)
- ワークバッファのサイズは「10」MB (13)
- 日付項目の基準年は「75」 (14)

### 6.2.7 ソート+定数分割

サンプルファイル名 : exsort\_sample15\_sortsplitdata.txt

-function sort	(1)
-cmdopt DUP	(2)
-key g+4-8 rj+10-3	(3)
-splitinp	
fileorg=A	(4)
file=INFILE	(5)
reclen=64	(6)
-splitout	
fileorg=A	(7)
prefix=OUTFILE	(8)
-sorttmpdir sorttemp	(9)
-splitdata	
recno=10000	(10)
nosplitdata=c+0-2	(11)
-worksize 10	(12)
-dateopt 75	(13)

- ソート機能を使用する (1)



- キーの比較は、JIS8 単位コードの体系に従う
- 重複レコードの入力順序保存機能を使用 (2)
- 第 1 キーでのソートは昇順、キーの属性は日付項目 (3)
- 第 1 キーの開始バイト位置は「4」バイト目、バイト長は「8」バイト (3)
- 第 2 キーでのソートは降順、キーの属性は文字 (3)
- 第 2 キーの開始バイト位置は「10」バイト目、バイト長は「3」バイト (3)
- 入力ファイルおよび出力ファイルは、テキストファイル (4), (7)
- ソートする入力ファイルは「INFILE」 (5)
- 入力ファイルのレコード長は「64」バイト (6)
- ソート結果の出力ファイルは「OUTFILE」 (8)
- ソート用一時ファイルディレクトリは、カレントディレクトリ下の「sorttemp」 (9)
- 分割後の出力ファイルのレコード件数の最大は「10000」 (10)
- 非分割データの属性は文字 (11)
- 非分割データのデータ位置は「0」、データ長は「2」 (11)
- ワークバッファのサイズは「10」MB (12)
- 日付項目の基準年は「75」 (13)

## 6.2.8 ソート+キー分割

サンプルファイル名 : exsort\_sample16\_sortsplitkey.txt

-function sort	(1)
-cmdopt DUP	(2)
-key c+0-2 g+4-8 rj+10-3	(3)
-splitinp	
fileorg=A	(4)
file=INFILE	(5)
reclen=64	(6)
-splitout	
fileorg=A	(7)
prefix=OUTFILE	(8)
-sorttmpdir sorttemp	(9)
-splitkey	
key=c+0-2	(10)
spno=10M	(11)
-worksize 10	(12)
-dateopt 75	(13)

- ソート機能を使用する (1)
- キーの比較は、JIS8 単位コードの体系に従う

## 6. SORT EE の使用例

- 重複レコードの入力順序保存機能を使用 (2)
- 第1キーでのソートは昇順、キーの属性は文字 (3)
- 第1キーの開始バイト位置は「0」バイト目、バイト長は「2」バイト (3)
- 第2キーでのソートは昇順、キーの属性は日付項目 (3)
- 第2キーの開始バイト位置は「4」バイト目、バイト長は「8」バイト (3)
- 第3キーでのソートは降順、キーの属性は文字 (3)
- 第3キーの開始バイト位置は「10」バイト目、バイト長は「3」バイト (3)
- 入力ファイルおよび出力ファイルは、テキストファイル (4), (7)
- ソートする入力ファイルは「INFILE」 (5)
- 入力ファイルのレコード長は「64」バイト (6)
- ソート結果の出力ファイルは「OUTFILE」 (8)
- ソート用一時ファイルディレクトリは、カレントディレクトリ下の「sorttemp」 (9)
- 分割キーは昇順でソートされ、分割キーの属性は文字 (10)
- 分割キーの開始バイト位置は「0」バイト目、バイト長は「2」バイト (10)
- 分割出力ファイルのファイルサイズは「10」MB (11)
- ワークバッファのサイズは「10」MB (12)
- 日付項目の基準年は「75」 (13)

### 6.2.9 ソート+ファイル連結

サンプルファイル名 : exsort\_sample17\_sortcat.txt

-function sort	(1)
-cmdopt DUP	(2)
-key g+4-8 rj+10-3	(3)
-catinp	
fileorg=A	(4)
file=INFILE1 INFILE2	(5)
reclen=64	(6)
-catout	
fileorg=A	(7)
file=OUTFILE	(8)
-sorttmpdir sorttemp	(9)
-worksize 10	(10)
-dateopt 75	(11)

- ソート機能を使用する (1)
- キーの比較は、JIS8 単位コードの体系に従う
- 重複レコードの入力順序保存機能を使用 (2)
- 第1キーでのソートは昇順、キーの属性は日付項目 (3)
- 第1キーの開始バイト位置は「4」バイト目、バイト長は「8」バイト (3)

- 第2キーでのソートは降順、キーの属性は文字（英数字） (3)
- 第2キーの開始バイト位置は「10」バイト目、バイト長は「3」バイト (3)
- 入力ファイルおよび出力ファイルは、テキストファイル (4), (7)
- ソートする入力ファイルは「INFILE1」と「INFILE2」 (5)
- 入力ファイルのレコード長は「64」バイト (6)
- ソート結果の出力ファイルは「OUTFILE」 (8)
- ソート用一時ファイルディレクトリは、カレントディレクトリ下の「sorttemp」 (9)
- ワークバッファのサイズは「10」MB (10)
- 日付項目の基準年は「75」 (11)

## 6.2.10 ソート+集約+CSVファイル

サンプルファイル名 : exsort\_sample18\_csvsum.txt

-function sort	(1)
-cmdopt INPCONSEPA:OUTCONSEPA:SUMNULKEY:SUMNONULL	(2)
-csvkey c+1-10 z+2-5,2	(3)
-csvsum z+3-8,3	(4)
-inpfiler	
fileorg=C	(5)
file=INFILE	(6)
recsepa=LFCRLF	(7)
cellsepa=TAB '\t' ','	(8)
-outfile	
fileorg=C	(9)
file=OUTFILE	(10)
recsepa=LF	(11)
cellsepa=TAB	(12)
-sorttmpdir sorttemp	(13)
-worksize 10	(14)

- ソート機能を使用する (1)
- CSV キーの比較は、JIS8 単位コードの体系に従う
- 入力レコードの連続した区切り文字（デリミタ）を1つのデリミタとして扱う「INPCONSEPA」 (2)
- 出力レコードの連続した区切り文字（デリミタ）を1つのデリミタに置き換えて出力する「OUTCONSEPA」 (2)
- ナル文字列（空のセル）の集約キーを同値と扱わず集約しない「SUMNULKEY」 (2)
- ナル文字列（空のセル）の集約項目を集約対象としない「SUMNONULL」 (2)
- 第1キーでのソートは昇順、キーの属性は文字（英数字） (3)

## 6. SORT EE の使用例

- 第1キーのセル番号は「1」、有効バイト長は「10」バイト (3)
- 第2キーでのソートは昇順、キーの属性は固定小数点数 (3)
- 第2キーのセル番号は「2」、総桁数は「5」桁、小数点以下の有効桁は「2」桁 (3)
- 集約項目の属性は固定小数点数 (4)
- 集約項目のセル番号は「3」、有効桁は「8」桁、小数点以下の有効桁は「3」桁 (4)
- 入力ファイルおよび出力ファイルは、CSV ファイル (5)、(9)
- ソートする入力ファイルは「INFILE」 (6)
- 入力ファイルの改行文字は「LF」と「CRLF」の混在 (7)
- 入力ファイルの区切り文字 (デリミタ) は「タブ」、「空白」および「コンマ」の3種類 (8)
- ソート結果の出力ファイルは「OUTFILE」 (10)
- 出力ファイルの改行文字を「LF」に変更 (11)
- 出力ファイルの区切り文字3種類をすべて「タブ」に変更 (12)
- ソート用一時ファイルディレクトリは、カレントディレクトリ下の「sorttemp」 (13)
- ワークバッファのサイズは「10」MB (14)

### 6.2.11 ソート+出力レコード選択+出力レコード編集+ファイル分割 (キー分割) + CSV ファイル

サンプルファイル名 : exsort\_sample22\_csvreorg.txt

<code=SJIS>	(1)
-function sort	(2)
-cmdopt DUP	(3)
-csvkey c+2-8 g+4-8,1 zr+6-10,2	(4)
-splitinp	
fileorg=C	(5)
file=INFILE	(6)
recsepa=CRLF	(7)
cellsepa=','	(8)
codetype=SJIS	(9)
-splitout	
prefix=OUTFILE_NULL	(10)
csvinclude=c+5-3 = NULL	(11)
-splitout	
prefix=OUTFILE_2011	(12)
csvinclude=c+5-3 != NULL AND	(13)
g+4-8,1 >= D'20110101' AND	(14)
(z+6-10,2 <= N'99999' OR	(15)
z+6-10,2 > N'99999')	(16)

csvreorg=1+C'DATE2011'-1	(17)
+2-1	(18)
3+NULL-1	(19)
4+4-3	(20)
9+7	(21)
-sorttmpdir sorttemp	(22)
-splitkey	
csvkey=c+2-8	(23)
recno=1000	(24)
-worksize auto,10	(25)

- パラメタファイルの記述コードは Shift\_JIS (1)
- ソート機能を使用する (2)
- 重複レコードの入力順序保存機能を使用 (3)
- CSV キーの比較は JIS8 単位コードの体系に従う (4)
- 第 1 キーでのソートは昇順, キーの属性は文字 (4)
- 第 1 キーのセル番号は「2」, 有効バイト長は「8」バイト (4)
- 第 2 キーでのソートは昇順, キーの属性は日付 (4)
- 第 2 キーのセル番号は「4」, 有効桁は「8」桁, 日付の書式は「1 (前置 4 桁年号)」 (4)
- 第 3 キーでのソートは降順, キーの属性は数値 (固定小数点数) (4)
- 第 3 キーのセル番号は「6」, 総桁数は「10」桁, 小数点以下の有効桁は「2」桁 (4)
- 入力ファイルは, CSV ファイル (5)
- ソートする入力ファイルは「INFILE」 (6)
- 入力レコードの改行文字は「CRLF」 (7)
- 入力レコードのデリミタ (セル区切り文字) は「, (コンマ)」 (8)
- 入力データの文字コードは「SJIS (Shift\_JIS)」 (9)
- ソート結果の出力ファイルは「OUTFILE\_NULL」 (10)
- 選択条件を満たすレコードだけを出力する (11)
- 比較項目の属性は文字 (11)
- 比較項目のセル番号は「5」, 有効バイト長は「3」バイト (11)
- 比較演算子は「=」 (11)
- 条件値は「NULL (ナル文字列 (空セル))」 (11)
- ソート結果の出力ファイルは「OUTFILE\_2011」 (12)
- 選択条件を満たすレコードだけを出力する (13)
- 比較項目の属性は文字 (13)
- 比較項目のセル番号は「5」, 有効バイト長は「3」バイト (13)
- 比較演算子は「!=」 (13)
- 条件値は「NULL (ナル文字列 (空セル))」 (13)
- 論理演算子は「AND」 (13)
- 比較項目の属性は日付 (14)
- 比較項目のセル番号は「4」, 有効桁は「8」桁, 日付の書式は「1 (前置 4 桁年号)」 (14)
- 比較演算子は「>=」 (14)

## 6. SORT EE の使用例

- 条件値は日付「20110101」 (14)
- 論理演算子は「AND」 (14)
- 比較項目の属性は固定小数点数 (15), (16)
- 比較項目のセル番号は「6」、総桁数は「10」桁、小数点以下の有効桁は「2」桁 (15), (16)
- 比較演算子は「<=」 (15)
- 条件値は固定小数点数「99999」 (15)
- 論理演算子は「OR」 (15)
- 比較演算子は「>」 (16)
- 条件値は固定小数点数「999999」 (16)
- 出力レコードを編集 (17)
- 編集セル「1」に、文字列'DATE2011'を1個挿入 (17)
- 編集セル「2」に、セル番号「2」のセルを挿入 (18)
- 編集セル「3」に、ナル文字列(空セル)を挿入 (19)
- 編集セル「4」に、セル番号「4」「5」「6」の3セルを挿入 (20)
- 編集セル「9」に、セル番号「7」以降のセルを挿入 (21)
- 編集セル「7」「8」(前の編集セルとの間に生じた間隔)に、ナル文字列(空セル)を挿入 (21)
- ソート用一時ファイルディレクトリは、カレントディレクトリ下の「sorttemp」 (22)
- 分割キーは昇順でソートされ、分割キーの属性は文字 (23)
- 分割キーのセル番号は「2」、有効バイト長は「8」バイト (23)
- 分割出力ファイルに出力する最大レコード件数は「1000」 (24)
- ワークバッファサイズは「auto (自動見積もり (アウトコアソート対応))」 (25)
- ワークバッファサイズの最大値は「10」MB (25)

### exsort コマンド実行結果

```
KBLS1007-I ソートが完了しました。Pid=27211 IN=50 OUT=9 WORKSIZE=incore,1,1
("OUTFILE_NULL")
KBLS1010-I ファイル分割が完了しました。 OUT=5 ("OUTFILE_NULL_00000001_00000001")
KBLS1010-I ファイル分割が完了しました。 OUT=4 ("OUTFILE_NULL_00000002_00000001")
KBLS1007-I ソートが完了しました。Pid=27211 IN=50 OUT=9 WORKSIZE=incore,1,1
("OUTFILE_2011")
KBLS1010-I ファイル分割が完了しました。 OUT=4 ("OUTFILE_2011_00000001_00000001")
KBLS1010-I ファイル分割が完了しました。 OUT=5 ("OUTFILE_2011_00000002_00000001")
```

### 入力ファイル (INFILE)

```

00000001, KANAGAWA, abcdefghijklmn123, 2010/06/12, ref, 0272161.00, Sou, 2721
00000002, TOKYO , opqrstuvwxyz67890, 2012/04/16,
00000003, KANAGAWA, abcdefghijklmn123, 2011/06/16, ref, 0602660.00, In , 6026
00000004, TOKYO , opqrstuvwxyz67890, 2010/05/16, add, 0729055.00, Set, 7290
00000005, KANAGAWA, abcdefghijklmn123, 2011/09/10, ref, 0306999.00, Set, 3069
00000006, TOKYO , opqrstuvwxyz67890, 2011/05/16, add, 0073850.01, DoI, 0738
00000007, KANAGAWA, abcdefghijklmn123, 2010/09/12, ref, 0698791.00, Cre, 6987
00000008, TOKYO , opqrstuvwxyz67890, 2011/08/10, ref, 0731084.00, Loa, 7310
00000009, KANAGAWA, abcdefghijklmn123, 2012/02/14, add, 1221225.90, Bui, 2212
00000010, TOKYO , opqrstuvwxyz67890, 2010/05/18, add, 0508754.80, Sta, 5087
00000011, KANAGAWA, abcdefghijklmn123, 2011/09/02,
00000012, TOKYO , opqrstuvwxyz67890, 2011/09/06, ref, 0138693.00, No , 1386
00000013, KANAGAWA, abcdefghijklmn123, 2010/05/06, add, 1143553.70, OEM, 1435
00000014, TOKYO , opqrstuvwxyz67890, 2011/01/02, add, 0273613.45, Fin, 2736
00000015, KANAGAWA, abcdefghijklmn123, 2011/06/08, del, 1828396.00, KB9, 2839
00000016, TOKYO , opqrstuvwxyz67890, 2010/03/16,
00000017, KANAGAWA, abcdefghijklmn123, 2011/05/18, add, 0296853.14, Ana, 2968
00000018, TOKYO , opqrstuvwxyz67890, 2011/08/08, ref, 0687585.00, Ana, 6875
00000019, KANAGAWA, abcdefghijklmn123, 2011/10/24, del, 0722003.00, Ana, 7220
00000020, TOKYO , opqrstuvwxyz67890, 2010/12/10, ref, 0768120.00, Dow, 7681
00000021, KANAGAWA, abcdefghijklmn123, 2011/12/31, add, 0867910.00, bPa, 8679
00000022, TOKYO , opqrstuvwxyz67890, 2011/01/31, del, 1261474.80, Inv, 2614
00000023, KANAGAWA, abcdefghijklmn123, 2010/11/16,
00000024, TOKYO , opqrstuvwxyz67890, 2011/10/21,
00000025, KANAGAWA, abcdefghijklmn123, 2012/04/02, ref, 0984847.00, Sou, 9848
00000026, TOKYO , opqrstuvwxyz67890, 2010/06/23, add, 0467068.00, Des, 4670
00000027, KANAGAWA, abcdefghijklmn123, 2012/03/25, ref, 1184660.50, In , 1846
00000028, TOKYO , opqrstuvwxyz67890, 2011/09/23, ref, 0881797.00, Set, 8817
00000029, KANAGAWA, abcdefghijklmn123, 2012/03/20,
00000030, TOKYO , opqrstuvwxyz67890, 2010/07/20, del, 0714902.00, DoI, 7149
00000031, KANAGAWA, abcdefghijklmn123, 2011/06/24,
00000032, TOKYO , opqrstuvwxyz67890, 2012/04/01, ref, 0017220.00, Zen, 0646
00000033, TOKYO , opqrstuvwxyz67890, 2010/07/29, ref, 0052672.00, Bui, 7526
00000034, TOKYO , opqrstuvwxyz67890, 2011/03/27,
00000035, TOKYO , opqrstuvwxyz67890, 2011/09/08, del, 0711790.00, Ana, 7117
00000036, TOKYO , opqrstuvwxyz67890, 2010/01/01, ref, 0167217.00, No , 1672
00000037, TOKYO , opqrstuvwxyz67890, 2012/01/31, add, 0857549.12, OEM, 8575
00000038, TOKYO , opqrstuvwxyz67890, 2012/02/29, ref, 0077126.00, Fin, 0771
00000039, KANAGAWA, abcdefghijklmn123, 2010/02/21, add, 0019899.00, KB9, 0198
00000040, KANAGAWA, abcdefghijklmn123, 2011/01/01, add, 0564334.19, Ana, 5643
00000041, KANAGAWA, abcdefghijklmn123, 2011/05/18, ref, 0067531.00, Ana, 0675
00000042, KANAGAWA, abcdefghijklmn123, 2010/11/06, ref, 0475823.00, Ana, 4758
00000043, KANAGAWA, abcdefghijklmn123, 2011/01/02,
00000044, KANAGAWA, abcdefghijklmn123, 2011/11/02, add, 0655974.00, Dow, 6559
00000045, KANAGAWA, abcdefghijklmn123, 2010/01/09, ref, 0214820.00, bPa, 2148
00000046, TOKYO , opqrstuvwxyz67890, 2011/03/15, ref, 0168228.00, Inv, 1682
00000047, KANAGAWA, abcdefghijklmn123, 2011/10/12, del, 0662431.00, Num, 6624
00000048, TOKYO , opqrstuvwxyz67890, 2010/12/12, add, 0923745.00, Ver, 9237
00000049, KANAGAWA, abcdefghijklmn123, 2011/11/29, ref, 0916980.00, KB9, 9169
00000050, TOKYO , opqrstuvwxyz67890, 2012/02/25, add, 3041596.00, Zen, 3415

```

出力ファイル

- OUTFILE\_NULL\_00000001\_00000001

```

00000023, KANAGAWA, abcdefghijklmn123, 2010/11/16,
00000043, KANAGAWA, abcdefghijklmn123, 2011/01/02,
00000031, KANAGAWA, abcdefghijklmn123, 2011/06/24,
00000011, KANAGAWA, abcdefghijklmn123, 2011/09/02,
00000029, KANAGAWA, abcdefghijklmn123, 2012/03/20,

```

- OUTFILE\_NULL\_00000002\_00000001

```

00000016, TOKYO , opqrstuvwxyz67890, 2010/03/16,
00000034, TOKYO , opqrstuvwxyz67890, 2011/03/27,
00000024, TOKYO , opqrstuvwxyz67890, 2011/10/21,
00000002, TOKYO , opqrstuvwxyz67890, 2012/04/16,

```

## 6. SORT EE の使用例

- OUTFILE\_2011\_00000001\_00000001

```
DATE2011,KANAGAWA,,2011/05/18,ref,0067531.00,,,Ana,0675
DATE2011,KANAGAWA,,2011/06/08,del,1828396.00,,,KB9,2839
DATE2011,KANAGAWA,,2012/02/14,add,1221225.90,,,Bui,2212
DATE2011,KANAGAWA,,2012/03/25,ref,1184660.50,,,In ,1846
```

- OUTFILE\_2011\_00000002\_00000001

```
DATE2011,TOKYO,,2011/01/31,del,1261474.80,,,Inv,2614
DATE2011,TOKYO,,2011/05/16,add,0073850.01,,,DoI,0738
DATE2011,TOKYO,,2012/02/25,add,3041596.00,,,Zen,3415
DATE2011,TOKYO,,2012/02/29,ref,0077126.00,,,Fin,0771
DATE2011,TOKYO,,2012/04/01,ref,0017220.00,,,Zen,0646
```

### 6.2.12 ソート+ CSV ファイル+ UCS-2LE

サンプルファイル名 : exsort\_sample23\_csvsort.txt

<code=SJIS>	(1)
-function sort	(2)
-cmdopt DUP	(3)
-csvkey rw+4-10	(4)
z+1-4,1	(5)
c+2-6	(6)
-infile	
codetype=UCS-2LE	(7)
fileorg=C	(8)
file=INFILE	(9)
recsepa=CRLF	(10)
cellsepa=', '	(11)
-outfile	
file=OUTFILE	(12)
-sorttmpdir sorttemp	(13)
-worksize auto,10	(14)

- パラメタファイルの記述コードは Shift\_JIS (1)
- ソート機能を使用する (2)
- 重複レコードの入力順序保存機能を使用 (3)
- 第1キーでのソートは降順、キーの属性はワイド文字 (4)
- 第1キーのセル番号は「4」、有効バイト長は「10」バイト (4)
- 第2キーでのソートは昇順、キーの属性は数値 (固定小数点数) (5)
- 第2キーのセル番号は「1」、有効桁は「4」桁、小数点以下の有効桁は「1」桁 (5)
- 第3キーでのソートは昇順、キーの属性は文字 (6)



- 第3キーのセル番号は「2」、有効バイト長は「6」バイト (6)
- 入力データのエンコードは「UCS-2LE (UCS-2 little endian)」 (7)
- 入力ファイルは、CSV ファイル (8)
- ソートする入力ファイルは「INFILE」 (9)
- 入力レコードの改行文字は「CRLF」 (10)
- 入力レコードのデリミタ (セル区切り文字) は「コンマ (0x2C00)」 (11)
- ソート結果の出力ファイルは「OUTFILE」 (12)
- ソート用一時ファイルディレクトリは、カレントディレクトリ下の「sorttemp」 (13)
- ワークバッファサイズは「auto (自動見積もり (アウトコアソート対応))」 (14)
- ワークバッファサイズの最大値は「10」 MB (14)

### 6.2.13 ソート+符号なし2進数のマスクビット指定キー

サンプルファイル名 : exsort\_sample24\_sortbitkey.txt

-function sort	(1)
-cmdopt DUP:OWWRT	(2)
-key rb+10-f0	(3)
b+10-0f	(4)
a+4-fffff000	(5)
-inpfiler	
fileorg=S	(6)
reclen=100	(7)
file=INFILE	(8)
-outfile	
fileorg=S	(9)
file=OUTFILE	(10)
-sorttmpdir sorttemp	(11)
-worksize auto	(12)

- ソート機能を使用する (1)
- 重複レコードの入力順序保存機能を使用 (2)
- 出力ファイルがすでに存在する場合、上書き出力される (2)
- 第1キーでのソートは降順、キーの属性は符号なし2進数 (バイトデータ) ビットキー (3)
- 第1キーの開始バイト番号は「10」バイト目、バイト長は「1」バイト、マスクビットは「0xF0」 (3)
- 第2キーでのソートは昇順、キーの属性は符号なし2進数 (バイトデータ) ビットキー (4)
- 第2キーの開始バイト位置は「10」バイト目、バイト長は「1」バイト、マスクビットは「0x0F」 (4)
- 第3キーでのソートは昇順、キーの属性は符号なし2進数 (0 から始まる整数) ビットキー (5)

## 6. SORT EE の使用例

- 第3キーの開始バイト位置は「4」バイト目、バイト長は「4」バイト、マスクビットは「0xFFFFF000」 (5)
- 入力ファイルおよび出力ファイルは、テキストファイル (6), (9)
- 入力ファイルのレコード長は「100」バイト (7)
- ソートする入力ファイルは「INFILE」 (8)
- ソート結果の出力ファイルは「OUTFILE」 (10)
- ソート用一時ファイルディレクトリは、カレントディレクトリ下の「sorttemp」 (11)
- ワークバッファサイズは「auto (自動見積もり (アウトコアソート対応))」 (12)

### 6.2.14 ソート+ CSV ファイル+ UTF-16LE

サンプルファイル名 : exsort\_sample25\_multibyte.txt

-function sort	(1)
-cmdopt OVWRT	(2)
-key u+8-4	(3)
v+16-10	(4)
-inpfiler	
fileorg=A	(5)
file=INFILE	(6)
codetype=UTF-16LE	(7)
recsepa=LF CRLF	(8)
-outfile	
file=OUTFILE	(9)
-sorttmpdir sorttemp	(10)
-worksize incore	(11)

- ソート機能を使用する (1)
- 出力ファイルがすでに存在する場合、上書き出力される (2)
- 第1キーでのソートは昇順、キーの属性はマルチバイト文字<sup>※1</sup> (3)
- 第1キーの開始バイト番号は「8」バイト目、長さは「4」文字 (3)
- 第2キーでのソートは昇順、キーの属性はマルチバイト文字<sup>※2</sup> (4)
- 第2キーの開始バイト位置は「16」バイト目、長さは「10」文字 (4)
- 入力ファイルは、テキストファイル (5)
- ソートする入力ファイルは「INFILE」 (6)
- 入力データのエンコードは「UTF-16LE (UTF-16 little endian)」 (7)
- 入力レコードの改行文字は「LF CRLF (LF と CRLF の改行文字の自動判定)」 (8)
- ソート結果の出力ファイルは「OUTFILE」 (9)
- ソート用一時ファイルディレクトリは、カレントディレクトリ下の「sorttemp」 (10)
- ワークバッファサイズは「incore (インコアソート)」 (11)

注※1

第 1 キーの属性コード「u」は、UTF-16 サロゲートペア文字を 1 文字として扱います。また、UCS コードが一致する文字であれば、Unicode の IVS (Ideographic Variation Sequence) の異体字であっても同じ文字として扱います。

注※ 2

第 2 キーの属性コード「v」は、UTF-16 サロゲートペア文字を 1 文字として扱います。また、UCS コードが一致する文字であっても、VS (Variation Selector, 異体字セレクタ) が異なる場合は、異なる文字として扱います。

## 6.3 マージ機能の使用例

マージ機能の使用例を次に示します。

### 6.3.1 マージ

サンプルファイル名 : exsort\_sample2\_merge.txt

-function merge	(1)
-cmdopt EBCJ:OVWRT	(2)
-key j+0-4	(3)
-inpfile	
fileorg=A	(4)
file=INFILE1 INFILE2 INFILE3	(5)
reclen=100	(6)
-outfile	
fileorg=A	(7)
file=OUTFILE	(8)

- マージ機能を使用する (1)
- キーの比較は EBCDIK コードの体系に従う (2)
- 出力ファイルがすでに存在する場合、上書き出力される (2)
- マージは昇順、キーの属性は文字 (英数字) (3)
- キーの開始バイト位置は「0」バイト目、バイト長は「4」バイト (3)
- 入力ファイルおよび出力ファイルは、テキストファイル (4), (7)
- マージする入力ファイルは「INFILE1」、「INFILE2」、および「INFILE3」 (5)
- 入力ファイルのレコード長は「100」バイト (6)
- マージ結果の出力ファイルは「OUTFILE」 (8)

### 6.3.2 マージ+集約

サンプルファイル名 : exsort\_sample6\_mergesum.txt

-function merge	(1)
-cmdopt OVWRT	(2)
-key c+0-2	(3)
-sum z+16-8	(4)
-inpfile	
fileorg=S	(5)

file=INFILE1 INFILE2 INFILE3	(6)
reclen=100	(7)
-outfile	
fileorg=S	(8)
file=OUTFILE	(9)

- マージ機能を使用する (1)
- キーの比較は JIS8 単位コードの体系に従う
- 出力ファイルがすでに存在する場合、上書き出力される (2)
- キーの属性は文字 (3)
- キーの開始バイト位置は「0」バイト目、バイト長は「2」バイト (3)
- 集約項目の属性は外部 10 進数 (4)
- 集約項目の開始バイト位置は「16」バイト目、バイト長は「8」バイト (4)
- 入力ファイルおよび出力ファイルは、固定長順編成ファイル (5), (8)
- 入力ファイルは「INFILE1」, 「INFILE2」, および「INFILE3」 (6)
- 入力ファイルのレコード長は「100」バイト (7)
- 出力ファイルは「OUTFILE」 (9)

### 6.3.3 マージ+出力選択+出力レコード編集

サンプルファイル名 : exsort\_sample10\_outfile.txt

-function merge	(1)
-cmdopt OVWRT	(2)
-key z+0-8	(3)
-inpfiler	
fileorg=A	(4)
file=INFILE1 INFILE2 INFILE3	(5)
reclen=300	(6)
-outfile	
fileorg=A	(7)
file=OUTFILE1 OUTFILE2	(8)
-outfile	
fileorg=A	(9)
file=OUTFILE3	(10)
reclen=200	(11)
minlen=10	(12)
reorg=+100	(13)

## 6. SORT EE の使用例

<code>-outfile</code>	
<code>fileorg=A</code>	(14)
<code>file=OUTFILE4</code>	(15)
<code>include=z+0-8 &gt; N'500'</code>	(16)

- マージ機能を使用する (1)
- キーの比較は JIS8 単位コードの体系に従う
- 出力ファイルがすでに存在する場合、上書き出力される (2)
- キーの属性は外部 10 進数 (3)
- キーの開始バイト位置は「0」バイト目、バイト長は「8」バイト (3)
- 入力ファイルおよび出力ファイルは、テキストファイル (4), (7), (9), (14)
- 入力ファイルは「INFILE1」, 「INFILE2」, および「INFILE3」 (5)
- 入力ファイルのレコード長は「300」バイト (6)
- 出力ファイルは「OUTFILE1」, 「OUTFILE2」 (8)
- 出力ファイルは「OUTFILE3」で、下記 (13) の編集をする (10)
- レコード編集後の最大レコード長は「200」バイト (11)
- レコード編集後の最小レコード長は「10」バイト (12)
- レコードの 100 バイト目から末尾までを、出力レコードの先頭から配置される (13)
- 出力ファイルは「OUTFILE4」 (15)
- 比較項目の属性は外部 10 進数 (16)
- 比較項目の開始バイト位置は「0」バイト目、バイト長は「8」バイト (16)
- 比較演算子は「>」 (16)
- 条件値は 10 進数「500」 (16)

## 6.4 コピー機能の使用例

コピー機能の使用例を次に示します。

### 6.4.1 選択

サンプルファイル名 : exsort\_sample3\_select.txt

-function copy	(1)
-cmdopt OVWRT	(2)
-inpfile	
fileorg=S	(3)
file=INFILE1 INFILE2 INFILE3	(4)
reclen=100	(5)
include=z+8-4 > N'999' AND z+16-4 < N'500'	(6)
-outfile	
fileorg=S	(7)
file=OUTFILE	(8)

- コピー機能を使用する (1)
- 項目の比較は JIS8 単位コードの体系に従う
- 出力ファイルがすでに存在する場合、上書き出力される (2)
- 入力ファイルおよび出力ファイルは、固定長順編成ファイル (3), (7)
- 選択する入力ファイルは「INFILE1」, 「INFILE2」, および「INFILE3」 (4)
- 選択条件 (第 1 比較項目と第 2 比較項目の論理積) を満たすレコードだけを入力する (6)
- 第 1 比較項目の属性は外部 10 進数 (6)
- 第 1 比較項目の開始バイト位置は「8」バイト目, バイト長は「4」バイト (6)
- 第 1 比較項目の比較演算子は「>」 (6)
- 第 1 比較項目の条件値は数値「999」 (6)
- 第 2 比較項目の属性は外部 10 進数 (6)
- 第 2 比較項目の開始バイト位置は「16」バイト目, バイト長は「4」バイト (6)
- 第 2 比較項目の比較演算子は「<」 (6)
- 第 2 比較項目の条件値は数値「500」 (6)
- 選択結果の出力ファイルは「OUTFILE」 (8)

### 6.4.2 集約

サンプルファイル名 : exsort\_sample4\_sum.txt

-function copy	(1)
-cmdopt OVFLW	(2)

## 6. SORT EE の使用例

<code>-key c+6-2</code>	(3)
<code>-sum z+28-8 p+48-4</code>	(4)
<code>-inpfiler</code>	
<code>fileorg=S</code>	(5)
<code>file=INFILE</code>	(6)
<code>reclen=128</code>	(7)
<code>-outfile</code>	
<code>fileorg=S</code>	(8)
<code>file=OUTFILE</code>	(9)

- コピー機能を使用する (1)
- キーの比較は JIS8 単位コードの体系に従う
- 集約結果のオーバーフローを桁数で判定する (2)
- キーの属性は文字 (3)
- キーの開始バイト位置は「6」バイト目、バイト長は「2」バイト (3)
- 1 番目の集約項目の属性は外部 10 進数 (4)
- 1 番目の集約項目の開始バイト位置は「28」バイト目、バイト長は「8」バイト (4)
- 2 番目の集約項目の属性は内部 10 進数 (4)
- 2 番目の集約項目の開始バイト位置は「48」バイト目、バイト長は「4」バイト (4)
- 入力ファイルおよび出力ファイルは、固定長順編成ファイル (5), (8)
- 集約する入力ファイルは「INFILE」 (6)
- 入力ファイルのレコード長は「128」バイト (7)
- 集約結果の出力ファイルは「OUTFILE」 (9)

### 6.4.3 ファイルコピー

(この例題のサンプルファイルはありません)

<code>-function copy</code>	(1)
<code>-cmdopt OVWRT</code>	(2)
<code>-inpfiler</code>	
<code>fileorg=S</code>	(3)
<code>file=INFILE</code>	(4)
<code>reclen=200</code>	(5)
<code>-outfile</code>	
<code>fileorg=S</code>	(6)
<code>file=OUTFILE</code>	(7)

- コピー機能を使用する (1)
- 出力ファイルがすでに存在する場合、上書き出力される (2)



- 入力ファイルおよび出力ファイルは、固定長順編成ファイル (3) (6)
- 入力ファイルは「INFILE」 (4)
- 入力ファイルのレコード長は「200」バイト (5)
- 出力ファイルは「OUTFILE」 (7)

#### 6.4.4 レコード編集

##### (1) 入力レコードに対して編集フォーマットを指定する場合

サンプルファイル名 : exsort\_sample8\_reorg.txt

-function copy	(1)
-inpfiler	
fileorg=S	(2)
file=INFILE	(3)
reclen=150	(4)
reorg=0+0-100	(5)
-outfile	
file=OUTFILE	(6)
reclen=100	(7)

- コピー機能を使用する (1)
- 入力ファイルは、固定長順編成ファイル (2)
- 入力ファイルは「INFILE」で、下記 (5) の編集をする (3)
- 入力ファイルのレコード長は「150」バイト (4)
- 入力レコードの先頭から 100 バイトまでが出力レコードの先頭から配置される (5)
- 出力ファイルは「OUTFILE」 (6)
- レコード編集後の最大レコード長は「100」バイト (7)

##### (2) 中間レコードに対して編集フォーマットを指定する場合

サンプルファイル名 : exsort\_sample9\_reorg.txt

-function copy	(1)
-inpfiler	
fileorg=A	(2)
file=INFILE	(3)
reclen=150	(4)
-reorg 0+0-100	(5)
-outfile	
file=OUTFILE	(6)

## 6. SORT EE の使用例

reclen=100	(7)
minlen=100	(8)

- コピー機能を使用する (1)
- 入力ファイルは、テキストファイル (2)
- 入力ファイルは「INFILE」で、下記 (5) の編集をする (3)
- 入力ファイルのレコード長は「150」バイト (4)
- 入力レコードの先頭から 100 バイトまでが出力レコードの先頭から配置される (5)
- 出力ファイルは「OUTFILE」 (6)
- レコード編集後の最大レコード長は「100」バイト (7)
- レコード編集後の最小レコード長は「100」バイト (8)

### 6.4.5 レコードの読み飛ばしと打ち切り

サンプルファイル名 : exsort\_sample11\_skiprec.txt

-function copy	(1)
-cmdopt OVWRT	(2)
-infile	
fileorg=A	(3)
file=INFILE	(4)
skiprec=100	(5)
stopaft=200	(6)
-outfile	
file=OUTFILE	(7)

- コピー機能を使用する (1)
- 出力ファイルがすでに存在する場合、上書き出力される (2)
- 入力ファイルは、テキストファイル (3)
- 入力ファイルは「INFILE」 (4)
- 読み飛ばしレコード件数は 100 (5)
- 打ち切りレコード件数は 200 (6)
- 出力ファイルは「OUTFILE」 (7)

### 6.4.6 コピー+選択+ CSV ファイル

サンプルファイル名 : exsort\_sample19\_csvsel.txt

-function copy	(1)
-infile	

fileorg=C	(2)
file=INFILE	(3)
recsepa=LF <sup>C</sup> RLF	(4)
cellsepa='△' TAB	(5)
csvinclude=z+4-5 = N'50000' OR g+1-8,1 >= D'20120327'	(6)
-outfile	
fileorg=C	(7)
file=OUTFILE	(8)

- コピー機能を使用する (1)
- CSV 比較項目の比較は JIS8 単位コードの体系に従う
- 入力ファイルおよび出力ファイルは、CSV ファイル (2), (7)
- コピーする入力ファイルは「INFILE」 (3)
- 入力ファイルの改行文字は「LF」と「CRLF」の混在 (4)
- 入力ファイルの区切り文字 (デリミタ) は「空白」と「タブ」の 2 種類 (5)
- 選択条件 (第 1 比較項目と第 2 比較項目の論理和) を満たすレコードだけを入力する (6)
- 第 1 比較項目の属性は固定小数点数 (6)
- 第 1 比較項目のセル番号は「4」、有効桁は「5」桁、小数点以下の有効桁はなし (6)
- 第 1 比較項目の比較演算子は「=」 (6)
- 第 1 比較項目の条件値は数値「50000」 (6)
- 第 2 比較項目の属性は日付 (6)
- 第 2 比較項目のセル番号は「1」、有効桁は「8」桁、前置 4 桁年号 (6)
- 第 2 比較項目の比較演算子は「>=」 (6)
- 第 2 比較項目の条件値は日付「20120327」(2012 年 3 月 27 日) (6)
- コピー結果の出力ファイルは「OUTFILE」 (8)

### 6.4.7 コピー+ファイル編成変更

サンプルファイル名 : exsort\_sample20\_fileorg.txt

-function copy	(1)
-infile	
fileorg=A	(2)
reclen=200	(3)
file=INFILE	(4)
-outfile	
fileorg=V	(5)
minlen=100	(6)
reclen=200	(7)

## 6. SORT EE の使用例

file=OUTFILE	(8)
--------------	-----

- コピー機能を使用する (1)
- 入力ファイルのファイル編成はテキストファイル (2)
- 入力レコードの最大レコード長は 200 バイト (3)
- コピーする入力ファイルは「INFILE」 (4)
- 出力ファイルのファイル編成は可変長順編成ファイル (5)
- 出力レコードの最小レコード長は 100 バイト (6)
- 出力レコードの最大レコード長は 200 バイト (7)
- コピー結果の出力ファイルは「OUTFILE」 (8)

### 6.4.8 コピー+レコード長変更

サンプルファイル名 : exsort\_sample21\_reclen.txt

-function copy	(1)
-infile	
fileorg=S	(2)
reclen=100	(3)
file=INFILE	(4)
-outfile	
fileorg=S	(5)
reclen=200	(6)
padding=C'*'	(7)
file=OUTFILE	(8)

- コピー機能を使用する (1)
- 入力ファイルと出力ファイルのファイル編成は固定長順編成ファイル (2), (5)
- 入力レコード長は 100 バイト (3)
- コピーする入力ファイルは「INFILE」 (4)
- 出力レコード長は 200 バイト (6)
- 出力レコードの足りない 100 バイトをパディング文字で埋める (7)
- 出力レコードのパディング文字は「\*」 (7)
- コピー結果の出力ファイルは「OUTFILE」 (8)

### 6.4.9 簡易コピー

サンプルファイル名 : exsort\_sample26\_simplecopy.txt

-function copy	(1)
-cpyinp	(2)

fileorg=A	(3)
INFILE1 INFILE2	(4)
-cpyout	(5)
OUTFILE	(6)

- コピー機能を使用する (1)
- 簡易コピー機能の入力ファイル情報を `-cpyinp` パラメタで指定する (2)
- 入力ファイルのファイル編成はテキスト (3)
- コピーする入力ファイルは「INFILE1」と「INFILE2」 (4)
- 簡易コピー機能の出力ファイル情報を `-cpyout` パラメタで指定する (5)
- コピー結果の出力ファイルは「OUTFILE」 (6)

#### 注意事項

- 簡易コピーのファイル情報を指定する `-cpyinp` パラメタと `-cpyout` パラメタでは、入力ファイル名 (3) と出力ファイル名 (6) を指定する時、「file=」オペランドは指定しません。
- BOM 付きの入力ファイルを指定する場合、BOM の読み飛ばしを指定する環境変数「EXSORTBOMLNGIN (64 ビット版は「EXSORT64BOMLNGIN」)」または「EXSORTBOMGETPUT (64 ビット版は「EXSORT64BOMGETPUT」)」を指定してください。これらの環境変数の指定がない場合、入力ファイルの BOM を読み飛ばさず、レコードデータとして処理します。なお、環境変数「EXSORTBOMLNGIN (64 ビット版は「EXSORT64BOMLNGIN」)」および「EXSORTBOMGETPUT (64 ビット版は「EXSORT64BOMGETPUT」)」については「14.1 SORT EE の環境変数の設定」を参照してください。
- 出力ファイルの BOM の有無と種類を指定する場合は、環境変数「EXSORTBOMLNGOUT (64 ビット版は「EXSORT64BOMLNGOUT」)」または「EXSORTBOMGETPUT (64 ビット版は「EXSORT64BOMGETPUT」)」を指定してください。なお、環境変数「EXSORTBOMLNGOUT (64 ビット版は「EXSORT64BOMLNGOUT」)」または「EXSORTBOMGETPUT (64 ビット版は「EXSORT64BOMGETPUT」)」については「14.1 SORT EE の環境変数の設定」を参照してください。

## 6.5 altseq 変換ファイルの作成例

サンプルファイル名 : exsort\_altseq\_sample.txt

```

00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f /* 00 - 0f */
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f /* 10 - 1f */
20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f /* 20 - 2f */
30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f /* 30 - 3f */
40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f /* 40 - 4f */
50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f /* 50 - 5f */
60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f /* 60 - 6f */
70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f /* 70 - 7f */
80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f /* 80 - 8f */
90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f /* 90 - 9f */
a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af /* a0 - af */
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf /* b0 - bf */
c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf /* c0 - cf */
d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df /* d0 - df */
e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef /* e0 - ef */
f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff /* f0 - ff */

```

サンプルファイル名 : exsort\_altseq\_ebcdic\_sample.txt

```

00 01 02 03 37 2d 2e 2f 16 05 15 0b 0c 0d 0e 0f /* 00 - 0f */
10 11 12 13 3c 3d 32 26 18 19 3f 27 1c 1d 1e 1f /* 10 - 1f */
40 4f 7f 7b e0 6c 50 7d 4d 5d 5c 4e 6b 60 4b 61 /* 20 - 2f */
f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 7a 5e 4c 7e 6e 6f /* 30 - 3f */
7c c1 c2 c3 c4 c5 c6 c7 c8 c9 d1 d2 d3 d4 d5 d6 /* 40 - 4f */
d7 d8 d9 e2 e3 e4 e5 e6 e7 e8 e9 4a 5b 5a 5f 6d /* 50 - 5f */
79 81 82 83 84 85 86 87 88 89 91 92 93 94 95 96 /* 60 - 6f */
97 98 99 a2 a3 a4 a5 a6 a7 a8 a9 c0 6a d0 a1 07 /* 70 - 7f */
80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f /* 80 - 8f */
90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f /* 90 - 9f */
a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af /* a0 - af */
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf /* b0 - bf */
c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf /* c0 - cf */
d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df /* d0 - df */
e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef /* e0 - ef */
f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff /* f0 - ff */

```

# 7

## ソートコマンドの使い方

この章では、Windows の場合と UNIX の場合に分けてソートコマンドの使い方について説明します。

---

7.1 ソートコマンドの使い方 (Windows の場合)

---

7.2 ソートコマンドの使い方 (UNIX の場合)

---

## 7.1 ソートコマンドの使い方（Windows の場合）

コマンドを入力して、ソート機能を使うこともできます。

ソートコマンドで、処理したいデータが納められているファイルを入力ファイルに指定し、処理結果をどのファイルに出力するのかを指定します。さらに、キーや項目の属性や位置などを指定します。

### 7.1.1 コマンドの使い方

コマンドを入力して、ソート機能を使う方法について説明します。

ソートコマンドのコマンド名を次に示します。

- `wnsort` (32 ビット版 SORT でソートコマンドを使う場合)
- `wnsort64` (64 ビット版 SORT でソートコマンドを使う場合)

ソートコマンドの入力方法について説明します。

#### (1) コマンドの形式

コマンドは、コマンド名（`wnsort` または `wnsort64`）とそのあとに続く各パラメタで構成されます。ソートコマンドの形式を次の図に示します。

図 7-1 ソートコマンドの形式

```

wnsort*1 Δ [ソートオプション]
          Δ /g入力ファイルのファイル編成[:出力ファイルのファイル編成]※2
          Δ [/lレコード長]
          Δ [/o出力ファイル名]
          Δ [/yワークバッファサイズ]
          Δ [/wディレクトリ名]
          Δ [/w2ディレクトリ名]
          Δ { キー属性Δ+キー位置-キー長 } . . .
          Δ { [入力ファイル名] } . . .

```

(凡例) Δ : スペース (空白) を示す。

[ ] : 省略できることを示す。

. . . : 直前の { } で囲まれた部分を複数繰り返して指定できることを示す。

注※1 使用している SORT が 64 ビット版対応の場合は、「`wnsort64`」となる。

注※2 入力ファイルが索引ファイルの場合、出力ファイルのファイル編成の指定は省略できない。

### 7.1.2 オペランドの意味

オペランドには、必ず指定するオペランドと任意に指定できるオペランドの 2 種類があります。



## (1) ソートオプション

ソート処理で使用する機能上のオプションを指定します。ソートオプションのパラメータを次の表に示します。

表 7-1 ソートオプションのパラメータ

パラメータ	意味
/d	ファイル中に重複したキーのレコードがある場合、それらのデータを入力順に並べることを指定する（重複レコードの入力順序保存機能）。
/i	出力ファイル名と同じ名前のファイルがすでにある場合、そのファイルの内容が書き換えられるのを防ぐ（既存ファイルの書き換え防止）。
/v	EBCDIK コードの体系に従ってキーを比較する。
/b <sup>※1</sup>	ソートキーが数値項目の場合、 <b>big endian</b> <sup>※2</sup> の数値として扱う。

### 注※1

ソートキーが2進項目「/n」「/m」、または内部浮動小数点項目「/f」の場合だけ有効となる。

### 注※2

**big endian**とは、上位バイトを下位番地とする数値のことです。

/bパラメータを指定しなかった場合は、**little endian**（上位バイトを上位番地とする数値）として扱います。

複数のオプションを指定する場合は、次に示すどちらかの方法で指定します。

- 「/d △ /i」のように分けて指定する（「△」は、スペース（空白）を示す）。
- 「/di」のようにつなげて指定する。

また、この指定は省略できます。省略した場合は、次のようになります。

- 「/d」の指定を省略した場合、重複レコードの入力順序は保存されません。
- 「/i」の指定を省略した場合、出力ファイル名と同じ名前のファイルがすでにあると、そのファイルの内容が書き換えられます。
- 「/v」の指定を省略した場合、JIS8 単位コードの体系に従ってキーが比較されます。

なお、ソートオプションを指定する場合は、次の点に注意してください。

- 入力ファイルが索引ファイルの場合は、「/d」を指定しても無効になります。
- キー属性に「/j」を指定していない場合は、「/v」を指定しても無効になります。
- ファイル中に重複したキーがある場合、ワークバッファサイズを変えることで、重複したキーの並びが変わる場合があります。重複したキーの並びを保つためには、「/d」を指定してください。

## (2) /g 入力ファイルのファイル編成

入力ファイルのファイル編成を「/g +オペランド」の形式で指定します。複数の入力ファイルを指定する場合は、すべて同じファイル編成にしなければなりません。

## 7. ソートコマンドの使い方

また、この指定は省略できません。入力ファイルのファイル編成のオペランドを次の表に示します。

表 7-2 入力ファイルのファイル編成のオペランド（ソート機能）

オペランド	意味
s	固定長順編成ファイルを指定する。
v	可変長順編成ファイルを指定する。
r	固定長相対ファイルを指定する。
w	可変長相対ファイルを指定する。
i	固定長索引ファイルを指定する。
x	可変長索引ファイルを指定する。
a	テキストファイルを指定する。

### (3) 出力ファイルのファイル編成

出力ファイルのファイル編成のオペランドを、コロンで区切って指定します。出力ファイルのファイルの形式（固定長、可変長またはテキスト）は、入力ファイルと同じにしなればなりません。

入力ファイルが索引ファイル以外の場合、この指定は省略できます。指定を省略した場合は、入力ファイルと同じファイル編成が設定されます。入力ファイルが索引ファイルの場合は、順ファイルまたは相対ファイルのどちらかを必ず指定してください。

出力ファイルのファイル編成のオペランドを次の表に示します。

表 7-3 出力ファイルのファイル編成のオペランド（ソート機能）

オペランド	意味
s	固定長順編成ファイルを指定する。
v	可変長順編成ファイルを指定する。
r	固定長相対ファイルを指定する。
w	可変長相対ファイルを指定する。
a	テキストファイルを指定する。

注

索引ファイルに出力する場合は、ISAM のユティリティを使用する。

### (4) //レコード長

入力ファイル編成ごとのレコード長の指定方法を次の表に示します。

表 7-4 入力ファイル編成ごとのレコード長の指定方法（ソート機能）

入力ファイル編成	レコード長の指定方法
固定長順編成ファイル	ファイルのレコード長を「1～65,535」バイトの範囲で指定します。
可変長順編成ファイル	指定は不要です。指定しても無視されます。 ただし、入力できるレコード長の範囲は、1～65,535 バイトです。
固定長相対ファイル	
可変長相対ファイル	
固定長索引ファイル	指定は不要です。指定しても無視されます。 ただし、入力できるレコード長の範囲は、1～65,503 バイトです。
可変長索引ファイル	
テキストファイル	ファイルの最大レコード長を「1～65,535」バイトの範囲で指定します。 レコード長には、改行文字「X'0A'」は含みません。 省略できます。省略した場合「65,535」バイトが設定されます。 レコード長を指定した方がソート時間が短くなります。

### (5) /o 出力ファイル名

ソート結果の出力先ファイルをファイル名で指定します。

この指定は、省略できます。指定を省略した場合は、ソート結果を標準出力ファイルに出力します。

### (6) /y ワークバッファサイズ

ワークバッファとして使用するメモリの容量を KB または MB 単位で指定します。ワークバッファサイズは、メモリ量と単位で指定します。

#### (a) メモリ量

メモリ量の指定範囲を次の表に示します。

表 7-5 メモリ量の指定範囲（ソート機能）

SORT の種別	レコード長※	指定範囲
32 ビット版	32,760 バイト以下	128KB 以上 2,000MB 以内の 1～7 桁の数字で指定します。
	32,760 バイトより大きい	256KB 以上 2,000MB 以内の 1～7 桁の数字で指定します。
64 ビット版	32,760 バイト以下	256KB 以上 2,000MB 以内の 1～7 桁の数字で指定します。
	32,760 バイトより大きい	384KB 以上 2,000MB 以内の 1～7 桁の数字で指定します。

注※

可変長ファイルおよびテキストファイルの場合、最大レコード長となります。

この指定は省略できます。この指定を省略した場合、または最小メモリ量に満たない値

## 7. ソートコマンドの使い方

が指定された場合は、1,024KB が設定されます。ただし、これらの場合で環境ファイルに初期値が設定されているときは、その値が設定されます。

なお、メモリ量の目安については、「17.1.1 ワークバッファサイズと処理時間」を参照してください。また、環境ファイルの設定方法については、「15. 環境ファイルの設定」を参照してください。

### (b) 単位

メモリ量の単位を、KB にするか MB にするか指定します。

- k : 単位を KB にします。
- m : 単位を MB にします。

単位を省略した場合は「KB (k)」が設定されます。

指定例を次に示します。

(例 1) 1,024KB → 「/y1024k」または「/y1024」

(例 2) 2MB → 「/y2m」

### (7) /w ディレクトリ名

ソート用一時ファイルを割り当てるディレクトリを指定します。

ソート用一時ファイルを使用する条件については、「16.6 SORT の一時ファイルの作成条件」を参照してください。

この指定は、省略できます。指定を省略した場合、使用するディレクトリの優先順位は、次の順となります。

1. 環境ファイルの `temp_dir1` で指定されているディレクトリ
2. 環境変数「TMP」で指定されているディレクトリ
3. 環境変数「TEMP」で指定されているディレクトリ
4. 次の表で示すディレクトリ

表 7-6 一時ファイルが割り当てられるディレクトリ (ソートコマンド)

OS	ディレクトリ
Windows	次の優先順位で使用するディレクトリが決まります。 1. 環境変数「USERPROFILE」で指定されているディレクトリ 2. 環境変数「SYSTEMROOT」で指定されているディレクトリ

環境ファイルの設定方法については、「15. 環境ファイルの設定」を参照してください。

## (8) /w2 ディレクトリ名

マージ用一時ファイルを割り当てるディレクトリを指定します。

マージ用一時ファイルを使用する条件については、「16. 一時ファイルの作成条件とファイル容量の算出方法」を参照してください。

指定を省略した場合、使用するディレクトリの優先順位は、次の順となります。

1. 環境ファイル `temp_dir3` に指定されているディレクトリ
2. /w で指定されているディレクトリ
3. 環境変数「TMP」で指定されているディレクトリ
4. 環境変数「TEMP」で指定されているディレクトリ
5. 表 7-6 で示すディレクトリ

なお、環境ファイルの設定方法については、「15. 環境ファイルの設定」を参照してください。

## (9) キー属性

ソートキーのキー属性を指定します。キー属性のパラメタを次の表に示します。

表 7-7 キー属性のパラメタ (ソート機能)

パラメタ	意味
/r	キーの降順でソートすることを指定する。
/j	キーが文字 (JIS8 単位コード※) であることを指定する。
/c	キーが文字であることを指定する。
/n	キーが符号付き固定小数点 2 進数であることを指定する。
/m	キーが符号なし 2 進数であることを指定する。
/p	キーが内部 10 進数であることを指定する。
/z	キーが外部 10 進数であることを指定する。
/x	キーが左符号付き外部 10 進数であることを指定する。
/s	キーが右分離符号付き外部 10 進数であることを指定する。
/t	キーが左分離符号付き外部 10 進数であることを指定する。
/f	キーが内部浮動小数点であることを指定する。
/e	キーが日付 (英数字) であることを指定する。

### 注※

ASCII コード (0x00 ~ 0x7F) を含む、JIS8 単位コード (0x00 ~ 0xFF) を意味します。入力データに日本語などのマルチバイト文字が含まれている場合、マルチバイト文字として扱わず、シングルバイト文字として扱います。EBCDIK コードの体系に従ってキー・項目を比較する「-v オプション (Windows は /v オプション)」を使用できます。

## 7. ソートコマンドの使い方

「r」を指定しないときは、キーの昇順でソートされます。また、ソートキーにするデータの属性を「j, lc, ln, lm, lp, lz, lx, ls, lt, lf, le」のパラメタのどれかで、必ず指定してください。降順でソートする場合は、「/re」のようにつけて指定します。なお、ソートオプションに「v」を指定した場合、キーの属性は必ず「j」で指定してください。

### (10) + キー位置 - キー長

キーの開始バイト位置を5桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置（先頭を0とする）とします。開始バイト位置に指定できる値の範囲は、「0～レコード長-1」です。

また、キーのバイト長を「1～4,096」の範囲で指定します。

この指定は、省略できません。

#### ソートキーの指定範囲

ソートキーは、次に示す範囲で指定してください。

- 入力ファイルが固定長ファイルのとき、キーは、レコード長の範囲
- 入力ファイルが可変長ファイルおよびテキストファイルのとき、キーは、最小レコード長の範囲

### (11) 入力ファイル名

ソートの入力ファイルを指定します。二つ以上のファイルを指定するときは、すべて同じファイル編成にしなければなりません。

入力ファイルが順ファイルまたは相対ファイルの場合、この指定は、省略できます。省略した場合は、入力ファイルに標準入力ファイルが設定されます。入力ファイルが索引ファイルの場合は、この指定は省略できません。

## 7.1.3 コマンド指定時の注意事項

ソートコマンドの指定時の注意事項を次に示します。

### (1) オペランドの入力順序

各パラメタのオペランドの入力順序は任意です。ただし、次の四つのパラメタのオペランドは、次の順序で入力してください。

- 入力ファイルのファイル編成と出力ファイルのファイル編成  
「入力ファイルのファイル編成」「出力ファイルのファイル編成」
- キー属性と+キー位置・キー長のパラメタ  
「キー属性」「+キー位置・キー長」

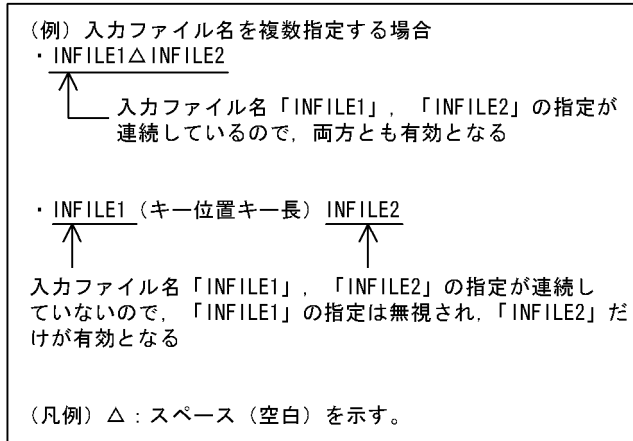
### (2) 同一パラメタのオペランドの複数指定

複数指定の不要なオペランドを複数指定した場合は、最後に指定したオペランドが有効

になります。

### (3) 複数指定できるパラメタ

複数指定できるパラメタを複数指定するときは、連続で指定してください。指定例を次に示します。



### (4) スペース (空白) の扱い

パラメタで使用する文字「/l, /o, /y, /w...」の/ (スラント) の前にスペース (空白) を入れるかどうかは任意です。しかし、+ キー位置・キー長のハイフンの前には、スペースを入れてはなりません。指定例を次に示します。

(例)

- 「△ /l128 △ /oOUTFILE」 ……有効となる
- 「+10 △ -8」 ……エラーとなる

(凡例) △ : スペース (空白) を示す。

また、パラメタで使用する文字「/l, /o, /y, /w...」とそのあとに続けるオペランドの間にスペース (空白) を入れるかどうかは任意です。指定例を次に示します。

(例)

- 「/y128」 ……有効となる
- 「/y △ 128」 ……有効となる

(凡例) △ : スペース (空白) を示す。

### (5) 英文字の扱い

各パラメタを大文字で入力するか、小文字で入力するかは任意です。また、大文字と小文字が混在してもかまいません。

## 7.1.4 使用例

ソートコマンドの使用例を示します。

### (1) 文字項目でソートする例

文字項目のキーを使ってソートする例を示します。

#### (a) 条件

ソートの条件は、次のとおりです。

- 32 ビット版 SORT を使用する
- キーの比較は JIS8 単位コードの体系に従う
- 入力ファイルおよび出力ファイルは、可変長順編成ファイル
- ソート結果の出力ファイル名は「OUTFILE」
- ソートは昇順、キーの属性は文字項目（文字）
- キーの開始バイト位置は「0」バイト目、バイト長は「5」バイト
- ソートするファイルのファイル名は「INFILE」

#### (b) 入力するコマンド

入力するソートコマンドを次の図に示します。

図 7-2 入力するソートコマンド（その 1）

```
wnsort△/gv△/oOUTFILE△/c△+0-5△ INFILE
```

（凡例） △：スペース（空白）を示す。

#### (c) コマンドの指定内容

ソートコマンドの指定内容を次の表に示します。

表 7-8 ソートコマンドの指定内容（その 1）

入力項目	指定内容
wnsort	ソート処理のコマンド名「wnsort」を指定する。
/gv	入力ファイルを「可変長順編成ファイル」で指定する。
/oOUTFILE	ソート結果の出力ファイル名「OUTFILE」を指定する。
/c	キーの整列順序を「昇順」、属性に文字項目の「文字」を指定する。
+0-5	キーの開始バイト位置に「0」バイト目、バイト長に「5」バイトを指定する。
INFILE	ソートするファイルのファイル名「INFILE」を指定する。

#### 注 1

キーの比較は JIS8 単位コード体系に従うので、指定を省略する。



注 2

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略する。

## (2) 2種類の数字項目でソートする例

2種類の数字項目をそれぞれ第1キーおよび第2キーにして、ソートする例を示します。

### (a) 条件

ソートの条件は次のとおりです。

- 32ビット版 SORT を使用する
- 重複レコードの順序は、入力順
- 入力ファイルおよび出力ファイルは、固定長順編成ファイル
- 入力ファイルおよび出力ファイルのレコード長は、128バイト
- ソート結果の出力ファイル名は、「OUTFILE」
- 使用するワークバッファサイズは、5,000KB
- ソート用一時ファイルを割り当てるディレクトリは、カレントディレクトリ下の「work」
- 第1キーでのソートは降順、キーの属性は数字項目（外部10進数）
- 第1キーの開始バイト位置は「0」バイト目、バイト長は「5」バイト
- 第2キーでのソートは昇順、キーの属性は数字項目（符号付き固定小数点2進数）
- 第2キーの開始バイト位置は「6」バイト目、バイト長は「8」バイト
- ソートするファイルのファイル名は「INFILE1」、「INFILE2」、および「INFILE3」

### (b) 入力するコマンド

入力するソートコマンドを次の図に示します。

図 7-3 入力するソートコマンド（その2）

```
wnsort△/d△/gs△/l128△/oOUTFILE△
/y5000△/w.¥work△/rz△+0-5△
/n△+6-8△INFILE1△INFILE2△
INFILE3
```

（凡例）

△：スペース（空白）を示す。

### (c) コマンドの指定内容

ソートコマンドの指定内容を次の表に示します。

表 7-9 ソートコマンドの指定内容（その2）

入力項目	指定内容
wnsort	ソート処理のコマンド名「wnsort」を指定する。
/d	重複レコードの順序は「入力順」と指定する。

## 7. ソートコマンドの使い方

入力項目	指定内容
/gs	入力ファイルに「固定長順編成ファイル」を指定する。
/l128	入力ファイルおよび出力ファイルのレコード長を「128」バイトと指定する。
/oOUTFILE	ソート結果の出力ファイル名「OUTFILE」を指定する。
/y5000	使用するワークバッファサイズを「5,000」KBと指定する。
/w.¥work	ソート用一時ファイルを割り当てるディレクトリにカレントディレクトリ下の「work」を指定する。
/rz	第1キーの整列順序に「降順」、属性に数字項目の「外部10進数」を指定する。
+0-5	第1キーの開始バイト位置に「0」バイト目、バイト長に「5」バイトを指定する。
/n	第2キーの整列順序に「昇順」、属性に数字項目の「符号付き固定小数点2進数」を指定する。
+6-8	第2キーの開始バイト位置に「6」バイト目、バイト長に「8」バイトを指定する。
INFILE1 INFILE2 INFILE3	ソートするファイルのファイル名「INFILE1」、「INFILE2」、および「INFILE3」を指定する。

### 注

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略する。

### (3) 索引ファイルのデータをソートし、順ファイルに出力する例

索引ファイルのデータをソートして、結果を順ファイルに出力する例を示します。

#### (a) 条件

ソートの条件は、次のとおりです。

- 32ビット版 SORT を使用する
- 出力ファイル名と同じ名前のファイルがすでにある場合は、そのファイルの内容を書き換ええない（既存ファイルの書き換え防止）
- キーの比較は JIS8 単位コードの体系に従う
- 入力ファイルは固定長索引ファイル
- 出力ファイルは固定長順編成ファイル
- ソート結果の出力ファイル名は「OUTFILE」
- ソートは昇順、キーの属性は文字項目（文字）
- キーの開始バイト位置は「0」バイト目、バイト長は「5」バイト
- ソートするファイルのファイル名は「INFILE」

#### (b) 入力するコマンド

入力するソートコマンドを次の図に示します。

図 7-4 入力するソートコマンド（その 3）

```
wnsort△/i△/gi:s△/oOUTFILE△/c△+0-5△INFILE
```

（凡例）△：スペース（空白）を示す。

### (c) コマンドの指定内容

ソートコマンドの指定内容を次の表に示します。

表 7-10 ソートコマンドの指定内容（その 3）

入力項目	指定内容
wnsort	ソート処理のコマンド名「wnsort」を指定する。
/i	「既存ファイルの書き換え防止」を指定する。
/gi	入力ファイルに「固定長索引ファイル」を指定する。
:s	出力ファイルに「固定長順編成ファイル」を指定する。
/oOUTFILE	ソート結果の出力先ファイル名「OUTFILE」を指定する。
/c	キーの整列順序に「昇順」、属性に文字項目の「文字」を指定する。
+0-5	キーの開始バイト位置に「0」バイト目、バイト長に「5」バイトを指定する。
INFILE	ソートするファイルのファイル名「INFILE」を指定する。

#### 注 1

キーの比較は JIS8 単位コードの体系に従うので、指定を省略する。

#### 注 2

入力ファイルが固定長索引ファイルなので、レコード長は指定しなくてよい。

## 7.2 ソートコマンドの使い方（UNIX の場合）

ソートコマンドを使うには、コマンドを入力します。

### 7.2.1 コマンドの使い方

ソートコマンドを使う場合のコマンド名を次に示します。

- `rsort`（32 ビット版 `SORT` でソート機能を使う場合）
- `rsort64`（64 ビット版 `SORT` でソート機能を使う場合）

ソートコマンドの入力方法について説明します。

#### (1) コマンドの形式

ソートコマンドは、コマンド名（`rsort` または `rsort64`）とそのあとに続く各オペランドで構成されます。ソートコマンドの形式を次の図に示します。

図 7-5 ソートコマンドの形式

```
rsort※1△ [ソートオプション]
  △ 入力ファイルのファイル編成[:出力ファイルのファイル編成]※2
  △ [-Lレコード長]
  △ [-o出力ファイル名]
  △ [-yワークバッファサイズ]
  △ [-Tディレクトリ名]
  △ [-T2ディレクトリ名2]
  △ { キー属性△+キー位置-キー長 } . . .
  △ { [入力ファイル名] } . . .
```

(凡例) △ : スペース（空白）を示す。

[ ] : 省略できることを示す。

. . . : 直前の { } で囲まれた部分を複数繰り返して指定できることを示す。

注※1 使用している `SORT` が 64 ビット版対応の場合は、「`rsort64`」となる。

注※2 入力ファイルが索引ファイルの場合、出力ファイル編成の指定は省略できない。

### 7.2.2 オペランドの意味

オペランドには、必ず指定するオペランドと任意に指定できるオペランドの 2 種類があります。

#### (1) ソートオプション

ソート処理で使用する機能上のオプションを指定します。ソートオプションのパラメタを次の表に示します。

表 7-11 ソートオプションのパラメタ

パラメタ	意味
-d	ファイル中に重複したキーのレコードがある場合、それらのデータを入力順に並べたことを指定する（重複レコードの入力順序保存）。
-e	入力ファイルを排他モードでオープンする（ファイル排他）。
-i	出力ファイル名と同じ名前のファイルがすでにある場合、そのファイルの内容が書き換えられるのを防ぐ（既存ファイルの書き換え防止）。
-v	EBCDIK コードの体系に従ってキーを比較する。
-q	出力データがキャッシュファイルに残っている場合、すべてのデータをファイル出力したあと、コマンドを終了する（キャッシュファイルの同期出力）。
-B※1	ソートキーが数値項目の場合、big endian※2の数値として扱う。

## 注

「-q」を指定した場合、コマンドの実行時間が長くなることもあるため注意してください。  
また、入力ファイルが索引ファイルの場合は、「-e」を指定した方が処理時間が短くなります。

## 注※1

Linux の場合だけ有効なオプションです。また、ソートキーが2進項目「-n」、-m」、または内部浮動小数点項目「-f」の場合だけ有効です。

## 注※2

big endian とは、上位バイトを下位番地とする数値のことです。

Linux で「-B」を指定しなかった場合は、little endian（上位バイトを上位番地とする数値）として扱います。

複数のオプションを指定する場合は、次に示すどちらかの方法で指定します。

- 「-d △ -e」のように分けて指定する（「△」は、スペース（空白）を示します）。
- 「-de」のようにつなげて指定する。

ソートオプションのパラメタの指定は、省略できます。省略した場合は、次のようになります。

- 「-d」の指定を省略した場合、重複レコードの入力順序は保存されません。
- 「-e」の指定を省略した場合、入力ファイルが共用モードでオープンされます。
- 「-i」の指定を省略した場合、出力ファイル名と同じ名前のファイルがすでにあると、そのファイルの内容が書き換えられます。
- 「-v」の指定を省略した場合、JIS8 単位コードの体系に従ってキーが比較されます。

なお、ソートオプションを指定する場合は、次の点に注意してください。

- 入力ファイルが索引ファイルの場合は、「-d」を指定しても無効になります。
- 入力ファイルの書き込みが許可されていない場合、および、入力ファイルがすでにほかの処理でオープンされている場合は、「-e」は指定できません。
- キー属性に「-j」を指定していない場合は、「-v」を指定しても無効になります。
- ファイル中に重複したキーがある場合、ワークバッファサイズを変えると、重複した

## 7. ソートコマンドの使い方

キーの順序が変わることがあります。重複したキーの順序が変わらないようにするには、「-d」を指定してください。

### (2) 入力ファイルのファイル編成

入力ファイルのファイル編成を指定します。

複数の入力ファイルを指定する場合は、指定するファイルをすべて同じファイル編成にしてください。

入力ファイルのファイル編成の指定は、省略できません。必ず指定してください。入力ファイルのファイル編成のパラメタを次の表に示します。

表 7-12 入力ファイルのファイル編成のパラメタ（ソート機能）

パラメタ	意味
-S	固定長順ファイルを指定する。
-V	可変長順ファイルを指定する。
-R	固定長相対ファイルを指定する。
-W	可変長相対ファイルを指定する。
-I	固定長索引ファイルを指定する。
-X	可変長索引ファイルを指定する。
-A	テキストファイルを指定する。

### (3) 出力ファイルのファイル編成

出力ファイルのファイル編成を指定します。出力ファイルのレコードの形式（固定長、可変長またはテキスト）は、入力ファイルと同じにしなければなりません。

入力ファイルが順ファイル、相対ファイルまたはテキストファイルの場合、この指定は省略できます。指定を省略した場合は、入力ファイルと同じファイル編成が設定されます。入力ファイルが索引ファイルの場合は、順ファイルまたは相対ファイルのどちらかを必ず指定してください。

出力ファイルのファイル編成のパラメタを次の表に示します。

表 7-13 出力ファイルのファイル編成のパラメタ（ソート機能）

パラメタ	意味
:S	固定長順ファイルを指定する。
:V	可変長順ファイルを指定する。
:R	固定長相対ファイルを指定する。
:W	可変長相対ファイルを指定する。
:A	テキストファイルを指定する。

## 注

索引ファイルに出力する場合は、いったん順ファイルにデータを出力し、索引ファイルに変換します。索引ファイルに変換するには、ISAMのユティリティコマンドをパイプで連結します。索引ファイルに出力する場合のコマンドの指定例を次に示します。

(例)

```

rsort -S -L128 -c +0-10 infile | isconv -tSI -outfile
      ↑           ↑           ↑
      ソートコマンド   パイプ   ISAMのユティリ
                          ティコマンド
  
```

## (4) -L レコード長

入力ファイル編成ごとのレコード長の指定方法を次の表に示します。

表 7-14 入力ファイル編成ごとのレコード長の指定方法（ソート機能）

入力ファイル編成	レコード長の指定方法
固定長順ファイル	ファイルのレコード長を「1～65,535」バイトの範囲で指定します。
可変長順ファイル	指定は不要です。指定しても無視されます。
固定長相対ファイル	ただし、入力できるレコード長の範囲は、1～65,535 バイトです。
可変長相対ファイル	
固定長索引ファイル	指定は不要です。指定しても無視されます。
可変長索引ファイル	ただし、入力できるレコード長の範囲は、1～65,503 バイトです。
テキストファイル	ファイルの最大レコード長を「1～65,535」バイトの範囲で指定します。 レコード長には、改行文字「X'0A'」は含みません。 この指定は省略できます。指定を省略した場合は、「65535」バイトが設定されます。 なお、レコード長を指定した方が、ソート時間が短くなります。

## (5) -o 出力ファイル名

ソート結果の出力先ファイルをファイル名で指定します。

出力先ファイルに、入力ファイルと同じファイルの指定もできます。

この指定は省略できます。指定を省略した場合は、ソート結果を標準出力ファイルに出力します。

## (6) -y ワークバッファサイズ

ワークバッファとして使用するメモリの容量を KB または MB 単位で指定します。ワークバッファサイズの指定は、メモリ量と単位で指定します。

## (a) メモリ量

メモリ量の指定範囲を次の表に示します。

## 7. ソートコマンドの使い方

表 7-15 メモリ量の指定範囲 (ソート機能)

SORT の種別	レコード長※	指定範囲
32 ビット版	32,760 バイト以下	128KB 以上 2,000MB 以内の値を 1 ~ 7 桁の数字で指定します。
	32,760 バイトより大きい	256KB 以上 2,000MB 以内の値を 1 ~ 7 桁の数字で指定します。
64 ビット版	32,760 バイト以下	256KB 以上 2,000MB 以内の値を 1 ~ 7 桁の数字で指定します。
	32,760 バイトより大きい	384KB 以上 2,000MB 以内の値を 1 ~ 7 桁の数字で指定します。

### 注※

可変長ファイルおよびテキストファイルの場合、最大レコード長となります。

この指定は省略できます。この指定を省略したとき、または最小メモリ量に満たない大きさが指定されたときは、1,024KB が設定されます。ただし、環境変数 SORTOPT に初期値が設定されている場合は、その値が設定されます。

環境変数 SORTOPT の設定方法については、「14.2 SORT の環境変数の設定 (UNIX の場合)」を参照してください。

### (b) 単位

メモリ量の単位が、KB か MB かを指定します。

メモリの単位指定を次の表に示します。

表 7-16 メモリの単位指定

単位	意味
k	単位を KB にする。
m	単位を MB にする。

また、単位を省略した場合は、「KB (k)」で指定されます。単位の指定例を次に示します。

### (例)

1,024KB → 「-y1024k」または「-y1024」

10MB → 「-y10m」

## (7) -T ディレクトリ名

ソート用一時ファイルを割り当てるディレクトリを指定します。なお、この指定は省略できます。

環境変数「TMPDIR」にディレクトリが設定されている場合は、-T で指定したディレク



トリではなく、環境変数「TMPDIR」で設定されているディレクトリ下にソート用一時ファイルが割り当てられます。

この環境変数「TMPDIR」の指定は、`-T`および環境変数「SORTOPT」の`-T`指定よりも優先されます。

ソート用一時ファイルを使用する条件については、「16. 一時ファイルの作成条件とファイル容量の算出方法」を参照してください。

使用するディレクトリの優先順位を次に示します。

1. 環境変数 TMPDIR で指定したディレクトリ
2. `rsort` コマンドの `-T` オペランドで指定したディレクトリ
3. 環境変数 SORTOPT の `-T` で指定したディレクトリ
4. 次の表で示すディレクトリ

表 7-17 ソート用一時ファイルが割り当てられるディレクトリ

OS	ディレクトリ
HP-UX	/var/tmp
AIX または Linux	/tmp

環境変数 SORTOPT の設定方法については、「14.2 SORT の環境変数の設定 (UNIX の場合)」を参照してください。

## (8) -T2 ディレクトリ名 2

マージ用一時ファイルを割り当てるディレクトリを指定します。なお、この指定は省略できます。

環境変数 TMPDIR にディレクトリが設定されている場合は、`-T2` で指定したディレクトリではなく、環境変数 TMPDIR で設定されているディレクトリ下にマージ用一時ファイルが割り当てられます。

この環境変数 TMPDIR の指定は、`-T2` および環境変数 SORTOPT の `-T2` 指定よりも優先されます。

マージ用一時ファイルを使用する条件については、「16. 一時ファイルの作成条件とファイル容量の算出方法」を参照してください。

使用するディレクトリの優先順位を次に示します。

1. 環境変数 TMPDIR で指定したディレクトリ
2. 環境変数 SORTOPT の `-T2` で指定したディレクトリ
3. `-T` で指定したディレクトリ
4. 環境変数 SORTOPT の `-T` で指定したディレクトリ

## 7. ソートコマンドの使い方

### 5. 表 7-17 で示すディレクトリ

環境変数 SORTOPT の設定方法については、「14.2 SORT の環境変数の設定 (UNIX の場合)」を参照してください。

## (9) キー属性

ソートキーの属性を指定します。キー属性のパラメタを次の表に示します。

表 7-18 キー属性のパラメタ (ソート機能)

パラメタ	意味
-r	キーを降順でソートすることを指定する。
-j	キーが文字 (JIS8 単位コード※) であることを指定する。
-c	キーが文字であることを指定する。
-n	キーが符号付き固定小数点 2 進数であることを指定する。
-m	キーが符号なし 2 進数であることを指定する。
-p	キーが内部 10 進数であることを指定する。
-z	キーが外部 10 進数であることを指定する。
-x	キーが左符号付き外部 10 進数であることを指定する。
-s	キーが右分離符号付き外部 10 進数であることを指定する。
-t	キーが左分離符号付き外部 10 進数であることを指定する。
-f	キーが内部浮動小数点であることを指定する。
-g	キーが日付 (英数字) であることを指定する。

### 注※

ASCII コード (0x00 ~ 0x7F) を含む, JIS8 単位コード (0x00 ~ 0xFF) を意味します。入力データに日本語などのマルチバイト文字が含まれている場合, マルチバイト文字として扱わず, シングルバイト文字として扱います。EBCDIK コードの体系に従ってキー・項目を比較する「-v オプション (Windows は /v オプション)」を使用できます。

「-r」を指定しないときは, キーの昇順でソートされます。また, ソートキーにするデータの属性を, 「-j, -c, -n, -m, -p, -z, -x, -s, -t, -f, -g」のパラメタのどれかで必ず指定しなければなりません。降順でソートする場合は, 「-rc」のようにつなげて指定します。なお, ソートオプションに「-v」を指定した場合, キーの属性を必ず「-j」で指定してください。

## (10) + キー位置 - キー長

キーの開始バイト位置を 5 桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置 (先頭を 0 とする) とします。開始バイト位置に指定できる値の範囲は次のとおりです。

- 順ファイルおよび相対ファイルの場合「0～65,534」
- 索引ファイルの場合「0～65,502」
- テキストファイルの場合「0～65,534」

また、キーのバイト長を4桁以内の数字「1～4,096」で指定します。

この指定は、省略できません。必ず指定してください。

#### キーの指定範囲

キーは、次に示す範囲で指定してください。

- 入力ファイルが固定長ファイルのとき、キーはレコード長の範囲
- 入力ファイルが可変長ファイルのとき、キーは最小レコード長の範囲
- 入力ファイルがテキストファイルのとき、キーは最小レコード長の範囲

### (11) 入力ファイル名

ソートの入力ファイルを指定します。同じファイルは2個以上指定できません。2個以上指定するときは、それぞれは互いに異ならなければなりません。

入力ファイルが順ファイル、相対ファイルまたはテキストファイルの場合、この指定は、省略できます。省略したときは、入力ファイルに標準入力ファイルが設定されます。入力ファイルが索引ファイルの場合は、この指定は省略できません。

## 7.2.3 コマンド指定時の注意事項

ソートコマンドの指定時の注意事項を次に示します。

### (1) パラメタの入力順序

各パラメタの入力順序は任意です。ただし、キー属性と+キー位置-キー長のパラメタ、および入力ファイルのファイル編成と出力ファイルのファイル編成のパラメタは、必ず、次に示す順序で入力してください。

- 「キー属性」「+キー位置-キー長」
- 「入力ファイルのファイル編成」「出力ファイルのファイル編成」

### (2) 同一パラメタの複数指定

同一パラメタを複数指定した場合は、最後に指定したパラメタが有効になります。

### (3) 複数指定できるパラメタ

複数指定できるパラメタを複数指定するときは、連続で指定してください。指定例を次に示します。

(例)



## (a) 条件

ソートの条件は、次のとおりです。

- 32ビット版 SORT を使用する
- キーの比較は JIS8 単位コードの体系に従う
- 入力ファイルおよび出力ファイルは、可変長順ファイル
- ソート結果の出力ファイル名は「OUTFILE」
- ソートは昇順、キーの属性は文字項目（文字）
- キーの開始バイト位置は「0」バイト目、バイト長は「5」バイト
- ソートするファイルのファイル名は「INFILE」

## (b) 入力するコマンド

入力するソートコマンドを次の図に示します。

図 7-6 入力するソートコマンド（その 1）

```
rsort△-V△-oOUTFILE△-c△+0-5△ INFILE
```

（凡例） △：スペース（空白）を示す。

## (c) コマンドの指定内容

ソートコマンドの指定内容を次の表に示します。

表 7-19 ソートコマンドの指定内容（その 1）

入力項目	指定内容
rsort	ソート処理のコマンド名「rsort」を指定する。
-V	入力ファイルを「可変長順ファイル」で指定する。
-oOUTFILE	ソート結果の出力ファイル名「OUTFILE」を指定する。
-c	キーの整列順序を「昇順」、属性を文字項目の「文字」で指定する。
+0-5	キーの開始バイト位置を「0」バイト目、バイト長を「5」バイトで指定する。
INFILE	ソートするファイルのファイル名「INFILE」を指定する。

## 注 1

キーの比較は JIS8 単位コード体系に従うので、指定を省略します。

## 注 2

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略します。

## (2) 2種類の数字項目でソートする例

2種類の数字項目をそれぞれ第 1 キーおよび第 2 キーにして、ソートする例を示します。

## 7. ソートコマンドの使い方

### (a) 条件

ソートの条件は次のとおりです。

- 32 ビット版 SORT を使用する
- 重複レコードの順序は、入力順
- 入力ファイルおよび出力ファイルは、固定長順ファイル
- 入力ファイルおよび出力ファイルのレコード長は、128 バイト
- ソート結果の出力ファイル名は、「OUTFILE」
- ワークバッファのメモリサイズは、5,000KB
- ソート用一時ファイルを割り当てるディレクトリは、カレントディレクトリ下の「st185」
- 第 1 キーでのソートは降順、キーの属性は数字項目（外部 10 進数）
- 第 1 キーの開始バイト位置は「0」バイト目、バイト長は「5」バイト
- 第 2 キーでのソートは昇順、キーの属性は数字項目（符号付き固定小数点 2 進数）
- 第 2 キーの開始バイト位置は「6」バイト目、バイト長は「8」バイト
- ソートするファイルのファイル名は「INFILE1」、「INFILE2」、および「INFILE3」

### (b) 入力するコマンド

入力するソートコマンドを次の図に示します。

図 7-7 入力するソートコマンド（その 2）

```
rsort△-d△-S△-L128△-oOUTFILE△  
-y5000△-T./st185△-rz△+0-5△  
-n△+6-8△INFILE1△INFILE2△  
INFILE3
```

（凡例） △ : スペース（空白）を示す。

### (c) コマンドの指定内容

ソートコマンドの指定内容を次の表に示します。

表 7-20 ソートコマンドの指定内容（その 2）

入力項目	指定内容
rsort	ソート処理のコマンド名「rsort」を指定する。
-d	重複レコードの順序は「入力順」と指定する。
-S	入力ファイルを「固定長順ファイル」で指定する。
-L128	入力ファイルおよび出力ファイルのレコード長を「128」バイトで指定する。
-oOUTFILE	ソート結果の出力ファイル名「OUTFILE」を指定する。
-y5000	ワークバッファのメモリサイズを「5,000」KBで指定する。

入力項目	指定内容
-T/st185	ソート用一時ファイルを割り当てるディレクトリをカレントディレクトリ下の「st185」で指定する。
-rz	第1キーの整列順序を「降順」、属性を数字項目の「外部10進数」で指定する。
+0-5	第1キーの開始バイト位置を「0」バイト目、バイト長を「5」バイトで指定する。
-n	第2キーの整列順序を「昇順」、属性を数字項目の「符号付き固定小数点2進数」で指定する。
+6-8	第2キーの開始バイト位置を「6」バイト目、バイト長を「8」バイトで指定する。
INFILE1 INFILE2 INFILE3	ソートするファイルのファイル名「INFILE1」、「INFILE2」、および「INFILE3」を指定する。

注

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略します。

### (3) 索引ファイルのデータをソートし、順ファイルに出力する例

索引ファイルのデータをソートして、結果を順ファイルに出力する例を示します。

#### (a) 条件

ソートの条件は、次のとおりです。

- 32ビット版 SORT を使用する
- 入力ファイルは排他モードでオープンする
- 出力ファイル名と同じ名前のファイルがすでにある場合は、そのファイルの内容を書き換えない（既存ファイルの書き換え防止）
- キーの比較は JIS8 単位コードの体系に従う
- 入力ファイルは固定長索引ファイル
- 出力ファイルは固定長順ファイル
- 入力ファイルおよび出力ファイルのレコード長は、128 バイト
- ソート結果の出力ファイル名は「OUTFILE」
- ソートは昇順、キーの属性は文字項目（文字）
- キーの開始バイト位置は「0」バイト目、バイト長は「5」バイト
- ソートするファイルのファイル名は「INFILE」

#### (b) 入力するコマンド

入力するソートコマンドを次の図に示します。

## 7. ソートコマンドの使い方

図 7-8 入力するソートコマンド (その 3)

```
rsort△-e△-i△-l:S△-oOUTFILE△-c△+0-5△ INFILE
```

(凡例) △ : スペース (空白) を示す。

### (c) コマンドの指定内容

ソートコマンドの指定内容を次の表に示します。

表 7-21 ソートコマンドの指定内容 (その 3)

入力項目	指定内容
rsort	ソート処理のコマンド名「rsort」を指定する。
-e	入力ファイルのオープンモードは「排他モード」と指定する。
-i	「既存ファイルの書き換え防止」を指定する。
-l	入力ファイルを「固定長索引ファイル」で指定する。
:S	出力ファイルを「固定長順ファイル」で指定する。
-oOUTFILE	ソート結果の出力先ファイル名「OUTFILE」を指定する。
-c	キーの整列順序を「昇順」、属性を文字項目の「文字」で指定する。
+0-5	キーの開始バイト位置を「0」バイト目、バイト長を「5」バイトで指定する。
INFILE	ソートするファイルのファイル名「INFILE」を指定する。

#### 注 1

キーの比較は JIS8 単位コードの体系に従うので、指定を省略します。

#### 注 2

入力ファイルが固定長索引ファイルなので、レコード長は指定しません。

### (4) 索引ファイルのデータをソートし、索引ファイルに出力する例

索引ファイルのデータをソートして、結果を索引ファイルに出力する例を示します。

#### (a) 条件

ソートの条件は、次のとおりです。

- 32 ビット版 SORT を使用する
- 入力ファイルは排他モードでオープンする
- キーの比較は EBCDIK コードの体系に従う
- 入力ファイルおよび出力ファイルは、固定長索引ファイル
- 入力ファイルおよび出力ファイルのレコード長は、128 バイト
- ソート結果の出力ファイル名は「OUTFILE」
- ソートは昇順、キーの属性は文字項目 (英数字)
- キーの開始バイト位置は「0」バイト目、バイト長は「5」バイト



- ソートするファイルのファイル名は「INFILE」

## (b) 入力するコマンド

索引ファイルに出力するので、ISAM のユーティリティコマンド「isconv」をパイプで連結します。入力するソートコマンドを次の図に示します。

図 7-9 入力するソートコマンド（その 4）

```
rsort△-e△-v△-l:S△-j△+0-5△ INFILE△
|△isconv△-tSI△-OUTFILE
```

（凡例） △：スペース（空白）を示す。

注 「isconv -tSI -OUTFILE」は、ISAM のユーティリティコマンドです。

## (c) コマンドの指定内容

ソートコマンドの指定内容を次の表に示します。

表 7-22 ソートコマンドの指定内容（その 4）

入力項目	指定内容
rsort	ソート処理のコマンド名「rsort」を指定する。
-e	入力ファイルのオープンモードは「排他モード」と指定する。
-v	キーの比較は「EBCDIK コードの体系に従う」と指定する。
-l	入力ファイルを「固定長索引ファイル」で指定する。
:S	出力ファイルを「固定長順ファイル」で指定する。
-j	キーの整列順序を「昇順」、属性を文字項目の「英数字」で指定する。
+0-5	キーの開始バイト位置を「0」バイト目、バイト長を「5」バイトで指定する。
INFILE	ソートするファイルのファイル名「INFILE」を指定する。

## 注 1

入力ファイルが固定長索引ファイルなので、レコード長は指定しません。



# 8

## マージコマンドの使い方

この章では、マージコマンドの使い方について説明します。

---

8.1 マージコマンドの使い方（Windows の場合）

---

8.2 マージコマンドの使い方（UNIX の場合）

---

## 8.1 マージコマンドの使い方（Windows の場合）

---

### 8.1.1 コマンドの使い方

コマンドを入力して、マージ機能を使う方法について説明します。

マージコマンドのコマンド名を次に示します。

- `wnmerge`（32 ビット版 SORT でマージコマンドを使う場合）
- `wnmerge64`（64 ビット版 SORT でマージコマンドを使う場合）

マージコマンドの形式について説明します。

#### (1) コマンドの形式

マージコマンドは、コマンド名（`wnmerge` または `wnmerge64`）とそのあとに続く各オペランドで構成されます。マージコマンドの形式を次の図に示します。

図 8-1 マージコマンドの形式

```
wnmerge*Δ [マージオプション]
Δ /g入力ファイルのファイル編成[:出力ファイルのファイル編成]
Δ [/lレコード長]
Δ [/o出力ファイル名]
Δ { キー属性Δ+キー位置-キー長 } . . .
Δ { [入力ファイル名] } . . .
```

(凡例) Δ : スペース（空白）を示す。

[ ] : 省略できることを示す。

. . . : 直前の { } で囲まれた部分を複数繰り返して指定できることを示す。

注※ 使用している SORT が 64 ビット版対応の場合は、「`wnmerge64`」となる。

### 8.1.2 オペランドの意味

オペランドには、必ず指定するオペランドと任意に指定できるオペランドの 2 種類があります。

#### (1) マージオプション

マージ処理で使用する機能上のオプションを指定します。マージオプションのパラメータを次の表に示します。

表 8-1 マージオプションのパラメタ

パラメタ	意味
/k	マージするファイルが昇順または降順にソートされていない場合に、強制的にマージすることを指定する（強制マージ）。
/i	出力ファイル名と同じ名前のファイルがすでにある場合、そのファイルの内容が書き換えられるのを防ぐ（既存ファイルの書き換え防止）。
/v	EBCDIK コードの体系に従ってキーを比較する。
/b <sup>※1</sup>	マージキーが数値項目の場合、big endian <sup>※2</sup> の数値として扱う。

## 注※1

マージキーが2進項目「/n」「/m」、または内部浮動小数点項目「/f」の場合だけ有効となる。

## 注※2

big endian とは、上位バイトを下位番地とする数値のことです。

/b パラメタを指定しなかった場合は、little endian（上位バイトを上位番地とする数値）として扱います。

複数のオプションを指定する場合は、次に示すどちらかの方法で指定します。

- 「/k Δ /i」のように分けて指定します。
  - 「/ki」のようにつなげて指定します。
- （凡例） Δ：スペース（空白）を示します。

また、この指定は省略できます。省略した場合は、次のようになります。

- 「/k」の指定を省略した場合、通常のマージをします（マージするファイルが昇順または降順に並んでいない場合は、エラーメッセージが表示されます）。
- 「/i」の指定を省略した場合、出力ファイル名と同じ名前のファイルがすでにあると、そのファイルの内容が書き換えられます。
- 「/v」の指定を省略した場合、JIS8 単位コードの体系に従ってキーが比較されます。

なお、キー属性に「/j」を指定していない場合は、「/v」を指定しても無効になります。

## (2) /g 入力ファイルのファイル編成

入力ファイルのファイル編成を「/g +オペランド」の形で指定します。複数の入力ファイルを指定する場合は、すべて同じファイル編成にしてください。

また、この指定は省略できません。入力ファイルのファイル編成のオペランドを次の表に示します。

表 8-2 入力ファイルのファイル編成のオペランド（マージ機能）

オペランド	意味
s	固定長順編成ファイルを指定する。
v	可変長順編成ファイルを指定する。

## 8. マージコマンドの使い方

オペランド	意味
r	固定長相対ファイルを指定する。
w	可変長相対ファイルを指定する。
a	テキストファイルを指定する。

### (3) 出力ファイルのファイル編成

出力ファイルのファイル編成のオペランドを、コロンで区切って指定します。出力ファイルのファイルの形式（固定長、可変長またはテキスト）は、入力ファイルと同じにしてください。

この指定は省略できます。指定を省略した場合は、入力ファイルと同じファイル編成が設定されます。出力ファイルのファイル編成のオペランドを次の表に示します。

表 8-3 出力ファイルのファイル編成のオペランド（マージ機能）

オペランド	意味
s	固定長順編成ファイルを指定する。
v	可変長順編成ファイルを指定する。
r	固定長相対ファイルを指定する。
w	可変長相対ファイルを指定する。
a	テキストファイルを指定する。

### (4) /l レコード長

入力ファイル編成ごとのレコード長の指定方法を次の表に示します。

表 8-4 入力ファイル編成ごとのレコード長の指定方法（マージ機能）

入力ファイル編成	レコード長の指定方法
固定長順編成ファイル	ファイルのレコード長を「1～65,535」バイトの範囲で指定します。 ただし、入力できるレコード長の範囲は、1～65,535 バイトです。
可変長順編成ファイル	
固定長相対ファイル	
可変長相対ファイル	
テキストファイル	ファイルの最大レコード長を「1～65,535」バイトの範囲で指定します。 レコード長には、改行文字「X'0A'」は含みません。 省略できます。省略した場合「65,535」バイトが設定されます。

### (5) /o 出力ファイル名

マージ結果の出力先ファイルをファイル名で指定します。出力先ファイルは、入力ファイルとは異なるファイルである必要があります。

この指定は、省略できます。指定を省略した場合は、マージ結果を標準出力ファイルに

出力します。

## (6) キー属性

マージキーのキー属性を指定します。キー属性のパラメタを次の表に示します。

表 8-5 キー属性のパラメタ (マージ機能)

パラメタ	意味
/r	キーの降順でマージすることを指定する。
/j	キーが文字 (JIS8 単位コード※) であることを指定する。
/c	キーが文字であることを指定する。
/n	キーが符号付き固定小数点 2 進数であることを指定する。
/m	キーが符号なし 2 進数であることを指定する。
/p	キーが内部 10 進数であることを指定する。
/z	キーが外部 10 進数であることを指定する。
/x	キーが左符号付き外部 10 進数であることを指定する。
/s	キーが右分離符号付き外部 10 進数であることを指定する。
/t	キーが左分離符号付き外部 10 進数であることを指定する。
/f	キーが内部浮動小数点であることを指定する。
/e	キーが日付 (英数字) であることを指定する。

### 注※

ASCII コード (0x00 ~ 0x7F) を含む, JIS8 単位コード (0x00 ~ 0xFF) を意味します。入力データに日本語などのマルチバイト文字が含まれている場合, マルチバイト文字として扱わず, シングルバイト文字として扱います。EBCDIK コードの体系に従ってキー・項目を比較する「-v オプション (Windows は /v オプション)」を使用できます。

「/r」を指定しないときは, キーの昇順でマージされます。また, マージキーにするデータの属性を, 「/j, /c, /n, /m, /p, /z, /x, /s, /t, /f」のパラメタのどれかで必ず指定してください。降順でマージする場合は, 「/rc」のようにつなげて指定します。なお, マージオプションに「/v」を指定した場合, キーの属性を必ず「/j」で指定してください。

## (7) + キー位置 - キー長

キーの開始バイト位置を 5 桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置 (先頭を 0 とする) とします。開始バイト位置に指定できる値の範囲は, 「0 ~ レコード長 - 1」です。

また, キーのバイト長を「1 ~ 4,096」から指定します。

この指定は, 省略できません。必ず, 指定してください。

### マージキーの指定範囲

## 8. マージコマンドの使い方

マージキーは、次に示す範囲で指定してください。

- 入力ファイルが固定長ファイルのとき、キーは、レコード長の範囲
- 入力ファイルが可変長ファイルおよびテキストファイルのとき、キーは、最小レコード長の範囲

### (8) 入力ファイル名

マージの入力ファイルを指定します。同じファイルを 2 個以上指定できません。2 個以上指定するときは、それぞれは互いに異ならなければなりません。

この指定は、省略できます。指定を省略した場合は、入力ファイルに標準入力ファイルが設定されます。

### 8.1.3 コマンド指定時の注意事項

マージコマンドの指定時の注意事項を次に示します。

#### (1) オペランドの入力順序

各パラメタのオペランドの入力順序は任意です。ただし、次に示す四つのパラメタのオペランドは、次の順序で入力してください。

- 入力ファイルのファイル編成と出力ファイルのファイル編成  
「入力ファイルのファイル編成」「出力ファイルのファイル編成」
- キー属性と + キー位置 - キー長のパラメタ  
「キー属性」「+ キー位置 - キー長」

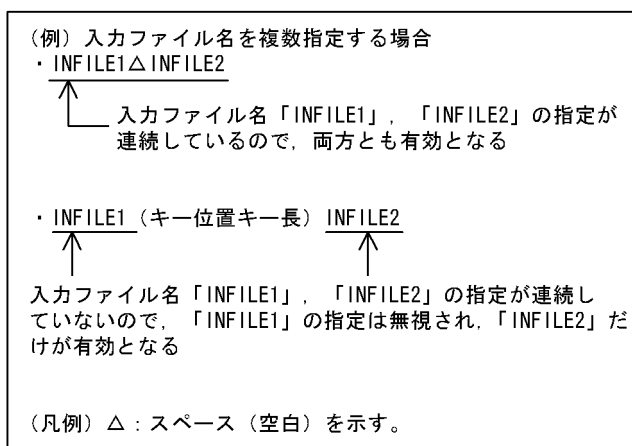
#### (2) 同一パラメタのオペランドの複数指定

複数指定が不要なオペランドを複数指定した場合は、最後に指定したオペランドが有効になります。

#### (3) 複数指定できるパラメタ

複数指定できるパラメタを複数指定するときは、連続で指定してください。指定例を次に示します。





#### (4) スペース (空白) の扱い

パラメタで使用する文字「/l, /o...」の / (スラント) の前にスペース (空白) を入れるかどうかは任意です。しかし, + キー位置 - キー長のハイフンの前には, スペースを入れな  
いでください。指定例を次に示します。

(例)

- 「△ /l128 △ /oOUTFILE」 ……有効となる
- 「+10 △ -8」 ……エラーとなる

(凡例) △: スペース (空白) を示す。

また, パラメタで使用する文字「/l, /o...」とそのあとに続けるオペランドの間にスペース (空白) を入れるかどうかは任意です。指定例を次に示します。

(例)

- 「/l128」 ……有効となる
- 「/l △ 128」 ……有効となる

(凡例) △: スペース (空白) を示す。

#### (5) 英文字の扱い

各パラメタを大文字で入力するか, 小文字で入力するかは任意です。また, 大文字と小文字が混在してもかまいません。

### 8.1.4 使用例

マージコマンドの使用例を示します。

#### (1) 文字項目でマージする例

文字項目のキーを使ってマージする例を示します。

## 8. マージコマンドの使い方

### (a) 条件

マージの条件は、次のとおりです。

- 32 ビット版 SORT を使用する
- キーの比較は JIS8 単位コードの体系に従う
- 入力ファイルおよび出力ファイルは、共に可変長順編成ファイル
- マージ結果の出力ファイル名は「OUTFILE」
- マージは昇順、キーの属性は文字項目（文字）
- キーの開始バイト位置は「0」バイト目、バイト長は「5」バイト
- マージする入力ファイルのファイル名は「INFILE1」、「INFILE2」

### (b) 入力するコマンド

入力するマージコマンドを次の図に示します。

図 8-2 入力するマージコマンド（その 1）

```
wnmerge△/gv△/oOUTFILE△/c△+0-5△ INFILE1△ INFILE2
```

（凡例）△：スペース（空白）を示す。

### (c) コマンドの指定内容

マージコマンドの指定内容を次の表に示します。

表 8-6 マージコマンドの指定内容（その 1）

入力項目	指定内容
wnmerge	マージ処理のコマンド名「wnmerge」を指定する。
/gv	入力ファイルのファイル編成を「可変長順編成ファイル」で指定する。
/oOUTFILE	マージ結果の出力ファイル名「OUTFILE」を指定する。
/c	キーの整列順序を「昇順」、属性を文字項目の「文字」で指定する。
+0-5	キーの開始バイト位置を「0」バイト目、バイト長を「5」バイトで指定する。
INFILE1 INFILE2	マージする入力ファイルのファイル名「INFILE1」および「INFILE2」を指定する。

注

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略する。

## (2) 複数の文字項目で順ファイルをマージする例

複数の文字項目のキーを使ってマージする例を示します。

### (a) 条件

マージの条件は、次のとおりです。

- 32 ビット版 SORT を使用する
- キーの比較は JIS8 単位コードの体系に従う
- 入力ファイルおよび出力ファイルは、共に固定長順編成ファイル
- 入力ファイルおよび出力ファイルのレコード長は、128 バイト
- マージ結果の出力ファイル名は「OUTFILE」
- 第 1 キーでのマージは昇順、キーの属性は文字項目（文字）
- 第 1 キーの開始バイト位置は「0」バイト目、バイト長は「5」バイト
- 第 2 キーでのマージは降順、キーの属性は文字項目（文字）
- 第 2 キーの開始バイト位置は「10」バイト目、バイト長は「6」バイト
- マージする入力ファイルのファイル名は「INFILE1」、「INFILE2」および「INFILE3」

## (b) 入力するコマンド

入力するマージコマンドを次の図に示します。

図 8-3 入力するマージコマンド（その 2）

```
wnmerge△/gs△/l128△/oOUTFILE
△/c△+0-5△/cr△+10-6
△INFILE1△INFILE2△INFILE3
```

（凡例）△：スペース（空白）を示す。

## (c) コマンドの指定内容

マージコマンドの指定内容を次の表に示します。

表 8-7 マージコマンドの指定内容（その 2）

入力項目	指定内容
wnmerge	マージ処理のコマンド名「wnmerge」を指定する。
/gs	入力ファイルのファイル編成を「固定長順編成ファイル」で指定する。
/l128	レコード長 128 バイトを指定する。
/oOUTFILE	マージ結果の出力ファイル名「OUTFILE」を指定する。
/c	第 1 キーの整列順序を「昇順」、属性を文字項目の「文字（英数字および漢字）」で指定する。
+0-5	第 1 キーの開始バイト位置を「0」バイト目、バイト長を「5」バイトで指定する。
/cr	第 2 キーの整列順序を「降順」、属性を文字項目の「文字」で指定する。
+10-6	第 2 キーの開始バイト位置を「10」バイト目、バイト長を「6」バイトで指定する。
INFILE1 INFILE2 INFILE3	マージする入力ファイルのファイル名「INFILE1」、「INFILE2」、および「INFILE3」を指定する。

注

## 8. マージコマンドの使い方

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略する。

## 8.2 マージコマンドの使い方（UNIX の場合）

マージコマンドを使うには、コマンドを入力します。

マージ機能を使う場合のコマンド名を次に示します。

- `rmerge`（32 ビット版 SORT でソート機能を使う場合）
- `rmerge64`（64 ビット版 SORT でソート機能を使う場合）

マージコマンドの入力方法について説明します。

### 8.2.1 コマンドの使い方

マージコマンドは、コマンド名（`rmerge` または `rmerge64`）とそのあとに続く各オペランドで構成されます。マージコマンドの形式を次の図に示します。

図 8-4 マージコマンドの形式

```
rmerge*1 Δ [マージオプション]
      Δ 入力ファイルのファイル編成 [: 出力ファイルのファイル編成]*2
      Δ [-Lレコード長]
      Δ [-o出力ファイル名]
      Δ { キー属性Δ+キー位置-キー長 } . . .
      Δ { [入力ファイル名] } . . .
```

(凡例)     Δ : スペース（空白）を示す。  
           [ ] : 省略できることを示す。  
           . . . : 直前の { } で囲まれた部分を複数繰り返して指定できることを示す。

注※1  使用しているSORTが64ビット版対応の場合は、「`rmerge64`」となる。

注※2  入力ファイルが索引ファイルの場合、出力ファイルのファイル編成の指定は省略できない。

### 8.2.2 オペランドの意味

オペランドには、必ず指定するオペランドと任意に指定できるオペランドの2種類があります。

#### (1) マージオプション

マージ処理で使用する機能上のオプションを指定します。マージオプションのパラメタを次の表に示します。

表 8-8 マージオプションのパラメタ

パラメタ	意味
<code>-k</code>	マージするファイルが昇順または降順にソートされていない場合に、強制的にマージすることを指定する（強制マージ）。

## 8. マージコマンドの使い方

パラメタ	意味
-e	入力ファイルを排他モードでオープンする（ファイル排他）。
-i	出力ファイル名と同じ名前のファイルがすでにある場合、そのファイルの内容が書き換えられるのを防ぐ（既存ファイルの書き換え防止）。
-v	EBCDIK コードの体系に従ってキーを比較する。
-q	出力データがキャッシュファイルに残っている場合、すべてのデータをファイル出力したあと、コマンドを終了する（キャッシュファイルの同期出力）。
-B <sup>※1</sup>	マージキーが数値項目の場合、big endian <sup>※2</sup> の数値として扱う。

### 注

「-q」を指定した場合、コマンドの実行時間が長くなることがあるため注意してください。  
また、入力ファイルが索引ファイルの場合は、「-e」を指定した方が処理時間が短くなります。

### 注※1

Linux の場合だけ有効なオプションです。また、マージキーが2進項目「-n」、 「-m」、または内部浮動小数点項目「-f」の場合だけ有効です。

### 注※2

big endian とは、上位バイトを下位番地とする数値のことです。

Linux で「-B」を指定しなかった場合は、little endian（上位バイトを上位番地とする数値）として扱います。

複数のオプションを指定する場合は、次に示すどちらかの方法で指定します。

- 「-k △ -e」のように分けて指定する（「△」は、スペース（空白）を示します）。
- 「-ke」のようにつなげて指定する。

マージオプションのパラメタの指定は、省略できます。省略した場合は、次のようになります。

- 「-k」の指定を省略した場合、通常のマージをします（マージするファイルが昇順または降順に並んでいない場合は、エラーメッセージが表示されます）。
- 「-e」の指定を省略した場合、入力ファイルが共用モードでオープンされます。
- 「-i」の指定を省略した場合、出力ファイル名と同じ名前のファイルがすでにあるときは、そのファイルの内容が書き換えられます。
- 「-v」の指定を省略した場合、JIS8 単位コードの体系に従ってキーが比較されます。

なお、マージオプションを指定する場合は、次の点に注意してください。

- 入力ファイルの書き込みが許可されていない場合、および入力ファイルがすでにほかの処理でオープンされている場合は、「-e」は指定できません。
- キー属性に「-j」を指定していない場合は、「-v」を指定しても無効になります。

## (2) 入力ファイルのファイル編成

入力ファイルのファイル編成を指定します。

複数の入力ファイルを指定する場合は、指定するファイルをすべて同じファイル編成にしてください。

入力ファイルのファイル編成の指定は、省略できません。必ず指定してください。入力ファイルのファイル編成のパラメタを次の表に示します。

表 8-9 入力ファイルのファイル編成のパラメタ（マージ機能）

パラメタ	意味
-S	固定長順ファイルを指定する。
-V	可変長順ファイルを指定する。
-R	固定長相対ファイルを指定する。
-W	可変長相対ファイルを指定する。
-I	固定長索引ファイルを指定する。
-X	可変長索引ファイルを指定する。
-A	テキストファイルを指定する。

### (3) 出力ファイルのファイル編成

出力ファイルのファイル編成を指定します。出力ファイルのレコードの形式（固定長、可変長またはテキスト）は、入力ファイルと同じにしなければなりません。

入力ファイルが順ファイル、相対ファイルまたはテキストファイルの場合、この指定は省略できます。指定を省略した場合は、入力ファイルと同じファイル編成が設定されます。入力ファイルが索引ファイルの場合は、順ファイルまたは相対ファイルのどちらかを必ず指定してください。

出力ファイルのファイル編成のパラメタを次の表に示します。

表 8-10 出力ファイルのファイル編成のパラメタ（マージ機能）

パラメタ	意味
:S	固定長順ファイルを指定する。
:V	可変長順ファイルを指定する。
:R	固定長相対ファイルを指定する。
:W	可変長相対ファイルを指定する。
:A	テキストファイルを指定する。

#### 注

索引ファイルに出力する場合は、いったん順ファイルにデータを出力し、索引ファイルに変換します。索引ファイルに変換するには、ISAMのユティリティコマンドをパイプで連結します。索引ファイルに出力する場合のコマンドの指定例を次に示します。

(例)

## 8. マージコマンドの使い方

```

rmerge -l:S -L128 -c +0-10 infile1 infile2 | isconv -tSI -outfile
          ↑           ↑           ↑
          マージコマンド   パイプ   ISAMのユティリティコマンド
    
```

### (4) -L レコード長

入力ファイル編成ごとのレコード長の指定方法を次の表に示します。

表 8-11 入力ファイル編成ごとのレコード長の指定方法（マージ機能）

入力ファイル編成	レコード長の指定方法
固定長順ファイル	ファイルのレコード長を「1～65,535」バイトの範囲で指定します。
可変長順ファイル	指定は不要です。指定しても無視されます。 ただし、入力できるレコード長の範囲は、1～65,535 バイトです。
固定長相対ファイル	
可変長相対ファイル	
固定長索引ファイル	指定は不要です。指定しても無視されます。 ただし、入力できるレコード長の範囲は、1～65,503 バイトです。
可変長索引ファイル	
テキストファイル	ファイルの最大レコード長を「1～65,535」バイトの範囲で指定します。 レコード長には、改行文字「X'0A'」は含みません。 この指定は省略できます。指定を省略した場合は、「65,535」バイトが設定されます。 なお、レコード長を指定した方が、ソート時間が短くなります。

### (5) -o 出力ファイル名

マージ結果の出力先ファイルをファイル名で指定します。出力先ファイルは、入力ファイルとは異なるファイルでなければなりません。

-o 出力ファイル名の指定は、省略できます。指定を省略した場合は、マージ結果を標準出力ファイルに出力します。

### (6) キー属性

マージキーの属性を指定します。キー属性のパラメタを次の表に示します。

表 8-12 キー属性のパラメタ（マージ機能）

パラメタ	内容
-r	キーの降順でマージすることを指定する。
-j	キーが文字（JIS8 単位コード※）であることを指定する。
-c	キーが文字であることを指定する。
-n	キーが符号付き固定小数点 2 進数であることを指定する。
-m	キーが符号なし 2 進数であることを指定する。



パラメタ	内容
-p	キーが内部 10 進数であることを指定する。
-z	キーが外部 10 進数であることを指定する。
-x	キーが左符号付き外部 10 進数であることを指定する。
-s	キーが右分離符号付き外部 10 進数であることを指定する。
-t	キーが左分離符号付き外部 10 進数であることを指定する。
-f	キーが内部浮動小数点であることを指定する。
-g	キーが日付（英数字）であることを指定する。

#### 注※

ASCII コード (0x00 ~ 0x7F) を含む、JIS8 単位コード (0x00 ~ 0xFF) を意味します。入力データに日本語などのマルチバイト文字が含まれている場合、マルチバイト文字として扱わず、シングルバイト文字として扱います。EBCDIK コードの体系に従ってキー・項目を比較する「-v オプション (Windows は /v オプション)」を使用できます。

「-r」を指定しないときは、キーの昇順でマージされます。また、マージキーにするデータの属性を、「-j, -c, -n, -m, -p, -z, -x, -s, -t, -f, -g」のパラメタのどれかで必ず指定しなければなりません。降順でマージする場合は、「-rc」のようにつなげて指定します。なお、マージオプションに「-v」を指定した場合、キーの属性を必ず「-j」で指定してください。

### (7) + キー位置 - キー長

キーの開始バイト位置を 5 桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置（先頭を 0 とする）とします。開始バイト位置に指定できる値の範囲は次のとおりです。

- 順ファイルおよび相対ファイルの場合「0 ~ 65,534」
- 索引ファイルの場合「0 ~ 65,502」
- テキストファイルの場合「0 ~ 65,534」

なお、入力ファイルが索引ファイルの場合、マージの第 1 キーは索引ファイルの主キーまたは副キーでなければなりません。

また、キーのバイト長を 4 桁以内の数字「1 ~ 4,096」で指定します。

この指定は、省略できません。必ず指定してください。

#### キーの指定範囲

キーは、次に示す範囲で指定してください。

- 入力ファイルが固定長ファイルのとき、キーはレコード長の範囲
- 入力ファイルが可変長ファイルのとき、キーは最小レコード長の範囲
- 入力ファイルがテキストファイルのとき、キーは最小レコード長の範囲

## 8. マージコマンドの使い方

### (8) 入力ファイル名

マージの入力ファイルを指定します。同じファイルを2個以上指定できません。2個以上指定するときは、それぞれは互いに異ならなければなりません。

入力ファイルが順ファイル、相対ファイルまたはテキストファイルの場合、この指定は省略できます。省略した場合は、入力ファイルに標準入力ファイルが設定されます。入力ファイルが索引ファイルの場合は、この指定は省略できません。

### 8.2.3 コマンド指定時の注意事項

マージコマンドの指定時の注意事項を次に示します。

#### (1) マージの条件

通常のマージでは、入力ファイルが次の条件を満たしていることが必要です。

- 昇順でマージする場合、入力ファイルのデータは昇順に並んでいなければなりません。
- 降順でマージする場合、入力ファイルのデータは降順に並んでいなければなりません。

この条件を満たしていないファイルをマージしようとするとうエラーになります。しかし、マージオプションで強制マージを指定すれば、マージできます。ただし、この場合のマージ後のデータの並びは保証しません。

#### (2) パラメタの入力順序

各パラメタの入力順序は任意です。ただし、キー属性と+キー位置・キー長のパラメタ、および入力ファイルのファイル編成と出力ファイルのファイル編成のパラメタは、次に示す順序で入力しなければなりません。

- 「キー属性」「+キー位置・キー長」
- 「入力ファイルのファイル編成」「出力ファイルのファイル編成」

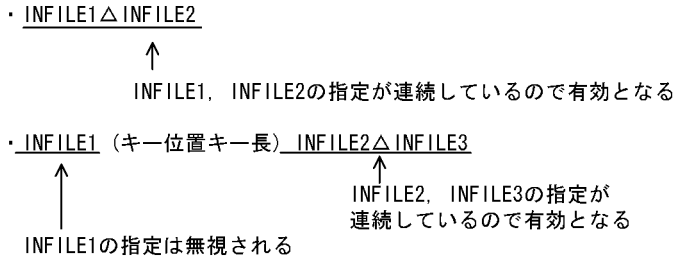
#### (3) 同一パラメタの複数指定

同一パラメタを複数指定した場合は、最後に指定したパラメタが有効になります。

#### (4) 複数指定できるパラメタ

複数指定できるパラメタを複数指定するときは、連続で指定してください。指定例を次に示します。

(例)



(凡例) Δ : スペース (空白) を示す。

### (5) スペース (空白) の扱い

オペランドで使用する文字「-L, -o」の・ (ハイフン) の前にはスペース (空白) を入れなければなりません。ただし、+キー位置・キー長のハイフンの前には、スペースは不要です。指定例を次に示します。

(例)

- 「Δ -L128 Δ -oOUTFILE」 ……有効となる
- 「Δ -L128-oOUTFILE」 ……エラーとなる

(凡例) Δ : スペース (空白) を示す。

また、オペランドで使用する文字「-L, -o」とそのあとに続けるパラメタの間にスペース (空白) を入れるかどうかは任意です。指定例を次に示します。

(例)

- 「-L128」 ……有効となる
- 「-L Δ 128」 ……有効となる

(凡例) Δ : スペース (空白) を示す。

### (6) 入力ファイルを複数指定する場合

固定長ファイルの場合、レコード長はすべて同じでなければなりません。

可変長ファイルまたは、テキストファイルの場合、入力ファイルの中で最も小さい最小レコード長を出力ファイルの最小レコード長に適用し、入力ファイルの中で最も大きい最大レコード長を出力ファイルの最大レコード長に適用します。

### (7) 入力ファイルが索引ファイルの場合

入力ファイルが索引ファイルの場合、マージの第1キーには、索引ファイルの主キーまたは副キーを指定する必要があります。

## 8.2.4 使用例

マージコマンドの使用例を示します。

### (1) 文字項目で相対ファイルをマージする例

文字項目のキーを使って、相対ファイルをマージする例を示します。

#### (a) 条件

マージの条件は、次のとおりです。

- 32 ビット版 SORT を使用する
- キーの比較は JIS8 単位コードの体系に従う
- 入力ファイルおよび出力ファイルは、共に可変長順ファイル
- マージ結果の出力ファイル名は「OUTFILE」
- マージは昇順、キーの属性は文字項目（文字）
- キーの開始バイト位置は「0」バイト目、バイト長は「5」バイト
- マージするファイルのファイル名は「INFILE1」、「INFILE2」

#### (b) 入力するコマンド

入力するマージコマンドを次の図に示します。

図 8-5 入力するマージコマンド（その 1）

```
rmerge△-V△-oOUTFILE△-c△+0-5△ INFILE1△ INFILE2
```

（凡例） △ : スペース（空白）を示す。

#### (c) コマンドの指定内容

マージコマンドの指定内容を次の表に示します。

表 8-13 マージコマンドの指定内容（その 1）

入力項目	指定内容
rmerge	マージ処理のコマンド名「rmerge」を指定する。
-V	入力ファイルのファイル編成を「可変長順ファイル」で指定する。
-oOUTFILE	マージ結果の出力ファイル名「OUTFILE」を指定する。
-c	キーの整列順序を「昇順」、属性を文字項目の「文字」で指定する。
+0-5	キーの開始バイト位置を「0」バイト目、バイト長を「5」バイトで指定する。
INFILE1 INFILE2	マージするファイルのファイル名「INFILE1」および「INFILE2」を指定する。

#### 注 1

キーの比較は JIS8 単位コード体系に従うため、指定を省略します。

注 2

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略します。

## (2) 2種類の文字項目で索引ファイルをマージする例

2種類の文字項目をそれぞれ第1キーおよび第2キーにして、索引ファイルをマージする例を示します。

### (a) 条件

マージの条件は次のとおりです。

- 32ビット版 SORT を使用する
- 入力ファイルは排他モードでオープンする
- 出力ファイル名と同じ名前のファイルがすでにある場合は、そのファイルの内容を書き換えない（既存ファイルの書き換え防止）
- キーの比較は EBCDIK コードの体系に従う
- 入力ファイルは固定長索引ファイル
- 出力ファイルは固定長順ファイル
- 入力ファイルおよび出力ファイルのレコード長は、128 バイト
- マージ結果の出力ファイル名は、「OUTFILE」
- 第1キー※でのマージは昇順、キーの属性は文字項目（英数字）
- 第1キーの開始バイト位置は「0」バイト目、バイト長は「5」バイト
- 第2キーでのマージは降順、キーの属性は文字項目
- 第2キーの開始バイト位置は「6」バイト目、バイト長は「8」バイト
- マージするファイルのファイル名は「INFILE1」、「INFILE2」、および「INFILE3」

注※

入力ファイルが索引ファイルの場合、マージの第1キーには、索引ファイルの主キーまたは副キーを指定してください。

### (b) 入力するコマンド

入力するマージコマンドを次の図に示します。

図 8-6 入力するマージコマンド（その2）

```
rmerge△-e△-i△-v△-l:△S△-oOUTFILE
△-j△+0-5△-jr△+6-8
△INFILE1△INFILE2△INFILE3
```

（凡例） △ : スペース（空白）を示す。

### (c) コマンドの指定内容

マージコマンドの指定内容を次の表に示します。

## 8. マージコマンドの使い方

表 8-14 マージコマンドの指定内容（その2）

入力項目	指定内容
rmerge	マージ処理のコマンド名「rmerge」を指定する。
-e	入力ファイルのオープンモードは「排他モード」と指定する。
-i	「既存ファイルの書き換え防止」を指定する。
-v	キーの比較は「EBCDIK コードの体系に従う」と指定する。
-l	入力ファイルのファイル編成を「固定長索引ファイル」で指定する。
:S	出力ファイルのファイル編成を「固定長順ファイル」で指定する。
-oOUTFILE	マージ結果の出力ファイル名「OUTFILE」を指定する。
-j	第1キーの整列順序を「昇順」、属性を文字項目の「英数字」で指定する。
+0-5	第1キーの開始バイト位置を「0」バイト目、バイト長を「5」バイトで指定する※。
-jr	第2キーの整列順序を「降順」、属性を文字項目の「英数字」で指定する。
+6-8	第2キーの開始バイト位置を「6」バイト目、バイト長を「8」バイトで指定する。
INFILE1 INFILE2 INFILE3	マージするファイルのファイル名「INFILE1」、「INFILE2」、および「INFILE3」を指定する。

### 注

入力ファイルが固定長索引ファイルなので、レコード長は指定しません。

### 注※

索引ファイルをマージする場合は、必ず、マージの第1キーに索引ファイルの主キーまたは副キーを指定してください。

# 9

## 選択コマンドの使い方

選択機能を使うには、コマンドを入力します。コマンドで、選択したいデータが収められているファイルを入力ファイルに指定し、選択結果をどのファイルに出力するのかを指定します。さらに、選択項目の属性や位置などを指定します。この章では、選択機能で使用するコマンドについて説明します。

---

9.1 選択コマンドの使い方 (Windows の場合)

---

9.2 選択コマンドの使い方 (UNIX の場合)

---

## 9.1 選択コマンドの使い方（Windows の場合）

選択コマンドのコマンド名を次に示します。

- `wselect`（32 ビット版 SORT で選択コマンドを使う場合）
- `wselect64`（64 ビット版 SORT で選択コマンドを使う場合）

選択コマンドの入力方法について説明します。

### 9.1.1 コマンドの使い方

選択コマンドは、コマンド名（`wselect` または `wselect64`）とそのあとに続く各オペランドで構成されます。選択機能のコマンドの形式を次の図に示します。

図 9-1 選択コマンドの形式

```
wselect※1 Δ [選択オプション]
           Δ /g入力ファイルのファイル編成[:出力ファイルのファイル編成]※2
           Δ [/lレコード長]
           Δ [/o出力ファイル名]
           Δ [/s|"選択条件式"]
           Δ { [入力ファイル名] } . . .
```

（凡例） Δ : スペース（空白）を示す。

[ ] : 省略できることを示す。

. . . : 直前の { } で囲まれた部分を複数繰り返し指定できることを示す。

注※1 使用している SORT が 64 ビット版対応の場合は、「`wselect64`」となる。

注※2 入力ファイルが索引ファイルの場合、出力ファイルのファイル編成の指定は省略できない。

### 9.1.2 オペランドの意味

オペランドには、必ず指定するオペランドと任意に指定できるオペランドの 2 種類があります。

#### (1) 選択オプション

選択処理で使用する機能上のオプションを指定します。選択オプションのパラメタを次の表に示します。

表 9-1 選択オプションのパラメタ

パラメタ	意味
<code>/i</code>	出力ファイル名と同じ名前のファイルがすでにある場合、そのファイルの内容が書き換えられるのを防ぐ（既存ファイルの書き換え防止）。



パラメタ	意味
/v 比較レベル	EBCDIK コードの体系に従って項目を比較する。 <ul style="list-style-type: none"> <li>• 選択条件式の比較項目が JIS8 単位コードの英数字の場合比較レベルに 1 を指定するか、または比較レベルを指定しない。</li> <li>• 選択条件式の比較項目が EBCDIK コードの英数字の場合比較レベルに 2 を指定する。</li> </ul>
/b ※ <sup>1</sup>	比較項目が数値項目の場合、big endian ※ <sup>2</sup> の数値として扱う。
/u	選択条件を満たさないレコードだけを入力する。

## 注※ 1

比較項目が 2 進項目「/n」「/m」、または内部浮動小数点項目「/f」の場合だけ有効となる。

## 注※ 2

big endian とは、上位バイトを下位番地とする数値のことです。

/b パラメタを指定しなかった場合は、little endian（上位バイトを上位番地とする数値）として扱います。

複数のオプションを指定する場合は、次に示すどちらかの方法で指定します。

- 「/b △ /i」のように分けて指定します。
  - 「/bi」のようにつなげて指定します。
- （凡例） △：スペース（空白）を示します。

また、この指定は省略できます。省略した場合は、次のようになります。

- 「/i」の指定を省略した場合、出力ファイル名と同じ名前のファイルがすでにあると、そのファイルの内容が書き換えられます。
- 「/v」の指定を省略した場合、JIS8 単位コードの体系に従って項目が比較されます。

なお、選択オプションを指定する場合は、次の点に注意してください。

- 比較項目の属性に「/j」を指定していない場合は、「/v」を指定しても無効になります。
- 「/v 比較レベル」の比較レベルには、指定値以外の値を指定できません。

## (2) /g 入力ファイルのファイル編成

入力ファイルのファイル編成を「/g +オペランド」の形式で指定します。複数の入力ファイルを指定する場合は、すべて同じファイル編成にしなければなりません。

また、この指定は省略できません。入力ファイルのファイル編成のオペランドを次の表に示します。

表 9-2 入力ファイルのファイル編成のオペランド（選択機能）

オペランド	意味
s	固定長順編成ファイルを指定する。
v	可変長順編成ファイルを指定する。

## 9. 選択コマンドの使い方

オペランド	意味
r	固定長相対ファイルを指定する。
w	可変長相対ファイルを指定する。
i	固定長索引ファイルを指定する。
x	可変長索引ファイルを指定する。
a	テキストファイルを指定する。

### (3) 出力ファイルのファイル編成

出力ファイルのファイル編成のオペランドを、コロンで区切って指定します。出力ファイルのレコードの形式（固定長、可変長またはテキスト）は、入力ファイルと同じにしなればなりません。

入力ファイルが索引ファイル以外の場合、この指定は省略できます。指定を省略した場合は、入力ファイルと同じファイル編成が設定されます。入力ファイルが索引ファイルの場合は、順ファイルまたは相対ファイルのどちらかを必ず指定してください。

出力ファイルのファイル編成のオペランドを次の表に示します。

表 9-3 出力ファイルのファイル編成のオペランド（選択機能）

オペランド	意味
s	固定長順編成ファイルを指定する。
v	可変長順編成ファイルを指定する。
r	固定長相対ファイルを指定する。
w	可変長相対ファイルを指定する。
a	テキストファイルを指定する。

注

索引ファイルに出力する場合は、ISAM のユティリティを使用する。

### (4) //レコード長

入力ファイルのファイル編成ごとのレコード長の指定方法を次の表に示します。

表 9-4 入力ファイルのファイル編成ごとのレコード長の指定方法（選択機能）

入力ファイル編成	レコード長の指定方法
固定長順編成ファイル	ファイルのレコード長を「1～65,535」バイトの範囲で指定します。
可変長順編成ファイル	指定は不要です。指定しても無視されます。
固定長相対ファイル	ただし、入力できるレコード長の範囲は、1～65,535 バイトです。
可変長相対ファイル	

入力ファイル編成	レコード長の指定方法
固定長索引ファイル	指定は不要です。指定しても無視されます。 ただし、入力できるレコード長の範囲は、1～65,503バイトです。
可変長索引ファイル	
テキストファイル	ファイルの最大レコード長を「1～65,535」バイトの範囲で指定します。 レコード長には、改行文字「X'0A'」は含みません。 省略できます。省略した場合「65,535」バイトが設定されます。

### (5) /o 出力ファイル名

選択結果の出力先ファイルをファイル名で指定します。

この指定は、省略できます。指定を省略した場合は、選択結果を標準出力ファイルに出力します。

### (6) /sl" 選択条件式 "

レコードを選択するための選択条件式を指定します。この指定は省略できません。選択条件式の形式は次のとおりです。

「比較項目の属性△+項目位置-項目長△比較演算子△条件値」

(凡例) △:スペース(空白)を示す。

選択条件式は、間を論理演算子で区切って、128個まで指定できます。

指定形式を次の図に示します。

図 9-2 選択条件式の指定形式

/sl"比較項目の属性△+項目位置-項目長△比較演算子△条件値  
[ [論理演算子△比較項目の属性△+項目位置-項目長△比較演算子△条件値] … ] "

(凡例)

△ : スペース(空白)を示す。

[ ] : 省略できることを示す。

… : 直前の [ ] で囲まれた部分を複数繰り返して指定できることを示す。

#### (a) 比較項目の属性

比較項目の属性を指定します。比較項目の属性のオペランドを次の表に示します。

表 9-5 比較項目の属性のオペランド

オペランド	意味
j	比較項目が文字 (JIS8 単位コード※) であることを指定する。
c	比較項目が文字であることを指定する。
n	比較項目が符号付き固定小数点 2 進数であることを指定する。

## 9. 選択コマンドの使い方

オペランド	意味
m	比較項目が符号なし 2 進数であることを指定する。
p	比較項目が内部 10 進数であることを指定する。
z	比較項目が外部 10 進数であることを指定する。
x	比較項目が左符号付き外部 10 進数であることを指定する。
s	比較項目が右分離符号付き外部 10 進数であることを指定する。
t	比較項目が左分離符号付き外部 10 進数であることを指定する。
f	比較項目が内部浮動小数点であることを指定する。
e	比較項目が日付（英数字）であることを指定する。

### 注※

ASCII コード (0x00 ~ 0x7F) を含む、JIS8 単位コード (0x00 ~ 0xFF) を意味します。入力データに日本語などのマルチバイト文字が含まれている場合、マルチバイト文字として扱わず、シングルバイト文字として扱います。EBCDIK コードの体系に従ってキー・項目を比較する「v オプション (Windows は /v オプション)」を使用できます。

### (b) + 項目位置 - 項目長

比較項目の開始バイト位置を 5 桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置（先頭を 0 とする）とします。開始バイト位置に指定できる値の範囲は、「0 ~ レコード長 - 1」です。

また、比較項目のバイト長を「1 ~ 4,096」の範囲で指定します。

### 比較項目の指定範囲

比較項目は、次に示す範囲で指定してください。

- 入力ファイルが固定長ファイルのとき、比較項目は、レコード長の範囲
- 入力ファイルが可変長ファイルおよびテキストファイルのとき、比較項目は、最小レコード長の範囲

### (c) 比較演算子

比較項目と条件値の値を比較、選択する方法を指定します。

比較演算子の意味を次の表に示します。

表 9-6 比較演算子の意味（選択コマンド）

比較演算子	意味
=	条件値と等しい値の入力レコードを選択することを指定する。
!=	条件値と異なる値の入力レコードを選択することを指定する。
>	条件値より大きい値の入力レコードを選択することを指定する。
>=	条件値より大きいか、または等しい値の入力レコードを選択することを指定する。

比較演算子	意味
<	条件値より小さい値の入力レコードを選択することを指定する。
<=	条件値より小さいか、または等しい値の入力レコードを選択することを指定する。

## (d) 条件値

比較の条件となる値を「c' 文字条件値」、 「n' 数値条件値」、 「x'16 進数条件値」または「d' 日付条件値」の形式で指定します。

「c' 文字条件値」は比較項目が文字項目の場合、「n' 数値条件値」は比較項目が数値項目の場合に指定できます。また、「x'16 進数条件値」は比較項目が文字の場合でも数値の場合でも指定できます。「d' 日付条件値」は比較項目が日付項目の場合に指定できません。

指定できる条件値の長さを次の表に示します。

表 9-7 指定できる条件値の長さ（選択コマンド）

項目	最小値	最大値	備考
文字条件値	1	4,096	—
数値条件値※1	1	19	符号なしの場合は 18
16 進数条件値※2	2	8,192	指定できるのは偶数桁のデータだけ
日付条件値※3	2	8	—

単位：バイト

(凡例)

—：なし

注※1

数値条件値として指定できるのは「0～9」、「+」および「-」だけである。

注※2

16 進数条件値として指定できるのは「0～9」および「A～F」だけである。

注※3

日付条件値として指定できる値は次のとおり。

- ・先頭の 1 文字はスペースまたは「0～9」の文字列
- ・2 文字目は「0～9」の文字列
- ・1 文字のスペースを指定した場合「00」を設定
- ・1 文字の「0～9」を指定した場合、先頭の 1 文字は「0」を設定

## 条件値の扱い

- ・条件値の長さが、指定した比較項目より短い場合  
文字条件値のとき  
下位にスペースが設定されます。

## 9. 選択コマンドの使い方

c +0-4 : c'ab' → ab △△

数値条件値のとき

上位に 0 が設定されます。

z +0-4 : n'12' → 0012

16 進数条件値のとき

下位に (00)<sub>16</sub> が設定されます。

c +0-4 : x'3031' → 30310000

日付条件値のとき

e +0-4 : D'12' → 12 △△

- 条件値の長さが、指定した比較項目より長い場合  
文字条件値、16 進数条件値または日付条件値のときは、下位のオーバーフロー部分が対象外になります。数値条件値のときは、上位のオーバーフロー部分が対象外になります。

文字条件値のとき

c +0-4 : c'abcdef' → abcd ef

数値条件値のとき

z +0-4 : n'123456' → 12 3456

16 進数条件値のとき

c +0-2 : x'303132' → 3031 32

日付条件値のとき

e +0-3 : D'12/03' → 12/ 03

(凡例)

—— : 比較対象となる部分を示す。

△ : スペース (空白) を示す。

### (e) 論理演算子

複数の選択条件式を指定する場合、論理演算子「AND」(論理積)または「OR」(論理和)を指定します。

選択条件中に「AND」と「OR」が混在した場合は、「AND」を判定後、「OR」を判定します。「AND」と「OR」が混在した場合の処理の例を次に示します。

(例)

n△+10-3>n'30'△AND△n△+23-3=n'10'△  
OR△n△+10-3<=n'10'△AND△n△+23-3=n'10'

(凡例) △ : スペース (空白) を示す。

この場合、次のうちどちらかの条件を満たしたレコードが選択されます。

- 「n △ +10-3」は 30 より大きく、「n △ +23-3」は 10
- 「n △ +10-3」は 10 か 10 より小さく、「n △ +23-3」は 10

## (7) 入力ファイル名

選択の対象となる入力ファイルを指定します。2 個以上指定するときは、すべて同じファイル編成にしなければなりません。

また、出力ファイルの指定と同じファイルの指定はできません。

入力ファイルが順ファイルまたは相対ファイルの場合、この指定は、省略できます。省略した場合は、入力ファイルに標準入力ファイルが設定されます。入力ファイルが索引ファイルの場合は、この指定は省略できません。

## 9.1.3 コマンド指定時の注意事項

選択コマンドの指定時の注意事項を次に示します。

### (1) オペランドの入力順序

各パラメタのオペランドの入力順序は任意です。ただし、次に示すパラメタのオペランドは、次の順序で入力してください。

- 入力ファイルのファイル編成と出力ファイルのファイル編成  
「入力ファイルのファイル編成」「出力ファイルのファイル編成」
- キー属性と + キー位置・キー長のパラメタ  
「キー属性」「+ キー位置・キー長」
- 選択条件式  
「比較項目の属性」「+ キー位置・キー長」「比較演算子」「条件値」

### (2) 同一パラメタのオペランドの複数指定

複数指定の不要なオペランドを複数指定した場合は、最後に指定したオペランドが有効になります。

### (3) 複数指定できるパラメタ

複数指定できるパラメタを複数指定するときは、連続で指定してください。指定例を次に示します。

## 9. 選択コマンドの使い方

集約項目を複数指定する場合

・  $\Delta/a^n\Delta+0-5\Delta m\Delta+11-5''$   
↑  
指定が連続しているので有効となる

・  $\Delta/a^n\Delta+0-5''\Delta/a^m\Delta+11-5''$   
↑                      ↑  
この指定は無視される    この指定が有効となる

(凡例) Δ : スペース (空白) を示す。

### (4) スペース (空白) の扱い

パラメタで使用する文字「/l, /o, /sl...」の/ (スラント) の前にスペース (空白) を入れるかどうかは任意です。しかし、+ キー位置・キー長のハイフンの前には、スペースを入れてはなりません。指定例を次に示します。

(例)

- ・ 「 $\Delta /l128 \Delta /oOUTFILE$ 」 ……有効となる
- ・ 「 $+10 \Delta -8$ 」 ……………エラーとなる

(凡例) Δ : スペース (空白) を示す。

また、パラメタで使用する文字「/l, /o, /sl...」とそのあとに続けるオペランドの間にスペース (空白) を入れるかどうかは任意です。指定例を次に示します。

(例)

- ・ 「/l128」 ……有効となる
- ・ 「/l  $\Delta$  128」 ……有効となる

(凡例) Δ : スペース (空白) を示す。

### (5) 英文字の扱い

各パラメタを大文字で入力するか、小文字で入力するかは任意です。また、大文字と小文字が混在してもかまいません。

### (6) 特殊記号の扱い

次の三つの特殊記号を文字として使用するときは、直前に「 $\text{¥}$ 」を指定してください。

- ・ 「 $\text{¥}$ 」
- ・ 「'」 (クォーテーションマーク)
- ・ 「"」 (ダブルクォーテーションマーク)

指定例を次に示します。



(例 1) 「A'」を指定する場合

- 「c'A¥'''」  
有効となる
- 「c'A\_」  
エラーとなる

(例 2) 「"A」を指定する場合

- 「c'¥"A'」  
有効となる
- 「c'"A'」  
エラーとなる

(例 3) 「A¥'」を指定する場合

- 「c'A¥¥¥¥'''」  
有効となる
- 「c'A¥¥'''」  
エラーとなる

注

円記号 (¥) を一つ指定し、このあとに、上記特殊記号以外を指定した場合も、円記号 (¥) は、エスケープ文字として扱われます。なお、円記号 (¥) を一つだけ指定した場合は、KBLS072-E のエラーとなります。

## 9.1.4 使用例

選択コマンドの使用例を示します。

### (1) 1 種類の条件式でデータを選択する例

1 種類の条件式でデータを選択する例を示します。ここでは、社員データファイルから、年齢が 40 歳以上のデータを持つレコードを選択する例を示します。

#### (a) 条件

選択の条件は、次のとおりです。

- 32 ビット版 SORT を使用する
- 入力ファイルおよび出力ファイルは、可変長順編成ファイル
- 選択結果の出力ファイル名は「OUTFILE」
- 比較項目の属性は数字項目 (外部 10 進数)
- 比較項目の開始バイト位置は「25」バイト目、バイト長は「2」バイト
- 比較演算子は「>=」
- 条件値は「40」
- 入力ファイル名は「INFILE」

## 9. 選択コマンドの使い方

### (b) 入力するコマンド

入力する選択コマンドを次の図に示します。

図 9-3 入力する選択コマンド（その 1）

```
wnselect△/gv△/oOUTFILE△/sl"z△+25-2△>=△n'40'"△INFILE
```

（凡例）△：スペース（空白）を示す。

### (c) コマンドの指定内容

選択コマンドの指定内容を次の表に示します。

表 9-8 選択コマンドの指定内容（その 1）

入力項目	指定内容
wnselect	選択処理のコマンド「wnselect」を指定する。
/gv	入力ファイルを「可変長順編成ファイル」で指定する。
/oOUTFILE	選択結果の出力ファイル名「OUTFILE」を指定する。
/sl	比較条件式のオペランド「sl」を指定する。
z	比較項目の属性を数字項目の「外部 10 進数」で指定する。
+25-2	比較項目の開始バイト位置を「25」バイト目、バイト長を「2」バイトで指定する。
>=	比較演算子を「条件値と等しいか、または大きい」で指定する。
n'40'	条件値を数値条件値の「40」で指定する。
INFILE	入力ファイル名「INFILE」を指定する。

注

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略する。

## (2) 2 種類の条件式でデータを選択する例

2 種類の条件式でデータを選択する例を、次に示します。この例では、社員データファイルから、年齢が 40 歳から 50 歳までのデータを持つレコードを選択します。

### (a) 条件

選択の条件は次のとおりです。

- 32 ビット版 SORT を使用する
- 入力ファイルおよび出力ファイルは可変長順編成ファイル
- 選択結果の出力ファイル名は「OUTFILE」
- 第 1 比較項目の属性は数字項目（外部 10 進数）
- 第 1 比較項目の開始バイト位置は「25」バイト目、バイト長は「2」バイト

- 第1比較演算子は「>=」
- 第1比較項目の条件値は「40」
- 第2比較項目の属性は数字項目（外部10進数）
- 第2比較項目の開始バイト位置は「25」バイト目、バイト長は「2」バイト
- 第2比較項目の比較演算子は「<=」
- 第2比較項目の条件値は「50」
- 入力ファイル名は「INFILE」

## (b) 入力するコマンド

入力する選択コマンドを次の図に示します。

図 9-4 入力する選択コマンド（その2）

```
wselect△/gv△/oOUTFILE
△/s!z△+25-2△>=△n'40'△and△z△+25-2△<=△n'50'△
△INFILE
```

（凡例）△：スペース（空白）を示す。

## (c) コマンドの指定内容

選択コマンドの指定内容を次の表に示します。

表 9-9 選択コマンドの指定内容（その2）

入力項目	指定内容
wselect	選択処理のコマンド「wselect」を指定する。
/gv	入力ファイルを「可変長順編成ファイル」で指定する。
/oOUTFILE	選択結果の出力ファイル名「OUTFILE」を指定する。
/s!	比較条件式オペランド「s!」を指定する。
z	第1比較項目の属性を数字項目の「外部10進数」で指定する。
+25-2	第1比較項目の開始バイト位置を「25」バイト目、バイト長を「2」バイトで指定する。
>=	第1比較演算子を「条件値と等しいか、または大きい」で指定する。
n'40'	第1比較項目の条件値を数値条件値の「40」で指定する。
and	論理演算子を「and」で指定する。
z	第2比較項目の属性を数字項目の「外部10進数」で指定する。
+25-2	第2比較項目の開始バイト位置を「25」バイト目、バイト長を「2」バイトで指定する。
<=	第2比較演算子を「条件値と等しいか、または小さい」で指定する。
n'50'	第2比較項目の条件値を数値条件値の「50」で指定する。
INFILE	入力ファイル名「INFILE」を指定する。

## 9. 選択コマンドの使い方

注

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略する。

## 9.2 選択コマンドの使い方（UNIX の場合）

選択コマンドの指定時の使い方を次に示します。

### 9.2.1 コマンドの使い方

選択コマンドを使うには、コマンドを入力します。

選択機能を使う場合のコマンド名を次に示します。

- `rselect`（32 ビット版 SORT でソート機能を使う場合）
- `rselect64`（64 ビット版 SORT でソート機能を使う場合）

選択コマンドの入力方法について説明します。

#### (1) コマンドの形式

選択コマンドは、コマンド名（`rselect` または `rselect64`）とそのあとに続く各オペランドで構成されます。選択コマンドの形式を次の図に示します。

図 9-5 選択コマンドの形式

```
rselect※1△ [選択オプション]
      △ 入力ファイルのファイル編成 [:出力ファイルのファイル編成]※2
      △ [-Lレコード長]
      △ [-o出力ファイル名]
      △ -s|“選択条件式”
      △ { [入力ファイル名] } . . .
```

(凡例)     △ : スペース（空白）を示す。  
           [ ] : 省略できることを示す。  
           . . . : 直前の { } で囲まれた部分を複数繰り返して指定できることを示す。

注※1  使用しているSORTが64ビット版対応の場合は、「`rselect64`」となる。

注※2  入力ファイルが索引ファイルの場合、出力ファイルのファイル編成の指定は省略できない。

### 9.2.2 オペランドの意味

オペランドには、必ず指定するオペランドと任意に指定できるオペランドの2種類があります。

#### (1) 選択オプション

選択処理で使用する機能上のオプションを指定します。選択オプションのパラメタを次の表に示します。

## 9. 選択コマンドの使い方

表 9-10 選択オプションのパラメタ

パラメタ	意味
-e	入力ファイルを排他モードでオープンする（ファイル排他）。
-i	出力ファイル名と同じ名前のファイルがすでにある場合、そのファイルの内容が書き換えられるのを防ぐ（既存ファイルの書き換え防止）。
-v 比較レベル	EBCDIK コードの体系に従ってキーを比較する。 <ul style="list-style-type: none"> <li>• 選択条件式の比較項目が JIS8 単位コードの英数字の場合 比較レベルに 1 を指定するか、または比較レベルを指定しない。</li> <li>• 選択条件式の比較項目が EBCDIK コードの英数字の場合 比較レベルに 2 を指定する。</li> </ul>
-u	選択条件を満たさないレコードだけを入力する。
-q	出力データがキャッシュファイルに残っている場合、すべてのデータをファイル出力したあと、コマンドを終了する（キャッシュファイルの同期出力）。
-B※1	比較項目が数値項目の場合、big endian※2 の数値として扱う。

### 注

「-q」を指定した場合、コマンドの実行時間が長くなることあるため注意してください。  
 また、入力ファイルが索引ファイルの場合は、「-e」を指定した方が処理時間が短くなります。

### 注※1

Linux の場合だけ有効なオプションです。また、比較項目が 2 進項目「-n」または「-m」の場合だけ有効です。

### 注※2

big endian とは、上位バイトを下位番地とする数値のことです。

Linux で「-B」を指定しなかった場合は、little endian（上位バイトを上位番地とする数値）として扱います。

複数のオプションを指定する場合は、次に示すどちらかの方法で指定します。

- 「-e Δ -i」のように分けて指定する（「Δ」は、スペース（空白）を示します）。
- 「-ei」のようにつなげて指定する。

この指定は、省略できます。省略した場合は、次のようになります。

- 「-e」の指定を省略した場合、入力ファイルが共用モードでオープンされます。
- 「-i」の指定を省略した場合、出力ファイル名と同じ名前のファイルがすでにあると、そのファイルの内容が書き換えられます。
- 「-v」の指定を省略した場合、JIS8 単位コードの体系に従って項目が比較されます。

なお、選択オプションを指定する場合は、次の点に注意してください。

- 入力ファイルの書き込みが許可されていない場合、および入力ファイルがすでにほかの処理でオープンされている場合は、「-e」は指定できません。
- 比較項目の属性に「-j」を指定していない場合は、「-v」を指定しても無効になります。
- 「-v 比較レベル」の比較レベルには、指定値以外の値を指定できません。指定値以外

の値を指定すると、メッセージ ID 「KBLS062-E」 のメッセージが出力され、エラーになります。

(a) **-v** パラメタについて

**-v** パラメタに指定した値による比較処理については、次のとおりです。

- 「-v1」 および 「-v2」 指定なし  
入力レコードと選択条件式の比較項目の値は、JIS8 単位コードとして扱い、JIS8 単位コードの体系に従って比較します。
- 「-v1」 指定あり  
入力レコードと選択条件式の比較項目の値を、JIS8 単位コードから EBCDIK コードの並びに変換して、EBCDIK コードの体系に従って比較します。
- 「-v2」 指定あり  
入力レコードと選択条件式の比較項目の値は、EBCDIK コードとして扱い、EBCDIK コードの体系に従って比較します。

JIS8 単位コード体系と EBCDIK コード体系の変換表は、「付録 A JIS8 単位コード体系と EBCDIK コード体系の変換表」を参照してください。

(b) 「-v1」と「-v2」の違い

「-v1」の場合は、入力レコードの比較項目と選択条件式の文字条件値どちらも JIS8 単位コードと見なし、EBCDIK コード体系に従って比較します。「-v2」の場合は、入力レコードの比較項目と選択条件式の文字条件値どちらも EBCDIK コードと見なし、EBCDIK コード体系に従って比較します。

**実行例**

次の条件での実行例を示します。

<条件>

- ・レコード長1バイトの固定長順編成ファイル
- ・入力レコード(8バイト)

0x80	0x82	0x83	0x84	0xb1	0xb2	0xb3	0xb4
------	------	------	------	------	------	------	------

- ・選択条件式

-j +0-1 = C'ウ'

-j +0-1 > C'ウ'

文字条件値の半角カタカナ"ウ"を、JIS8単位コードとEBCDIKコードで表した場合の値は、次のとおりです。

JIS8単位コード	EBCDIKコード
0xb3	0x83

- 「-v1」, および 「-v2」 指定なし

## 9. 選択コマンドの使い方

入力レコード：JIS8単位コードとして扱います。

0x80	0x82	0x83	0x84	0xb1	0xb2	0xb3	0xb4
------	------	------	------	------	------	------	------

文字条件値の値：JIS8単位コードとして扱います。

0xb3

上記の値で、入力レコードと文字条件値をバイナリ比較します。

・“-j +0-1 = C'ウ”の結果

入力レコード

値	0x80	0x82	0x83	0x84	0xb1	0xb2	0xb3	0xb4
比較結果	×	×	×	×	×	×	○	×

文字条件値

= 0xb3

出力レコード

0xb3

・“-j +0-1 > C'ウ”の結果

入力レコード

値	0x80	0x82	0x83	0x84	0xb1	0xb2	0xb3	0xb4
比較結果	×	×	×	×	×	×	×	○

文字条件値

> 0xb3

出力レコード

0xb4

- ・「-v1」指定あり



入力レコード：JIS8単位コードからEBCDIKコードの並びに内部変換します  
 (内部変換により、出力レコードの内容が変わることはありません)。

0x80 0x82 0x83 0x84 0xb1 0xb2 0xb3 0xb4



0x00 0xff 0xff 0xff 0x81 0x82 0x83 0x84

文字条件値の値：JIS8単位コードからEBCDIKコードの並びに変換します。

0xb3



0x83

上記の変換後の値で、入力レコードと文字条件値をバイナリ比較します。

・“-j +0-1 = C'ウ'”の結果

内部変換後の入力レコード

文字条件値

値	0x00	0xff	0xff	0xff	0x81	0x82	0x83	0x84	=	0x83
比較結果	×	×	×	×	×	×	○	×		

出力レコード

0xb3

・“-j +0-1 > C'ウ'”の結果

内部変換後の入力レコード

文字条件値

値	0x00	0xff	0xff	0xff	0x81	0x82	0x83	0x84	>	0x83
比較結果	×	○	○	○	×	×	×	○		

出力レコード

0x82 0x83 0x84 0xb4

- 「-v2」指定あり

## 9. 選択コマンドの使い方

入力レコード：内部変換しないで、EBCDICコードとして扱います。

0x80	0x82	0x83	0x84	0xb1	0xb2	0xb3	0xb4
------	------	------	------	------	------	------	------

文字条件値の値：'ウ'はEBCDICコードとして扱います。

0x83
------

上記の値で、入力レコードと文字条件値をバイナリ比較します。

・"-j +0-1 = C'ウ'"の結果

入力レコード

文字条件値

値	0x80	0x82	0x83	0x84	0xb1	0xb2	0xb3	0xb4	=	0x83
比較結果	×	×	○	×	×	×	×	×		

出力レコード

0x83
------

・"-j +0-1 > C'ウ'"の結果

入力レコード

文字条件値

値	0x80	0x82	0x83	0x84	0xb1	0xb2	0xb3	0xb4	>	0x83
比較結果	×	×	×	○	○	○	○	○		

出力レコード

0x84	0xb1	0xb2	0xb3	0xb4
------	------	------	------	------

コマンドの実行結果を次に示します。

```

$od -tx in/inp1.txt
0000000 80828384 b1b2b3b4
0000010
$/opt/HISORTlib/bin/rselect -S -L1 -o out/out1.txt -sl"-j +0-1 = C'ウ'" in/inp1.txt
KBLS061-l number of data items selected: 1
$od -tx out/out1.txt
0000000 b3000000
0000001
$/opt/HISORTlib/bin/rselect -S -L1 -o out/out1.txt -sl"-j +0-1 > C'ウ'" in/inp1.txt
KBLS061-l number of data items selected: 1
$od -tx out/out1.txt
0000000 b4000000
0000001
$/opt/HISORTlib/bin/rselect -S -L1 -o out/out1.txt -sl"-j +0-1 = C'ウ'" in/inp1.txt -v1
KBLS061-l number of data items selected: 1
$od -tx out/out1.txt
0000000 b3000000
0000001
$/opt/HISORTlib/bin/rselect -S -L1 -o out/out1.txt -sl"-j +0-1 > C'ウ'" in/inp1.txt -v1
KBLS061-l number of data items selected: 4
$od -tx out/out1.txt
0000000 828384b4
0000004
$/opt/HISORTlib/bin/rselect -S -L1 -o out/out1.txt -sl"-j +0-1 = C'ウ'" in/inp1.txt -v2
KBLS061-l number of data items selected: 1
$od -tx out/out1.txt
0000000 83000000
0000001
$/opt/HISORTlib/bin/rselect -S -L1 -o out/out1.txt -sl"-j +0-1 > C'ウ'" in/inp1.txt -v2
KBLS061-l number of data items selected: 5
$od -tx out/out1.txt
0000000 84b1b2b3 b4000000
0000005

```

## (2) 入力ファイルのファイル編成

入力ファイルのファイル編成を指定します。

複数の入力ファイルを指定する場合は、指定するファイルをすべて同じファイル編成にしてください。

この指定は、省略できません。必ず指定してください。入力ファイルのファイル編成のパラメタを次の表に示します。

表 9-11 入力ファイルのファイル編成のパラメタ（選択機能）

パラメタ	意味
-S	固定長順ファイルを指定する。
-V	可変長順ファイルを指定する。
-R	固定長相対ファイルを指定する。
-W	可変長相対ファイルを指定する。
-I	固定長索引ファイルを指定する。

## 9. 選択コマンドの使い方

パラメタ	意味
-X	可変長索引ファイルを指定する。
-A	テキストファイルを指定する。

### (3) 出力ファイルのファイル編成

出力ファイルのファイル編成を指定します。出力ファイルのレコードの形式（固定長、可変長またはテキスト）は、入力ファイルと同じにしなければなりません。

入力ファイルが順ファイル、相対ファイルまたはテキストファイルの場合、この指定は省略できます。指定を省略した場合は、入力ファイルと同じファイル編成が設定されます。入力ファイルが索引ファイルの場合は、順ファイルまたは相対ファイルのどちらかを必ず指定してください。

出力ファイルのファイル編成のパラメタを次の表に示します。

表 9-12 出力ファイルのファイル編成のパラメタ（選択機能）

パラメタ	意味
:S	固定長順ファイルを指定する。
:V	可変長順ファイルを指定する。
:R	固定長相対ファイルを指定する。
:W	可変長相対ファイルを指定する。
:A	テキストファイルを指定する。

#### 注

索引ファイルに出力する場合は、いったん順ファイルにデータを出力し、索引ファイルに変換します。索引ファイルに変換するには、ISAMのユーティリティコマンドをパイプで連結します。索引ファイルに出力する場合のコマンドの指定例を次に示します。

(例)

```
rselect -s -L128 -sl"-z +0-10 != C' ffffffff' " infile | isconv -tSI -outfile
```

↑
↑
↑  
選択コマンド
パイプ
ISAMのユーティリティ  
コマンド

### (4) -L レコード長

入力ファイル編成ごとのレコード長の指定方法を次の表に示します。

表 9-13 入力ファイル編成ごとのレコード長の指定方法（選択機能）

入力ファイル編成	レコード長の指定方法
固定長順ファイル	ファイルのレコード長を「1～65,535」バイトの範囲で指定します。

入力ファイル編成	レコード長の指定方法
可変長順ファイル	指定は不要です。指定しても無視されます。 ただし、入力できるレコード長の範囲は、1 ～ 65,535 バイトです。
固定長相対ファイル	
可変長相対ファイル	
固定長索引ファイル	指定は不要です。指定しても無視されます。 ただし、入力できるレコード長の範囲は、1 ～ 65,503 バイトです。
可変長索引ファイル	
テキストファイル	ファイルの最大レコード長を「1 ～ 65,535」バイトの範囲で指定します。 レコード長には、改行文字「X'0A'」は含みません。 この指定は省略できます。指定を省略した場合は、「65535」バイトが設定されます。 なお、レコード長を指定した方が、ソート時間が短くなります。

### (5) -o 出力ファイル名

選択結果の出力先ファイルをファイル名で指定します。

この指定は、省略できます。指定を省略した場合は、選択結果を標準出力ファイルに出力します。

### (6) -sl" 選択条件式 "

レコードを選択するための選択条件式を指定します。指定できる条件式の個数は1から32個です。

選択条件式の形式を次の図に示します。

図 9-6 選択条件式の形式

```
-sl"比較項目の属性△+項目位置-項目長△比較演算子△条件値
  [ [論理演算子△比較項目の属性△+項目位置-項目長△比較演算子△条件値] …] "
```

(凡例)     △ : スペース (空白) を示す。  
          [ ] : 省略できることを示す。  
          ... : 直前の {} で囲まれた部分を複数繰り返して指定できることを示す。

#### (a) 比較項目の属性

比較項目に関する属性を指定します。比較項目の属性のパラメタを次の表に示します。

表 9-14 比較項目の属性のパラメタ

パラメタ	意味
-j	比較項目が文字 (JIS8 単位コード※) であることを指定する。
-c	比較項目が文字であることを指定する。
-n	比較項目が符号付き固定小数点 2 進数であることを指定する。

## 9. 選択コマンドの使い方

パラメタ	意味
-m	比較項目が符号なし 2 進数であることを指定する。
-p	比較項目が内部 10 進数であることを指定する。
-z	比較項目が外部 10 進数であることを指定する。
-x	比較項目が左符号付き外部 10 進数であることを指定する。
-s	比較項目が右分離符号付き外部 10 進数であることを指定する。
-t	比較項目が左分離符号付き外部 10 進数であることを指定する。
-g	比較項目が日付（英数字）であることを指定する。

### 注※

ASCII コード (0x00 ~ 0x7F) を含む、JIS8 単位コード (0x00 ~ 0xFF) を意味します。入力データに日本語などのマルチバイト文字が含まれている場合、マルチバイト文字として扱わず、シングルバイト文字として扱います。EBCDIK コードの体系に従ってキー・項目を比較する「-v オプション (Windows は /v オプション)」を使用できます。

### (b) + 項目位置 - 項目長

比較項目の開始バイト位置を 5 桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置（先頭を 0 とする）とします。開始バイト位置に指定できる値の範囲を次に示します。

- 順ファイル、相対ファイルおよびテキストファイルの場合「0 ~ 65,534」
- 索引ファイルの場合「0 ~ 65,502」

また、比較項目のバイト長を 4 桁以内の数字「1 ~ 4,096」で指定します。この指定は、省略できません。必ず指定してください。

### 比較項目の指定範囲

比較項目は、次に示す範囲で指定してください。

- 入力ファイルが固定長ファイルのとき、比較項目は、レコード長の範囲
- 入力ファイルが可変長ファイルのとき、比較項目は、最小レコード長の範囲
- 入力ファイルがテキストファイルのとき、比較項目は、最小レコード長の範囲

### (c) 比較演算子

比較項目で指定した値と入力レコードの値を比較、選択する方法を指定します。

この指定は、省略できません。必ず指定してください。

比較演算子のパラメタを次の表に示します。

表 9-15 比較演算子のパラメタ

比較演算子	意味
=	条件値と等しい値の入力レコードを選択することを指定する。
!=	条件値と異なる値の入力レコードを選択することを指定する。

比較演算子	意味
>	条件値より大きい値の入力レコードを選択することを指定する。
>=	条件値より大きいかまたは等しい値の入力レコードを選択することを指定する。
<	条件値より小さい値の入力レコードを選択することを指定する。
<=	条件値より小さいかまたは等しい値の入力レコードを選択することを指定する。

#### (d) 条件値

比較の条件となる値を「C' 文字条件値 '」、 「N' 数値条件値 '」または「D' 日付条件値 '」の形で指定します。

「C' 文字条件値 '」は比較項目が文字項目の場合、「N' 数値条件値 '」は比較項目が数値項目の場合に指定できます。また、「D' 日付条件値 '」は比較項目が日付項目の場合に指定できます。

指定できる条件値の長さを次の表に示します。

表 9-16 指定できる条件値の長さ

項目	最小値	最大値	備考
文字条件値	1	4,096	—
数値条件値	1	19	符号なし数値は 18
日付条件値※	2	8	—

単位：バイト

(凡例)

—：なし

注※

日付条件値として指定できる値を次に示します。

- ・先頭の 1 文字はスペースまたは「0～9」の文字列
- ・2 文字目が「0～9」の文字列
- ・1 文字のスペースを指定した場合「00」を設定
- ・1 文字の「0～9」を指定した場合、先頭の 1 文字は「0」を設定

#### 条件値の扱い

条件値の長さが、指定した比較項目より短い場合、文字条件値のときは、下位にスペースが設定されます。数値条件値のときは、上位に 0 が設定されます。例を次に示します。

(例)

- ・文字条件値のとき  
-c +0-4 : C'ab' → ab△△
- ・数値条件値のとき  
-z +0-4 : N'12' → 0012

## 9. 選択コマンドの使い方

- 日付条件値のとき

`-g +0-4 : D'12' → 12 △△`

(凡例) `_____` : 比較対象となる部分

`△` : スペース (空白) を示す。

また、条件値の長さが指定した比較項目より長い場合、文字条件値のときは、下位の桁あふれ部分が対象外になります。数値条件値のときは、上位の桁あふれ部分が対象外になります。例を次に示します。

(例)

- 文字条件値のとき

`-c +0-4 : C'abcdef' → abcd ef`

- 数値条件値のとき

`-z +0-4 : N'123456' → 12 3456`

- 日付条件値のとき

`-g +0 -4 : D'10/03' → 10/03`

(凡例) `_____` : 比較対象となる部分

`△` : スペース (空白) を示す。

### (e) 論理演算子

複数の選択条件式を指定する場合、論理演算子「and」または「or」を指定します。

選択条件中に「and」と「or」が混在した場合は、「and」を判定後、「or」を判定します。「and」と「or」が混在した場合の処理の例を次に示します。

(例)

比較項目 A>30 △ and △比較項目 B=10 △ or △比較項目 A<=10 △ and △比較項目 B=10

(凡例) `△` : スペース (空白) を示す。

この場合、次のうちどちらかの条件を満たしたレコードが選択されます。

- 比較項目 A は 30 より大きく、比較項目 B は 10
- 比較項目 A は 10 か 10 より小さく、比較項目 B は 10

### (7) 入力ファイル名

選択の対象となる入力ファイルを指定します。同じファイルを 2 個以上指定できません。2 個以上指定するときは、すべて同じファイル編成とし、それぞれのファイル名は互いに異ならなければなりません。

また、出力ファイルの指定と同じファイルの指定はできません。

入力ファイルが順ファイル、相対ファイルまたはテキストファイルの場合、この指定は、省略できます。省略した場合は、入力ファイルに標準入力ファイルが設定されます。入力ファイルが索引ファイルの場合は、この指定は省略できません。



### 9.2.3 コマンド指定時の注意事項

選択コマンドの指定時の注意事項を次に示します。

#### (1) パラメタの入力順序

各パラメタの入力順序は任意です。ただし、入力ファイルのファイル編成と出力ファイルのファイル編成のパラメタ、および選択条件式は、次に示す順序で入力しなければなりません。

- 「入力ファイルのファイル編成」「出力ファイルのファイル編成」
- 「比較項目の属性」「+ 項目位置 - 項目長」「比較演算子」「条件値」

#### (2) 同一パラメタの複数指定

同一パラメタを複数指定した場合は、最後に指定したパラメタが有効になります。

#### (3) 複数指定できるパラメタ

複数指定できるパラメタを複数指定するときは、連続して指定してください。指定例を次に示します。

(例)

```

・ INFILE1△INFILE2
      ↑
      INFILE1, INFILE2の指定が連続しているので
      有効となる

・ INFILE1 (選択条件式) INFILE2△INFILE3
      ↑                               ↑
      INFILE1の指定は無視される     INFILE2, INFILE3の指定が
      連続しているため有効となる
  
```

(凡例) △ : スペース (空白) を示す。

#### (4) スペース (空白) の扱い

オペランドで使用する文字「-L, -o, -sl」の・ (ハイフン) の前にはスペース (空白) を入れなければなりません。ただし、+項目位置-項目長のハイフンの前には、スペースは不要です。指定例を次に示します。

(例)

- 「△ -L128 △ -oOUTFILE」 ……有効となる
- 「△ -L128-oOUTFILE」 ……エラーとなる

(凡例) △ : スペース (空白) を示す。

また、オペランドで使用する文字「-L, -o, -sl」とそのあとに続けるパラメタの間にス

## 9. 選択コマンドの使い方

ペースを入れるかどうかは任意です。指定例を次に示します。

(例)

- 「-L128」 ……有効となる
- 「-L △ 128」 ……有効となる

(凡例) △ : スペース (空白) を示す。

### (5) 特殊記号の扱い

次の三つの特殊記号を文字として使用するときは、直前に「¥」を指定してください。

- 「¥」
- 「'」 (クォーテーションマーク)
- 「"」 (ダブルクォーテーションマーク)

選択コマンドを C シェル以外のシェルで実行する場合、「¥」が認識できないときがあります。この場合は「14.2 SORT の環境変数の設定 (UNIX の場合)」を参照して、対処してください。

指定例を次に示します。

(例 1) 「A'」を指定する場合

- 「C'A¥'」  
有効となる
- 「C'A'」  
エラーとなる

(例 2) 「"A」を指定する場合

- 「C'¥"A'」  
有効となる
- 「C'"A'」  
エラーとなる

(例 3) 「A¥'」を指定する場合

- 「C'A¥¥¥'」  
有効となる
- 「C'A¥¥'」  
エラーとなる

注

円記号 (¥) を一つ指定し、このあとに、上記特殊記号以外を指定した場合も、円記号 (¥) は、エスケープ文字として扱われます。なお、円記号 (¥) を一つだけ指定した場合は、KBL5072-E のエラーとなります。

## 9.2.4 使用例

選択コマンドの使用例を示します。

### (1) 1種類の条件式でデータを選択する例

1種類の条件式でデータを選択する例を示します。ここでは、社員データファイルから、年齢が40歳以上のデータを持つレコードを選択する例を示します。

#### (a) 条件

選択の条件は、次のとおりです。

- 32ビット版 SORT を使用する
- 入力ファイルおよび出力ファイルは、可変長順ファイル
- 選択結果の出力ファイル名は「OUTFILE」
- 比較項目の属性は数字項目（外部10進数）
- 比較項目の開始バイト位置は「25」バイト目、バイト長は「2」バイト
- 比較演算子は「>=」
- 条件値は「40」
- 入力ファイル名は「INFILE」

#### (b) 入力するコマンド

入力する選択コマンドを次の図に示します。

図 9-7 入力する選択コマンド（その1）

```
rselect△-V△-oOUTFILE△-s|"-z△+25-2△>=△N'40'△ INFILE
```

（凡例） △ : スペース（空白）を示す。

#### (c) コマンドの指定内容

選択コマンドの指定内容を次の表に示します。

表 9-17 選択コマンドの指定内容（その1）

入力項目	指定内容
rselect	選択処理のコマンド「rselect」を指定する。
-V	入力ファイルを「可変長順ファイル」で指定する。
-oOUTFILE	選択結果の出力ファイル名「OUTFILE」を指定する。
-s	比較条件式のオペランド「s 」を指定する。
-z	比較項目の属性を数字項目の「外部10進数」で指定する。
+25-2	比較項目の開始バイト位置を「25」バイト目、バイト長を「2」バイトで指定する。

## 9. 選択コマンドの使い方

入力項目	指定内容
>=	比較演算子を「条件値と等しいか、または大きい」で指定する。
N'40'	条件値を数値条件値の「40」で指定する。
INFILE	入力ファイル名「INFILE」を指定する。

注

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略します。

### (2) 2種類の条件式でデータを選択する例

2種類の条件式でデータを選択する例を示します。ここでは、社員データファイルから、年齢が40歳から50歳までのデータを持つレコードを選択する例を示します。

#### (a) 条件

選択の条件は次のとおりです。

- 32ビット版 SORT を使用する
- 入力ファイルおよび出力ファイルは可変長順ファイル
- 選択結果の出力ファイル名は「OUTFILE」
- 第1比較項目の属性は数字項目（外部10進数）
- 第1比較項目の開始バイト位置は「25」バイト目、バイト長は「2」バイト
- 第1比較演算子は「>=」
- 第1比較演算子は「40」
- 第2比較項目の属性は数字項目（外部10進数）
- 第2比較項目の開始バイト位置は「25」バイト目、バイト長は「2」バイト
- 第2比較項目の比較演算子は「<=」
- 第2比較項目の条件値は「50」
- 入力ファイル名は「INFILE」

#### (b) 入力するコマンド

入力する選択コマンドを次の図に示します。

図 9-8 入力する選択コマンド（その2）

```
rselect△-V△-oOUTFILE△-s|"-z△+25-2△>=△N' 40'  
△and△-z△+25-2△<=△N' 50'△ INFILE
```

（凡例） △ : スペース（空白）を示す。

#### (c) コマンドの指定内容

選択コマンドの指定内容を次の表に示します。

表 9-18 選択コマンドの指定内容 (その2)

項番	入力項目	指定内容
1	rselect	選択処理のコマンド「rselect」を指定する。
2	-V	入力ファイルを「可変長順ファイル」で指定する。
3	-oOUTFILE	選択結果の出力ファイル名「OUTFILE」を指定する。
4	-sl	比較条件式のオペランド「sl」を指定する。
5	-z	第1比較項目の属性を数字項目の「外部10進数」で指定する。
6	+25-2	第1比較項目の開始バイト位置を「25」バイト目、バイト長を「2」バイトで指定する。
7	>=	第1比較演算子を「条件値と等しいか、または大きい」で指定する。
8	N'40'	第1条件値を数値条件値の「40」で指定する。
9	and	論理演算子を「and」で指定する。
10	-z	第2比較項目の属性を数字項目の「外部10進数」で指定する。
11	+25-2	第2比較項目の開始バイト位置を「25」バイト目、バイト長を「2」バイトで指定する。
12	<=	第2比較演算子を「条件値と等しいか、または小さい」で指定する。
13	N'50'	第2条件値を数値条件値の「50」で指定する。
14	INFILE	入力ファイル名「INFILE」を指定する。

注

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略します。

### (3) ソート処理と組み合わせて選択する例

選択結果をソートするには、ソートコマンドをパイプで連結します。選択結果をソートする例を示します。ここでは、社員データファイルから、年齢が30歳以上のデータを持つレコードを選択したあと、社員番号でソートする例を示します。

#### (a) 条件

選択の条件は、次のとおりです。

- 32ビット版 SORT を使用する
- 入力ファイルおよび出力ファイルは、固定長順ファイル
- 入力ファイルおよび出力ファイルのレコード長は、240 バイト
- 選択結果の出力ファイルは標準出力ファイル
- 比較項目の属性は数字項目 (外部10進数)
- 比較項目の開始バイト位置は「25」バイト目、バイト長は「2」バイト
- 比較演算子は「>=」
- 条件値は「30」
- 入力ファイル名は「INFILE」

## 9. 選択コマンドの使い方

ソートの条件は次のとおりです。

- 32 ビット版 SORT を使用する
- キーの比較は JIS8 単位コードの体系に従う
- 入力ファイルおよび出力ファイルは固定長順ファイル※
- 入力ファイルおよび出力ファイルのレコード長は、240 バイト※
- ソート結果の出力ファイル名は「OUTFILE」
- ソートは昇順、キーの属性は文字項目（英数字）
- キーの開始バイト位置は「10」バイト目、バイト長は「8」バイト
- 入力ファイルは標準入力ファイル

注※

選択処理と同じ値を指定しなければなりません。

### (b) 入力するコマンド

入力する選択コマンドを次の図に示します。

図 9-9 入力する選択コマンド（その 3）

```
rselect△-S△-L240△-sl"-z△+25-2△>=△N'30'△ INFILE
| rsort△-S△-L240△-oOUTFILE△-j△+10-8
```

(凡例)     △ : スペース（空白）を示す。  
          | : パイプを示す。

### (c) コマンドの指定内容

選択コマンドの指定内容を次の表に示します。

表 9-19 選択コマンドの指定内容（その 3）

項番	入力項目	指定内容
1	rselect	選択処理のコマンド「rselect」を指定する。
2	-S	入力ファイルを「固定長順ファイル」で指定する。
3	-L240	入力ファイルおよび出力ファイルのレコード長を「240」バイトで指定する。
4	-sl	比較条件式のオペランド「sl」を指定する。
5	-z	比較項目の属性を数字項目の「外部 10 進数」で指定する。
6	+25-2	比較項目の開始バイト位置を「25」バイト目、バイト長を「2」バイトで指定する。
7	>=	比較演算子を「条件値と等しいか、または大きい」で指定する。
8	N'30'	条件値を数値条件値の「30」で指定する。
9	INFILE	入力ファイル名「INFILE」を指定する。
10	rsort	ソート処理のコマンド「rsort」を指定する。

項番	入力項目	指定内容
11	-S	入力ファイルを「固定長順ファイル」で指定する。
12	-L240	入力ファイルおよび出力ファイルのレコード長を「240」バイトで指定する。
13	-oOUTFILE	ソート結果の出力ファイル名「OUTFILE」を指定する。
14	-j	キーの整列順序を「昇順」、属性を「英数字」で指定する。
15	+10-8	キーの開始バイト位置を「10」バイト目、バイト長を「8」バイトで指定する。

## 注 1

キーの比較は JIS8 単位コード体系に従うので、指定を省略します。

## 注 2

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略します。





# 10 集約コマンドの使い方

この章では、コマンドを入力して集約機能を使う方法について説明します。

---

10.1 集約コマンドの使い方 (Windows の場合)

---

10.2 集約コマンドの使い方 (UNIX の場合)

---

## 10.1 集約コマンドの使い方（Windows の場合）

集約コマンドのコマンド名を次に示します。

- wnsun (32 ビット版 SORT で集約コマンドを使う場合)
- wnsun64 (64 ビット版 SORT で集約コマンドを使う場合)

集約コマンドの入力方法について説明します。

### 10.1.1 コマンドの使い方

集約コマンドは、コマンド名（wnsun または wnsun64）とそのあとに続く各オペランドで構成されます。集約コマンドの形式を次の図に示します。

図 10-1 集約コマンドの形式

```
wnsun*Δ [集約オプション]
Δ /g入力ファイルのファイル編成[:出力ファイルのファイル編成]
Δ [/lレコード長]
Δ [/o出力ファイル名]
Δ { キー属性Δ+キー位置-キー長 } . . .
Δ /a" { 集約項目の属性Δ+項目位置-項目長 } . . ."
Δ [入力ファイル名]
```

(凡例) Δ : スペース (空白) を示す。

[ ] : 省略できることを示す。

. . . : 直前の { } で囲まれた部分を複数繰り返して指定できることを示す。

注※ 使用している SORT が 64 ビット版対応の場合は、「wnsun64」となる。

### 10.1.2 オペランドの意味

オペランドには、必ず指定するオペランドと任意に指定できるオペランドの 2 種類があります。

#### (1) 集約オプション

集約処理で使用する機能上のオプションを指定します。集約オプションのパラメタを次の表に示します。

表 10-1 集約オプションのパラメタ

パラメタ	意味
/i	出力ファイルと同じ名前のファイルがすでにある場合、そのファイルの内容が書き換えられるのを防ぐ（既存ファイルの書き換え防止）。
/v	EBCDIK コードの体系に従ってキーを比較する。

パラメータ	意味
/b <sup>※1</sup>	集約項目が数値項目の場合、big endian <sup>※2</sup> の数値として扱う。
/h	集約項目が2進数の場合、10進数の桁数でオーバーフロー判定をする。
/if	集約項目に内部10進数を指定した場合、演算符号が(F) <sub>16</sub> (絶対値)である項目の集約結果を同じ演算符号(絶対値)にする。 このパラメータを指定しない場合、集約した結果の演算符号は(C) <sub>16</sub> (正)になる。

## 注※1

集約キーが2進項目「/n」「/m」、または内部浮動小数点項目「/f」の場合だけ有効となる。

## 注※2

big endian とは、上位バイトを下位番地とする数値のことです。

/b パラメータを指定しなかった場合は、little endian (上位バイトを上位番地とする数値)として扱います。

複数のオプションを指定する場合は、次に示すどれかの方法で指定します。

- 「/b △ /i」のように分けて指定します。
- 「/bi」のようにつなげて指定します。
- 「/if」をほかのオプションとつなげて指定する場合は、「/ifiv」のように「/if」を必ず先頭に指定します。  
(凡例) △:スペース(空白)を示します。

また、この指定は省略できます。省略した場合は、次のようになります。

- 「/i」の指定を省略した場合、出力ファイル名と同じ名前のファイルがすでにあると、そのファイルの内容が書き換えられます。
- 「/v」の指定を省略した場合、JIS8 単位コードの体系に従ってキーが比較されます。
- 「/if」の指定を省略した場合、演算符号が(F)<sub>16</sub>(絶対値)の集約項目を集約した結果の演算符号は(C)<sub>16</sub>(正)となります。

なお、集約オプションを指定する場合は、次の点に注意してください。

- キー属性に「/j」を指定していない場合は、「/v」を指定しても無効になります。
- 「/if」を指定し、かつ演算符号に(C)<sub>16</sub>(正)と(F)<sub>16</sub>(絶対値)が混在している場合は、集約した結果の演算符号は保証しません。

「/h」の指定による、集約結果の範囲の違いを次の表に示します。

「/h」の指定は、集約項目が2進項目「/n」「/m」の場合だけ有効となります。集約結果がこの範囲を超えると、オーバーフローとなります。

## 10. 集約コマンドの使い方

表 10-2 集約結果の範囲の違い

集約項目のバイト長	集約結果の値の範囲	
	「/h」指定なし	「/h」指定あり
2	-32,768 ~ +32,767	-9,999 ~ +9,999
4	-2,147,483,648 ~ +2,147,483,647	-999,999,999 ~ +999,999,999
8	-9,223,372,036,854,775,808 ~ +9,223,372,036,854,775,807	-999,999,999,999,999,999 ~ +999,999,999,999,999,999


### 集約結果がオーバーフローした場合

集約結果がオーバーフローした場合は、それまで集約した結果を出力したあと、処理中のレコードから新規に集約します。

集約結果がオーバーフローした場合の例を次の図に示します。

図 10-2 集約結果がオーバーフローした場合の例

支店コード	担当者名	売上数量
03	田中	2
03	田中	7
04	渡辺	5
05	日立	8
06	山田	999
06	山田	10



支店コード	担当者名	売上数量
03	田中	9
04	渡辺	5
05	日立	8
06	山田	999
06	山田	10

(凡例)  : 集約結果がオーバーフローしたレコードを示す。

## (2) /g 入力ファイルのファイル編成

入力ファイルのファイル編成を「/g +オペランド」の形式で指定します。

また、この指定は、省略できません。入力ファイルのファイル編成のオペランドを次の表に示します。

表 10-3 入力ファイルのファイル編成のオペランド (集約機能)

オペランド	意味
s	固定長順編成ファイルを指定する。
v	可変長順編成ファイルを指定する。
r	固定長相対ファイルを指定する。
w	可変長相対ファイルを指定する。
a	テキストファイルを指定する。

### (3) 出力ファイルのファイル編成

出力ファイルのファイル編成のオペランドを、コロンで区切って指定します。出力ファイルのレコードの形式（固定長、可変長またはテキスト）は、入力ファイルと同じにしてください。

この指定は省略できます。指定を省略した場合は、入力ファイルと同じファイル編成が設定されます。

出力ファイルのファイル編成のオペランドを次の表に示します。

表 10-4 出力ファイルのファイル編成のオペランド（集約機能）

オペランド	意味
s	固定長順編成ファイルを指定する。
v	可変長順編成ファイルを指定する。
r	固定長相対ファイルを指定する。
w	可変長相対ファイルを指定する。
a	テキストファイルを指定する。

### (4) //レコード長

入力ファイルのファイル編成ごとのレコード長の指定方法を次の表に示します。

表 10-5 入力ファイルのファイル編成ごとのレコード長の指定方法（集約機能）

入力ファイル編成	レコード長の指定方法
固定長順編成ファイル	ファイルのレコード長を「1～65,535」バイトの範囲で指定します。
可変長順編成ファイル	指定は不要です。指定しても無視されます。 ただし、入力できるレコード長の範囲は、1～65,535 バイトです。
固定長相対ファイル	
可変長相対ファイル	
テキストファイル	ファイルの最大レコード長を「1～65,535」バイトの範囲で指定します。 レコード長には、改行文字「X'0A'」は含みません。 省略できます。省略した場合「65,535」バイトが設定されます。 レコード長を指定した方がソート時間が短くなります。

### (5) /o 出力ファイル名

集約結果の出力先ファイルをファイル名で指定します。

この指定は、省略できます。指定を省略した場合は、集約結果を標準出力ファイルに出力します。

### (6) キー属性

集約キーのキー属性を指定します。キー属性のパラメタを次の表に示します。

## 10. 集約コマンドの使い方

表 10-6 キー属性のパラメタ (集約機能)

パラメタ	意味
/c	キーが文字であることを指定する。
/j	キーが文字 (JIS8 単位コード※) であることを指定する。
/n	キーが符号付き固定小数点 2 進数であることを指定する。
/m	キーが符号なし 2 進数であることを指定する。
/p	キーが内部 10 進数であることを指定する。
/z	キーが外部 10 進数であることを指定する。
/x	キーが左符号付き外部 10 進数であることを指定する。
/s	キーが右分離符号付き外部 10 進数であることを指定する。
/t	キーが左分離符号付き外部 10 進数であることを指定する。
/f	キーが内部浮動小数点であることを指定する。
/e	キーが日付 (英数字) であることを指定する。

### 注※

ASCII コード (0x00 ~ 0x7F) を含む, JIS8 単位コード (0x00 ~ 0xFF) を意味します。入力データに日本語などのマルチバイト文字が含まれている場合, マルチバイト文字として扱わず, シングルバイト文字として扱います。EBCDIK コードの体系に従ってキー・項目を比較する「-v オプション (Windows は /v オプション)」を使用できます。

### (7) + キー位置 - キー長

キーの開始バイト位置を 5 桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置 (先頭を 0 とする) とします。開始バイト位置に指定できる値の範囲は, 「0 ~ レコード長 - 1」です。

また, キーのバイト長を 「1 ~ 4,096」 の範囲で指定します。

この指定は, 省略できません。

#### 集約キーの指定範囲

集約キーは, 次に示す範囲で指定してください。

- 入力ファイルが固定長ファイルのとき, キーは, レコード長の範囲
- 入力ファイルが可変長ファイルおよびテキストファイルのとき, キーは, 最小レコード長の範囲

### (8) /a

無条件レコード抽出機能を使用する場合は, 「/a」だけを指定します。また, 集約項目は, 集約項目の属性と「+ 項目位置 - 項目長」で指定します。

## (a) 集約項目の属性

集約項目の属性を指定します。集約項目の属性のオペランドを次の表に示します。

表 10-7 集約項目の属性のオペランド

オペランド	意味
n	集約項目が符号付き固定小数点 2 進数であることを指定する。
m	集約項目が符号なし 2 進数であることを指定する。
p	集約項目が内部 10 進数であることを指定する。
z	集約項目が外部 10 進数であることを指定する。
x	集約項目が左符号付き外部 10 進数であることを指定する。
s	集約項目が右分離符号付き外部 10 進数であることを指定する。
t	集約項目が左分離符号付き外部 10 進数であることを指定する。

## (b) + 項目位置 - 項目長

集約項目の開始バイト位置を 5 桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置（先頭を 0 とする）とします。開始バイト位置に指定できる値の範囲は、「0 ～レコード長 -1」です。

また、集約項目のバイト長を「1 ～ 19」の範囲で指定します。

この指定は、省略できません。

集約項目を複数指定する場合は、集約項目の属性と「+ 項目位置 - 項目長」をひとまとめにして指定してください。

## 集約項目の指定範囲

集約項目は、次の範囲で指定してください。

- 入力ファイルが固定長ファイルのとき、集約項目は、レコード長の範囲
- 入力ファイルが可変長ファイルおよびテキストファイルのとき、集約項目は、最小レコード長の範囲

集約項目を複数個指定する場合は、集約項目を重複して指定しないでください。

また、集約項目は、集約キーと重複して指定しないでください。

## (9) 入力ファイル名

集約機能の入力ファイルを指定します。出力ファイルの指定と同じファイルの指定はできません。

入力ファイルが順ファイルまたは相対ファイルの場合、この指定は、省略できます。省略した場合は、入力ファイルに標準入力ファイルが設定されます。

### 10.1.3 コマンド指定時の注意事項

集約コマンドの指定時の注意事項を次に示します。

#### (1) オペランドの入力順序

各パラメタのオペランドの入力順序は任意です。ただし、次に示すパラメタのオペランドは、次の順序で入力してください。

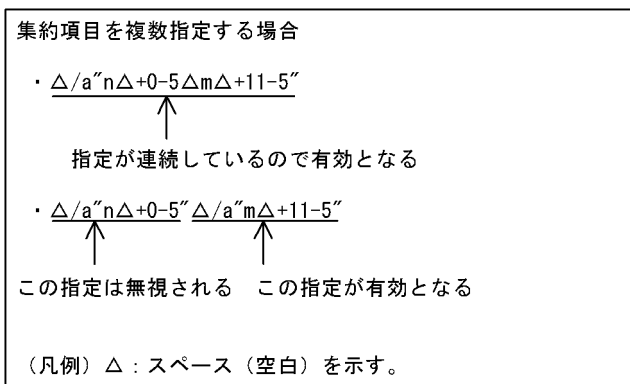
- 入力ファイルのファイル編成と出力ファイルのファイル編成  
「入力ファイルのファイル編成」「出力ファイルのファイル編成」
- キー属性と + キー位置 - キー長のパラメタ  
「キー属性」「+ キー位置 - キー長」
- 集約項目の属性と + キー位置 - キー長のパラメタ  
「集約項目の属性」「+ キー位置 - キー長」

#### (2) 同一パラメタのオペランドの複数指定

複数指定が不要なオペランドを複数指定した場合は、最後に指定したオペランドが有効になります。

#### (3) 複数指定できるパラメタ

複数指定できるパラメタを複数指定するときは、連続で指定してください。指定例を次に示します。



#### (4) スペース (空白) の扱い

パラメタで使用する文字「/l, /o, /a...」の / (スラント) の前にスペース (空白) を入れるかどうかは任意です。しかし、+ キー位置 - キー長のハイフンの前には、スペースを入れないでください。指定例を次に示します。

(例)

- 「Δ /l128 Δ /oOUTFILE」 ……有効となる



- 「+10 △ -8」 ……エラーとなる

(凡例) △ : スペース (空白) を示す。

また、パラメタで使用する文字「/l, /o, /a…」とそのあとに続けるオペランドの間にスペース (空白) を入れるかどうかは任意です。指定例を次に示します。

(例)

- 「/l128」 ……有効となる
- 「/l △ 128」 ……有効となる

(凡例) △ : スペース (空白) を示す。

## (5) 英文字の扱い

各パラメタを大文字で入力するか、小文字で入力するかは任意です。また、大文字と小文字が混在してもかまいません。

### 10.1.4 使用例

集約コマンドの使用例を示します。

#### (1) 文字項目をキーとして集約する例

文字項目をキーとして集約する例を示します。この例では、商品コードをキーとして売上数を集約します。

##### (a) 条件

集約の条件は、次のとおりです。なお、入力ファイルは、あらかじめ集約キーでソートしておくことを前提とします。

- 32 ビット版 SORT を使用する
- 集約キーの比較は、JIS8 単位コードの体系に従う
- 入力ファイルおよび出力ファイルは、固定長順編成ファイル
- 入力ファイルおよび出力ファイルのレコード長は、128 バイト
- 集約結果の出力ファイル名は「OUTFILE」
- 集約キーの属性は文字項目 (文字)
- 集約キーの開始バイト位置は「5」バイト目、バイト長は「4」バイト
- 集約項目の属性は数字項目 (内部 10 進数)
- 集約項目の開始バイト位置は「17」バイト目、バイト長は「4」バイト
- 入力ファイル名は「INFILE」

##### (b) 入力するコマンド

入力する集約コマンドを次の図に示します。

## 10. 集約コマンドの使い方

図 10-3 入力する集約コマンド

```
wncsum△/gs△/l128△/oOUTFILE  
△/c△+5-4△/a"p△+17-4"△INFILE
```

(凡例)

△：スペース（空白）を示す。

### (c) コマンドの指定内容

集約コマンドの指定内容を次の表に示します。

表 10-8 集約コマンドの指定内容

入力項目	指定内容
wncsum	集約処理のコマンド「wncsum」を指定する。
/gs	入力ファイルを「固定長順編成ファイル」で指定する。
/l128	入力ファイルおよび出力ファイルのレコード長を「128」バイトで指定する。
/oOUTFILE	集約結果の出力先ファイル名「OUTFILE」を指定する。
/c	集約キーの属性を文字項目の「文字」で指定する。
+5-4	集約キーの開始バイト位置を「5」バイト目、バイト長を「4」バイトで指定する。
/a	集約項目の属性を指定する。
p	集約項目の属性を数字項目の「内部 10 進数」で指定する。
+17-4	集約項目の開始バイト位置を「17」バイト目、バイト長を「4」バイトで指定する。
INFILE	入力ファイル名「INFILE」を指定する。

注 1

集約キーの比較は JIS8 単位コード体系に従うので、集約オプションの指定を省略する。

注 2

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略する。

## 10.2 集約コマンドの使い方（UNIX の場合）

集約コマンドの使い方を次に示します。

### 10.2.1 コマンドの使い方

集約コマンドのコマンド名を次に示します。

- `rsum`（32 ビット版 SORT で集約機能を使う場合）
- `rsum64`（64 ビット版 SORT で集約機能を使う場合）

集約コマンドの入力方法について説明します。

#### (1) コマンドの形式

集約コマンドは、コマンド名（`rsum` または `rsum64`）とそのあとに続く各オペランドで構成されます。集約コマンドの形式を次の図に示します。

図 10-4 集約コマンドの形式

```
rsum*Δ [集約オプション]
Δ 入力ファイルのファイル編成 [:出力ファイルのファイル編成]
Δ [-Lレコード長]
Δ [-o出力ファイル名]
Δ { キー属性Δ+キー位置-キー長 } . . .
Δ -a [集約項目情報]
Δ [入力ファイル名]
```

- (凡例)     Δ : スペース（空白）を示す。  
           [ ] : 省略できることを示す。  
           . . . : 直前の {} で囲まれた部分を複数繰り返して指定できることを示す。
- 注※ 使用している SORT が 64 ビット版対応の場合は、「`rsum64`」となる。

### 10.2.2 オペランドの意味

オペランドには、必ず指定するオペランドと任意に指定できるオペランドの 2 種類があります。

#### (1) 集約オプション

集約処理で使用する機能上のオプションを指定します。集約オプションのパラメタを次の表に示します。

表 10-9 集約オプションのパラメタ

パラメタ	意味
<code>-e</code>	入力ファイルを排他モードでオープンする（ファイル排他）。

## 10. 集約コマンドの使い方

パラメタ	意味
-i	出力ファイルと同じ名前のファイルがすでにある場合、そのファイルの内容が書き換えられるのを防ぐ（既存ファイルの書き換え防止）。
-v	EBCDIK コードの体系に従ってキーを比較する。
-b	集約項目に 2 進項目を指定した場合、集約結果のオーバーフローの判定をバイト長から桁数に変更する（2 進項目の 10 進表示桁判定）。
-q	出力データがキャッシュファイルに残っている場合、すべてのデータをファイル出力したあと、コマンドを終了する（キャッシュファイルの同期出力）。
-if	集約項目に内部 10 進数を指定した場合、演算符号が (F) <sub>16</sub> （絶対値）である項目の集約結果を同じ演算符号（絶対値）にする。 このパラメタを指定しない場合、集約した結果の演算符号は (C) <sub>16</sub> （正）になる。
-B <sup>※1</sup>	集約キーおよび集約項目が数値項目の場合、big endian <sup>※2</sup> の数値として扱う。

### 注

「-q」を指定した場合、コマンドの実行時間が長くなることもあるため注意してください。  
また、入力ファイルが索引ファイルの場合は、「-e」を指定した方が処理時間が短くなります。

### 注※1

Linux の場合だけ有効なオプションです。また、集約キーは、2 進項目「-n」、「-m」、または内部浮動小数点項目「-f」の場合だけ有効です。集約項目は、2 進項目「-n」または「-m」の場合だけ有効です。

### 注※2

big endian とは、上位バイトを下位番地とする数値のことです。

Linux で「-B」を指定しなかった場合は、little endian（上位バイトを上位番地とする数値）として扱います。

複数のオプションを指定する場合は、次に示すどれかの方法で指定します。

- 「-e Δ -i」のように分けて指定する（「Δ」は、スペース（空白）を示します）。
- 「-ei」のようにつなげて指定する。
- 「-if」をほかのオプションとつなげて指定する場合は、「-ifiv」のように「-if」を必ず先頭に指定する。

この指定は省略できます。省略した場合は、次のようになります。

- 「-e」の指定を省略した場合、入力ファイルが共用モードでオープンされます。
- 「-i」の指定を省略した場合、出力ファイル名と同じ名前のファイルがすでにあると、そのファイルの内容が書き換えられます。
- 「-v」の指定を省略した場合、JIS8 単位コードの体系に従ってキーが比較されます。
- 「-if」の指定を省略した場合、演算符号が (F)<sub>16</sub>（絶対値）の集約項目を集約した結果の演算符号は (C)<sub>16</sub>（正）となります。

なお、集約オプションを指定する場合は、次の点に注意してください。

- 入力ファイルの書き込みが許可されていない場合、および入力ファイルがすでにほか

の処理でオープンされている場合は、「-e」は指定できません。

- キー属性に「-j」を指定していない場合は、「-v」を指定しても無効になります。
- 集約項目のオプションに2進項目（n, m）を指定していない場合、「-b」を指定しても無効になります。
- 「-if」を指定し、かつ演算符号に(C)<sub>16</sub>（正）と(F)<sub>16</sub>（絶対値）が混在している場合は、集約した結果の演算符号は保証しません。

## (2) 入力ファイルのファイル編成

入力ファイルのファイル編成を指定します。

この指定は、省略できません。必ず指定してください。入力ファイルのファイル編成のパラメタを次の表に示します。

表 10-10 入力ファイルのファイル編成のパラメタ（集約機能）

パラメタ	意味
-S	固定長順ファイルを指定する。
-V	可変長順ファイルを指定する。
-R	固定長相対ファイルを指定する。
-W	可変長相対ファイルを指定する。
-A	テキストファイルを指定する。

## (3) 出力ファイルのファイル編成

出力ファイルのファイル編成を指定します。出力ファイルのレコードの形式（固定長、可変長またはテキスト）は、入力ファイルと同じにしなければなりません。

この指定は、省略できます。指定を省略した場合は、入力ファイルと同じファイル編成が設定されます。

出力ファイルのファイル編成のパラメタを次の表に示します。

表 10-11 出力ファイルのファイル編成のパラメタ（集約機能）

パラメタ	意味
:S	固定長順ファイルを指定する。
:V	可変長順ファイルを指定する。
:R	固定長相対ファイルを指定する。
:W	可変長相対ファイルを指定する。
:A	テキストファイルを指定する。

## (4) -L レコード長

入力ファイル編成ごとのレコード長の指定方法を次の表に示します。

## 10. 集約コマンドの使い方

表 10-12 入力ファイル編成ごとのレコード長の指定方法（集約機能）

入力ファイル編成	レコード長の指定方法
固定長順ファイル	ファイルのレコード長を「1～65,535」バイトの範囲で指定します。
可変長順ファイル	指定は不要です。指定しても無視されます。
固定長相対ファイル	ただし、入力できるレコード長の範囲は、1～65,535 バイトです。
可変長相対ファイル	
テキストファイル	ファイルの最大レコード長を「1～65,535」バイトの範囲で指定します。 レコード長には、改行文字「X'0A'」は含みません。 この指定は省略できます。指定を省略した場合は、「65535」バイトが設定されます。 なお、レコード長を指定した方が、ソート時間が短くなります。

### (5) -o 出力ファイル名

集約結果の出力先ファイルをファイル名で指定します。

この指定は、省略できます。指定を省略した場合は、集約結果を標準出力ファイルに出力します。

### (6) キー属性

集約キーの属性を指定します。キー属性のパラメタを次の表に示します。

表 10-13 キー属性のパラメタ（集約機能）

パラメタ	意味
-c	キーが文字であることを指定する。
-j	キーが文字（JIS8 単位コード※）であることを指定する。
-n	キーが符号付き固定小数点 2 進数であることを指定する。
-m	キーが符号なし 2 進数であることを指定する。
-p	キーが内部 10 進数であることを指定する。
-z	キーが外部 10 進数であることを指定する。
-x	キーが左符号付き外部 10 進数であることを指定する。
-s	キーが右分離符号付き外部 10 進数であることを指定する。
-t	キーが左分離符号付き外部 10 進数であることを指定する。
-f	キーが内部浮動小数点であることを指定する。
-g	キーが日付（英数字）であることを指定する。

注※

ASCII コード（0x00～0x7F）を含む、JIS8 単位コード（0x00～0xFF）を意味します。入力データに日本語などのマルチバイト文字が含まれている場合、マルチバイト文字として扱わず、シングルバイト文字として扱います。EBCDIK コードの体系に従ってキー・項目を比較する

「-v オプション (Windows は /v オプション)」を使用できます。

### (7) + キー位置 - キー長

キーの開始バイト位置を 5 桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置 (先頭を 0 とする) とします。開始バイト位置に指定できる値の範囲は、「0 ~ 65,534」です。

また、キーのバイト長を 4 桁以内の数字「1 ~ 4,096」で指定します。

この指定は、省略できません。必ず指定してください。

#### キーの指定範囲

キーは、次に示す範囲で指定してください。

- 入力ファイルが固定長ファイルのとき、キーは、レコード長の範囲
- 入力ファイルが可変長ファイルのとき、キーは、最小レコード長の範囲
- 入力ファイルがテキストファイルのとき、キーは、最小レコード長の範囲

### (8) -a 集約項目情報

レコードを集約するための集約項目情報を指定します。指定できる集約項目の個数は、0 ~ 64 個です。集約項目情報の形式を次の図に示します。

図 10-5 集約項目情報の形式

-a {集約項目の属性△+項目位置-項目長} . . .

- (凡例) △ : スペース (空白) を示す。  
 . . . : 直前の { } で囲まれた部分を複数繰り返して指定できることを示す。

無条件レコード抽出機能を使用する場合、集約項目の個数を 0 個 (-a だけを指定) にします。無条件レコード抽出機能については、「2.4 無条件レコード抽出機能」を参照してください。

#### (a) -a 集約項目の属性

集約項目の属性を指定します。集約項目の属性のパラメタを次の表に示します。

集約項目を複数個指定する場合は、集約項目同士が重なってはなりません。

また、集約項目は、集約キーと重なってはなりません。

表 10-14 集約項目の属性のパラメタ

パラメタ	意味
n	集約項目が符号付き固定小数点 2 進数であることを指定する。
m	集約項目が符号なし 2 進数であることを指定する。

## 10. 集約コマンドの使い方

パラメタ	意味
p	集約項目が内部 10 進数であることを指定する。
z	集約項目が外部 10 進数であることを指定する。
x	集約項目が左符号付き外部 10 進数であることを指定する。
s	集約項目が右分離符号付き外部 10 進数であることを指定する。
t	集約項目が左分離符号付き外部 10 進数であることを指定する。

### (b) + 項目位置 - 項目長

集約項目の開始バイト位置を 5 桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置（先頭を 0 とする）とします。開始バイト位置に指定できる値の範囲は、「0 ~ 65,534」です。

また、集約項目のバイト長を 2 桁以内の数字「1 ~ 19」で指定します。

この指定は、省略できません。必ず指定してください。

#### 集約項目の指定範囲

集約項目は、次の範囲で指定してください。

- 入力ファイルが固定長ファイルのとき、集約項目はレコード長の範囲
- 入力ファイルが可変長ファイルのとき、集約項目は最小レコード長の範囲
- 入力ファイルがテキストファイルのとき、集約項目は最小レコード長の範囲

### (9) 入力ファイル名

集約機能の入力ファイルを指定します。出力ファイルの指定と同じファイルの指定はできません。

この指定は、省略できます。省略した場合は、入力ファイルに標準入力ファイルが設定されます。

## 10.2.3 コマンド指定時の注意事項

集約コマンドの指定時の注意事項を次に示します。

### (1) パラメタの入力順序

各パラメタの入力順序は任意です。ただし次の三つのパラメタは、この順序で入力しなければなりません。

- キー属性と + キー位置 - キー長のパラメタ  
「キー属性」「+ キー位置 - キー長」
- 集約項目の属性と + 項目位置 - 項目長のパラメタ  
「集約項目の属性」「+ 項目位置 - 項目長」
- 入力ファイルのファイル編成と出力ファイルのファイル編成のパラメタ



「入力ファイルのファイル編成」「出力ファイルのファイル編成」

## (2) 同一パラメタの複数指定

同一パラメタを複数指定した場合は、最後に指定したパラメタが有効になります。

## (3) 複数指定できるパラメタ

複数指定できるパラメタを複数指定するときは、連続で指定してください。指定例を次に示します。

(例)

・ -c△+0-5△-z△+5-4

↑

指定が連続しているのが有効となる

・ -c△+0-5 (レコード長) -z△+5-4△-x△+9-8

↑

この指定は無視される

↑

指定が連続しているのが有効となる

(凡例) △ : スペース (空白) を示す。

## (4) スペース (空白) の扱い

オペランドで使用する文字「-L, -o, -a」の- (ハイフン) の前にはスペース (空白) を入れなければなりません。ただし、+項目位置-項目長のハイフンの前には、スペースは不要です。指定例を次に示します。

(例)

- 「△ -L128 △ -oOUTFILE」 ……有効となる
- 「△ -L128-oOUTFILE」 ……エラーとなる

(凡例) △ : スペース (空白) を示す。

また、オペランドで使用する文字「-L, -o, -a」とそのあとに続けるパラメタの間にスペースを入れるかどうかは任意です。指定例を次に示します。

(例)

- 「-L128」 ……有効となる
- 「-L △ 128」 ……有効となる

(凡例) △ : スペース (空白) を示す。

## (5) 集約項目情報の指定

### (a) 集約項目情報の複数指定

集約項目情報を複数指定するときは、連続で指定してください。その場合、先頭の集約項目のオプションにだけ「-a」を付けます。指定例を次に示します。

(例)

・ -an△+0-5△m△+11-5

↑

先頭の集約項目属性だけに「-a」を付けているので有効となる

・ -an△+0-5△-am△+11-5

↑

↑

この指定は無視される

先頭以外に「-a」が付いているため、この指定が有効となる

(凡例) △ : スペース (空白) を示す。

### (b) 数字以外の指定

集約項目属性に数字項目 (n, m, p, z, x, s, t) 以外を指定すると、入力ファイルと見なされます。

(例)

・ -an△+0-5△c△+11-5

↑

↑

入力ファイル名とみなされる

(凡例) △ : スペース (空白) を示す。

## 10.2.4 使用例

集約コマンドの使用例を示します。

### (1) 文字項目をキーとして集約する例

文字項目をキーとして集約する例を示します。ここでは、商品コードをキーとして売上数を集約する例を示します。

#### (a) 条件

集約の条件は、次のとおりです。

- 32 ビット版 SORT を使用する

- 集約キーの比較は、JIS8 単位コードの体系に従う
- 入力ファイルおよび出力ファイルは、固定長順ファイル
- 入力ファイルおよび出力ファイルのレコード長は、128 バイト
- 集約結果の出力ファイル名は「OUTFILE」
- 集約キーの属性は文字項目（文字）
- 集約キーの開始バイト位置は「5」バイト目、バイト長は「4」バイト
- 集約項目の属性は数字項目（内部 10 進数）
- 集約項目の開始バイト位置は「17」バイト目、バイト長は「4」バイト
- 入力ファイルは標準入力ファイル

また、ソートの条件は、次のとおりです。

- 32 ビット版 SORT を使用する
- ソートキーの比較は JIS8 単位コードの体系に従う
- 入力ファイルおよび出力ファイルは、固定長順ファイル※
- 入力ファイルおよび出力ファイルのレコード長は、128 バイト※
- ソート結果の出力ファイルは標準出力ファイル
- ソートは昇順、キーの属性は文字項目（文字）※
- ソートキーの開始バイト位置は「5」バイト目、バイト長は「4」バイト※
- ソートするファイルのファイル名は「INFILE」

注※

集約処理と同じ値を指定しなければなりません。

#### (b) 入力するコマンド

入力する集約コマンドを次の図に示します。

図 10-6 入力する集約コマンド（その 1）

```
rsort△-S△-L128△-c△+5-4△INFILE
|rsum△-S△-L128△-oOUTFILE△-c△+5-4△-ap△+17-4
```

（凡例） △：スペース（空白）を示す。  
|：パイプを示す。

#### (c) コマンドの指定内容

集約コマンドの指定内容を次の表に示します。

表 10-15 集約コマンドの指定内容（その 1）

項番	入力項目	指定内容
1	rsort	ソート処理のコマンド「rsort」を指定する。
2	-S	入力ファイルを「固定長順ファイル」で指定する。

## 10. 集約コマンドの使い方

項番	入力項目	指定内容
3	-L128	入力ファイルおよび出力ファイルのレコード長を「128」バイトで指定する。
4	-c	キーの整列順序を「昇順」、属性を文字項目の「文字」で指定する。
5	+5-4	キーの開始バイト位置を「5」バイト目、バイト長を「4」バイトで指定する。
6	-INFILE	ソートするファイルのファイル名「INFILE」を指定する。
7	rsum	集約処理のコマンド「rsum」を指定する。
8	-S	入力ファイルを「固定長順ファイル」で指定する。
9	-L128	入力ファイルおよび出力ファイルのレコード長を「128」バイトで指定する。
10	-oOUTFILE	集約結果の出力先ファイル名「OUTFILE」を指定する。
11	-c	集約キーの属性を文字項目の「文字」で指定する。
12	+5-4	集約キーの開始バイト位置を「5」バイト目、バイト長を「4」バイトで指定する。
13	-ap	集約項目の属性を数字項目の「内部 10 進数」で指定する。
14	+17-4	集約項目の開始バイト位置を「17」バイト目、バイト長を「4」バイトで指定する。

### 注 1

ソートキーおよび集約キーの比較は JIS8 単位コード体系に従うので、指定を省略します。

### 注 2

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略します。

## (2) 集約項目を複数指定して集約する例

文字項目をキーとして、集約項目を 2 個指定して集約する例を示します。ここでは、ソート済みの商品名を集約キーとして、売上数および売上金額を集計する例を示します。

### (a) 条件

集約の条件は、次のとおりです。

- 32 ビット版 SORT を使用する
- 集約キーの比較は、JIS8 単位コードの体系に従う
- 入力ファイルおよび出力ファイルは、固定長順ファイル
- 入力ファイルおよび出力ファイルのレコード長は、128 バイト
- 集約するファイルのファイル名は「INFILE」
- 集約結果の出力ファイル名は「OUTFILE」
- 集約キーの属性は文字項目（文字）
- 集約キーの開始バイト位置は「9」バイト目、バイト長は「24」バイト
- 集約項目（売上数）の属性は数字項目（内部 10 進数）
- 集約項目（売上数）の開始バイト位置は「37」バイト目、バイト長は「4」バイト
- 集約項目（売上金額）の属性は数字項目（外部 10 進数）

- 集約項目（売上金額）の開始バイト位置は「41」バイト目、バイト長は「8」バイト

## (b) 入力するコマンド

入力する集約コマンドを次の図に示します。

図 10-7 入力する集約コマンド（その 2）

```
rsum△-S△-L128△-c△+9-24△ INFILE△-oOUTFILE△-ap△+37-4△z
△+41-8
```

（凡例） △：スペース（空白）を示す。

## (c) コマンドの指定内容

集約コマンドの指定内容を次の表に示します。

表 10-16 集約コマンドの指定内容（その 2）

入力項目	指定内容
rsum	集約処理のコマンド名「rsum」を指定する。
-S	入力ファイルを「固定長順ファイル」で指定する。
-L128	入力ファイルおよび出力ファイルの最大レコード長を「128」バイトで指定する。
-c	集約キーの属性を文字項目の「文字」で指定する。
+9-24	集約キーの開始バイト位置を「9」バイト目、バイト長を「24」バイトで指定する。
INFILE	集約するファイルのファイル名「INFILE」を指定する。
-oOUTFILE	集約結果の出力ファイル名「OUTFILE」を指定する。
-ap	集約項目（売上数）の属性を数字項目の「内部 10 進」で指定する。
+37-4	集約項目（売上数）の開始バイト位置を「37」バイト目、バイト長を「4」バイトで指定する。
z	集約項目（売上金額）の属性を数字項目の「外部 10 進」で指定する。
+41-8	集約項目（売上金額）の開始バイト位置を「41」バイト目、バイト長を「8」バイトで指定する。

## 注 1

集約キーの比較は JIS8 単位コード体系に従うので、指定を省略します。

## 注 2

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略します。



# 11 拡張 SORT コマンドの使い方 (Windows の場合)

この章では、SORT の拡張 SORT コマンドの使い方について説明します。

- 
- 11.1 コマンドの使い方
  - 11.2 オペランドの意味
  - 11.3 コマンド指定時の注意事項
  - 11.4 使用例
-

## 11.1 コマンドの使い方

---

ソート機能, 選択機能および集約機能を組み合わせて使用する場合は, 拡張 SORT コマンドを使用します。拡張 SORT コマンド使用時は, パラメタファイルを指定します。

拡張 SORT コマンドのコマンド名を次に示します。


- wnsortex (32 ビット版 SORT で拡張 SORT コマンドを使う場合)
- wnsortex64 (64 ビット版 SORT で拡張 SORT コマンドを使う場合)

拡張 SORT コマンドとパラメタファイルの形式について説明します。

### 11.1.1 拡張 SORT コマンドの形式

コマンドは, コマンド名 (wnsortex または wnsortex64) とそのあとに続くパラメタファイル名で構成されます。コマンドの形式を次の図に示します。

図 11-1 拡張 SORT コマンドの形式



```
wnsortex*Δパラメタファイル名
```

(凡例) Δ : スペース (空白) を示す。

注※ 使用している SORT が 64 ビット版対応の場合は, 「wnsortex64」となる。

「パラメタファイル名」は, 拡張子「.WSP」を付けて指定します。



## 11.2 オペランドの意味

### 11.2.1 パラメタファイルの形式

パラメタファイルは、キーワードとオペランドで、改行文字を区切りとして構成します。パラメタファイルの形式を次の図に示します。また、各パラメタの指定の組み合わせを次の表に示します。

図 11-2 パラメタファイルの形式

```
[WNSORTEX] ※1
exec_opt=実行オプション
sort_opt=機能オプション
file_org=入力ファイルのファイル編成 [:出力ファイルのファイル編成]
record_leng=入力ファイルのレコード長
temp_dir1=ソート用一時ファイルディレクトリ
temp_dir2=中間ファイルディレクトリ
temp_dir3=マージ用一時ファイルディレクトリ
select_exp=選択条件式
work_size=ワークバッファサイズ
sort_key={ソートキー属性+キー位置-キー長} . . . ※2
sum_key={集約キー属性+キー位置-キー長} . . . ※2
sum_field={集約項目の属性+項目位置-項目長}
output_file=出力ファイル名
input_file={入力ファイル名} . . . ※3
```

(凡例) [ ] : 省略できることを示す。  
 . . . : 直前の { } で囲まれた部分を複数繰り返して指定できることを示す。

- 注※1 [WNSORTEX]は必ず先頭に指定する。  
 注※2 ソートキーおよび集約キーを複数指定する場合は、「,」で区切って指定する。  
 (例) j+0-5.p+10-20  
 注※3 入力ファイル名を複数指定する場合は、「|」で区切って指定する。  
 (例) INFILE1|INFILE2

表 11-1 パラメタの指定の組み合わせ (拡張 SORT コマンド)

キーワード		exec_opt (実行オプション)		
		select (選択)	sort (ソート)	sum (集約)
[WNSORTEX] (パラメタファイルの先頭)		◎	◎	◎
sort_opt=	i (既存ファイルの書き換え防止)	△	△	△
	v (EBCDIK コードに従った処理)	△	△	△
	b (数値は big endian)	△	△	△
	u (選択条件を満足しないレコードを選択)	△	×	×
	d (重複レコードの入力順序保存機能)	×	△	×
	h (10進数の桁数でオーバーフロー判定)	×	×	△

## 11. 拡張 SORT コマンドの使い方 (Windows の場合)

キーワード		exec_opt (実行オプション)		
		select (選択)	sort (ソート)	sum (集約)
file_org=	入力ファイルのファイル編成	◎	◎	◎
	: 出力ファイルのファイル編成	○※1	○※1	△
record_leng= (入力ファイルのレコード長)		○※2	○※2	○※2
temp_dir1= (ソート用一時ファイルディレクトリ)		×	△	×
temp_dir2= (中間ファイルディレクトリ)		△※3	△※3	△※3
temp_dir3= (マージ用一時ファイルディレクトリ)		×	△	×
select_exp= (選択条件式)		◎	×	×
work_size= (ワークバッファサイズ)		×	△	×
sort_key= (ソートキー属性 + キー位置・キー長)		×	◎	○※4
sum_key= (集約キー属性 + キー位置・キー長)		×	×	○※4
sum_field= (集約項目の属性 + 項目位置・項目長)		×	×	△
output_file= (出力ファイル名)		△	△	△
input_file= (入力ファイル名)	先頭	△	△	△
	2 番目以降	△	△	×

(凡例)

- ◎ : 必ず指定するパラメタであることを示す。
- : 場合によって指定が必要なパラメタであることを示す。
- △ : 任意に指定できるパラメタであることを示す。
- × : 指定しても無視されるパラメタであることを示す。

注※1

入力ファイルが索引ファイルの場合必ず指定する。

注※2

入力ファイルが固定長順編成ファイルまたはテキストファイルの場合必ず指定する。

注※3

実行上のオプションを 2 個以上指定した場合に有効。

注※4

sort\_key= または sum\_key= のどちらかを必ず指定する。

なお、任意に指定できるパラメタは、指定方法が間違っている場合、その指定は無視されます。

パラメタの指定の組み合わせの例を次に示します。

(例) 索引編成のファイルから選択したレコードをソートし、出力する場合  
必ず指定する項目

- 予約語「[WNSORTEX]」
- 実行オプションの「選択」および「ソート」
- 入力ファイルのファイル編成
- 出力ファイルのファイル編成
- 選択条件式
- ソートキー属性 + キー位置 - キー長

指定が任意の項目

- 機能オプションの「既存ファイルの書き換え防止」
- 機能オプションの「EBCDIK コードに従った処理」
- 機能オプションの「数値は big endian」
- 機能オプションの「選択条件を満足しないレコードを選択」
- 機能オプションの「重複レコードの入力順序保存機能」
- ソート用一時ファイルディレクトリ
- 中間ファイルディレクトリ
- マージ用一時ファイルディレクトリ
- ワークバッファサイズ
- 入力ファイル名
- 出力ファイル名

指定しても無視される項目

- 機能オプションの「10 進数の桁数でオーバーフロー判定」
- 集約キー属性 + キー位置 - キー長
- 集約項目の属性 + 項目位置 - 項目長

## 11.2.2 拡張 SORT コマンドのパラメタの意味

パラメタには、必ず指定するパラメタと任意に指定できるパラメタの 2 種類があります。

### (1) [WNSORTEX]

`wnsortex` コマンドのパラメタファイルの先頭を意味します。この指定は省略できません。

### (2) `exec_opt=` 実行オプション

`wnsortex` コマンドで実行する機能を「`exec_opt=`」 + オペランドの形式で指定します。実行オプションの指定形式を次の図に、オペランドを次の表に示します。

図 11-3 実行オプションの指定形式 (拡張 SORT コマンド)

```
exec_opt = { select [:sort] [:sum]
            sort [:sum]
            sum }
```

(凡例)

{ } : どれか一つを選択することを示す。

[ ] : 省略できることを示す。

表 11-2 実行オプションのオペランド (拡張 SORT コマンド)

オペランド	意味
select	レコードを選択する。
sort	レコードをソートする。
sum	レコードを集約する。

オペランドを複数指定した場合は、選択、ソート、集約の順に実行します。例えば、ソートと選択を指定した場合は、選択されたデータをソートします。

### (3) sort\_opt= 機能オプション

wnsortex コマンドで実行する機能オプションを「sort\_opt=」+オペランドの形式で指定します。機能オプションのオペランドを次の表に示します。

表 11-3 機能オプションのオペランド (拡張 SORT コマンド)

オペランド	内容
d	ファイル中に重複したキーのレコードがある場合、それらのデータを入力順に並べることを指定する (重複レコードの入力順序保存機能)。
i	出力ファイルと同じ名前のファイルがすでにある場合、そのファイルの内容が書き換えられるのを防ぐ (既存ファイルの書き換え防止)。
v	EBCDIK コードの体系に従ってキーを比較する。
b <sup>※1</sup>	キーが数値項目の場合、big endian <sup>※2</sup> の数値として扱う。
u	選択条件を満足しないレコードだけを入力する。
h	集約項目の属性に2進数を指定した場合、10進数の桁数でオーバーフローの判定をする。

注※1

キーが2進項目「n」、「m」、または内部浮動小数点項目「f」の場合だけ有効となる。

注※2

big endian とは、上位バイトを下位番地とする数値のことです。

b オペランドを指定しなかった場合は、little endian (上位バイトを上位番地とする数値) として扱います。

複数のオペランドを指定する場合は、「di」のようにつなげて指定します。「d △ i」のように分けて指定した場合は、最初に指定したオペランドが有効になります。

また、この指定は省略できます。省略した場合は、次のようになります。

- 「d」の指定を省略した場合、重複レコードの入力順序は保存されません。
- 「i」の指定を省略した場合、出力ファイルと同じ名前のファイルがすでにあると、そのファイルの内容が書き換えられます。
- 「v」の指定を省略した場合、JIS8 単位コードの体系に従ってキーが比較されます。

なお、機能オプションを指定する場合は、次の点に注意してください。

- 入力ファイルが索引ファイルの場合は、「d」を指定しても無効になります。
- ソートキーのキー属性に「j」を指定していない場合は、「v」を指定しても無効になります。
- ファイル中に重複したキーがある場合、ワークバッファサイズを変えることで、重複したキーの並びが変わる場合があります。重複したキーの並びを保つためには、「d」を指定してください。

#### (4) file\_org= 入力ファイルのファイル編成

入力ファイルのファイル編成を「file\_org=」+オペランドの形式で指定します。複数の入力ファイルを指定する場合は、すべて同じファイル編成にしてください。また、この指定は省略できません。入力ファイルのファイル編成のオペランドを次の表に示します。

表 11-4 入力ファイルのファイル編成のオペランド (拡張 SORT コマンド)

オペランド	内容
s	固定長順編成ファイルを指定する。
v	可変長順編成ファイルを指定する。
r	固定長相対ファイルを指定する。
w	可変長相対ファイルを指定する。
i	固定長索引ファイルを指定する。
x	可変長索引ファイルを指定する。
a	テキストファイルを指定する。

#### (5) 出力ファイルのファイル編成

出力ファイルのファイル編成のオペランドを、コロン (:) で区切って指定します。出力ファイルのファイルの形式 (固定長, 可変長またはテキスト) は、入力ファイルと同じにしてください。

入力ファイルが順ファイル, 相対ファイルまたはテキストファイルの場合, この指定は省略できます。指定を省略した場合は, 入力ファイルと同じファイル編成が設定されます。入力ファイルが索引ファイルの場合は, 順ファイルまたは相対ファイルのどちらか

## 11. 拡張 SORT コマンドの使い方 (Windows の場合)

を必ず指定してください。

なお、指定を省略するときは、コロンも省略します。省略時にコロンを指定した場合はエラーとなります。

出力ファイルのファイル編成のオペランドを次の表に示します。

表 11-5 出力ファイルのファイル編成のオペランド (拡張 SORT コマンド)

オペランド	内容
s	固定長順編成ファイルを指定する。
v	可変長順編成ファイルを指定する。
r	固定長相対ファイルを指定する。
w	可変長相対ファイルを指定する。
a	テキストファイルを指定する。

### (6) record\_leng= レコード長

入力ファイルのファイル編成ごとのレコード長の指定方法を次の表に示します。

表 11-6 入力ファイルのファイル編成ごとのレコード長の指定方法 (拡張 SORT コマンド)

入力ファイル編成	レコード長の指定方法
固定長順編成ファイル	ファイルのレコード長を「1～65,535」バイトの範囲で指定します。
可変長順編成ファイル	指定は不要です。指定しても無視されます。 ただし、入力できるレコード長の範囲は、1～65,535 バイトです。
固定長相対ファイル	
可変長相対ファイル	
固定長索引ファイル	指定は不要です。指定しても無視されます。 ただし、入力できるレコード長の範囲は、1～65,503 バイトです。
可変長索引ファイル	
テキストファイル	ファイルの最大レコード長を「1～65,535」バイトの範囲で指定します。 レコード長には、改行文字「X'0A'」は含みません。 省略できます。省略した場合「65,535」バイトが設定されます。 レコード長を指定した方がソート時間が短くなります。

### (7) temp\_dir1= ソート用一時ファイルディレクトリ

ソート用一時ファイルを割り当てるディレクトリを指定します。この指定は、実行オプションで「sort」を指定した場合だけ有効となります。

ソート用一時ファイルを使用する条件については、「16. 一時ファイルの作成条件とファイル容量の算出方法」を参照してください。

また、この指定は省略できます。指定を省略した場合、使用するディレクトリの優先順位は、次の順となります。

1. 環境ファイルの `temp_dir1` で指定されているディレクトリ
2. 環境変数「TMP」で指定されているディレクトリ
3. 環境変数「TEMP」で指定されているディレクトリ
4. 次の表に示すディレクトリ

表 11-7 一時ファイルが割り当てられるディレクトリ (拡張 SORT コマンド)

OS	ディレクトリ
Windows	次の優先順位で使用するディレクトリが決まります。 1. 環境変数「USERPROFILE」で指定されているディレクトリ 2. 環境変数「SYSTEMROOT」で指定されているディレクトリ

環境ファイルの設定方法については、「15. 環境ファイルの設定」を参照してください。

### (8) temp\_dir2= 中間ファイルディレクトリ

中間ファイルを割り当てるディレクトリを指定します。この指定は、実行オプションを 2 個以上指定したときだけ有効となります。

また、この指定は省略できます。指定を省略した場合、使用するディレクトリの優先順位は、次の順となります。

1. 環境ファイルの `temp_dir2` で指定されているディレクトリ
2. 環境変数「TMP」で指定されているディレクトリ
3. 環境変数「TEMP」で指定されているディレクトリ
4. 表 11-7 に示すディレクトリ

### (9) temp\_dir3= マージ用一時ファイルディレクトリ

マージ用一時ファイルを割り当てるディレクトリを指定します。この指定は、実行オプションで「sort」を指定し、ソート実行時にマージ用一時ファイルを使用する場合だけ有効となります。

マージ用一時ファイルを使用する条件については「16. 一時ファイルの作成条件とファイル容量の算出方法」を参照してください。

また、この指定は省略できます。指定を省略した場合、使用するディレクトリの優先順位は、次の順となります。

1. ソート用一時ファイルディレクトリ
2. 環境ファイルの `temp_dir3` で指定されているディレクトリ
3. 環境ファイルの `temp_dir1` で指定されているディレクトリ
4. 環境変数「TMP」で指定されているディレクトリ
5. 環境変数「TEMP」で指定されているディレクトリ
6. 表 11-7 に示すディレクトリ

## (10) select\_exp= 選択条件式

レコードを選択するための選択条件式を指定します。この指定は、実行オプションで「select」を指定した場合だけ有効となります。また、この指定は省略できません。指定できる条件式の個数は 1 ～ 128 個です。

選択条件式の形式を次の図に示します。

図 11-4 選択条件式の形式 (拡張 SORT コマンド)

```
select_exp=比較項目の属性+項目位置-項目長△比較演算子△条件値
           [ {△論理演算子△比較項目の属性+項目位置-項目長△比較演算子△条件値}・・・]
```

- (凡例)    △    : スペース (空白) を示す。  
          [ ]    : 省略できることを示す。  
          …… : 直前の { } で囲まれた部分を複数繰り返して指定できることを示す。

なお、スペースを入れてよいのは、論理演算子の前後と比較演算子の前後だけです。

## (a) 比較項目の属性

比較項目の属性を指定します。比較項目の属性のオペランドを次の表に示します。

表 11-8 比較項目の属性のオペランド (拡張 SORT コマンド)

オペランド	内容
j	比較項目が文字 (JIS8 単位コード※) であることを指定する。
c	比較項目が文字であることを指定する。
n	比較項目が符号付き固定小数点 2 進数であることを指定する。
m	比較項目が符号なし 2 進数であることを指定する。
p	比較項目が内部 10 進数であることを指定する。
z	比較項目が外部 10 進数であることを指定する。
x	比較項目が左符号付き外部 10 進数であることを指定する。
s	比較項目が右分離符号付き外部 10 進数であることを指定する。
t	比較項目が左分離符号付き外部 10 進数であることを指定する。
f	比較項目が内部浮動小数点項目であることを指定する。
e	比較項目が日付 (英数字) であることを指定する。

## 注※

ASCII コード (0x00 ～ 0x7F) を含む、JIS8 単位コード (0x00 ～ 0xFF) を意味します。入力データに日本語などのマルチバイト文字が含まれている場合、マルチバイト文字として扱わず、シングルバイト文字として扱います。EBCDIK コードの体系に従ってキー・項目を比較する「-v オプション (Windows は /v オプション)」を使用できます。



## (b) + 項目位置 - 項目長

比較項目の開始バイト位置を 5 桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置（先頭を 0 とする）とします。開始バイト位置に指定できる値の範囲は、「0 ~ レコード長 - 1」です。

また、比較項目のバイト長を「1 ~ 4,096」の範囲で指定します。

## (c) 比較演算子

比較項目で指定した値と入力レコードの値を比較、選択する方法を指定します。比較演算子のオペランドを次の表に示します。

表 11-9 比較演算子のオペランド (拡張 SORT コマンド)

比較演算子	意味
=	条件値と等しい値の入力レコードを選択することを指定する。
!=	条件値と異なる値の入力レコードを選択することを指定する。
>	条件値より大きい値の入力レコードを選択することを指定する。
>=	条件値より大きいか、または等しい値の入力レコードを選択することを指定する。
<	条件値より小さい値の入力レコードを選択することを指定する。
<=	条件値より小さいか、または等しい値の入力レコードを選択することを指定する。

## (d) 条件値

比較の条件となる値を「c' 文字条件値 '」、 「n' 数値条件値 '」、 「x'16 進数条件値 '」または「d' 日付条件値 '」の形式で指定します。「c' 文字条件値 '」は比較項目が文字項目の場合、「n' 数値条件値 '」は比較項目が数値項目の場合に指定できます。また、「x'16 進数条件値 '」は比較項目が文字の場合でも数値の場合でも指定できます。「d' 日付条件値 '」は比較項目が日付項目の場合に指定できます。

指定できる条件値の長さを次の表に示します。

表 11-10 指定できる条件値の長さ (拡張 SORT コマンド)

項目	最小値	最大値	備考
文字条件値	1	4,096	—
数値条件値※ <sup>1</sup>	1	19	符号なしの場合は 18
16 進数条件値※ <sup>2</sup>	2	8,192	指定できるのは偶数桁のデータだけ
日付条件値※ <sup>3</sup>	2	8	—

単位：バイト

## 11. 拡張 SORT コマンドの使い方 (Windows の場合)

(凡例)

- : なし

注※ 1

数値条件値として指定できるのは「0～9」,「+」および「-」だけである。

注※ 2

16進数条件値として指定できるのは「0～9」および「A～F」だけである。

注※ 3

日付条件値として指定できる値は次のとおり。

- ・先頭の1文字はスペースまたは「0～9」の文字列
- ・2文字目は「0～9」の文字列
- ・1文字のスペースを指定した場合「00」を設定
- ・1文字の「0～9」を指定した場合、先頭の1文字は「0」を設定

### (e) 論理演算子

複数の選択条件式を指定する場合、論理演算子「AND」(論理積)または「OR」(論理和)を指定します。

選択条件式中に「AND」と「OR」が混在した場合は、「AND」を判定後、「OR」を判定します。「AND」と「OR」が混在した場合の処理の例を次に示します。

(例)

```
n+10-3△>△n'30'△AND△n+23-3△=△n'10'△  
OR△n+10-3△<=△n'10'△AND△n+23-3△=△n'10'
```

(凡例) △ : スペース (空白) を示す。

この場合、次のうちどちらかの条件を満たしたレコードが選択されます。

- ・「n+10-3」は30より大きく、「n+23-3」は10
- ・「n+10-3」は10か10より小さく、「n+23-3」は10

## (11) work\_size= ワークバッファサイズ

ワークバッファとして使用するメモリの容量を KB または MB 単位で指定します。ワークバッファサイズは、メモリ量と単位で指定します。

### (a) メモリ量

ワークバッファとして使用するメモリの容量を指定します。指定できる値を次の表に示します。

表 11-11 メモリ量の指定範囲 (拡張 SORT コマンド)

SORT の種別	レコード長※	指定範囲
32 ビット版	32,760 バイト以下	128KB ~ 2,000MB
	32,760 バイトより大きい	256KB ~ 2,000MB

SORT の種別	レコード長※	指定範囲
64 ビット版	32,760 バイト以下	256KB ~ 2,000MB
	32,760 バイトより大きい	384KB ~ 2,000MB

## 注※

可変長ファイルおよびテキストファイルの場合、最大レコード長となります。

この指定は省略できます。この指定を省略した場合、または最小メモリ量に満たない値が指定された場合は、1,024KB が設定されます。ただし、これらの場合で環境ファイルに初期値が設定されているときは、その値が設定されます。

なお、メモリ量の目安については、「17.1.1 ワークバッファサイズと処理時間」を参照してください。また、環境ファイルの設定方法については、「15. 環境ファイルの設定」を参照してください。

## (b) 単位

メモリ量の単位を、KB にするか MB にするか指定します。

- k : 単位を KB にします。
- m : 単位を MB にします。

また、単位を省略した場合は「KB (k)」が設定されます。

指定例を次に示します。

(例 1) 1,024KB → 「work\_size=1024k」または「work\_size=1024」

(例 2) 2MB → 「work\_size=2m」

## (12) sort\_key= ソートキー属性 + キー位置 - キー長

ソート実行に必要なソートキーを指定します。この指定は、実行上のオプションで「SORT」を指定した場合だけ有効となります。また、この指定は省略できません。

## (a) キー属性

ソートキーのキー属性を「sort\_key=」+ オペランドの形式で指定します。キー属性のオペランドを次の表に示します。

表 11-12 ソートキー属性のオペランド (拡張 SORT コマンド)

オペランド	内容
r	キーの降順でソートすることを指定する。
c	キーが文字であることを指定する。
j	キーが文字 (JIS8 単位コード※) であることを指定する。
n	キーが符号付き固定小数点 2 進数であることを指定する。

## 11. 拡張 SORT コマンドの使い方 (Windows の場合)

オペランド	内容
m	キーが符号なし 2 進数であることを指定する。
p	キーが内部 10 進数であることを指定する。
z	キーが外部 10 進数であることを指定する。
x	キーが左符号付き外部 10 進数であることを指定する。
s	キーが右分離符号付き外部 10 進数であることを指定する。
t	キーが左分離符号付き外部 10 進数であることを指定する。
f	キーが内部浮動小数点であることを指定する。
e	キーが日付 (英数字) であることを指定する。

### 注※

ASCII コード (0x00 ~ 0x7F) を含む, JIS8 単位コード (0x00 ~ 0xFF) を意味します。入力データに日本語などのマルチバイト文字が含まれている場合, マルチバイト文字として扱わず, シングルバイト文字として扱います。EBCDIK コードの体系に従ってキー・項目を比較する「v オプション (Windows は /v オプション)」を使用できます。

「r」を指定しないときは, キーの昇順でソートされます。また, ソートキーにするデータの属性を「c, j, n, m, p, z, x, s, t, f, e」のパラメタのどれかで必ず指定してください。降順でソートする場合は, 「rc」のようにつけて指定します。なお, 機能オプションに「v」を指定した場合, キーの属性を必ず「j」で指定してください。

### (b) + キー位置 - キー長

キーの開始バイト位置を 5 桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置 (先頭を 0 とする) とします。開始バイト位置に指定できる値の範囲は, 「0 ~ レコード長 - 1」です。

また, キーのバイト長を「1 ~ 4,096」の範囲で指定します。

## (13) sum\_key= 集約キー属性 + キー位置 - キー長

レコードを集約するための集約キーを指定します。この指定は, 実行オプションで「sum」を指定した場合だけ有効となります。

複数指定する場合は, 間をコンマ (,) で区切って指定します。

また, この指定は, 「sort\_key= ソートキー属性 + キー位置 - キー長」を指定した場合は省略できます。指定を省略した場合は「sort\_key=」と同じ値が設定されます。どちらの指定もない場合はエラーとなります。

### (a) キー属性

集約キーのキー属性を「sum\_key=」 + オペランドの形式で指定します。キー属性のオペランドを次の表に示します。

表 11-13 集約キー属性のオペランド (拡張 SORT コマンド)

オペランド	内容
c	キーが文字であることを指定する。
j	キーが文字 (JIS8 単位コード <sup>※</sup> ) であることを指定する。
n	キーが符号付き固定小数点 2 進数であることを指定する。
m	キーが符号なし 2 進数であることを指定する。
p	キーが内部 10 進数であることを指定する。
z	キーが外部 10 進数であることを指定する。
x	キーが左符号付き外部 10 進数であることを指定する。
s	キーが右分離符号付き外部 10 進数であることを指定する。
t	キーが左分離符号付き外部 10 進数であることを指定する。
f	キーが内部浮動小数点であることを指定する。
e	キーが日付 (英数字) であることを指定する。

## 注※

ASCII コード (0x00 ~ 0x7F) を含む, JIS8 単位コード (0x00 ~ 0xFF) を意味します。入力データに日本語などのマルチバイト文字が含まれている場合, マルチバイト文字として扱わず, シングルバイト文字として扱います。EBCDIK コードの体系に従ってキー・項目を比較する「-v オプション (Windows は /v オプション)」を使用できます。

## (b) + キー位置 - キー長

キーの開始バイト位置を 5 桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置 (先頭を 0 とする) とします。開始バイト位置に指定できる値の範囲は, 「0 ~ レコード長 -1」です。

また, キーのバイト長を 「1 ~ 4,096」 の範囲で指定します。

## (14) sum\_field= 集約項目の属性 + 項目位置 - 項目長

集約する項目を指定します。複数指定する場合は, 間をコンマ (,) で区切って指定します。

無条件レコード抽出機能を使用する場合は, 指定しません。無条件レコード抽出機能については, 「2.4 無条件レコード抽出機能」を参照してください。

## (a) 集約項目属性

集約項目の属性を 「sum\_field=」 + オペランドの形式で指定します。集約項目の属性のオペランドを次の表に示します。

表 11-14 集約項目の属性のオペランド (拡張 SORT コマンド)

オペランド	内容
n	集約項目が符号付き固定小数点 2 進数であることを指定する。
m	集約項目が符号なし 2 進数であることを指定する。
p	集約項目が内部 10 進数であることを指定する。
z	集約項目が外部 10 進数であることを指定する。
x	集約項目が左符号付き外部 10 進数であることを指定する。
s	集約項目が右分離符号付き外部 10 進数であることを指定する。
t	集約項目が左分離符号付き外部 10 進数であることを指定する。

**(b) + 項目位置 - 項目長**

集約項目の開始バイト位置を 5 桁以内の数字で指定します。開始バイト位置はレコードの先頭からのバイト位置 (先頭を 0 とする) とします。開始バイト位置に指定できる値の範囲は、「0 ～レコード長 -1」です。

また、集約項目のバイト長を「1 ～ 19」の範囲で指定します。

**(15) output\_file= 出力ファイル名**

SORT の出力先のファイル名を指定します。この指定は省略できます。省略した場合は、標準出力ファイルに出力します。

**(16) input\_file= 入力ファイル名**

SORT の入力ファイル名を指定します。複数指定する場合は、間を「|」で区切って指定します。

また、この指定は省略できます。省略した場合は、標準入力ファイルが設定されます。

実行オプションで `select` または `sum` を単独で指定した場合、出力ファイル名と同じ指定はできません。

## 11.3 コマンド指定時の注意事項

---

### 11.3.1 パラメタファイル作成時の注意事項

パラメタファイルを作成するときの注意事項を次に示します。

#### (1) 同一パラメタの複数指定

同じパラメタを複数指定した場合は、最初に指定したオペランドが有効になります。

#### (2) 改行文字

パラメタファイルを作成する場合、各パラメタの最後に改行文字を付けてください。ただし、最後のパラメタには、改行文字の代わりに EOF (**E**nd **O**f **F**ile) コードを付けることもできます。

## 11.4 使用例

拡張 SORT コマンドの使用例を示します。

### 11.4.1 レコードを選択したあとにソートする例

選択したレコードをソートする例を示します。この例では、複数の社員データファイルから、営業部の社員レコードを選択し、社員番号順にソートします。

#### (1) 条件

条件は、次のとおりです。

- 32 ビット版 SORT を使用する
- キーの比較は、JIS8 単位コードの体系に従う
- 入力ファイルおよび出力ファイルは、固定長順編成ファイル
- 入力ファイルのレコード長は、320 バイト
- 比較項目の属性は文字項目 (営業部の部署コードを「D」とする)
- 比較項目の開始バイト位置は「6」バイト目、バイト長は「1」バイト
- 条件値は「D」
- 比較演算子は「=」
- ソートは昇順、キーの属性は文字項目 (英数字)
- キーの開始バイト位置は「0」バイト目、バイト長は「5」バイト
- 結果の出力ファイル名は「OUTFILE」
- 入力ファイル名は「INFILE1」、「INFILE2」

#### (2) パラメタファイルの形式

「wmsortex」コマンドに続けて指定するパラメタファイルの形式を次の図に示します。

図 11-5 パラメタファイルの形式 (その 1)

```
[WMSORTEX]
exec_opt=select:sort
file_org=s
record_leng=320
select_exp=c+6-1△=△c'D'
sort_key=c+0-5
output_file=OUTFILE
input_file=INFILE1|INFILE2
```

(凡例) △ : スペース (空白) を示す。

#### (3) パラメタの指定内容

パラメタの指定内容を次の表に示します。



表 11-15 パラメタの指定内容 (その 1)

入力項目	指定内容
[WNSORTEX]	パラメタファイルの先頭である予約語を指定する。
exec_opt=select:sort	選択機能およびソート機能を使用することを指定する。
file_org=s	入力ファイルに「固定長順編成ファイル」を指定する。
record_leng=320	入力ファイルおよび出力ファイルのレコード長を「320」バイトと指定する。
select_exp=	比較条件式を示す予約語を指定する。
c+6-1	比較項目の属性を「文字 (英数字)」, 比較項目の開始バイト位置を「6」バイト目, バイト長を「1」バイトと指定する。
=	比較演算子を「条件値と等しい」と指定する。
c'D'	条件値に文字項目「D」を指定する。
sort_key=	ソートキーを示す予約語を指定する。
c+0-5	キーの属性を「文字 (英数字)」, キーの開始バイト位置を「0」バイト目, キー長を「5」バイトと指定する。
output_file=OUTFILE	結果の出力ファイル名を「OUTFILE」と指定する。
input_file=INFILE1 INFILE2	入力ファイル名を「INFILE1」, 「INFILE2」と指定する。

注 1

キーおよび項目の比較は JIS8 単位コード体系に従うので、機能オプションの指定を省略する。

注 2

出力ファイルのファイル編成は入力ファイルと同じなので、指定を省略する。

## 11.4.2 レコードをソートしたあとに集約する例

ソートしたファイルのレコードを集約する例を示します。ここでは、支店コードをキーとして、売上金額を集約する例を示します。

### (1) 条件

条件は、次のとおりです。

- 32 ビット版 SORT を使用する
- キーの比較は JIS8 単位コードの体系に従う
- 重複レコードの順序は入力順
- 入力ファイルおよび出力ファイルは、可変長順編成ファイル
- ソートは昇順, キーの属性は文字項目 (英数字)
- ソートキーの開始バイト位置は「3」バイト目, バイト長は「6」バイト
- 集約キーの属性は文字項目 (英数字)
- 集約キーの開始バイト位置は「3」バイト目, バイト長は「6」バイト

## 11. 拡張 SORT コマンドの使い方 (Windows の場合)

- 集約項目の属性は数字項目 (内部 10 進数)
- 集約項目の開始バイト位置は「12」バイト目, バイト長は「8」バイト
- 結果の出力ファイル名は「OUTFILE」
- 入力ファイル名は「INFILE」

### (2) パラメタファイルの形式

「wmsortex」コマンドに続けて指定するパラメタファイルの形式を次の図に示します。

図 11-6 パラメタファイルの形式 (その 2)

```
[WMSORTEX]
exec_opt=sort:sum
sort_opt=d
file_org=v
sort_key=c+3-6
sum_field=p+12-8
output_file=OUTFILE
input_file=INFILE
```

### (3) パラメタの指定内容

パラメタの指定内容を次の表に示します。

表 11-16 パラメタの指定内容 (その 2)

入力項目	指定内容
[WMSORTEX]	パラメタファイルの先頭である予約語を指定する。
exec_opt=sort:sum	ソート機能および集約機能を使用することを指定する。
sort_opt=d	重複レコードの順序は「入力順」と指定する。
file_org=v	入力ファイルに「可変長順編成ファイル」を指定する。
sort_key=c+3-6	ソートキーの属性を「文字 (英数字)」, ソートキーの開始バイト位置を「3」バイト目, バイト長を「6」バイトと指定する。
sum_field=p+12-8	集約項目の属性を「内部 10 進項目」, 集約項目の開始バイト位置を「12」バイト目, バイト長を「8」バイトと指定する。
output_file=OUTFILE	結果の出力ファイル名を「OUTFILE」と指定する。
input_file=INFILE	入力ファイル名を「INFILE」と指定する。

注 1

キーおよび項目の比較は JIS8 単位コード体系に従うので, 機能オプションの指定を省略する。

注 2

出力ファイルのファイル編成は入力ファイルと同じなので, 指定を省略する。

注 3

集約キーはソートキーと同じなので, 指定を省略する。

# 12 シェル・スクリプトコマンドの使い方（UNIX の場合）

幾つかのコマンドをファイルに格納しておき，コマンドとして実行する機能としてシェル・スクリプトがあります。この機能を使用することによって，コマンドの組み合わせなどによる複雑な処理を簡単に指定できます。

この章では，ソート機能と集約機能を組み合わせたコマンドについて説明します。

---

12.1 コマンドの使い方

---

12.2 オペランドの意味

---

12.3 コマンド指定時の注意事項

---

12.4 使用例

---

## 12.1 コマンドの使い方

32 ビット版の場合、`rsortsum` コマンドは、`rsort` コマンドと `rsum` コマンドを組み合わせたシェル・スクリプトです。

64 ビット版の場合、`rsortsum64` コマンドは、`rsort64` コマンドと `rsum64` コマンドを組み合わせたシェル・スクリプトです。

ソート結果を集約する場合、ソート機能のソートキーと集約機能の集約キーが同じキー属性、位置、および長さのとき、コマンドを一度に指定できます。

### 12.1.1 コマンドの形式

`rsortsum` (`rsortsum64`) コマンドは、コマンド名 (`rsortsum` または `rsortsum64`) とそのあとに続く各オペランドで構成されます。`rsortsum` (`rsortsum64`) コマンドの形式を次の図に示します。

図 12-1 `rsortsum` (`rsortsum64`) コマンドの形式

```
rsortsum*1Δ [機能オプション]
  Δ 入力ファイルのファイル編成 [ : 出力ファイルのファイル編成 ] ※2
  Δ [-Lレコード長]
  Δ [-o出力ファイル名]
  Δ [-yワークバッファサイズ]
  Δ [-Tソート用一時ファイルのディレクトリ名]
  Δ [-T2マージ用一時ファイルのディレクトリ名2]
  Δ [-w中間ファイルのディレクトリ名]
  Δ { キー属性Δ+キー位置-キー長 } . . .
  Δ -a [集約項目情報]
  Δ { [入力ファイル名] } . . .
```

(凡例) Δ : スペース (空白) を示す。  
 [ ] : 省略できることを示す。  
 . . . : 直前の { } で囲まれた部分を複数繰り返して指定できることを示す。

注※1 使用しているSORTが64ビット版対応の場合は、「`rsortsum64`」となる。

注※2 入力ファイルが索引ファイルの場合、出力ファイル編成の指定は省略できない。

## 12.2 オペランドの意味

---

オペランドには、必ず指定するオペランドと任意に指定できるオペランドの 2 種類があります。

### 12.2.1 機能オプション

ソートオプションおよび集約オプションを指定します。ソートオプションについては、「7.2.2 (1) ソートオプション」を、集約オプションについては、「10.2.2 (1) 集約オプション」を参照してください。

### 12.2.2 入力ファイルのファイル編成

入力ファイルのファイル編成を指定します。指定方法については、「7.2.2 (2) 入力ファイルのファイル編成」を参照してください。

この指定は、省略できません。必ず指定してください。

### 12.2.3 出力ファイルのファイル編成

出力ファイルのファイル編成を指定します。指定方法については、「7.2.2 (3) 出力ファイルのファイル編成」を参照してください。

### 12.2.4 -L レコード長

入力ファイルに固定長ファイルを指定したとき、そのファイルのレコード長を指定します。指定方法については、「10.2.2 (4) -L レコード長」を参照してください。

### 12.2.5 -o 出力ファイル名

集約結果の出力先ファイルをファイル名で指定します。指定方法については、「10.2.2 (5) -o 出力ファイル名」を参照してください。

### 12.2.6 -y ワークバッファサイズ

ワークバッファとして使用するメモリの容量を KB 単位で指定します。指定できる値については、「7.2.2 (6) -y ワークバッファサイズ」を参照してください。

### 12.2.7 -T ソート用一時ファイルのディレクトリ名

ソート用一時ファイルを割り当てるディレクトリを指定します。指定方法については、「7.2.2 (7) -T ディレクトリ名」を参照してください。

## 12.2.8 -T2 マージ用一時ファイルのディレクトリ名 2

マージ用一時ファイルを割り当てる場合に、使用するディレクトリを指定します。指定方法については、「7.2.2 (8) -T2 ディレクトリ名 2」を参照してください。

## 12.2.9 -w 中間ファイルのディレクトリ

`rsortsum` (`rsortsum64`) コマンドが使用する中間ファイルを割り当てるディレクトリを指定します。`rsortsum` (`rsortsum64`) コマンドは、このディレクトリにソートの出力ファイルを一時的に作成します。

この指定は、省略できます。指定を省略した場合に使用するディレクトリの優先順位を次に示します。

1. 環境変数 `TMPDIR` に設定されているディレクトリ
2. `-w` パラメタに設定されているディレクトリ
3. 環境変数 `SORTOPT` の `-T` パラメタに設定されているディレクトリ
4. 環境変数 `SORTOPT` の `-T2` パラメタに設定されているディレクトリ
5. `/tmp`

### 環境変数 `TMPDIR` の指定

環境変数 `TMPDIR` を指定した場合、`-T` パラメタ、`-T2` パラメタ、`-W` パラメタおよび環境変数 `SORTOPT` で指定したディレクトリ名は無効となり、`TMPDIR` で指定したディレクトリに一時ファイルが割り当てられます。

## 12.2.10 キー属性

ソートキーの属性を指定します。キー属性のパラメタについては、「7.2.2 (9) キー属性」を参照してください。

## 12.2.11 + キー位置 - キー長

キーの開始バイト位置を 5 桁以内の数字で指定します。指定方法については、「7.2.2 (10) + キー位置 - キー長」を参照してください。

## 12.2.12 -a 集約項目情報

集約項目の情報を指定します。集約項目の情報のパラメタについては、「10.2.2 (8) -a 集約項目情報」を参照してください。

## 12.2.13 入力ファイル名

ソートの入力ファイルを指定します。指定方法については、「7.2.2 (11) 入力ファイル

名」を参照してください。

## 12.3 コマンド指定時の注意事項

---

シェル・スクリプトコマンドの指定時の注意事項を次に示します。

### 12.3.1 スクリプト・ファイル

`rsortsum` (`rsortsum64`) コマンドはシェル・スクリプトなので、スクリプト・ファイルを変更し処理内容を変更できます。ただし、変更する場合には、シェル・スクリプトのコマンドを理解した上で実行するようにしてください。

また、`SORT` を上書きインストールすると、`rsortsum` (`rsortsum64`) コマンドも上書きされます。`rsortsum` (`rsortsum64`) コマンドの処理内容を変更した場合は、上書きインストールする前に `rsortsum` (`rsortsum64`) コマンドのバックアップを取得してください。

### 12.3.2 中間ファイルの削除

コマンドの実行時、`-w` パラメタで指定したディレクトリ下に「`rsort`」+「プロセス番号」の名称で中間ファイルが生成されます。中間ファイルは、コマンドが正常に終了した場合、自動的に削除されますが、異常終了した場合には削除されないことがあります。その場合、`rm` コマンドで削除してください。

### 12.3.3 中間ファイルの容量

中間ファイルが生成されるときは、入力ファイルと同じだけの容量が必要です。

### 12.3.4 中間ファイルのエラー

中間ファイルの入出力時にエラーが発生した場合、中間ファイルのエラーメッセージとしては表示されません。出力されるメッセージは、実行中のコマンドによって、次のようになります。

#### ● ソートコマンド実行中

中間ファイルのエラーメッセージは、出力ファイルに対するエラーメッセージとして表示されます。

#### ● 集約コマンド実行中

中間ファイルのエラーメッセージは、入力ファイルに対するエラーメッセージとして表示されます。



## 12.4 使用例

rsortsum コマンドの使用例を示します。

### 12.4.1 文字項目をキーとしてソートし集約する例

文字項目をキーとしてソートし集約する例を示します。ここでは、商品コードをキーとして売上数を集約する例を示します。

#### (1) 条件

ソートし集約する条件は次のとおりです。

- 32 ビット版 SORT を使用する
- ソートキーの比較は JIS8 単位コードの体系に従う
- 入力ファイルおよび出力ファイルは、固定長順ファイル
- 入力ファイルおよび出力ファイルのレコード長は、128 バイト
- ソートは昇順、キーの属性は文字項目（文字）
- ソートキー（商品コード）の開始バイト位置は「5」バイト目、バイト長は「4」バイト
- ソートし集約するファイルのファイル名は「INFILE」
- ソートし集約した結果の出力ファイル名は「OUTFILE」
- 集約項目（売上数）の属性は数字項目（内部 10 進）
- 集約項目（売上数）の開始バイト位置は「17」バイト目、バイト長は「4」バイト
- 中間ファイルのディレクトリとして「/sort/tmp」を使用する

#### (2) 入力するコマンド

入力する rsortsum コマンドを次の図に示します。

図 12-2 入力する rsortsum コマンド

```
rsortsum△-S△-L128△-c△+5-4△-ap△+17-4△-w/sort/tmp  
△-oOUTFILE△INFILE
```

（凡例） △：スペース（空白）を示す。

#### (3) コマンドの指定内容

rsortsum コマンドの指定内容を次の表に示します。

## 12. シェル・スクリプトコマンドの使い方（UNIX の場合）

表 12-1 rsortsum コマンドの指定内容

項番	入力項目	指定内容
1	rsortsum	ソートし集約する処理のコマンド「rsortsum」を指定する。
2	-S	入力ファイルおよび出力ファイルを「固定長順ファイル」で指定する。
3	-L128	入力ファイルおよび出力ファイルのレコード長を「128」バイトで指定する。
4	-c	ソートキーの属性を文字項目の「文字」で指定する。
5	+5-4	ソートキーの開始バイト位置を「5」バイト目、バイト長を「4」バイトで指定する。
6	-ap	集約項目（売上数）の属性を数字項目の「内部 10 進」で指定する。
7	+17-4	集約項目（売上数）の開始バイト位置を「17」バイト目、バイト長を「4」バイトで指定する。
8	-w/sort/tmp	rsortsum コマンドで使用する中間ファイルのディレクトリを「/sort/tmp」に指定する。
9	-oOUTFILE	ソートし集約する結果の出力先ファイル名「OUTFILE」を指定する。
10	INFILE	ソートし集約するファイルのファイル名「INFILE」を指定する。

### 注 1

ソートキーの比較は JIS8 単位コード体系に従うので、指定を省略します。

### 注 2

出力ファイルのファイル編成は、入力ファイルと同じなので、指定を省略します。

## (4) rsortsum コマンドを使用しない場合

rsortsum コマンドを使用しないで実行する場合のコマンドの例を次の図に示します。

図 12-3 rsortsum コマンドを使用しない場合の例

```
rsort△-S△-L128△-c△+5-4△ INFILE△-o/sort/tmp/WORKFILE  
rsum△-S△-L128△-c△+5-4△-ap△+17-4△/sort/tmp/WORKFILE  
△-oOUTFILE  
rm△/sort/tmp/WORKFILE
```

（凡例） △：スペース（空白）を示す。

# 13 C 言語プログラムでの使い方 (Windows の場合)

この章では、C 言語で作成したプログラムから SORT を呼び出すためのライブラリ関数について説明します。

---

13.1 関数の機能と形式

---

13.2 コーディング例

---

13.3 コンパイル方法

---

13.4 詳細エラーコード

---

## 13.1 関数の機能と形式

関数を使用して、C 言語で作成したプログラムから SORT を呼び出して実行できます。

### 13.1.1 関数の一覧

関数の一覧を、次の表に示します。

表 13-1 関数一覧

種類	関数名	機能
テーブル管理関数	WNS_Create	ハンドル管理テーブルを確保し、初期化する
	WNS_Release	ハンドル管理テーブルを解放する
ファイル情報関数	WNS_File_Create	ファイル情報領域を確保し、初期化する
	WNS_File_Set	ファイル情報を設定する
	WNS_File_Init	ファイル情報領域を初期化する
	WNS_File_Release	ファイル情報領域を解放する
キー情報関数	WNS_Key_Create	キー情報領域を確保し、初期化する
	WNS_Key_Set	キー情報を設定する
	WNS_Key_Init	キー情報領域を初期化する
	WNS_Key_Release	キー情報領域を解放する
選択条件式関数	WNS_Sel_Create	選択条件式領域を確保し、初期化する
	WNS_Sel_Set	選択条件式情報を設定する
	WNS_Sel_Init	選択条件式領域を初期化する
	WNS_Sel_Release	選択条件式領域を解放する
集約項目関数	WNS_Sum_Create	集約項目領域を確保し、初期化する
	WNS_Sum_Set	集約項目情報を設定する
	WNS_Sum_Init	集約項目領域を初期化する
	WNS_Sum_Release	集約項目領域を解放する
実行関数	WNS_Sort	ソート機能を実行する
	WNS_Select	選択機能を実行する
	WNS_Sum	集約機能を実行する
	WNS_SortEX	拡張 SORT 機能を実行する
エラー取得関数	WNS_GetError	詳細エラー情報を取得する
	WNS_GetMsg	エラーメッセージテキストを取得する
環境設定関数	WNS_Date_Set	日付項目使用時の基準年度を設定する
	WNS_Work_Set	ソート実行用の拡張一時ファイル情報を設定する

13. C言語プログラムでの使い方 (Windows の場合)

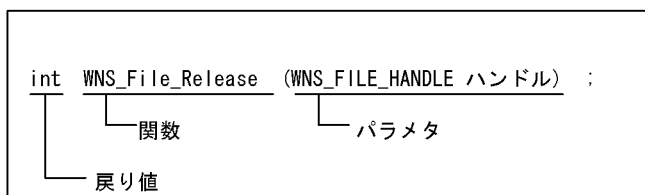
それぞれの関数を使用する順序は任意ですが、前提となる関数があります。それぞれの関数と、その前提となる関数の組み合わせを、次の表に示します。

関数	前提となる関数																											
	WNS_Create	WNS_Release	WNS_File_Create	WNS_File_Set	WNS_File_Init	WNS_File_Release	WNS_Key_Create	WNS_Key_Set	WNS_Key_Init	WNS_Key_Release	WNS_Sel_Create	WNS_Sel_Set	WNS_Sel_Init	WNS_Sel_Release	WNS_Sum_Create	WNS_Sum_Set	WNS_Sum_Init	WNS_Sum_Release	WNS_Sort	WNS_Select	WNS_Sum	WNS_SortEX	WNS_GetError	WNS_GetMsg	WNS_Date_Set	WNS_Work_Set		
WNS_Create																												
WNS_Release	○																											
WNS_File_Create	○																											
WNS_File_Set	○	○																										
WNS_File_Init	○	○																										
WNS_File_Release	○	○																										
WNS_Key_Create	○																											
WNS_Key_Set	○						○																					
WNS_Key_Init	○						○																					
WNS_Key_Release	○						○																					
WNS_Sel_Create	○																											
WNS_Sel_Set	○										○																	
WNS_Sel_Init	○										○																	
WNS_Sel_Release	○										○																	
WNS_Sum_Create	○	○	○				○	○																				
WNS_Sum_Set	○	○	○								○	○																
WNS_Sum_Init	○	○	○				○	○																				
WNS_Sum_Release	○																											
WNS_Sort	○																		△	△	△	△						
WNS_Select	○																		△	△	△	△	○					
WNS_Sum	○																											
WNS_SortEX	○																						○					
WNS_GetError	○																						○					
WNS_GetMsg	○																						○					
WNS_Date_Set	○																											
WNS_Work_Set	○																											

(凡例) ○ : 必要  
△ : どれか一つ必要

## 13.1.2 関数の詳細

このマニュアルでは、次に示す形式で関数を記載します。



なお、各関数での指定内容の詳細は、各機能のコマンドの使い方、および「3.3 指定できる値の範囲」を参照してください。

また、詳細エラーコードについては、「13.4 詳細エラーコード」を参照してください。

### (1) WNS\_Create

#### 形式

```
int WNS_Create (int ハンドル数);
```

#### 機能

ハンドル管理テーブルを確保し、初期化します。この関数で確保されたハンドル管理テーブルは、WNS\_Release が呼び出されるまで有効です。

#### 戻り値

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

#### パラメタ

int ハンドル数

ハンドル管理テーブルの総数 (エラーハンドルを含め、Create 関数で同時に取得できるハンドル数) を「16 ~ 65,535」の範囲で指定します。ハンドル管理テーブルは、Create 関数で使用され、Release 関数で解放されます。指定を省略した場合は「16」が設定されます。

### (2) WNS\_File\_Create

#### 形式

```
WNS_FILE_HANDLE WNS_File_Create
(int ファイル数,
WNS_FILEORG ファイル編成,
int ファイルレコード長,
WNS_FILEOPT オプション情報);
```

#### 機能

ファイル情報領域を確保して初期化します。この関数で確保されたファイル情報領域は、WNS\_File\_Release または WNS\_Release が呼び出されるまで有効です。

#### 戻り値

##### WNS\_FILE\_HANDLE

確保したファイル情報領域のハンドルです。確保できなかった場合は、「0」を返します。

#### パラメタ

##### int ファイル数

ファイル数を「1～256」の範囲で指定します。ただし、出力ファイル情報の場合は「1」を指定します。

##### WNS\_FILEORG ファイル編成

ファイルのファイル編成を指定します。

- WNS\_FIX\_SAM : 固定長順編成ファイル
- WNS\_VAR\_SAM : 可変長順編成ファイル
- WNS\_FIX\_RSAM : 固定長相対ファイル
- WNS\_VAR\_RSAM : 可変長相対ファイル
- WNS\_FIX\_ISAM : 固定長索引ファイル (入力ファイルのときだけ有効)
- WNS\_VAR\_ISAM : 可変長索引ファイル (入力ファイルのときだけ有効)
- WNS\_TEXT : テキストファイル

##### int ファイルレコード長

固定長順編成ファイルまたはテキストファイルの場合、レコード長 (テキストファイルの場合、最大レコード長) を「1～65,535」の範囲で指定します。それ以外のファイルの場合は指定を省略できます。

##### WNS\_FILEOPT オプション情報

ファイルに対するオプション情報を指定します。

- WNS\_ORDER : 重複レコードの入力順序保存機能を指定 (入力ファイルのときだけ有効)
- WNS\_NOTRW : 出力ファイルの上書き禁止を指定 (出力ファイルのときだけ有効)

### (3) WNS\_File\_Init

#### 形式

int WNS\_File\_Init (WNS\_FILE\_HANDLE ファイルハンドル);

#### 機能

ファイル情報領域を、WNS\_File\_Create 実行直後の状態に戻します (ファイル情報領域の初期化)。この関数で初期化されたファイル情報領域は、WNS\_File\_Release または WNS\_Release が呼び出されるまで有効です。

## 13. C 言語プログラムでの使い方 (Windows の場合)

### 戻り値

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

### パラメタ

WNS\_FILE\_HANDLE ファイルハンドル

ファイル情報領域のハンドルを指定します。

## (4) WNS\_File\_Release

### 形式

int WNS\_File\_Release (WNS\_FILE\_HANDLE ファイルハンドル);

### 機能

ファイル情報領域を解放します。

### 戻り値

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

### パラメタ

WNS\_FILE\_HANDLE ファイルハンドル

ファイル情報領域のハンドルを指定します。

## (5) WNS\_File\_Set

### 形式

int WNS\_File\_Set (WNS\_FILE\_HANDLE ファイルハンドル,  
WNS\_FILENAME ファイル名);

### 機能

ファイル情報領域にファイル名を指定します。この関数を複数呼び出した場合は、すべてのファイル名が有効です。ただしファイル数は、WNS\_File\_Create で設定したファイル数より多く設定できません。

この関数で設定したファイル名は、WNS\_File\_Init, WNS\_File\_Release または WNS\_Release が呼び出されるまで有効です。

### 戻り値

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

### パラメタ



**WNS\_FILE\_HANDLE** ファイルハンドル

ファイル情報領域のハンドルを指定します。

**WNS\_FILENAME** ファイル名

ファイル名を設定した文字列バッファポインタを指定します。

## (6) WNS\_GetError

### 形式

```
int WNS_GetError (WNS_ERR_HANDLE エラーハンドル,
                 WNS_ERR_INF エラー情報テーブルポインタ);
```

### 機能

WNS\_Sort, WNS\_Sum, WNS\_Select および WNS\_SortEX でエラーとなった場合の詳細エラー情報テーブルを取得します。

この関数を実行したあと、使用したエラーハンドルは削除されます。

### 戻り値

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

### パラメタ

**WNS\_ERR\_HANDLE** エラーハンドル

エラー発生時のエラーハンドルを指定します。

**WNS\_ERR\_INF** エラー情報テーブルポインタ

詳細エラー情報を設定するためのエラー情報テーブルへのポインタを指定します。エラー情報テーブルを次に示します。

```
typedef
struct WNS_ERR_INFORMATION{
    int          WNS_ERR_CODE   : /* エラーコード          */
    int          WNS_ERR_CODE2  : /* エラー詳細コード     */
    int          WNS_ERR_FILE   : /* エラー発生ファイル番号 */
    int          WNS_ERR_RECORD  : /* エラー発生レコード番号 */
    int          WNS_ERR_KEY     : /* エラー発生キー番号    */
    char         WNS_ERR_DMY[12]: /* 予備                  */
} A_ERR_INFORMATION;
```

## (7) WNS\_GetMsg

### 形式

```
int WNS_GetMsg (WNS_ERR_INF エラー情報テーブルポインタ,
                void メッセージ領域);
```

### 機能

WNS\_GetError で取得したエラー情報テーブルを基に、エラーメッセージテキスト

### 13. C 言語プログラムでの使い方 (Windows の場合)

を取得します。

#### 戻り値

int

エラーメッセージテキストの長さを返します。エラーコードに該当するメッセージテキストがない場合は「0」を返します。

#### パラメタ

WNS\_ERR\_INF エラー情報テーブルポインタ

WNS\_GetError で取得した詳細エラー情報テーブルへのポインタを指定します。

void メッセージ領域

メッセージテキストを設定するバッファ (256 バイト以上) へのポインタを指定します。

## (8) WNS\_Key\_Create

#### 形式

```
WNS_KEY_HANDLE WNS_Key_Create (int キー数,  
                                WNS_KEYOPT オプション情報);
```

#### 機能

キー情報領域を確保して初期化します。

この関数で確保されたキー情報領域は、WNS\_Key\_Release または WNS\_Release が呼び出されるまで有効です。

#### 戻り値

WNS\_KEY\_HANDLE

確保したキー情報領域のハンドルです。確保できなかった場合は「0」を返します。

#### パラメタ

int キー数

ソートキーおよび集約キー項目の数を「1～64」の範囲で指定します。

WNS\_KEYOPT オプション情報

ソートキーおよび集約キーに対するオプション情報を指定します。

WNS\_BIGENDIAN: 数値項目が big endian

## (9) WNS\_Key\_Init

#### 形式

```
int WNS_Key_Init (WNS_KEY_HANDLE キーハンドル);
```

#### 機能

キー情報領域を、WNS\_Key\_Create 実行直後の状態に戻します (キー情報領域の初期化)。

この関数で初期化されたキー情報領域は、WNS\_Key\_Release または WNS\_Release が呼び出されるまで有効です。

#### 戻り値

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

#### パラメタ

WNS\_KEY\_HANDLE キーハンドル

キー情報領域のハンドルを指定します。

### (10) WNS\_Key\_Release

#### 形式

```
int WNS_Key_Release (WNS_KEY_HANDLE キーハンドル);
```

#### 機能

キー情報領域を解放します。

#### 戻り値

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

#### パラメタ

WNS\_KEY\_HANDLE キーハンドル

キー情報領域のハンドルを指定します。

### (11) WNS\_Key\_Set

#### 形式

```
int WNS_Key_Set (WNS_KEY_HANDLE キーハンドル,  
                WNS_KEYATTR キー属性,  
                int キー位置,  
                int キー長,  
                WNS_KEYORDER キー並び);
```

#### 機能

キー情報領域にソートキーおよび集約キーの並び、属性、位置および長さを設定します。この関数を複数回呼び出した場合は、すべてのキーが有効です。ただし、キーの数は、WNS\_Key\_Create で設定したキー数より多く設定できません。

この関数で指定したソートキーおよび集約キーの情報は、WNS\_Key\_Init、WNS\_Key\_Release または WNS\_Release が呼び出されるまで有効です。

## 13. C 言語プログラムでの使い方 (Windows の場合)

### 戻り値

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

### パラメタ

**WNS\_KEY\_HANDLE** キーハンドル

キー情報領域のハンドルを指定します。

**WNS\_KEYATTR** キー属性

ソートキーおよび集約キーの属性を指定します。

- **WNS\_CHAR** : 文字項目
- **WNS\_SIGN\_BIN** : 符号付き 2 進項目
- **WNS\_UNSIGN\_BIN** : 符号なし 2 進項目
- **WNS\_PACK\_DEC** : 内部 10 進項目
- **WNS\_ZONE\_DEC** : 外部 10 進項目
- **WNS\_L\_ZONE\_DEC** : 左符号外部 10 進項目
- **WNS\_R\_SEP\_DEC** : 右分離符号外部 10 進項目
- **WNS\_L\_SEP\_DEC** : 左分離符号外部 10 進項目
- **WNS\_FLOAT** : 内部浮動小数点項目
- **WNS\_EBCDIK** : 文字 (EBCDIK 順) 項目
- **WNS\_DATE** : 日付 (英数字) 項目

int キー位置

ソートキーおよび集約キー項目のバイト位置を「0 ~ 65,534」の範囲で指定します。

int キー長

ソートキーおよび集約キーのバイト長を「1 ~ 4,096」の範囲で指定します。

**WNS\_KEYORDER** キー並び

昇順, 降順のどちらでソートするかを指定します。

- **WNS\_ASC** : 昇順ソート
- **WNS\_DESC** : 降順ソート

## (12) WNS\_Release

### 形式

```
WNS_Release ();
```

### 機能

**WNS\_Create** で確保したハンドル管理テーブルを解放します。

この関数を呼び出したあとは、**WNS\_Create** が呼び出されるまで、すべての関数が無効になります。

戻り値

なし。

パラメタ

なし。

### (13) WNS\_Sel\_Create

形式

```
WNS_SEL_HANDLE WNS_Sel_Create (int 選択条件式数,
                                WNS_SELOPT オプション情報);
```

機能

選択条件式の情報領域を確保して初期化します。

この関数で確保された選択条件式の情報領域は、WNS\_Sel\_Release または WNS\_Release が呼び出されるまで有効です。

戻り値

WNS\_SEL\_HANDLE

確保した選択条件式情報の領域のハンドルです。確保できなかった場合は「0」を返します。

パラメタ

int 選択条件式数

選択条件式の数を「1～64」の範囲で指定します。

WNS\_SELOPT オプション情報

選択条件式に対するオプション情報を指定します。

- WNS\_NOT\_SELECT : 選択条件式を満たさないデータを入力
- WNS\_BIGENDIAN : 数値項目が big endian

### (14) WNS\_Select

形式

```
WNS_ERR_HANDLE WNS_Select (WNS_FILE_HANDLE 入力ファイルハンドル,
                            WNS_FILE_HANDLE 出力ファイルハンドル,
                            WNS_SEL_HANDLE 選択条件式ハンドル);
```

機能

選択機能を実行します。

戻り値

WNS\_ERR\_HANDLE

関数が正常に終了した場合は「0」、エラーが発生した場合はそのエラーハンドルを返します。

パラメタ

### 13. C 言語プログラムでの使い方 (Windows の場合)

#### WNS\_FILE\_HANDLE 入力ファイルハンドル

入力ファイル情報領域のハンドルを指定します。

#### WNS\_FILE\_HANDLE 出力ファイルハンドル

出力ファイル情報領域のハンドルを指定します。入力ファイル領域のハンドルと同じ値は指定できません。

#### WNS\_SEL\_HANDLE 選択条件式ハンドル

選択条件式の情報領域のハンドルを指定します。

### (15) WNS\_Sel\_Init

#### 形式

```
int WNS_Sel_Init (WNS_SEL_HANDLE 選択条件式ハンドル);
```

#### 機能

選択条件式の情報領域を、WNS\_Sel\_Create 実行直後の状態に戻します (選択条件式の情報領域の初期化)。

この関数で初期化された選択条件式の情報領域は、WNS\_Sel\_Release または WNS\_Release が呼び出されるまで有効です。

#### 戻り値

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

#### パラメタ

##### WNS\_SEL\_HANDLE 選択条件式ハンドル

選択条件式の領域のハンドルを指定します。

### (16) WNS\_Sel\_Release

#### 形式

```
int WNS_Sel_Release (WNS_SEL_HANDLE 選択条件式ハンドル);
```

#### 機能

選択条件式の情報領域を解放します。

#### 戻り値

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

#### パラメタ

##### WNS\_SEL\_HANDLE 選択条件式ハンドル

選択条件式の領域のハンドルを指定します。

## (17) WNS\_Sel\_Set

## 形式

```
int WNS_Sel_Set (WNS_SEL_HANDLE 選択条件式ハンドル,
                WNS_LOGOPER 論理演算子,
                WNS_CMPATTR 比較項目の属性,
                int 比較項目のバイト位置,
                int 比較項目のバイト長,
                WNS_COMPARE 比較演算子,
                WNS_VALATTR 条件値の属性,
                WNS_VALADDR 条件値エリアのアドレス,
                int 条件値エリアのバイト長);
```

## 機能

選択条件式の情報領域に、選択条件式の内容を設定します。この関数を複数呼び出した場合、すべての選択条件式が有効です。ただし、選択条件式の設定数は、WNS\_Sel\_Create で設定した選択条件式の数より少なくしてください。

この関数で指定した選択条件式の情報には、WNS\_Sel\_Init、WNS\_Sel\_Release または WNS\_Release が呼び出されるまで有効です。

## 戻り値

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

## パラメタ

**WNS\_SEL\_HANDLE 選択条件式ハンドル**

選択条件式領域のハンドルを指定します。

**WNS\_LOGOPER 論理演算子**

選択条件式を接続する論理演算子を指定します。

- WNS\_SL\_NON : なし (先頭の選択条件式にだけ有効)
- WNS\_SL\_AND : AND (論理積)
- WNS\_SL\_OR : OR (論理和)

**WNS\_CMPATTR 比較項目の属性**

比較項目の属性を指定します。

- WNS\_CHAR : 文字項目
- WNS\_SIGN\_BIN : 符号付き 2 進項目
- WNS\_UNSIGN\_BIN : 符号なし 2 進項目
- WNS\_PACK\_DEC : 内部 10 進項目
- WNS\_ZONE\_DEC : 外部 10 進項目
- WNS\_L\_ZONE\_DEC : 左符号外部 10 進項目
- WNS\_R\_SEP\_DEC : 右分離符号外部 10 進項目
- WNS\_L\_SEP\_DEC : 左分離符号外部 10 進項目
- WNS\_FLOAT : 内部浮動小数点項目

### 13. C 言語プログラムでの使い方 (Windows の場合)

- WNS\_EBCDIK : 文字 (EBCDIK 順) 項目
- WNS\_DATE : 日付 (英数字) 項目

#### int 比較項目のバイト位置

比較項目のバイト位置を「0 ~ 65,534」の範囲で指定します。

#### int 比較項目のバイト長

比較項目のバイト長を「1 ~ 4,096」の範囲で指定します。

#### WNS\_COMPARE 比較演算子

比較演算子を指定します。

- WNS\_SL\_EQ : 比較項目 = 条件値
- WNS\_SL\_NE : 比較項目 ≠ 条件値
- WNS\_SL\_GT : 比較項目 > 条件値
- WNS\_SL\_GE : 比較項目 ≥ 条件値
- WNS\_SL\_LT : 比較項目 < 条件値
- WNS\_SL\_LE : 比較項目 ≤ 条件値

#### WNS\_VALATTR 条件値の属性

条件値の属性を設定します。

- WNS\_VAL\_CHAR : 文字条件値
- WNS\_VAL\_NUM : 数値条件値
- WNS\_VAL\_HEX : 16 進数条件値
- WNS\_VAL\_DATE : 日付条件値

#### WNS\_VALADDR 条件値エリアのアドレス

条件値エリアの先頭ポインタを指定します。

#### int 条件値エリアのバイト長

条件値エリアのバイト長を「1 ~ 8,192」の範囲で指定します。

## (18) WNS\_Sort

### 形式

```
WNS_ERR_HANDLE WNS_Sort  
(WNS_FILE_HANDLE 入力ファイルハンドル,  
WNS_FILE_HANDLE 出力ファイルハンドル,  
WNS_KEY_HANDLE ソートキーハンドル,  
int ワークバッファサイズ,  
WNS_WORKDIR ソート用一時ファイルディレクトリ);
```

### 機能

ソート機能を実行します。

### 戻り値

#### WNS\_ERR\_HANDLE

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーハンドルを返します。



**パラメタ**

**WNS\_FILE\_HANDLE 入力ファイルハンドル**  
 入力ファイル領域のハンドルを指定します。

**WNS\_FILE\_HANDLE 出力ファイルハンドル**  
 出力ファイル領域のハンドルを指定します。

**WNS\_KEY\_HANDLE ソートキーハンドル**  
 ソートキー情報領域のハンドルを指定します。

**int ワークバッファサイズ**  
 ソート実行時に使用するワークバッファサイズを KB 単位で、「128 ~ 2,000,000」の範囲で指定します。指定を省略した場合は「1024」が設定されます。

**WNS\_WORKDIR ソート用一時ファイルディレクトリ**  
 ソート用一時ファイルを生成するディレクトリ名を設定した、文字列バッファポインタを指定します。指定を省略した場合は、環境変数「temp」か「tmp」で指定したディレクトリ、またはカレントディレクトリが使用されます。

**(19) WNS\_SortEX****形式**

**WNS\_ERR\_HANDLE WNS\_SortEX (WNS\_PARM\_FILE パラメタファイル名);**

**機能**

拡張 SORT 機能を実行します。

**戻り値**

**WNS\_ERR\_HANDLE**  
 正常に終了した場合は「0」、エラーが発生した場合はそのエラーハンドルを返します。

**パラメタ**

**WNS\_PARM\_FILE パラメタファイル名**  
 パラメタファイル名がある文字列バッファへのポインタを指定します。  
 パラメタファイルについては、「11. 拡張 SORT コマンドの使い方 (Windows の場合)」を参照してください。

**(20) WNS\_Sum****形式**

**WNS\_ERR\_HANDLE WNS\_Sum**  
 (**WNS\_FILE\_HANDLE 入力ファイルハンドル,**  
**WNS\_FILE\_HANDLE 出力ファイルハンドル,**  
**WNS\_KEY\_HANDLE 集約キーハンドル,**  
**WNS\_SUM\_HANDLE 集約項目ハンドル);**

### 13. C 言語プログラムでの使い方 (Windows の場合)

#### 機能

集約機能を実行します。

#### 戻り値

**WNS\_ERR\_HANDLE**

正常に終了した場合は「0」、エラーが発生した場合はそのエラーハンドルを返します。

#### パラメタ

**WNS\_FILE\_HANDLE** 入力ファイルハンドル

入力ファイル情報領域のハンドルを指定します。

**WNS\_FILE\_HANDLE** 出力ファイルハンドル

出力ファイル情報領域のハンドルを指定します。入力ファイル領域のハンドルと同じ値は指定できません。

**WNS\_KEY\_HANDLE** 集約キーハンドル

集約キー情報領域のハンドルを指定します。

**WNS\_SUM\_HANDLE** 集約項目ハンドル

集約項目情報領域のハンドルを指定します。無条件レコード抽出機能を使用する場合は「0」を指定します。

### (21) WNS\_Sum\_Create

#### 形式

```
WNS_SUM_HANDLE WNS_Sum_Create  
                (int 集約項目数,  
                 WNS_SUMOPT オプション情報,  
                 WNS_SUMOPT値);
```

#### 機能

集約項目の情報領域を確保して初期化します。

この関数で確保された集約項目の情報領域は、WNS\_Sum\_Release または WNS\_Release が呼び出されるまで有効です。

#### 戻り値

**WNS\_SUM\_HANDLE**

確保した集約項目の情報領域のハンドルです。確保できなかった場合は「0」を返します。

#### パラメタ

**int** 集約項目数

集約項目数を「1～64」の範囲で指定します。

**WNS\_SUMOPT** オプション情報

集約項目に対するオプション情報を指定します。

**WNS\_SUMOPT 値**

- WNS\_OVERFLOW\_10 : 2 進数の桁あふれを 10 進表示の桁数で判定
- WNS\_BIGENDIAN : 数値項目が big endian

**(22) WNS\_Sum\_Init****形式**

int WNS\_Sum\_Init (WNS\_SUM\_HANDLE 集約項目ハンドル);

**機能**

集約項目の情報領域を、WNS\_Sum\_Create 実行直後の状態に戻します (集約項目の情報領域の初期化)。

この関数で初期化された集約項目の情報領域は、WNS\_Sum\_Release, または WNS\_Release が呼び出されるまで有効です。

**戻り値**

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

**パラメタ**

WNS\_SUM\_HANDLE 集約項目ハンドル

集約項目の情報領域のハンドルを指定します。

**(23) WNS\_Sum\_Release****形式**

int WNS\_Sum\_Release (WNS\_SUM\_HANDLE 集約項目ハンドル);

**機能**

集約項目の情報領域を解放します。

**戻り値**

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

**パラメタ**

WNS\_SUM\_HANDLE 集約項目ハンドル

集約項目の情報領域のハンドルを指定します。

**(24) WNS\_Sum\_Set****形式**

int WNS\_Sum\_Set (WNS\_SUM\_HANDLE 集約項目ハンドル,  
WNS\_SUMATTR 項目属性,

### 13. C 言語プログラムでの使い方 (Windows の場合)

```
int 項目位置,  
int 項目長);
```

#### 機能

集約項目の情報領域に集約項目の属性、位置、および長さを設定します。この関数を複数回呼び出した場合は、すべての集約項目が有効です。ただし、集約項目の数は、WNS\_Sum\_Create で設定した集約項目数より多く設定できません。

この関数で設定された集約項目の情報は、WNS\_Sum\_Init, WNS\_Sum\_Release または WNS\_Release が呼び出されるまで有効です。

#### 戻り値

int

正常に終了した場合は「0」、エラーが発生した場合はその詳細エラーコードを返します。

#### パラメタ

**WNS\_SUM\_HANDLE 集約項目ハンドル**

集約項目情報領域のハンドルを指定します。

**WNS\_SUMATTR 項目属性**

集約項目の属性を指定します。

- WNS\_SIGN\_BIN : 符号付き 2 進項目
- WNS\_UNSIGN\_BIN : 符号なし 2 進項目
- WNS\_PACK\_DEC : 内部 10 進項目
- WNS\_ZONE\_DEC : 外部 10 進項目
- WNS\_L\_ZONE\_DEC : 左符号外部 10 進項目
- WNS\_R\_SEP\_DEC : 右分離符号外部 10 進項目
- WNS\_L\_SEP\_DEC : 左分離符号外部 10 進項目

int 項目位置

集約項目のバイト位置を「0 ~ 65,534」の範囲で指定します。

int 項目長

集約項目のバイト長を「1 ~ 19」の範囲で指定します。

### (25) WNS\_Date\_Set

#### 形式

```
int WNS_Date_Set (int 基準年度);
```

#### 機能

日付項目使用時の基準年度を設定します。この関数を省略した場合、基準年度「70」を仮定します。

#### 戻り値

int

正常終了した場合は「0」、エラーが発生した場合はその詳細エラーコード (負) を返します。

#### パラメタ

int 基準年度

日付項目の基準年度 (0 ~ 99) を指定します。

### (26) WNS\_Work\_Set

#### 形式

int WNS\_Work\_Set (WNS\_WORKDIR マージ用一時ファイルディレクトリ);

#### 機能

ソート実行で、マージ用一時ファイルを使用するとき、マージ用一時ファイルの情報を設定します。

#### 戻り値

int

正常終了した場合は「0」、エラーが発生した場合はその詳細エラーコード (負) を返します。

#### パラメタ

WNS\_WORKDIR マージ用一時ファイルディレクトリ

マージ用一時ファイルを生成するディレクトリ名を設定した、文字列バッファポインタを指定します。指定を省略した場合は、WNS\_SORT, WNS\_SORTEX で指定したソート用一時ファイルディレクトリ、またはカレントディレクトリが使用されます。

## 13.2 コーディング例

C 言語で関数を使用して作成したプログラムのコーディング例を、次の図に示します。この例では、入力ファイル「IN\_FILE」から 2 個の選択条件式で選択したデータを 2 個のキーでソートし、結果を出力ファイル「OUT\_FILE」に出力します。

図 13-1 コーディング例

```
#include "WNS_SORT.h"
    WNS_FILE_HANDLE  input_handle;          /* 入力ファイルハンドル */
    WNS_FILE_HANDLE  output_handle;        /* 出力ファイルハンドル */
    WNS_SEL_HANDLE   sel_handle;          /* 選択条件式ハンドル */
    WNS_KEY_HANDLE   key_handle;         /* キーハンドル */
    WNS_ERR_HANDLE   error_handle;       /* エラーハンドル */
/* SORT終了処理 (ハンドルの解放) */
WNS_END ()
{
    WNS_File_Release (input_handle);
    WNS_File_Release (output_handle);
    WNS_Sel_Release (sel_handle);
    WNS_Key_Release (Key_handle);
    WNS_Release();
    return(0) ;
}
/* SORT機能 */
int sortexe()
{
    WNS_ERR_INF      error_inf;          /* エラー情報 */
    WNS_ERR_TXT      error_msg[WNS_MSGSIZ]; /* エラーメッセージ */
    char             in_file[] = "IN_FILE"; /* 入力ファイル */
    char             out_file[] = "OUT_FILE"; /* 出力ファイル */
    char             sel_val1[] = "AA0001"; /* 条件値1 */
    char             sel_val2[] = "AA9999"; /* 条件値2 */
    int              r_code;            /* リターンコード */
    input_handle = 0;                  /* 入力ファイルハンドル */
    output_handle = 0;                 /* 出力ファイルハンドル */
    sel_handle = 0;                    /* 選択条件式ハンドル */
    key_handle = 0;                    /* キーハンドル */
    error_handle = 0;                  /* エラーハンドル */
    if ((r_code = WNS_Create(16)) != 0) { /* SORT領域確保 */
        return(r_code);
    }
}
```

```

/* 入力ファイル情報 */
input_handle = WNS_File_Create(1,
                               WNS_FIX_SAM, /* ファイル数:1 */
                               80,          /* 固定長順編成 */
                               0);         /* レコード長:80 */
/* */

if (input_handle <= 0) {
    WNS_END;
    return(input_handle);
}
r_code = WNS_File_Set(input_handle,
                      in_file);          /* 入力ファイルハンドル */
/* ファイル名称 */

if (r_code != 0) {
    WNS_END;
    return(r_code);
}

/* 出力ファイル情報 */
output_handle = WNS_File_Create(1,
                                WNS_FIX_SAM, /* ファイル数:1 */
                                80,          /* 固定長順編成 */
                                WNS_ORDER); /* レコード長:80 */
/* 入力順序保存 */

if (output_handle <= 0) {
    WNS_END;
    return(output_handle);
}
r_code = WNS_File_Set(output_handle,
                      out_file);        /* 出力ファイルハンドル */
/* ファイル名称 */

if (r_code != 0) {
    WNS_END;
    return(r_code);
}

/* 選択条件情報 */
sel_handle = WNS_Sel_Create(2, 0);      /* 選択条件数:2 */

if (sel_handle <= 0) {
    WNS_END;
    return(sel_handle);
}
r_code = WNS_Sel_Set (sel_handle,
                     WNS_SL_NON,        /* 選択条件式ハンドル */
                     WNS_CHAR, 10, 6,  /* 論理演算子なし */
                     WNS_SL_GE,        /* 文字, 位置:10, 長さ:6 */
                     WNS_VAL_CHAR, sel_val1, 6); /* 文字, "AA0001", 長さ:6 */

if (r_code != 0) {
    WNS_END;
    return(r_code);
}
r_code = WNS_Sel_Set (sel_handle,
                     WNS_SL_AND,        /* 選択条件式ハンドル */
                     WNS_CHAR, 10, 6,  /* AND */
                     WNS_SL_LE,        /* 文字, 位置:10, 長さ:6 */
                     WNS_VAL_CHAR, sel_val1, 6); /* 文字, "AA9999", 長さ:6 */

if (r_code != 0) {
    WNS_END;
    return(r_code);
}

```

### 13. C 言語プログラムでの使い方 (Windows の場合)

```

/* キー情報 */
key_handle = WNS_Key_Create(2, 0); /* キー数:2 */
if (key_handle <= 0) {
    WNS_END;
    return(key_handle);
}
r_code = WNS_Key_Set (key_handle, /* キーハンドル */
                    WNS_CHAR, 10, 6, /* 文字,位置:10,長さ:6 */
                    WNS_ASC); /* 昇順 */

if (r_code != 0) {
    WNS_END;
    return(r_code);
}
r_code = WNS_Key_Set (key_handle, /* キーハンドル */
                    WNS_ZONE_DEC, 20, 8, /* 外部10進,位置:20,長さ:8 */
                    WNS_DES); /* 降順 */

if (r_code != 0) {
    WNS_END;
    return(r_code);
}
/* 選択実行 */
if ((error_handle = WNS_Select(input_handle, /* 入力ファイルハンドル */
                             output_handle, /* 出力ファイルハンドル */
                             sel_handle)) != 0) { /* 選択条件式ハンドル */
    r_code = WNS_GetError(error_handle, /* エラーハンドル */
                        &error_inf); /* エラー情報領域 */
    r_code = WNS_GetMsg(&error_inf, /* エラー情報領域 */
                      error_msg); /* エラーメッセージ領域 */
    WNS_END;
    return(error_inf.WNS_ERR_CODE);
}
/* ソート実行 */
if ((error_handle = WNS_Sort(output_handle, /* 入力(出力)ファイルハンドル */
                             output_handle, /* 出力ファイルハンドル */
                             key_handle, 0, 0)) != 0) { /*キーハンドル */
    r_code = WNS_GetError(error_handle, /* エラーハンドル */
                        &error_inf); /* エラー情報領域 */
    r_code = WNS_GetMsg(&error_inf, /* エラー情報領域 */
                      error_msg); /* エラーメッセージ領域 */
    WNS_END;
    return(error_inf.WNS_ERR_CODE);
}
return(0);
}

```



## 13.3 コンパイル方法

SORT では、C 言語用に、次の表に示すヘッダファイルとインポートライブラリを提供しています。

表 13-2 ヘッダファイルとインポートライブラリ

SORT の種別	ヘッダファイル	インポートライブラリ
32 ビット版	WNS_SORT.H	WNS_SORT.LIB
64 ビット版	WNS_SORT.H	WNS_SORT64.LIB

作成したプログラムにリンクージする場合、インポートライブラリを指定します。

コンパイル方法、およびリンクージ方法の詳細は、使用する C 言語のマニュアルを参照してください。

## 13.4 詳細エラーコード

関数実行時にエラーが発生した場合に返されるエラーコードの内容を、次の表に示します。

表 13-3 詳細エラーコードの内容

詳細エラーコード	エラー内容
-1	メモリが確保できない。
-2	これ以上ハンドルの割り当てができない。
-3	マルチスレッドで SORT を実行しようとした。
-11	WNS_Create が呼び出されていない。
-12	ハンドルが指定されていない。
-21	ファイル数の指定が間違っている。
-22	ファイル編成の指定が間違っている。
-23	レコード長の指定が間違っている。
-24	ファイル情報以外のハンドルを指定している。
-25	ファイル情報数が、確保した数より多い。
-26	ファイル名が指定されていない。
-31	キー数の指定が間違っている。
-32	キー情報以外のハンドルを指定している。
-33	キー属性の指定が間違っている。
-34	キー位置の指定が間違っている。
-35	キーの並びの指定が間違っている。
-36	キー情報数が、確保した数より多い。
-37	キー長の指定が間違っている。
-40	条件値が指定されていない。
-41	選択条件式数の指定が間違っている。
-42	選択条件式情報以外のハンドルを指定している。
-43	論理演算子の指定が間違っている。
-44	比較項目の属性の指定が間違っている。
-45	比較項目の位置の指定が間違っている。
-46	比較演算子の指定が間違っている。
-47	条件値の属性の指定が間違っている。
-48	条件値の長さの指定が間違っている。
-49	選択条件式の数が、確保した数より多い。
-50	比較項目の長さの指定が間違っている。
-51	集約項目数の指定が間違っている。

詳細エラーコード	エラー内容
-52	集約項目情報以外のハンドルを指定している。
-53	集約項目の属性の指定が間違っている。
-54	集約項目の位置の指定が間違っている。
-55	集約項目数が、確保した数より多い。
-56	集約項目の長さの指定が間違っている。
-61	エラー情報以外のハンドルを指定している。
-62	メッセージ情報テーブルアドレスが指定されていない。
-71	キー情報が設定されていない。
-72	入力ファイル情報が設定されていない。
-73	出力ファイル情報が設定されていない。
-74	入力ファイル情報と出力ファイル情報で同じハンドルを指定した。
-75	選択条件式の情報が設定されていない。
-76	出力ファイル情報の数が間違っている。
-77	使用メモリ量の指定が間違っている。
-81	基準年度の指定が間違っている。



# 14 環境変数の設定

この章では、環境変数の設定を、SORT EE と SORT に分けて説明します。

---

14.1 SORT EE の環境変数の設定

---

14.2 SORT の環境変数の設定 (UNIX の場合)

---

## 14.1 SORT EE の環境変数の設定

SORT EE では、コマンドのパラメタ省略時に設定するデフォルト値を定義する環境変数を用意しています。

また、SORT EE の環境変数の指定がない場合は、SORT の環境変数（Windows では環境ファイル）の指定があれば、SORT の環境変数（Windows では環境ファイル）の指定を有効とします。

32 ビット版 SORT EE では、「EXSORT」で始まる環境変数名は SORT EE の予約変数名となります。また、64 ビット版 SORT EE では、「EXSORT64」で始まる環境変数名が SORT EE の予約変数名となります。SORT の設定以外で「EXSORT～」および「EXSORT64～」の環境変数名を使用しないでください。

SORT EE に関する環境変数を次の表に示します。括弧内は、64 ビット版の環境変数名を示します。

表 14-1 SORT EE に関する環境変数（Windows の場合）

環境変数名	対応する コマンドパ ラメタ	SORT 環境 ファイ ル	ソ ー ト	マ ー ジ	コ ピ ー	簡 易 コ ピ ー	選 択	集 約	編 集	分 割	連 結
EXSORTDATEOPT (EXSORT64DATEOPT)	-dateopt	date_opt	○	○	×	×	○	○ ※ 1	×	○ ※ 2	×
EXSORTSORTTMPDIR (EXSORT64SORTTMPDIR)	-sorttmpdir	temp_dir1	○	×	×	×	×	×	×	×	×
EXSORTMERGETMPDIR (EXSORT64MERGETMPDIR)	-mergetmpdir	temp_dir3	○	×	×	×	×	×	×	×	×
EXSORTWORKSIZE (EXSORT64WORKSIZE)	-worksize	work_size	○	×	×	×	×	×	×	×	×
EXSORTMSGOPT (EXSORT64MSGOPT)	—	msg_opt	○	○	○	○	○	○	○	○	○
EXSORTMSGFILE (EXSORT64MSGFILE)	—	msg_file	○	○	○	○	○	○	○	○	○
EXSORTMGTOPT (EXSORT64MGTOPT)	-cmdopt	—	—	—	—	—	—	—	—	—	—
・ KEY_INVALID_NOCHK	KEYNOCHK	—	○	○	×	×	○	○ ※ 1	×	○ ※ 2	×
・ SUM_INVALID_NOCHK	SUMNOCHK	—	×	×	×	×	×	○	×	×	×

環境変数名	対応する コマンドパ ラメタ	SORT 環境 ファイ ル	ソ ー ト	マ ー ジ	コ ピ ー	簡 易 コ ピ ー	選 択	集 約	編 集	分 割	連 結
・VKEY_POS_NOCHK	POSNOCHK	—	○	○	×	×	×	○ ※ 1	×	○ ※ 2	×
・DATE_INVALID_NOCHK	DATENOCK	—	○	○	×	×	○	○ ※ 1	×	○ ※ 2	×
・DATETIME_INVALID_NOCHK	DATETIME NOCHK	—	○	○	×	×	○	○ ※ 1	×	○ ※ 2	×
・LOCK_INVALID	NOLOCK	—	○	○	○	×	×	×	×	×	×
EXSORTDUPOPT (EXSORT64DUPOPT)	-cmdopt DUP	—	○	×	×	×	×	×	×	×	×
EXSORTIOSIZE (EXSORT64IOSIZE)	-iosize	—	○	○ ※ 3	○ ※ 3	○	×	×	×	○ ※ 5	×
EXSORTINPIOSIZE (EXSORT64INPIOSIZE)	-iosize inpiosize	—	○	○	○	○	×	×	×	×	×
EXSORTOUTIOSIZE (EXSORT64OUTIOSIZE)	-iosize outiosize	—	○	○	○	○	×	×	×	×	×
EXSORTSORTTMPIOSIZE (EXSORT64SORTTMPIOSIZE)	-iosize sorttmpiosiz e	—	○	×	×	×	×	×	×	×	×
EXSORTMERGETMPIOSIZE (EXSORT64MERGETMPIOSIZ E)	-iosize mergetmpio size	—	○	×	×	×	×	×	×	×	×
EXSORT_HI_WORKSIZE (EXSORT64_HI_WORKSIZE)	—	—	○	×	×	×	×	×	×	×	×
EXSORTBOMLNGIN (EXSORT64BOMLNGIN)	—	—	○	○	○	○	×	×	×	×	×
EXSORTBOMLNGOUT (EXSORT64BOMLNGOUT)	—	—	○	○	○	○	×	×	×	×	×
EXSORTBOMGETPUT (EXSORT64BOMGETPUT)	—	—	○	○	○	○	×	×	×	×	×
EXSORTSTRINGDIR (EXSORT64STRINGDIR)	—	—	○	×	×	×	×	×	×	×	×
EXSORTSUPPRESSSSPLITMSG (EXSORT64SUPPRESSSSPLITM SG)	—	—	×	×	×	×	×	×	×	○	×
EXSORTSUPPRESSCATMSG (EXSORT64SUPPRESSCATMS G)	—	—	×	×	×	×	×	×	×	×	○

14. 環境変数の設定

環境変数名	対応する コマンドパ ラメタ	SORT 環境 ファイ ル	ソ ー ト	マ ー ジ	コ ピ ー	簡 易 コ ピ ー	選 択	集 約	編 集	分 割	連 結
EXSORTNOCREATEZEROFIL E (EXSORT64NOCREATEZEROF ILE)	—	—	×	×	×	×	×	×	×	○ ※ 4	×
EXSORTCATINPNOERR (EXSORT64CATINPNOERR)	—	—	×	×	×	×	×	×	×	×	○
EXSORTFILESPLITNUMB (EXSORT64FILESPLITNUMB)	—	—	×	×	×	×	×	×	×	○	×
EXSORTSPLITNUMBONE (EXSORT64SPLITNUMBONE)	—	—	×	×	×	×	×	×	×	○	×
EXSORTSPLITTMPDIR (EXSORT64SPLITTMPDIR)	-splittmpdir	—	×	×	×	×	×	×	×	○	×
EXSORTSPLITMPIOSIZE (EXSORT64SPLITMPIOSIZE)	-iosize splittmpiosi ze	—	×	×	×	×	×	×	×	○	×
EXSORTCATNAMESEQ (EXSORT64CATNAMESEQ)	-catinp nameseq	—	×	×	×	×	×	×	×	×	○
EXSORTOVFERROPT (EXSORT64OVFERROPT)	-cmdopt OVFERR	—	×	×	×	×	×	○	×	×	×
EXSORTGOPTION (EXSORT64GOPTION)	—	—	○	○	×	×	○	○ ※ 1	×	○ ※ 2	×
EXSORTPACKSIGNTYPE (EXSORT64PACKSIGNTYPE)	—	—	○	○	×	×	○	○	×	○ ※ 2	×
EXSORTUNPACKDATATYPE (EXSORT64UNPACKDATATY PE)	—	—	○	○	×	×	○	○	×	○ ※ 2	×
EXSORTDELBLANKLINE (EXSORT64DELBLANKLINE)	-cmdopt DELBLAN KLINE	—	○	○	○	×	×	×	×	×	×
EXSORTOUTFILEENUMUNLIM IT (EXSORT64OUTFILEENUMUN LIMIT)	—	—	○	○	○	×	—	—	—	—	—
EXSORTPRMSIZEUNLIMIT (EXSORT64PRMSIZEUNLIMI T)	—	—	○	○	○	—	○	○	○	○	○
EXSORTOUTFILENODUPCHK (EXSORT64OUTFILENODUP CHK)	—	—	○	○	○	×	—	—	—	○	○



環境変数名	対応する コマンドパラ メタ	SORT 環境 ファイル	ソ ー ト	マ ー ジ	コ ピ ー	簡 易 コ ピ ー	選 択	集 約	編 集	分 割	連 結
EXSORTLOGSUPPRESS (EXSORT64LOGSUPPRESS)	—	—	○	○	○	○	○	○	○	○	○

(凡例)

○：有効    ×：無効    —：該当しない

注※ 1

集約キーが対象となります。集約項目は対象ではありません。

注※ 2

分割キーおよび非分割データが対象となります。

注※ 3

マージおよびコピー機能の場合、ソート用一時ファイルとマージ用一時ファイルは対象ではありません。

注※ 4

-cmdopt パラメタの OVWRT, および INPOVWRT オプションと同時に指定できません。

注※ 5

分割機能を使用しない場合、分割用一時ファイルは対象ではありません。

表 14-2 SORT EE に関する環境変数 (UNIX の場合)

環境変数名	対応する コマンドパラ メタ	SORT 環境変数 (SORTO PT)	ソ ー ト	マ ー ジ	コ ピ ー	簡 易 コ ピ ー	選 択	集 約	編 集	分 割	連 結
EXSORTDATEOPT (EXSORT64DATEOPT)	-dateopt	-g	○	○	×	×	○	○ ※ 1	×	○ ※ 2	×
EXSORTSORTTMPDIR (EXSORT64SORTTMPDIR)	-sorttmpdir	-T	○	×	×	×	×	×	×	×	×
EXSORTMERGETMPDIR (EXSORT64MERGETMPDIR)	-mergetmpdir	-T2	○	×	×	×	×	×	×	×	×
EXSORTWORKSIZE (EXSORT64WORKSIZE)	-worksize	-y	○	×	×	×	×	×	×	×	×
EXSORTMSGOPT (EXSORT64MSGOPT)	—	—	○	○	○	○	○	○	○	○	○
EXSORTMSGFILE (EXSORT64MSGFILE)	—	—	○	○	○	○	○	○	○	○	○
EXSORTMGTOPT (EXSORT64MGTOPT)	-cmdopt	—	—	—	—	—	—	—	—	—	—

14. 環境変数の設定

環境変数名	対応する コマンドパラ メタ	SORT 環境変数 (SORTO PT)	ソ ー ト	マ ー ジ	コ ピ ー	簡 易 コ ピ ー	選 択	集 約	編 集	分 割	連 結
・ KEY_INVALID_NOCHK	KEYNOCHK	—	○	○	×	×	○	○ ※ 1	×	○ ※ 2	×
・ SUM_INVALID_NOCHK	SUMNOCHK	—	×	×	×	×	×	○	×	×	×
・ VKEY_POS_NOCHK	POSNOCHK	—	○	○	×	×	×	○ ※ 1	×	○ ※ 2	×
・ DATE_INVALID_NOCHK	DATENOCHK	—	○	○	×	×	○	○ ※ 1	×	○ ※ 2	×
・ DATETIME_INVALID_NOCHK	DATETIME NOCHK	—	○	○	×	×	○	○ ※ 1	×	○ ※ 2	×
・ LOCK_INVALID	NOLOCK	—	○	○	○	×	×	×	×	×	×
EXSORTDUPOPT (EXSORT64DUPOPT)	-cmdopt DUP	—	○	×	×	×	×	×	×	×	×
EXSORTIOSIZE (EXSORT64IOSIZE)	-iosize	—	○	○ ※ 3	○ ※ 3	○	×	×	×	○ ※ 5	×
EXSORTINPIOSIZE (EXSORT64INPIOSIZE)	-iosize inpiosize	—	○	○	○	○	×	×	×	×	×
EXSORTOUTIOSIZE (EXSORT64OUTIOSIZE)	-iosize outiosize	—	○	○	○	○	×	×	×	×	×
EXSORTSORTTMPIOSIZE (EXSORT64SORTTMPIOSIZE )	-iosize sorttmpiosiz e	—	○	×	×	×	×	×	×	×	×
EXSORTMERGETMPIOSIZE (EXSORT64MERGETMPIOSI ZE)	-iosize mergetmpios ize	—	○	×	×	×	×	×	×	×	×
EXSORT_HI_WORKSIZE (EXSORT64_HI_WORKSIZE)	—	—	○	×	×	×	×	×	×	×	×
EXSORTBOMLNGIN (EXSORT64BOMLNGIN)	—	—	○	○	○	○	×	×	×	×	×
EXSORTBOMLNGOUT (EXSORT64BOMLNGOUT)	—	—	○	○	○	○	×	×	×	×	×
EXSORTBOMGETPUT (EXSORT64BOMGETPUT)	—	—	○	○	○	○	×	×	×	×	×
EXSORTSTRINGDIR (EXSORT64STRINGDIR)	—	—	○	×	×	×	×	×	×	×	×

環境変数名	対応する コマンドパラ メタ	SORT 環境変数 (SORTO PT)	ソ ー ト	マ ー ジ	コ ピ ー	簡 易 コ ピ ー	選 択	集 約	編 集	分 割	連 結
EXSORTSUPPRESSSPLITMSG (EXSORT64SUPPRESSSPLITMSG)	—	—	×	×	×	×	×	×	×	○	×
EXSORTSUPPRESSCATMSG (EXSORT64SUPPRESSCATMSG)	—	—	×	×	×	×	×	×	×	×	○
EXSORTNOCREATEZEROFILE (EXSORT64NOCREATEZEROFILE)	—	—	×	×	×	×	×	×	×	○ ※ 4	×
EXSORTCATINPNOERR (EXSORT64CATINPNOERR)	—	—	×	×	×	×	×	×	×	×	○
EXSORTFILESPLITNUMB (EXSORT64FILESPLITNUMB)	—	—	×	×	×	×	×	×	×	○	×
EXSORTSPLITNUMBONE (EXSORT64SPLITNUMBONE)	—	—	×	×	×	×	×	×	×	○	×
EXSORTSPLITTMPDIR (EXSORT64SPLITTMPDIR)	-splittmpdir	—	×	×	×	×	×	×	×	○	×
EXSORTSPLITMPIOSIZE (EXSORT64SPLITMPIOSIZE)	-iosize splittmpiosize	—	×	×	×	×	×	×	×	○	×
EXSORTOVFERROPT (EXSORT64OVFERROPT)	-cmdopt OVFERR	—	×	×	×	×	×	○	×	×	×
EXSORTGOPTION (EXSORT64GOPTION)	—	—	○	○	×	×	○	○ ※ 1	×	○ ※ 2	×
EXSORTPACKSIGNTYPE (EXSORT64PACKSIGNTYPE)	—	—	○	○	×	×	○	○	×	○ ※ 2	×
EXSORTUNPACKDATATYPE (EXSORT64UNPACKDATATYPE)	—	—	○	○	×	×	○	○	×	○ ※ 2	×
EXSORTDELBLANKLINE (EXSORT64DELBLANKLINE)	-cmdopt DELBLANKLINE	—	○	○	○	×	×	×	×	×	×
EXSORT64OUTFILENUMUNLIMIT	—	—	○	○	○	×	—	—	—	—	—
EXSORT64PRMSIZEUNLIMIT	—	—	○	○	○	—	○	○	○	○	○

## 14. 環境変数の設定

環境変数名	対応する コマンドパラ メタ	SORT 環境変数 (SORTO PT)	ソ ー ト	マ ー ジ	コ ピ ー	簡 易 コ ピ ー	選 択	集 約	編 集	分 割	連 結
EXSORT64OUTFILENODUP CHK	—	—	○	○	○	×	—	—	—	○	○
EXSORT64LOGSUPPRESS	—	—	○	○	○	○	○	○	○	○	○
EXSORT64SORTLIBOPTION	—	—	○	○	○	○	○	○	○	○	○

(凡例)

○：有効    ×：無効    —：該当しない

注※1

集約キーが対象となります。集約項目は対象ではありません。

注※2

分割キーおよび非分割データが対象となります。

注※3

マージおよびコピー機能の場合、ソート用一時ファイルとマージ用一時ファイルは対象ではありません。

注※4

-cmdopt パラメタの OVWRT, および INPOVWRT オプションと同時に指定できません。

注※5

分割機能を使用しない場合、分割用一時ファイルは対象ではありません。

### 14.1.1 32 ビット版 SORT EE の環境変数

#### (1) EXSORTDATEOPT

日付（英数字）項目でソートするときに最小値とする基準年を指定できます。この指定は、キーまたは比較項目に日付属性「g」を指定した場合だけ有効となります。

環境変数名	: EXSORTDATEOPT
指定値	: 基準年度

基準年度には、西暦下 2 桁の数値を、1～2 桁の 10 進数で指定します。

基準年度はキーの比較時に、指定した値を年度の先頭に位置づけ、それより小さい値には、100 を加えた値を設定します。

環境変数 EXSORTDATEOPT および -dateopt パラメタの両方を省略した場合は、「70」が設定されます。

「0」を指定した場合は、文字項目で処理した結果と同じとなります。

コマンドでの指定方法については、「5.14 -dateopt パラメタ」を参照してください。

## (2) EXSORTSORTTMPDIR

ソート用一時ファイルを割り当てるディレクトリを指定できます。この指定は、ソート機能の実行時だけ有効となります。

コマンドでの指定方法については、「5.11 `-sorttmpdir` パラメタ」を参照してください。

環境変数名	: EXSORTSORTTMPDIR
指定値	: ソート用一時ファイルのディレクトリ名

## (3) EXSORTMERGETMPDIR

マージ用一時ファイルを割り当てるディレクトリを指定できます。この指定は、ソート機能の実行時だけ有効となります。

コマンドでの指定方法については、「5.12 `-mergetmpdir` パラメタ」を参照してください。

環境変数名	: EXSORTMERGETMPDIR
指定値	: マージ用一時ファイルのディレクトリ名

## (4) EXSORTWORKSIZE

ソート機能で使用するワークバッファのサイズを指定します。この指定は、ソート機能の実行時だけ有効となります。

コマンドでの指定方法、優先順位、および注意事項については、「5.13 `-worksize` パラメタ」を参照してください。

環境変数名	: EXSORTWORKSIZE
指定値	: { ワークバッファサイズ   AUTO [ , 最大ワークバッファサイズ ]   INCORE [ , 最大ワークバッファサイズ ] }

### ワークバッファサイズ

ワークバッファサイズはMB単位の1～4桁の数字で指定します。  
指定できる範囲は、1～2,000です。

### AUTO

自動で見積もりしたワークバッファサイズが適用されます(アウトコアソート対応)。

### INCORE

自動で見積もりしたワークバッファサイズが適用されます(インコアソート対応)。

### 最大ワークバッファサイズ

ソート処理で使用するワークバッファサイズの最大値を指定します。  
最大ワークバッファサイズはMB単位の1～4桁の数字で指定します。

## 14. 環境変数の設定

指定できる範囲は、1～2,000 です。

指定を省略した場合は、2,000 (MB) を仮定します。

### (5) EXSORTMSGOPT

コマンド実行時のメッセージ出力オプションを指定します。指定値と意味は、OSによって異なります。

環境変数名	: EXSORTMSGOPT
指定値	: オプション番号

#### ● Windows

環境変数 EXSORTMSGOPT を省略した場合は、SORT 環境ファイルの msg\_opt オプションの指定を有効とします。環境変数および SORT 環境ファイルのどちらも省略した場合は、0 (ゼロ) を設定します。

各オプションの意味を、次の表に示します。

オプション番号	意味
0	すべてのメッセージを画面に表示する。
1	エラーメッセージだけを画面に表示する。
2	すべてのメッセージを画面に表示しない。
3	メッセージを画面に表示しないで、環境変数 EXSORTMSGFILE または環境ファイルの msg_file オプションで指定したファイルに、すべてのメッセージを出力する。
4	メッセージを画面に表示しないで、環境変数 EXSORTMSGFILE または環境ファイルの msg_file オプションで指定したファイルに、エラーメッセージだけを出力する。
5	イベントログファイルにすべてのメッセージを出力する。
6	イベントログファイルにエラーメッセージだけを出力する。
7	標準エラーファイルにすべてのメッセージを出力する。
8	標準エラーファイルにエラーメッセージだけを出力する。
9	すべてのメッセージを標準出力ファイル <sup>※</sup> に出力する。
10	エラーメッセージだけを標準出力ファイル <sup>※</sup> に出力する。
11	正常終了メッセージだけを標準出力ファイル <sup>※</sup> に出力する。
12	正常終了メッセージだけを画面に表示する。
13	正常終了メッセージだけを環境変数 EXSORTMSGFILE または環境ファイルの msg_file オプションで指定したファイルに出力する。
14	正常終了メッセージだけをイベントログファイルに出力する。
15	正常終了メッセージだけを標準エラーファイルに出力する。

オプション番号	意味
16	正常終了メッセージを標準出力ファイル <sup>※</sup> に、エラーメッセージを標準エラーファイルに出力する。

## 注※

出力ファイルが標準出力ファイルでない場合だけ、メッセージを標準出力ファイルに出力します。出力ファイルが標準出力ファイルの場合は、標準エラーファイルに表示します。

## ● UNIX

指定を省略した場合は、7を設定します。各オプションの意味を、次の表に示します。

オプション番号	意味
2	すべてのメッセージを画面に表示しない。
3	環境変数 EXSORTMSGFILE で指定したファイルに、すべてのメッセージを出力する。
4	環境変数 EXSORTMSGFILE で指定したファイルに、エラーメッセージだけを出力する。
7	標準エラーファイルにすべてのメッセージを出力する。
8	標準エラーファイルにエラーメッセージだけを出力する。
9	すべてのメッセージを標準出力ファイル <sup>※</sup> に出力する。
10	エラーメッセージだけを標準出力ファイル <sup>※</sup> に出力する。
11	正常終了メッセージだけを標準出力ファイル <sup>※</sup> に出力する。
13	正常終了メッセージだけを環境変数 EXSORTMSGFILE で指定したファイルに出力する。
15	正常終了メッセージだけを標準エラーファイルに出力する。
16	正常終了メッセージを標準出力ファイル <sup>※</sup> に、エラーメッセージを標準エラーファイルに出力する。

## 注※

出力ファイルが標準出力ファイルでない場合だけ、メッセージを標準出力ファイルに出力します。出力ファイルが標準出力ファイルの場合は、標準エラーファイルに表示します。

また、オプション番号「3」または「4」を指定したときの、ファイル出力時のメッセージ形式を次に示します。

```
yyyy/mm/dd △ hh:mm:ss △ [SORT △ EE] △ [xx··xx] △ KBL$nnnn-x △△ zzz··zzz
```

## (凡例)

△ : 1文字の全角スペース

yyyy/mm/dd : メッセージ発行日 (年/月/日)

hh:mm:ss : メッセージ発行時刻 (時:分:秒)

## 14. 環境変数の設定

[SORT △ EE] : 製品名  
[xx…xx] : コマンドを実行したコンピュータ名称  
KBL\$nnnn-x : メッセージ ID  
zzz…zzz : メッセージテキスト

### (6) EXSORTMSGFILE

コマンド実行時のメッセージの出力先ファイル名を指定します。この指定は、環境変数 EXSORTMSGOPT で「3」または「4」を指定した場合にだけ有効となります。

環境変数名	: EXSORTMSGFILE
指定値	: メッセージの出力先ファイル名

環境変数 EXSORTMSGOPT が指定されている場合は、この指定を有効とします。

環境変数 EXSORTMSGOPT が省略で、SORT 環境ファイル msg\_opt の指定を有効とした場合、SORT 環境ファイル msg\_file オプションに指定されたファイル名を有効とします。

### (7) EXSORTMGTOPT

チェック機能の緩和に関するオプションを指定します。

各オプションの指定順序は任意です。複数のオプションを指定する場合、コンマ (,) で区切って指定します。コンマの前後に空白またはタブを入れないでください。

#### 【指定例】

EXSORTMGTOPT=KEY_INVALID_NOCHK,SUM_INVALID_NOCHK
--

(a) ~ (e) に示すオプションは、対応するコマンドオプションが用意されています。

コマンドの指定は EXSORTMGTOPT よりも優先度が高いため、これらのオプションのうち、どれかを exsort コマンドに指定した場合、EXSORTMGTOPT の指定は exsort コマンド指定に置き換わります。

例えば、「EXSORTMGTOPT=KEY\_INVALID\_NOCHK,DATE\_INVALID\_NOCHK」が指定されているシステムで、「-cmdopt SUMNOCHK」を指定した exsort コマンドを実行した場合、コマンド指定の「SUMNOCHK」だけが有効となります。

コマンドオプションについては、「表 14-1 SORT EE に関する環境変数 (Windows の場合)」または「表 14-2 SORT EE に関する環境変数 (UNIX の場合)」を参照してください。

#### (a) KEY\_INVALID\_NOCHK オプション

キーまたは比較項目の属性が外部 10 進数または内部 10 進数の場合、入力データの形式



が不正であってもエラーにしないオプションです。

機能の詳細は、「5.5.9 10進キーのチェックオプション (KEYNOCHK オプション)」を参照してください。

環境変数名	: EXSORTMGTOPT
指定値	: KEY_INVALID_NOCHK

#### (b) SUM\_INVALID\_NOCHK オプション

集約項目の属性が外部 10 進数または内部 10 進数の場合、入力データの形式が不正であってもエラーにしないオプションです。

機能の詳細は、「5.5.10 10進集約項目のチェックオプション (SUMNOCHK オプション)」を参照してください。

環境変数名	: EXSORTMGTOPT
指定値	: SUM_INVALID_NOCHK

#### (c) VKEY\_POS\_NOCHK オプション

キーの指定範囲のチェックを緩和するオプションです。

- 可変長順編成ファイルのキーの指定範囲が最小レコード長を超えていても、エラーにしません。
- テキストファイルのキーの指定範囲が実レコード長を超えていても、エラーにしません。

機能の詳細は、「5.5.11 キー位置のチェックオプション (POSNOCHK オプション)」を参照してください。

環境変数名	: EXSORTMGTOPT
指定値	: VKEY_POS_NOCHK

#### (d) DATE\_INVALID\_NOCHK オプション

年号に空白、Low\_value (x'0000'), High\_value (x'FFFF') などの値が設定されていても、エラーとしないオプションです。

機能の詳細は、「5.5.12 日付データのチェックオプション (DATENOCHK オプション)」を参照してください。

環境変数名	: EXSORTMGTOPT
指定値	: DATE_INVALID_NOCHK

## 14. 環境変数の設定

### (e) DATETIME\_INVALID\_NOCHK オプション

キーまたは比較項目の属性が日付時刻の場合、入力データの形式が不正であってもエラーにしないオプションです。

機能の詳細は、「5.5.13 日付時刻データのチェックオプション (DATETIMENOCK オプション)」を参照してください。

環境変数名	: EXSORTMGTOPT
指定値	: DATETIME_INVALID_NOCHK

### (f) LOCK\_INVALID オプション

SORT, SORT EE, COBOL, ISAM プログラムの処理間で、入出力ファイルのオープンモードを指定します。この指定は UNIX だけ有効となります。

機能の詳細は、「5.5.14 入出力ファイル共用・排他オプション (NOLOCK | EXC オプション)」を参照してください。

環境変数名	: EXSORTMGTOPT
指定値	: LOCK_INVALID

### (8) EXSORTDUPOPT

重複レコードの入力順序保存機能を有効とします。この指定は、ソート機能の実行時だけ有効となります。

環境変数名	: EXSORTDUPOPT
指定値	: YES

### (9) EXSORTIOSIZE

入力ファイル、出力ファイル、ソート用一時ファイル、およびマージ用一時ファイルに対する 1 回の I/O サイズを指定します。

環境変数名	: EXSORTIOSIZE
指定値	: I/O サイズ

I/O サイズは 10 進数（単位は KB）で指定します。

指定できる範囲は、8 ~ 32,768 です。

### (10) EXSORTINPIOSIZE

入力ファイルに対する 1 回の I/O サイズを指定します。

環境変数名	: EXSORTINPIOSIZE
指定値	: <i>I/O サイズ</i>

I/O サイズは 10 進数（単位は KB）で指定します。

指定できる範囲は、8 ～ 32,768 です。

#### (11) EXSORTOUTIOSIZE

出力ファイルに対する 1 回の I/O サイズを指定します。

環境変数名	: EXSORTOUTIOSIZE
指定値	: <i>I/O サイズ</i>

I/O サイズは 10 進数（単位は KB）で指定します。

指定できる範囲は、8 ～ 32,768 です。

#### (12) EXSORTSORTTMPIOSIZE

ソート用一時ファイルに対する 1 回の I/O サイズを指定します。

環境変数名	: EXSORTSORTTMPIOSIZE
指定値	: <i>I/O サイズ</i>

I/O サイズは 10 進数（単位は KB）で指定します。

指定できる範囲は、8 ～ 32,768 です。

#### (13) EXSORTMERGETMPIOSIZE

マージ用一時ファイルに対する 1 回の I/O サイズを指定します。

環境変数名	: EXSORTMERGETMPIOSIZE
指定値	: <i>I/O サイズ</i>

I/O サイズは 10 進数（単位は KB）で指定します。

指定できる範囲は、8 ～ 32,768 です。

#### (14) EXSORT\_HI\_WORKSIZE

ソート機能で使用するワークバッファのサイズを指定します。

exsort コマンドの `-worksize` パラメタ、または環境変数「EXSORTWORKSIZE」の指定があっても、環境変数「EXSORT\_HI\_WORKSIZE」で指定された値を、ワークサイズとして置き換えます。

## 14. 環境変数の設定

なお、この指定はソート機能の実行時だけ有効となります。

環境変数名	: EXSORT_HI_WORKSIZE
指定値	: {ワークバッファサイズ   AUTO [, 最大ワークバッファサイズ]   INCORE [, 最大ワークバッファサイズ]}

指定値は、「14.1.1 (4) EXSORTWORKSIZE」を参照してください。

### (15) EXSORTBOMLNGIN

Unicode シグニチャ (以降, BOM と呼びます) の読み飛ばし機能を使用できます。

環境変数名	: EXSORTBOMLNGIN
指定値	: UNICODE

- この環境変数を指定すると、すべての入力ファイルの BOM を読み飛ばします。
- 環境変数「EXSORTBOMGETPUT」と、この環境変数のどちらの指定もない場合、入力ファイルの BOM を読み飛ばさずに、レコードデータとして処理します。
- 出力ファイルに BOM を付けたい場合は、環境変数「EXSORTBOMLNGOUT」の指定が必要です。環境変数「EXSORTBOMLNGOUT」については、「14.1.1(16) EXSORTBOMLNGOUT」を参照してください。
- 出力ファイルの BOM の種類と有無を、入力ファイルに合わせたい場合は、環境変数「EXSORTBOMGETPUT」を指定します。環境変数「EXSORTBOMGETPUT」については、「14.1.1(17) EXSORTBOMGETPUT」を参照してください。
- この環境変数に「UNICODE」を指定すると、次の形式の BOM を読み飛ばします。

BOM 形式	バイト長	説明
0xEFBBBF	3	UTF-8
0xFFFE	2	UTF-16LE (UTF-16 little endian)
0xFEFF	2	UTF-16BE (UTF-16 big endian)
0xFFFE0000	4	UTF-32LE (UTF-32 little endian)
0x0000FEFF	4	UTF-32BE (UTF-32 big endian)

### (16) EXSORTBOMLNGOUT

Unicode シグニチャ (以降, BOM と呼びます) の付加機能を使用できます。

環境変数名	: EXSORTBOMLNGOUT
指定値	: {UTF8   UTF16LE   UTF16BE   UTF32LE   UTF32BE}

- 環境変数「EXSORTBOMGETPUT」の指定がある場合、この環境変数の指定は無視されます。環境変数「EXSORTBOMGETPUT」については、「14.1.1(17) EXSORTBOMGETPUT」を参照してください。
- 環境変数「EXSORTBOMGETPUT」と、この環境変数のどちらの指定もない場合、出力ファイルには BOM は付けられません。
- 入力ファイルの BOM を読み飛ばしたい場合は、環境変数「EXSORTBOMLNGIN」または「EXSORTBOMGETPUT」の指定が必要です。環境変数「EXSORTBOMLNGIN」については、「14.1.1(15) EXSORTBOMLNGIN」を、環境変数「EXSORTBOMGETPUT」については、「14.1.1(17) EXSORTBOMGETPUT」を参照してください。
- この環境変数の指定値に対応する BOM を、出力ファイルに付けます。指定値に対応する BOM 形式を次に示します。

指定値	BOM 形式	バイト長	説明
UTF8	0xEFBBBF	3	UTF-8
UTF16LE	0xFFFE	2	UTF-16 little endian
UTF16BE	0xFEFF	2	UTF-16 big endian
UTF32LE	0xFFFE0000	4	UTF-32 little endian
UTF32BE	0x0000FEFF	4	UTF-32 big endian

### (17) EXSORTBOMGETPUT

Unicode シグニチャ（以降、BOM と呼びます）の読み飛ばしおよび付加機能を使用できません。

環境変数名	: EXSORTBOMGETPUT
指定値	: AUTOBOM

- この環境変数の指定は、環境変数「EXSORTBOMLNGOUT」よりも優先されます。環境変数「EXSORTBOMLNGOUT」については、「14.1.1(16) EXSORTBOMLNGOUT」を参照してください。
- SORT EE で使用できる BOM 形式を次に示します。

BOM 形式	バイト長	説明
0xEFBBBF	3	UTF-8
0xFFFE	2	UTF-16LE (UTF-16 little endian)
0xFEFF	2	UTF-16BE (UTF-16 big endian)
0xFFFE0000	4	UTF-32LE (UTF-32 little endian)

## 14. 環境変数の設定

BOM 形式	バイト長	説明
0x0000FEFF	4	UTF-32BE (UTF-32 big endian)

- この環境変数を指定した場合の動作は次のとおりです。
  - すべての入力ファイルの BOM を読み飛ばします。
  - 入力ファイルの BOM と同じ形式の BOM を、すべての出力ファイルに付けます。入力ファイルが複数個の場合は、先頭の入力ファイルの BOM 形式を付けます。
  - 入力ファイルに BOM がなければ、すべての出力ファイルに BOM を付けません。入力ファイルが複数個の場合は、先頭の入力ファイルの BOM の有無が該当します。2 個目以降の入力ファイルに BOM があっても、先頭の入力ファイルに BOM がなければ出力ファイルには付けません。

環境変数の指定の有無による、入力ファイルの BOM の読み飛ばし、および出力ファイルへの BOM の付加を次の表に示します。

表 14-3 環境変数の指定の有無による、入力ファイルの BOM の読み飛ばし、および出力ファイルへの BOM の付加（先頭の入力ファイルに BOM がある場合）

環境変数の指定の有無			BOM の読み飛ばし	BOM の付加
EXSORTBOMLNGIN	EXSORTBOMLN GOUT	EXSORTBOMG ETPUT		
無	無	無	—	×
有	無	無	○	×
有	有	無	○	◎
有	無	有	○	●
有	有	有	○	●
無	有	無	—	◎
無	有	有	○	●
無	無	有	○	●

表 14-4 環境変数の指定の有無による、入力ファイルの BOM の読み飛ばし、および出力ファイルへの BOM の付加（先頭の入力ファイルに BOM がない場合）

環境変数の指定の有無			BOM の読み飛ばし	BOM の付加
EXSORTBOMLNGIN	EXSORTBOMLN GOUT	EXSORTBOMG ETPUT		
無	無	無	—	×
有	無	無	—	×
有	有	無	—	◎
有	無	有	—	×

環境変数の指定の有無			BOM の読 み飛 ばし	BOM の 付加
EXSORTBOMLNGIN	EXSORTBOMLNGOUT	EXSORTBOMGETPUT		
有	有	有	—	×
無	有	無	—	◎
無	有	有	—	×
無	無	有	—	×

(凡例)

無：環境変数を指定しない

有：環境変数を指定する

○：入力ファイルの BOM を読み飛ばす

—：入力ファイルの BOM を読み飛ばさない

◎：出力ファイルに BOM を付ける (EXSORTBOMLNGOUT に指定した BOM 形式)

●：出力ファイルに BOM を付ける (先頭の入力ファイルの BOM 形式)

×：出力ファイルに BOM を付けない

## (18) EXSORTSTRINGDIR

ソート用ワークバッファが不足した場合に、ワークバッファ内のstring情報を退避する一時ファイルを作成するディレクトリを指定します。この環境変数を指定しない場合、ソート用一時ファイルのディレクトリに、string情報格納一時ファイルが作成されます。この指定は、ソート機能の実行時だけ有効となります。また、ワークバッファが不足した場合にだけ、一時ファイルを作成します。

ワークバッファが不足してstring情報格納一時ファイルを作成した場合、「KBLS1133-I ソート用ワークバッファサイズが不足しています。」のメッセージを出力します。KBLS1133-I メッセージが出力された場合の原因と対処方法の詳細は、「付録 K.4 SORT のメッセージ一覧」を参照してください。

環境変数名	: EXSORTSTRINGDIR
指定値	: string情報格納一時ファイルディレクトリ

## (19) EXSORTSUPPRESSSPLITMSG

ファイル分割終了時、KBLS1010-I メッセージの出力を抑止します。この指定は、ファイル分割機能だけで有効となります。

環境変数名	: EXSORTSUPPRESSSPLITMSG
指定値	: YES

## 14. 環境変数の設定

### (20) EXSORTSUPPRESSCATMSG

ファイル連結終了時、KBL51011-I メッセージの出力を抑制します。この指定は、ファイル連結機能だけで有効となります。

環境変数名	: EXSORTSUPPRESSCATMSG
指定値	: YES

### (21) EXSORTNOCREATEZEROFILE

ファイル分割後の出力ファイルに格納するレコード件数が 0 件の場合、出力ファイルの作成を抑制します。また、0 件レコードであったことを示す KBL51010-I メッセージも抑制します。この指定は、ファイル分割機能だけで有効となります。

環境変数名	: EXSORTNOCREATEZEROFILE
指定値	: YES

- 環境変数 EXSORTNOCREATEZEROFILE は、ファイル分割機能使用時に `-cmdopt` パラメタの `OVWRT`、および `INPOVWRT` オプションと同時に指定できません。
- 環境変数 EXSORTOUTFILENODUPCHK は同時に指定できません。

### (22) EXSORTCATINPNOERR

`-catinp` パラメタの `file` オペランドで指定した入力ファイルの一部が存在しない場合でも、連結処理を続行します。この指定は、ファイル連結機能だけで有効となります。

環境変数名	: EXSORTCATINPNOERR
指定値	: YES

### (23) EXSORTFILESPLITNUMB

ファイル分割時、出力ファイル名に付加する追番の桁数を指定します。この指定がない場合は、追番の桁数は 8 を設定します。

環境変数名	: EXSORTFILESPLITNUMB
指定値	: <i>追番の桁数</i>

追番の桁数は 1 桁の数字で指定します。

指定できる値は、1 ～ 8 です。

### (24) EXSORTSPLITNUMBONE

ファイル分割時、分割キーを指定しても出力ファイル名に付加する追番の個数は一個とします。



環境変数名	: EXSORTSPLITNUMBONE
指定値	: YES

### (25) EXSORTSPLITTMPDIR

ファイル分割用一時ファイルのディレクトリを指定します。この指定は、均等分割情報によるファイル分割時だけで有効となります。

環境変数名	: EXSORTSPLITTMPDIR
指定値	: ファイル分割用一時ファイルのディレクトリ名

### (26) EXSORTSPLITMPIOsize

ファイル分割用一時ファイルに対する 1 回の I/O サイズを指定します。

環境変数名	: EXSORTSPLITMPIOsize
指定値	: I/O サイズ

I/O サイズは 10 進数（単位は KB）で指定します。

指定できる範囲は、8 ~ 32,768 です。

### (27) EXSORTCATNAMESEQ (Windows の場合)

ファイル連結時、`-catinp` パラメタで `suffix` または `prefix` オペランドを指定した場合の入力ファイルの入力順対応コードを指定します。

環境変数名	: EXSORTCATNAMESEQ
指定値	: 入力ファイルの入力順対応コード

入力ファイルの入力順対応コードの指定値と意味については、「5.26.4 `nameseq` オペランド (Windows の場合)」を参照してください。

### (28) EXSORTOVFERROPT

集約機能で集約結果がオーバーフローした場合にエラーにします。

この指定は、集約機能だけで有効となります。

環境変数名	: EXSORTOVFERROPT
指定値	: YES

### (29) EXSORTGOPTION

CSV 日付属性「g」の桁修飾文字を変更するオプションです。

## 14. 環境変数の設定

環境変数名	: EXSORTGOPTION
指定値	: YES

- このオプションの指定有無によって、「年月日 {T | 空白} 時分秒」形式の入力データの扱いが異なります。詳細を次の表に示します。

EXSORTGOPTION オプションの指定有無	日付と時刻の切り分け	日付データの 桁修飾文字
指定なし	Tおよび空白を日付と時刻の区切り文字として扱いません。	ハイフン, スラント
指定あり	Tおよび空白を日付と時刻の区切り文字として扱います。「年月日」部分の日付データだけを有効とし、「時分秒」部分の時刻データは無視します。区切り文字 {T   空白} がない場合、入力データを「年月日」の日付データとして扱います。	ハイフン, スラント, およびピリオド

- このオプションは、CSV 日付属性「g」を指定したときだけ有効です。
- このオプションの有無にかかわらず、CSV 日付属性「g」では、日付またがり調整しません。日付またがり調整したい場合は、CSV 日付時刻属性「h」を使用してください。

### 日付またがりの例

「2013-9-31」は翌月に、「2013-13-1」は翌年にまたがるデータですが、CSV 日付属性「g」ではまたがり調整しないで、入力データをそのまま比較します。なお、CSV 日付時刻属性「h」では、「2013-9-31」を翌月1日の「2013-10-1」として、「2013-13-1」を翌年の「2014-1-1」として扱います。

## (30) EXSORTPACKSIGNTYPE

内部 10 進数の符号の範囲を変更するオプションです。

環境変数名	: EXSORTPACKSIGNTYPE
指定値	: {SE   MF}

- このオプションは、属性コード「p」で指定した内部 10 進数の符号に対してだけ有効です。
- この環境変数の指定がない場合、環境変数ファイル「EXS\_PACK\_SIGNTYPE」の指定を仮定します。
- この環境変数と、環境変数ファイル「EXS\_PACK\_SIGNTYPE」の両方の指定がない場合は、「SE」を仮定します。
- 環境ファイルの指定方法については、環境ファイル「15.1.1(22) EXS\_PACK\_SIGNTYPE」を参照してください。

オプション指定	符号	キー・比較項目	集約項目	集約結果	KEYNOCHK 指定時の不正データの扱い
SE	正	C	C	C	<ul style="list-style-type: none"> <li>数値部が 0～9 以外の場合、JIS8 単位コード体系に従って比較します。</li> <li>符号部が C, D, F 以外の場合、内部 10 進数を正値として扱います。</li> </ul>
	負	D	D	D	
	符号なし (絶対値)	F※1	F※1	C (F※2)	
MF	正	0, 2, 4, 6, 8, A, C, E	A, C, E	C	<ul style="list-style-type: none"> <li>数値部が 0～9 以外の場合、JIS8 単位コード体系に従って比較します。</li> </ul>
	負	1, 3, 5, 7, 9, B, D	B, D	D	
	符号なし (絶対値)	F※1	F※1	C (F※2)	

## 注※1

絶対値 (F) は正値として扱います。

## 注※2

内部 10 進数符号保障オプション (-cmdopt ABS) の指定があり、最初の集約項目の演算符号が F であるとき、以降の集約項目の集約結果に対して同じ演算符号 F を適用します。

## (31) EXSORTUNPACKDATATYPE

外部 10 進数のデータ形式を変更するオプションです。

環境変数名	: EXSORTUNPACKDATATYPE
指定値	: {ASCII-1   ASCII-2   EBCDIK}

- このオプションが有効な属性コードは次のとおりです。
  - z (外部 10 進数)
  - x (左符号付き外部 10 進数)
  - s (右分離符号付き外部 10 進数)
  - t (左分離符号付き外部 10 進数)
  - d (左分離符号付き数字)
  - g (日付) で指定した 2 桁年号部分
- CSV の属性コード「z (固定小数点数)」、「g (日付)」および「h (日付時刻)」には適用されません。指定しても無効となります。
- 入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」、「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE,

## 14. 環境変数の設定

UTF-16BE, UTF-32LE または UTF-32BE)」を指定した場合、このオプションは無効となります。

- この環境変数の指定がない場合、環境変数ファイル「EXS\_UNPACK\_DATATYPE」の指定を仮定します。
- この環境変数と、環境変数ファイル「EXS\_UNPACK\_DATATYPE」の両方の指定がない場合は、「ASCII-1」を仮定します。
- 環境ファイルの指定方法については、環境ファイル「15.1.1(23) EXS\_UNPACK\_DATATYPE」を参照してください。
- 指定したデータ形式に合致しない場合、不正データ形式で KBL51211-E エラーとなります。
- 10進数の不正データをエラーにしない「10進キーのキーチェック緩和機能オプション (-cmdopt KEYNOCHK)」と一緒に指定できます。KEYNOCHK オプションを指定した場合の不正データの扱いを表 14-6, 表 14-7 に示します。
- 日付の不正データをエラーにしない「日付データのチェック緩和機能オプション (-cmdopt DATENOCHK)」と一緒に指定できます。DATENOCHK オプションを指定した場合の不正データの扱いを表 14-8 に示します。

表 14-5 外部 10 進データ形式

数値		オプション指定		
		ASCII-1	ASCII-2	EBCDIK
数字 (絶対値)	0	0 (0x30)	0 (0x30)	0 (0xF0)
	1	1 (0x31)	1 (0x31)	1 (0xF1)
	2	2 (0x32)	2 (0x32)	2 (0xF2)
	3	3 (0x33)	3 (0x33)	3 (0xF3)
	4	4 (0x34)	4 (0x34)	4 (0xF4)
	5	5 (0x35)	5 (0x35)	5 (0xF5)
	6	6 (0x36)	6 (0x36)	6 (0xF6)
	7	7 (0x37)	7 (0x37)	7 (0xF7)
	8	8 (0x38)	8 (0x38)	8 (0xF8)
	9	9 (0x39)	9 (0x39)	9 (0xF9)

数値		オプション指定		
		ASCII-1	ASCII-2	EBCDIK
正値	+0	0 (0x30)	{ (0x7B)	{ (0xC0)
	+1	1 (0x31)	A (0x41)	A (0xC1)
	+2	2 (0x32)	B (0x42)	B (0xC2)
	+3	3 (0x33)	C (0x43)	C (0xC3)
	+4	4 (0x34)	D (0x44)	D (0xC4)
	+5	5 (0x35)	E (0x45)	E (0xC5)
	+6	6 (0x36)	F (0x46)	F (0xC6)
	+7	7 (0x37)	G (0x47)	G (0xC7)
	+8	8 (0x38)	H (0x48)	H (0xC8)
	+9	9 (0x39)	I (0x49)	I (0xC9)
負数	-0	p (0x70)	} (0x7D)	} (0xD0)
	-1	q (0x71)	J (0x4A)	J (0xD1)
	-2	r (0x72)	K (0x4B)	K (0xD2)
	-3	s (0x73)	L (0x4C)	L (0xD3)
	-4	t (0x74)	M (0x4D)	M (0xD4)
	-5	u (0x75)	N (0x4E)	N (0xD5)
	-6	v (0x76)	O (0x4F)	O (0xD6)
	-7	w (0x77)	P (0x50)	P (0xD7)
	-8	x (0x78)	Q (0x51)	Q (0xD8)
	-9	y (0x79)	R (0x52)	R (0xD9)
分離符号	正	+ (0x2B)	+ (0x2B)	+ (0x4E)
	負	- (0x2D)	- (0x2D)	- (0x60)

表 14-6 外部 10 進数 (z, x) 不正データの扱い

オプション指定値	KEYNOCHK 指定時の不正データの扱い
ASCII-1	<ul style="list-style-type: none"> <li>ゾーン部は比較対象外です。</li> <li>数値部が 0 ~ 9 以外の場合、JIS8 単位コード体系に従って比較します。</li> <li>符号部が 3, 7 以外の場合、外部 10 進数を正値として扱います。</li> </ul>
ASCII-2	<ul style="list-style-type: none"> <li>ゾーン部は比較対象外です。</li> <li>数値部が 0 ~ 9 以外の場合、JIS8 単位コード体系に従って比較します。</li> <li>符号部が 4, 5 以外の場合、外部 10 進数を正値として扱います。</li> </ul>

## 14. 環境変数の設定

オプション指定値	KEYNOCHK 指定時の不正データの扱い
EBCDIK	<ul style="list-style-type: none"> <li>ゾーン部は比較対象外です。</li> <li>数値部が 0～9 以外の場合、EBCDIK コード体系に従って比較します。</li> <li>符号部が C, D 以外の場合は、外部 10 進数を次のとおり扱います。 (正) 0, 2, 4, 6, 8, A, C, E, F (負) 1, 3, 5, 7, 9, B, D</li> </ul>

表 14-7 分離符号付き外部 10 進数 (s, t) 不正データの扱い

オプション指定値	KEYNOCHK 指定時の不正データの扱い
ASCII-1 ASCII-2	<ul style="list-style-type: none"> <li>数値部が 0x30～0x39 以外の場合、JIS8 単位コード体系に従って比較します。</li> <li>符号部が 0x2B, 0x2D 以外の場合は、外部 10 進数を正值として扱います。</li> </ul>
EBCDIK	<ul style="list-style-type: none"> <li>数値部が 0xF0～0xF9 以外の場合、EBCDIK コード体系に従って比較します。</li> <li>符号部が 0x4E, 0x60 以外の場合は、外部 10 進数を正值として扱います。</li> </ul>

表 14-8 日付 (g) 不正データの扱い

オプション指定値	DATENOCHK 指定時の不正データの扱い
ASCII-1 ASCII-2	<ul style="list-style-type: none"> <li>キーの先頭 2 文字 (西暦下 2 桁の数値) が指すデータ内容が 0x30～0x39 以外の場合、JIS8 単位コード体系に従って比較します。</li> </ul>
EBCDIK	<ul style="list-style-type: none"> <li>キーの先頭 2 文字 (西暦下 2 桁の数値) が指すデータ内容が 0xF0～0xF9 以外の場合、EBCDIK コード体系に従って比較します。</li> </ul>

### (32) EXSORTDELBLANKLINE

改行だけのデータがないレコード (空行) を削除して入力できるオプションです。

このオプションは、入力ファイル編成がテキストファイル、または CSV ファイルのときに有効となります。入力ファイル編成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は、指定しても無視されます。

環境変数名	: EXSORTDELBLANKLINE
指定値	: YES

### (33) EXSORTOUTFILENUMUNLIMIT 【08-50 以降】

出力ファイル (-outfile) パラメタの指定個数を制限なしにします。

指定できる -outfile パラメタの最小値および最大値については、「3.3.1 パラメタ」の表 3-5 を参照してください。環境ファイルの指定方法については、環境ファイル「15.1.1(24) EXS\_OUTFILENUMUNLIMIT 【08-50 以降】」を参照してください。

環境変数名	: EXSORTOUTFILENUMUNLIMIT
指定値	: YES

### (34) EXSORTPRMSIZEUNLIMIT 【08-50 以降】

exsort パラメタファイルのサイズを制限なしにします。

exsort パラメタファイルの形式については、「4.4.1 ファイル形式」を参照してください。環境ファイルの指定方法については、環境ファイル「15.1.1(25) EXS\_PRMSIZEUNLIMIT 【08-50 以降】」を参照してください。

環境変数名	: EXSORTPRMSIZEUNLIMIT
指定値	: YES

### (35) EXSORTOUTFILENODUPCHK 【08-50 以降】

出力ファイルパラメタ (-outfile, -catout または -splitout パラメタ) で指定された出力ファイル名の重複チェックを抑止します。

環境変数名	: EXSORTOUTFILENODUPCHK
指定値	: YES

- 抑止する重複チェック項目は「入力ファイル名と出力ファイル名」、「出力ファイル名同士」および「exsort パラメタファイル名と出力ファイル名」です。「入力ファイル名と exsort パラメタファイル名」の重複はチェックします。
- ファイルの上書きオプション (-cmdopt OVWRT) との同時指定を必須とします。OVWRT オプションの指定がない場合、KBLS1122-E エラーとなります。
- 環境変数 EXSORTNOCREATEZEROFILE は、同時に指定できません。
- この環境変数を指定した場合、同一名の出力ファイルを指定してもエラーになりません。また、同一名の出力ファイルを指定した場合、出力ファイルの内容は保証されません。

### (36) EXSORTLOGSUPPRESS 【08-50 以降】

exsort コマンドのコマンドログ情報、および exsort コマンド実行時のリソース使用量などのレポート情報の出力を抑止し、ログ出力量の削減を図ります。抑止する情報は、指定された exsort パラメタファイル、出力ファイル情報、および完了メッセージの一部です。

環境変数名	: EXSORTLOGSUPPRESS
指定値	: YES

## 14.1.2 64 ビット版 SORT EE の環境変数

### (1) EXSORT64DATEOPT

日付（英数字）項目でソートするときに最小値とする基準年を指定できます。この指定は、キーまたは比較項目に日付属性「g」を指定した場合だけ有効となります。

環境変数名	: EXSORT64DATEOPT
指定値	: 基準年度

基準年度には、西暦下 2 桁の数値を 1 ～ 2 桁の 10 進数で指定します。

基準年度はキーの比較時に指定した値を年度の先頭に位置づけ、それより小さい値には、100 を加えた値を設定します。

環境変数 EXSORT64DATEOPT および `-dateopt` パラメタの両方を省略した場合は、「70」が設定されます。「0」を指定した場合は、文字項目で指定した結果と同じとなります。コマンドでの指定方法については、「5.14 `-dateopt` パラメタ」を参照してください。

### (2) EXSORT64SORTTMPDIR

ソート用一時ファイルを割り当てるディレクトリを指定できます。この指定は、ソート機能の実行時だけ有効となります。コマンドでの指定方法については、「5.11 `-sorttmpdir` パラメタ」を参照してください。

環境変数名	: EXSORT64SORTTMPDIR
指定値	: ソート用一時ファイルのディレクトリ名

### (3) EXSORT64MERGETMPDIR

マージ用一時ファイルを割り当てるディレクトリを指定できます。この指定は、ソート機能の実行時だけ有効となります。コマンドでの指定方法については、「5.12 `-mergetmpdir` パラメタ」を参照してください。

環境変数名	: EXSORT64MERGETMPDIR
指定値	: マージ用一時ファイルのディレクトリ名

### (4) EXSORT64WORKSIZE

ソート機能で使用するワークバッファのサイズを指定します。この指定は、ソート機能の実行時だけ有効となります。

コマンドでの指定方法、優先順位、および注意事項については、「5.13 `-worksize` パラメタ」を参照してください。



環境変数名	: EXSORT64WORKSIZE
指定値	: { ワークバッファサイズ   AUTO [, 最大ワークバッファサイズ]   INCORE [, 最大ワークバッファサイズ] }

#### ワークバッファサイズ

ワークバッファサイズは M (MB) または G (GB) 単位の 1 ~ 7 桁の数字で指定します。

指定できる範囲は、1 ~ 2,048,000 です。

単位を省略した場合は、M (MB) が設定されます。

#### AUTO

自動で見積もりしたワークバッファサイズが適用されます (アウトコアソート対応)。

#### INCORE

自動で見積もりしたワークバッファサイズが適用されます (インコアソート対応)。

#### 最大ワークバッファサイズ

ソート処理で使用するワークバッファサイズの最大値を指定します。

最大ワークバッファサイズは M (MB) または G (GB) 単位の 1 ~ 7 桁の数字で指定します。

指定できる範囲は、1 ~ 2,048,000 です。

単位を省略した場合は M (MB) が設定されます。

単位、および数字を省略した場合は、128 (GB) を仮定します。

### (5) EXSORT64MSGOPT

コマンド実行時のメッセージ出力オプションを指定します。指定値と意味は、OS によって異なります。

環境変数名	: EXSORT64MSGOPT
指定値	: オプション番号

#### ● Windows

環境変数 EXSORT64MSGOPT を省略した場合は、SORT 環境ファイルの msg\_opt オプションの指定を有効とします。環境変数および SORT 環境ファイルのどちらも省略した場合は、0 (ゼロ) を設定します。各オプションの意味を次の表に示します。

オプション番号	意味
0	すべてのメッセージを画面に表示する。
1	エラーメッセージだけを画面に表示する。

## 14. 環境変数の設定

オプション番号	意味
2	すべてのメッセージを画面に表示しない。
3	メッセージを画面に表示しないで、環境変数 EXSORT64MSGFILE または環境ファイルの msg_file オプションで指定したファイルに、すべてのメッセージを出力する。
4	メッセージを画面に表示しないで、環境変数 EXSORT64MSGFILE または環境ファイルの msg_file オプションで指定したファイルに、エラーメッセージだけを出力する。
5	イベントログファイルにすべてのメッセージを出力する。
6	イベントログファイルにエラーメッセージだけを出力する。
7	標準エラーファイルにすべてのメッセージを出力する。
8	標準エラーファイルにエラーメッセージだけを出力する。
9	すべてのメッセージを標準出力ファイル※に出力する。
10	エラーメッセージだけを標準出力ファイル※に出力する。
11	正常終了メッセージだけを標準出力ファイル※に出力する。
12	正常終了メッセージだけを画面に表示する。
13	正常終了メッセージだけを環境変数 EXSORT64MSGFILE または環境ファイルの msg_file オプションで指定したファイルに出力する。
14	正常終了メッセージだけをイベントログファイルに出力する。
15	正常終了メッセージだけを標準エラーファイルに出力する。
16	正常終了メッセージを標準出力ファイル※に、エラーメッセージを標準エラーファイルに出力する。

### 注※

出力ファイルが標準出力ファイルでない場合だけ、メッセージを標準出力ファイルに出力します。出力ファイルが標準出力ファイルの場合は、標準エラーファイルに表示します。

### ● UNIX

指定を省略した場合は、7を設定します。各オプションの意味を、次の表に示します。

オプション番号	意味
2	すべてのメッセージを画面に表示しない。
3	環境変数 EXSORT64MSGFILE で指定したファイルに、すべてのメッセージを出力する。
4	環境変数 EXSORT64MSGFILE で指定したファイルに、エラーメッセージだけを出力する。
7	標準エラーファイルにすべてのメッセージを出力する。
8	標準エラーファイルにエラーメッセージだけを出力する。
9	すべてのメッセージを標準出力ファイル※に出力する。

オプション番号	意味
10	エラーメッセージだけを標準出力ファイル <sup>※</sup> に出力する。
11	正常終了メッセージだけを標準出力ファイル <sup>※</sup> に出力する。
13	正常終了メッセージだけを環境変数 EXSORT64MSGFILE で指定したファイルに出力する。
15	正常終了メッセージだけを標準エラーファイルに出力する。
16	正常終了メッセージを標準出力ファイル <sup>※</sup> に、エラーメッセージを標準エラーファイルに出力する。

## 注※

出力ファイルが標準出力ファイルでない場合だけ、メッセージを標準出力ファイルに出力します。出力ファイルが標準出力ファイルの場合は、標準エラーファイルに出力します。

オプション番号「3」または「4」を指定したときの、ファイル出力時のメッセージ形式を次に示します。

```
yyyy/mm/dd △ hh:mm:ss △ [SORT △ EE(64)] △ [xx··xx] △ KBLSnxxx-x △ zzz··zzz
```

## (凡例)

△ : 1文字の全角スペース

yyyy/mm/dd : メッセージ発行日 (年 / 月 / 日)

hh:mm:ss : メッセージ発行時刻 (時 : 分 : 秒)

[SORT △ EE(64)] : 製品略称

[xx··xx] : コマンドを実行したコンピュータ名称

KBLSnxxx-x : メッセージ ID

zzz··zzz : メッセージテキスト

## (6) EXSORT64MSGFILE

コマンド実行時のメッセージの出力先ファイル名を指定します。この指定は、環境変数 EXSORT64MSGOPT で「3」または「4」を指定した場合にだけ有効となります。

```
環境変数名 : EXSORT64MSGFILE
指定値 : メッセージの出力先ファイル名
```

環境変数 EXSORT64MSGOPT が指定されている場合は、この指定を有効とします。環境変数 EXSORT64MSGOPT を省略して、SORT 環境ファイル msg\_opt の指定を有効とした場合、SORT 環境ファイル msg\_file オプションに指定されたファイル名を有効とします。

## 14. 環境変数の設定

### (7) EXSORT64MGTOPT

チェック機能の緩和に関するオプションを指定します。各オプションの指定順序は任意です。複数のオプションを指定する場合、コンマ (,) で区切って指定します。コンマの前後に空白またはタブを入れないでください。

#### 【指定例】

```
EXSORT64MGTOPT=KEY_INVALID_NOCHK,SUM_INVALID_NOCHK
```

次の (a) ~ (e) に示すオプションには、対応するコマンドオプションが用意されています。コマンドの指定は EXSORT64MGTOPT よりも優先度が高いため、これらのオプションのうち、どれかを `exsort64` コマンドに指定した場合、EXSORT64MGTOPT の指定は `exsort64` コマンド指定に置き換わります。例えば、

「EXSORT64MGTOPT=KEY\_INVALID\_NOCHK,DATE\_INVALID\_NOCHK」が指定されているシステムで、「`-cmdopt SUMNOCHK`」を指定した `exsort64` コマンドを実行した場合、コマンド指定の「SUMNOCHK」だけが有効となります。

コマンドオプションについては、「表 14-1 SORT EE に関する環境変数 (Windows の場合)」または「表 14-2 SORT EE に関する環境変数 (UNIX の場合)」を参照してください。

#### (a) KEY\_INVALID\_NOCHK オプション

キーまたは比較項目の属性が外部 10 進数または内部 10 進数の場合、入力データの形式が不正であってもエラーにしないオプションです。機能の詳細は、「5.5.9 10 進キーのチェックオプション (KEYNOCHK オプション)」を参照してください。

```
環境変数名 : EXSORT64MGTOPT  
指定値 : KEY_INVALID_NOCHK
```

#### (b) SUM\_INVALID\_NOCHK オプション

集約項目の属性が外部 10 進数または内部 10 進数の場合、入力データの形式が不正であってもエラーにしないオプションです。

機能の詳細は、「5.5.10 10 進集約項目のチェックオプション (SUMNOCHK オプション)」を参照してください。

```
環境変数名 : EXSORT64MGTOPT  
指定値 : SUM_INVALID_NOCHK
```

#### (c) VKEY\_POS\_NOCHK オプション

キーの指定範囲のチェックを緩和するオプションです。

- 可変長順編成ファイルのキーの指定範囲が最小レコード長を超えていても、エラーにしません。
- テキストファイルのキーの指定範囲が実レコード長を超えていても、エラーにしません。

機能の詳細は、「5.5.11 キー位置のチェックオプション (POSNOCHK オプション)」を参照してください。

環境変数名	: EXSORT64MGTOPT
指定値	: VKEY_POS_NOCHK

#### (d) DATE\_INVALID\_NOCHK オプション

年号に空白, Low\_value (x'0000'), High\_value (x'FFFF') などの値が設定されていても、エラーとしないオプションです。

機能の詳細は、「5.5.12 日付データのチェックオプション (DATENOCHK オプション)」を参照してください。

環境変数名	: EXSORT64MGTOPT
指定値	: DATE_INVALID_NOCHK

#### (e) DATETIME\_INVALID\_NOCHK オプション

キーまたは比較項目の属性が日付時刻の場合、入力データの形式が不正であってもエラーにしないオプションです。

機能の詳細は、「5.5.13 日付時刻データのチェックオプション (DATETIMENOCHK オプション)」を参照してください。

環境変数名	: EXSORT64MGTOPT
指定値	: DATETIME_INVALID_NOCHK

#### (f) LOCK\_INVALID オプション

SORT, SORT EE, COBOL, ISAM プログラムの処理間で、入出力ファイルのオープンモードを指定します。この指定は UNIX だけ有効となります。

機能の詳細は、「5.5.14 入出力ファイル共用・排他オプション (NOLOCK | EXC オプション)」を参照してください。

環境変数名	: EXSORT64MGTOPT
指定値	: LOCK_INVALID

### (8) EXSORT64DUPOPT

重複レコードの入力順序保存機能を有効とします。この指定は、ソート機能の実行時だ

## 14. 環境変数の設定

け有効となります。

環境変数名	: EXSORT64DUPOPT
指定値	: YES

### (9) EXSORT64IOSIZE

入力ファイル，出力ファイル，ソート用一時ファイル，およびマージ用一時ファイルに対する1回のI/Oサイズを指定します。

環境変数名	: EXSORT64IOSIZE
指定値	: <i>I/O サイズ</i>

I/O サイズは10進数（単位はKB）で指定します。

指定できる範囲は，8～32,768です。

### (10) EXSORT64INPIOSIZE

入力ファイルに対する1回のI/Oサイズを指定します。

環境変数名	: EXSORT64INPIOSIZE
指定値	: <i>I/O サイズ</i>

I/O サイズは10進数（単位はKB）で指定します。

指定できる範囲は，8～32,768です。

### (11) EXSORT64OUTIOSIZE

出力ファイルに対する1回のI/Oサイズを指定します。

環境変数名	: EXSORT64OUTIOSIZE
指定値	: <i>I/O サイズ</i>

I/O サイズは10進数（単位はKB）で指定します。

指定できる範囲は，8～32,768です。

### (12) EXSORT64SORTTMPPIOSIZE

ソート用一時ファイルに対する1回のI/Oサイズを指定します。

環境変数名	: EXSORT64SORTTMPPIOSIZE
指定値	: <i>I/O サイズ</i>

I/O サイズは10進数（単位はKB）で指定します。

指定できる範囲は、8～32,768 です。

### (13) EXSORT64MERGETMPIOSIZE

マージ用一時ファイルに対する1回のI/Oサイズを指定します。

環境変数名	: EXSORT64MERGETMPIOSIZE
指定値	: I/O サイズ

I/Oサイズは10進数(単位はKB)で指定します。

指定できる範囲は、8～32,768 です。

### (14) EXSORT64\_HI\_WORKSIZE

ソート機能で使用するワークバッファのサイズを指定します。

exsort64 コマンドの `-worksize` パラメタ、または環境変数「EXSORT64WORKSIZE」の指定があっても、環境変数「EXSORT64\_HI\_WORKSIZE」で指定された値を、ワークサイズとして置き換えます。

なお、この指定はソート機能の実行時だけ有効となります。

環境変数名	: EXSORT64_HI_WORKSIZE
指定値	: { ワークバッファサイズ   AUTO [ , 最大ワークバッファサイズ ]   INCORE [ , 最大ワークバッファサイズ ] }

指定値は、「14.1.2 (4) EXSORT64WORKSIZE」を参照してください。

### (15) EXSORT64BOMLNGIN

Unicode シグニチャ (以降、BOM と呼びます) の読み飛ばし機能を使用できます。

環境変数名	: EXSORT64BOMLNGIN
指定値	: UNICODE

- この環境変数を指定すると、すべての入力ファイルの BOM を読み飛ばします。
- 環境変数「EXSORT64BOMGETPUT」と、この環境変数のどちらの指定もない場合、入力ファイルの BOM を読み飛ばさずに、レコードデータとして処理します。
- 出力ファイルに BOM を付けたい場合は、環境変数「EXSORT64BOMLNGOUT」の指定が必要です。環境変数「EXSORT64BOMLNGOUT」については、「14.1.2(16) EXSORT64BOMLNGOUT」を参照してください。
- 出力ファイルの BOM の種類と有無を、入力ファイルに合わせたい場合は、環境変数「EXSORT64BOMGETPUT」を指定します。環境変数「EXSORT64BOMLNGOUT」については、「14.1.2(17) EXSORT64BOMGETPUT」を参照してください。

## 14. 環境変数の設定

- この環境変数に「UNICODE」を指定すると、次の形式の BOM を読み飛ばします。

BOM 形式	バイト長	説明
0xEFBBBF	3	UTF-8
0xFFFE	2	UTF-16LE (UTF-16 little endian)
0xFEFF	2	UTF-16BE (UTF-16 big endian)
0xFFFE0000	4	UTF-32LE (UTF-32 little endian)
0x000FEFF	4	UTF-32BE (UTF-32 big endian)

### (16) EXSORT64BOMLNGOUT

Unicode シグニチャ（以降、BOM と呼びます）の付加機能を使用できます。

環境変数名	: EXSORT64BOMLNGOUT
指定値	: {UTF8   UTF16LE   UTF16BE   UTF32LE   UTF32BE}

- 環境変数「EXSORT64BOMGETPUT」の指定がある場合、この環境変数の指定は無視されます。環境変数「EXSORT64BOMGETPUT」については、「14.1.2(17) EXSORT64BOMGETPUT」を参照してください。
- 環境変数「EXSORT64BOMGETPUT」と、この環境変数のどちらの指定もない場合、出力ファイルには BOM は付けられません。
- 入力ファイルの BOM を読み飛ばしたい場合は、環境変数「EXSORT64BOMLNGIN」または「EXSORT64BOMGETPUT」の指定が必要です。環境変数「EXSORT64BOMLNGIN」については、「14.1.2(15) EXSORT64BOMLNGIN」を、環境変数「EXSORT64BOMGETPUT」については、「14.1.2(17) EXSORT64BOMGETPUT」を参照してください。
- この環境変数の指定値に対応する BOM を、出力ファイルに付けます。指定値に対応する BOM 形式を次に示します。

指定値	BOM 形式	バイト長	説明
UTF8	0xEFBBBF	3	UTF-8
UTF16LE	0xFFFE	2	UTF-16 little endian
UTF16BE	0xFEFF	2	UTF-16 big endian
UTF32LE	0xFFFE0000	4	UTF-32 little endian
UTF32BE	0x000FEFF	4	UTF-32 big endian

### (17) EXSORT64BOMGETPUT

Unicode シグニチャ（以降、BOM と呼びます）の読み飛ばしまたは付加機能を使用でき



ます。

環境変数名	: EXSORT64BOMGETPUT
指定値	: AUTOBOM

- この環境変数の指定は、環境変数「EXSORT64BOMLANGOUT」よりも優先されます。環境変数「EXSORT64BOMLANGOUT」については、「14.1.2(16) EXSORT64BOMLANGOUT」を参照してください。
- SORT EE で使用できる BOM 形式を次に示します。

BOM 形式	バイト長	説明
0xEFBBBF	3	UTF-8
0xFFFE	2	UTF-16LE (UTF-16 little endian)
0xFEFF	2	UTF-16BE (UTF-16 big endian)
0xFFFE0000	4	UTF-32LE (UTF-32 little endian)
0x0000FEFF	4	UTF-32BE (UTF-32 big endian)

- この環境変数を指定した場合の動作は次のとおりです。
  - すべての入力ファイルの BOM を読み飛ばします。
  - 入力ファイルの BOM と同じ形式の BOM を、すべての出力ファイルに付けます。入力ファイルが複数個の場合は、先頭の入力ファイルの BOM 形式を付けます。
  - 入力ファイルに BOM がなければ、すべての出力ファイルに BOM を付けません。入力ファイルが複数個の場合は、先頭の入力ファイルの BOM の有無が該当します。2 個目以降の入力ファイルが BOM ありでも、先頭の入力ファイルに BOM がなければ出力ファイルには付けません。

環境変数の指定の有無による、入力ファイルの BOM の読み飛ばし、および出力ファイルへの BOM の付加を次の表に示します。

表 14-9 環境変数の指定の有無による、入力ファイルの BOM の読み飛ばし、および出力ファイルへの BOM の付加（先頭の入力ファイルに BOM がある場合）

環境変数の指定の有無			BOM の読み飛ばし	BOM の付加
EXSORTBOMLANGIN	EXSORTBOMLANGOUT	EXSORTBOMGETPUT		
無	無	無	—	×
有	無	無	○	×
有	有	無	○	◎
有	無	有	○	●
有	有	有	○	●
無	有	無	—	◎

## 14. 環境変数の設定

環境変数の指定の有無			BOM の読み 飛ばし	BOM の 付加
EXSORTBOMLNGIN	EXSORTBOMLN GOUT	EXSORTBOMG ETPUT		
無	有	有	○	●
無	無	有	○	●

表 14-10 環境変数の指定の有無による、入力ファイルの BOM の読み飛ばし、および出力ファイルへの BOM の付加（先頭の入力ファイルに BOM がない場合）

環境変数の指定の有無			BOM の読み 飛ばし	BOM の 付加
EXSORTBOMLNGIN	EXSORTBOMLN GOUT	EXSORTBOMG ETPUT		
無	無	無	—	×
有	無	無	—	×
有	有	無	—	◎
有	無	有	—	×
有	有	有	—	×
無	有	無	—	◎
無	有	有	—	×
無	無	有	—	×

（凡例）

無：環境変数を指定しない

有：環境変数を指定する

○：入力ファイルの BOM を読み飛ばす

—：入力ファイルの BOM を読み飛ばさない

◎：出力ファイルに BOM を付ける（EXSORTBOMLNGOUT に指定した BOM 形式）

●：出力ファイルに BOM を付ける（先頭の入力ファイルの BOM 形式）

×：出力ファイルに BOM を付けない

### (18) EXSORT64STRINGDIR

ソート用ワークバッファが不足した場合に、ワークバッファ内のストリング情報を退避する一時ファイルを作成するディレクトリを指定します。この環境変数を指定しない場合、ソート用一時ファイルのディレクトリに、ストリング情報格納一時ファイルが作成されます。この指定は、ソート機能の実行時だけ有効となります。また、ワークバッファが不足した場合にだけ、一時ファイルを作成します。

ワークバッファが不足してストリング情報格納一時ファイルを作成した場合、「KBLS1133-I ソート用ワークバッファサイズが不足しています。」のメッセージを出力

します。KBL51133-I メッセージが出力された場合の原因と対処方法の詳細は、「付録 K.4 SORT のメッセージ一覧」を参照してください。

環境変数名	: EXSORT64STRINGDIR
指定値	: スtring情報格納一時ファイルディレクトリ

### (19) EXSORT64SUPPRESSPLITMSG

ファイル分割終了時、KBL51010-I メッセージの出力を抑止します。この指定は、ファイル分割機能だけで有効となります。

環境変数名	: EXSORT64SUPPRESSPLITMSG
指定値	: YES

### (20) EXSORT64SUPPRESSCATMSG

ファイル連結終了時、KBL51011-I メッセージの出力を抑止します。この指定は、ファイル連結機能だけで有効となります。

環境変数名	: EXSORT64SUPPRESSCATMSG
指定値	: YES

### (21) EXSORT64NOCREATEZEROFILE

ファイル分割後の出力ファイルに格納するレコード件数が 0 件の場合、出力ファイルの作成を抑止します。また、0 件レコードであったことを示す KBL51010-I メッセージも抑止します。この指定は、ファイル分割機能だけで有効となります。

環境変数名	: EXSORT64NOCREATEZEROFILE
指定値	: YES

- 環境変数 EXSORT64NOCREATEZEROFILE は、ファイル分割機能使用時に `-cmdopt` パラメタの `OVWRT`、および `INPOVWRT` オプションと同時に指定できません。
- 環境変数 EXSORT64OUTFILENODUPCHK は同時に指定できません。

### (22) EXSORT64CATINPNOERR

`-catinp` パラメタの `file` オペランドで指定した入力ファイルの一部が存在しない場合でも、連結処理を続行します。この指定は、ファイル連結機能だけで有効となります。

環境変数名	: EXSORT64CATINPNOERR
指定値	: YES

## 14. 環境変数の設定

### (23) EXSORT64FILESPLITNUMB

ファイル分割時、出力ファイル名に付加する追番の桁数を指定します。この指定がない場合は、追番の桁数は 8 を設定します。

環境変数名	: EXSORT64FILESPLITNUMB
指定値	: <i>追番の桁数</i>

追番の桁数は 1 桁の数字で指定します。

指定できる値は、1 ～ 8 です。

### (24) EXSORT64SPLITNUMBONE

ファイル分割時、分割キーを指定しても出力ファイル名に付加する追番の個数は一個とします。

環境変数名	: EXSORT64SPLITNUMBONE
指定値	: YES

### (25) EXSORT64SPLITTMPDIR

ファイル分割用一時ファイルのディレクトリを指定します。この指定は、均等分割情報によるファイル分割時だけで有効となります。

環境変数名	: EXSORT64SPLITTMPDIR
指定値	: <i>ファイル分割用一時ファイルのディレクトリ名</i>

### (26) EXSORT64SPLITTMPIOSIZE

ファイル分割用一時ファイルに対する 1 回の I/O サイズを指定します。

環境変数名	: EXSORT64SPLITTMPIOSIZE
指定値	: <i>I/O サイズ</i>

I/O サイズは 10 進数（単位は KB）で指定します。

指定できる範囲は、8 ～ 32,768 です。

### (27) EXSORT64CATNAMESEQ (Windows の場合)

ファイル連結時、`-catinp` パラメタで `suffix` または `prefix` オペランドを指定した場合の入力ファイルの入力順対応コードを指定します。

環境変数名	: EXSORT64CATNAMESEQ
指定値	: <i>入力ファイルの入力順対応コード</i>

入力ファイルの入力順対応コードの指定値と意味については、「5.26.4 nameseq オペランド (Windows の場合)」を参照してください。

## (28) EXSORT64OVFERROPT

集約機能で集約結果がオーバーフローした場合にエラーにします。

この指定は、集約機能だけ有効となります。

環境変数名	: EXSORT64OVFERROPT
指定値	: YES

## (29) EXSORT64GOPTION

CSV 日付属性「g」の桁修飾文字を変更するオプションです。

環境変数名	: EXSORT64GOPTION
指定値	: YES

- このオプションの指定有無によって、「年月日 {T | 空白} 時分秒」形式の入力データの扱いが異なります。詳細を次の表に示します。

EXSORTGOPTION オプションの指定有無	日付と時刻の切り分け	日付データの 桁修飾文字
指定なし	T および空白を日付と時刻の区切り文字として扱いません。	ハイフン、スラント
指定あり	T および空白を日付と時刻の区切り文字として扱います。「年月日」部分の日付データだけを有効とし、「時分秒」部分の時刻データは無視します。区切り文字 {T   空白} がない場合、入力データを「年月日」の日付データとして扱います。	ハイフン、スラント、およびピリオド

- このオプションは、CSV 日付属性「g」を指定したときだけ有効です。
- このオプションの有無にかかわらず、CSV 日付属性「g」では、日付またがり調整しません。日付またがり調整したい場合は、CSV 日付時刻属性「h」を使用してください。

### 日付またがりの例

「2013-9-31」は翌月に、「2013-13-1」は翌年にまたがるデータですが、CSV 日付属性「g」ではまたがり調整しないで、入力データをそのまま比較します。なお、CSV 日付時刻属性「h」では、「2013-9-31」を翌月 1 日の「2013-10-1」として、「2013-13-1」を翌年の「2014-1-1」として扱います。

## (30) EXSORT64PACKSIGNTYPE

内部 10 進数の符号の範囲を変更するオプションです。

## 14. 環境変数の設定

環境変数名 : EXSORT64PACKSIGNTYPE  
 指定値 : {SE | MF}

- このオプションは、属性コード「p」で指定した内部 10 進数の符号に対してだけ有効です。
- この環境変数の指定がない場合、環境変数ファイル「EXS\_PACK\_SIGNTYPE」の指定を仮定します。
- この環境変数と、環境変数ファイル「EXS\_PACK\_SIGNTYPE」の両方の指定がない場合は、「SE」を仮定します。
- 環境ファイルの指定方法については、環境ファイル「15.1.1(22) EXS\_PACK\_SIGNTYPE」を参照してください。

オプション指定	符号	キー・比較項目	集約項目	集約結果	KEYNOCHK 指定時の不正データの扱い
SE	正	C	C	C	<ul style="list-style-type: none"> <li>• 数値部が 0～9 以外の場合、JIS8 単位コード体系に従って比較する。</li> <li>• 符号部が C, D, F 以外の場合は、内部 10 進数を正值として扱う。</li> </ul>
	負	D	D	D	
	符号なし (絶対値)	F※1	F※1	C (F※2)	
MF	正	0, 2, 4, 6, 8, A, C, E	A, C, E	C	<ul style="list-style-type: none"> <li>• 数値部が 0～9 以外の場合、JIS8 単位コード体系に従って比較する。</li> </ul>
	負	1, 3, 5, 7, 9, B, D	B, D	D	
	符号なし (絶対値)	F※1	F※1	C (F※2)	

### 注※1

絶対値 (F) は正值として扱う。

### 注※2

内部 10 進数符号保障オプション (-cmdopt ABS) の指定があり、最初の集約項目の演算符号が F であるとき、以降の集約項目の集約結果に対して同じ演算符号 F を適用する。

## (31) EXSORT64UNPACKDATATYPE

外部 10 進数のデータ形式を変更するオプションです。

環境変数名 : EXSORT64UNPACKDATATYPE  
 指定値 : {ASCII-1 | ASCII-2 | EBCDIK}

- このオプションが有効な属性コードは次のとおりです。

z (外部 10 進数)  
 x (左符号付き外部 10 進数)  
 s (右分離符号付き外部 10 進数)  
 t (左分離符号付き外部 10 進数)  
 d (左分離符号付き数字)  
 g (日付) で指定した 2 桁年号部分

- CSV の属性コード「z (固定小数点数)」、**「g (日付)」** および **「h (日付時刻)」** には適用されません。指定しても無効となります。
- 入力データのエンコードに「UCS-2LE」、**「UCS-2BE」**、**「UTF-16LE」**、**「UTF-16BE」**、**「UCS-4LE」**、**「UCS-4BE」**、**「UTF-32LE」**、**「UTF-32BE」** または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE、UTF-16BE、UTF-32LE または UTF-32BE)」を指定した場合、このオプションは無効となります。
- この環境変数の指定がない場合、環境変数ファイル「EXS\_UNPACK\_DATATYPE」の指定を仮定します。
- この環境変数と、環境変数ファイル「EXS\_UNPACK\_DATATYPE」の両方の指定がない場合は、「ASCII-1」を仮定します。
- 環境ファイルの指定方法については、環境ファイル「15.1.1(23) EXS\_UNPACK\_DATATYPE」を参照してください。
- 指定したデータ形式に合致しない場合、不正データ形式で KBL51211-E エラーとなります。
- 10 進数の不正データをエラーにしない「10 進キーのキーチェック緩和機能オプション (-cmdopt KEYNOCHK)」と一緒に指定できます。KEYNOCHK オプションを指定した場合の不正データの扱いを表 14-12、表 14-13 に示します。
- 日付の不正データをエラーにしない「日付データのチェック緩和機能オプション (-cmdopt DATENOCHK)」と一緒に指定できます。DATENOCHK オプションを指定した場合の不正データの扱いを表 14-14 に示します。

## 14. 環境変数の設定

表 14-11 外部 10 進データ形式

数値		オプション指定		
		ASCII-1	ASCII-2	EBCDIK
数字 (絶対値)	0	0 (0x30)	0 (0x30)	0 (0xF0)
	1	1 (0x31)	1 (0x31)	1 (0xF1)
	2	2 (0x32)	2 (0x32)	2 (0xF2)
	3	3 (0x33)	3 (0x33)	3 (0xF3)
	4	4 (0x34)	4 (0x34)	4 (0xF4)
	5	5 (0x35)	5 (0x35)	5 (0xF5)
	6	6 (0x36)	6 (0x36)	6 (0xF6)
	7	7 (0x37)	7 (0x37)	7 (0xF7)
	8	8 (0x38)	8 (0x38)	8 (0xF8)
	9	9 (0x39)	9 (0x39)	9 (0xF9)
正值	+0	0 (0x30)	{ (0x7B)	{ (0xC0)
	+1	1 (0x31)	A (0x41)	A (0xC1)
	+2	2 (0x32)	B (0x42)	B (0xC2)
	+3	3 (0x33)	C (0x43)	C (0xC3)
	+4	4 (0x34)	D (0x44)	D (0xC4)
	+5	5 (0x35)	E (0x45)	E (0xC5)
	+6	6 (0x36)	F (0x46)	F (0xC6)
	+7	7 (0x37)	G (0x47)	G (0xC7)
	+8	8 (0x38)	H (0x48)	H (0xC8)
	+9	9 (0x39)	I (0x49)	I (0xC9)
負数	-0	p (0x70)	} (0x7D)	} (0xD0)
	-1	q (0x71)	J (0x4A)	J (0xD1)
	-2	r (0x72)	K (0x4B)	K (0xD2)
	-3	s (0x73)	L (0x4C)	L (0xD3)
	-4	t (0x74)	M (0x4D)	M (0xD4)
	-5	u (0x75)	N (0x4E)	N (0xD5)
	-6	v (0x76)	O (0x4F)	O (0xD6)
	-7	w (0x77)	P (0x50)	P (0xD7)
	-8	x (0x78)	Q (0x51)	Q (0xD8)
	-9	y (0x79)	R (0x52)	R (0xD9)



数値		オプション指定		
		ASCII-1	ASCII-2	EBCDIK
分離符号	正	+ (0x2B)	+ (0x2B)	+ (0x4E)
	負	- (0x2D)	- (0x2D)	- (0x60)

表 14-12 外部 10 進数 (z, x) 不正データの扱い

オプション指定値	KEYNOCHK 指定時の不正データの扱い
ASCII-1	<ul style="list-style-type: none"> <li>ゾーン部は比較対象外です。</li> <li>数値部が 0 ~ 9 以外の場合、JIS8 単位コード体系に従って比較します。</li> <li>符号部が 3, 7 以外の場合、外部 10 進数を正值として扱います。</li> </ul>
ASCII-2	<ul style="list-style-type: none"> <li>ゾーン部は比較対象外です。</li> <li>数値部が 0 ~ 9 以外の場合、JIS8 単位コード体系に従って比較します。</li> <li>符号部が 4, 5 以外の場合、外部 10 進数を正值として扱います。</li> </ul>
EBCDIK	<ul style="list-style-type: none"> <li>ゾーン部は比較対象外です。</li> <li>数値部が 0 ~ 9 以外の場合、EBCDIK コード体系に従って比較します。</li> <li>符号部が C, D 以外の場合、外部 10 進数を次のとおり扱います。 (正) 0, 2, 4, 6, 8, A, C, E, F (負) 1, 3, 5, 7, 9, B, D</li> </ul>

表 14-13 分離符号付き外部 10 進数 (s, t) 不正データの扱い

オプション指定値	KEYNOCHK 指定時の不正データの扱い
ASCII-1 ASCII-2	<ul style="list-style-type: none"> <li>数値部が 0x30 ~ 0x39 以外の場合、JIS8 単位コード体系に従って比較します。</li> <li>符号部が 0x2B, 0x2D 以外の場合、外部 10 進数を正值として扱います。</li> </ul>
EBCDIK	<ul style="list-style-type: none"> <li>数値部が 0xF0 ~ 0xF9 以外の場合、EBCDIK コード体系に従って比較します。</li> <li>符号部が 0x4E, 0x60 以外の場合、外部 10 進数を正值として扱います。</li> </ul>

表 14-14 日付 (g) 不正データの扱い

オプション指定値	DATENOCHK 指定時の不正データの扱い
ASCII-1 ASCII-2	<ul style="list-style-type: none"> <li>キーの先頭 2 文字 (西暦下 2 桁の数値) が指すデータ内容が 0x30 ~ 0x39 以外の場合、JIS8 単位コード体系に従って比較します。</li> </ul>
EBCDIK	<ul style="list-style-type: none"> <li>キーの先頭 2 文字 (西暦下 2 桁の数値) が指すデータ内容が 0xF0 ~ 0xF9 以外の場合、EBCDIK コード体系に従って比較します。</li> </ul>

### (32) EXSORT64DELBLANKLINE

改行だけのデータがないレコード (空行) を削除して入力できるオプションです。

## 14. 環境変数の設定

このオプションは、入力ファイル編成がテキストファイル、または CSV ファイルのときに有効となります。入力ファイル編成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は、指定しても無視されます。

環境変数名	: EXSORT64DELBLANKLINE
指定値	: YES

### (33) EXSORT64OUTFILENUMUNLIMIT 【08-50 以降】

出力ファイル (-outfile) パラメタの指定個数を制限なしにします。

指定できる -outfile パラメタの最小値および最大値については、「3.3.1 パラメタ」の表 3-5 を参照してください。環境ファイルの指定方法については、環境ファイル「15.1.1(24) EXS\_OUTFILENUMUNLIMIT 【08-50 以降】」を参照してください。

環境変数名	: EXSORT64OUTFILENUMUNLIMIT
指定値	: YES

### (34) EXSORT64PRMSIZEUNLIMIT 【08-50 以降】

exsort パラメタファイルのサイズを制限なしにします。

exsort パラメタファイルの形式については、「4.4.1 ファイル形式」を参照してください。環境ファイルの指定方法については、環境ファイル「15.1.1(25) EXS\_PRMSIZEUNLIMIT 【08-50 以降】」を参照してください。

環境変数名	: EXSORT64PRMSIZEUNLIMIT
指定値	: YES

### (35) EXSORT64OUTFILENODUPCHK 【08-50 以降】

出力ファイルパラメタ (-outfile, -catout または -splitout パラメタ) で指定された出力ファイル名の重複チェックを抑制します。

環境変数名	: EXSORT64OUTFILENODUPCHK
指定値	: YES

- 抑制する重複チェック項目は「入力ファイル名と出力ファイル名」、「出力ファイル名同士」および「exsort パラメタファイル名と出力ファイル名」です。「入力ファイル名と exsort パラメタファイル名」の重複はチェックします。
- ファイルの上書きオプション (-cmdopt OVWRT) との同時指定を必須とします。OVWRT オプションの指定がない場合、KBLS1122-E エラーとなります。
- 環境変数 EXSORT64NOCREATEZEROFILE は、同時に指定できません。

- この環境変数を指定した場合、同一名の出力ファイルを指定してもエラーにはなりません。また、同一名の出力ファイルを指定した場合、出力ファイルの内容は保証されません。

### (36) EXSORT64LOGSUPPRESS 【08-50 以降】

exsort64 コマンドのコマンドログ情報、および exsort64 コマンド実行時のリソース使用量などのレポート情報の出力を抑制し、ログ出力量の削減を図ります。抑制する情報は、指定された exsort パラメタファイル、出力ファイル情報、および完了メッセージの一部です。

環境変数名	: EXSORT64LOGSUPPRESS
指定値	: YES

### (37) EXSORT64SORTLIBOPTION 【Linux(x64) SORT EE(64) 08-41 以降】

最適化したソートライブラリを使用できます。

環境変数名	: EXSORT64SORTLIBOPTION
指定値	: オプション番号

指定を省略した場合は、「1」を設定します。

オプションの意味を次の表に示します。

オプション番号	意味
1	最適化していないソートライブラリを使用する。
2	最適化したソートライブラリを使用する。

#### 注意事項

- オプション番号「2」を指定した場合、オプション番号「1」の指定と比べて、ソート機能、マージ機能、選択機能、集約機能、およびキーによる分割機能の処理時間が短縮できる可能性があります。ただし、入力ファイル（レコード件数）、指定されたキー（ソートキー、マージキー、集約キー、分割キー、非分割データ）、または比較項目の値などによって短縮できる処理時間は変動します。そのため、オプション番号「2」を適用する際は、事前に処理時間を測定し、問題ないかどうかを検証することを推奨します。
- オプション番号「1」および「2」以外を指定してもエラーにはなりません。この場合はオプション番号「1」を設定します。

## 14.1.3 ワークサイズ設定の優先順位

ワークサイズ設定の優先順位を以下に示します。

### ● SORT EE

1. 環境変数「EXSORT\_HI\_WORKSIZE (64 ビット版は「EXSORT64\_HI\_WORKSIZE)」で指定された値
2. `-worksize` パラメタで指定された値
  - 2.1 コマンドラインで指定された `-worksize`
  - 2.2 パラメタファイルで指定された `-worksize`
3. 環境変数「EXSORTWORKSIZE (64 ビット版は「EXSORT64WORKSIZE)」で指定された値
4. Windows の場合は、SORT の環境ファイルの `work_size` オプションに設定された値  
UNIX の場合は、SORT の環境変数 SORTOPT の `-y` オプションに指定された値
5. `-worksize auto` オペランドを設定 (SORT EE の省略値)

### ● SORT

SORT の場合、SORT EE を呼び出す機能拡張オプションの有無により、設定値が異なります。機能拡張オプションがあれば SORT EE の環境変数を参照し、なければ従来どおりとなります。

#### 【機能拡張オプションあり】

1. 環境変数「EXSORT\_HI\_WORKSIZE (64 ビット版は「EXSORT64\_HI\_WORKSIZE)」で指定された値
2. Windows の場合は、`wnsort` コマンドの `/y` パラメタ、または `wnsortex` コマンドの `work_size` パラメタで指定された値  
UNIX の場合は、`-y` パラメタで指定された値
3. Windows の場合は、環境ファイルの `work_size` オプションに設定された値  
UNIX の場合は、環境変数 SORTOPT の `-y` オプションに指定された値
4. SORT EE の環境変数「EXSORTWORKSIZE (64 ビット版は「EXSORT64WORKSIZE)」で指定された値
5. `-worksize auto` オペランドを設定 (SORT EE の省略値)

#### 【機能拡張オプションなし】

1. Windows の場合は、`wnsort` コマンドの `/y` パラメタ、または `wnsortex` コマンドの `work_size` パラメタで指定された値  
UNIX の場合は、`-y` パラメタで指定された値
2. Windows の場合は、環境ファイルの `work_size` オプションに設定された値  
UNIX の場合は、環境変数 SORTOPT の `-y` オプションに指定された値
3. 1,024KB を設定 (SORT の省略値)

## 14.1.4 入出力ファイル名の指定

パラメタファイル中で定義する入出力ファイル名を、環境変数名で指定できます。「%〜%」で変数名を囲んで指定します。

## 【例 1】 exsort パラメタファイルの指定例

```

:
-inpfile file=%SORTIN%
-outfile file=%SORTOUT%
:

```

入力ファイル名に「SORTIN」、出力ファイル名に「SORTOUT」の環境変数を指定します。

## 【例 2】 環境変数の指定例（UNIX B シェル系の場合）

```

SORTIN="INPUTDATA 01:INPUTDATA 02:INPUTDATA 03"
SORTOUT=OUTDATA

```

環境変数「SORTIN」に、「INPUTDATA 01」、「INPUTDATA 02」、「INPUTDATA 03」の 3 個のファイル名を指定します。

空白を含むファイル名の場合は、ファイル名をダブルクォーテーション (") で囲んで指定します。

環境変数「SORTOUT」に、「OUTDATA」の 1 個のファイル名を指定します。

## 【例 3】 環境変数の指定例（Windows）

```

set SORTIN=INPUTDATA 01:INPUTDATA 02:INPUTDATA 03
set SORTOUT=OUTDATA

```

環境変数名に複数のファイル名を指定する場合、UNIX ではコロン (:), Windows ではセミコロン (;) でファイル名を区切って指定します。

ファイル名の書き方については、「4.3.5 ファイル名およびディレクトリ名の指定形式」を参照してください。

## 14.2 SORT の環境変数の設定（UNIX の場合）

### 14.2.1 SORT の環境変数

#### (1) SORTOPT

環境変数 SORTOPT を指定することによって、コマンド省略時に設定される値を変更できます。

SORTOPT で使用できるオプションを次の表に示します。

表 14-15 SORTOPT のオプション一覧

オプション	機能	コマンド			
		ソート	マージ	選択	集約
-y	ワークバッファサイズの初期値設定	○	×	×	×
-T	ソート用一時ファイルを割り当てるディレクトリ	○	×	×	×
-T2	マージ用一時ファイルを複数割り当てる場合のディレクトリ	○	×	×	×
-g	日付項目の基準年度	○	○	○	○
-E	選択条件値の特殊記号「¥」の有効化	×	×	○	×
-SORT_EE	機能拡張オプションを有効にする	○	○	○	○

(凡例) ○ : 有効    × : 無効

#### (a) 複数のオプションの指定例

各オプションの指定順序は任意です。複数のオプションを指定する場合、コンマ (,) で区切って指定します。コンマの前後に空白またはタブを入れないでください。

(例)

```
SORTOPT=-y2m,-T/tmp/SORTWORK,-g90
```

#### (b) -y オプション（ワークバッファサイズの初期値指定）

ワークバッファサイズは KB (k) または MB (m) 単位の 1 ~ 7 桁の数字で指定します。指定できるメモリ量を次に示します。

- レコード長（可変長ファイルの場合、最大レコード長）が 32,760 バイト以下の場合
  - 32 ビット版では 128KB 以上
  - 64 ビット版では 256KB 以上
- レコード長が 32,760 バイトを超える場合
  - 32 ビット版では 256KB 以上

- 64 ビット版では 384KB 以上

なお、指定できるメモリ量の最大値は、2,000MB です。

また、単位を省略した場合は、KB (k) が設定されます。

指定方法を次に示します。

`SORTOPT=-y`ワークバッファサイズ

(c) **-T** オプション (ソート用一時ファイルを割り当てるディレクトリ)

ソート用一時ファイルを割り当てるディレクトリを指定できます。この指定は、ソートの実行時だけ有効となります。指定方法を次に示します。

`SORTOPT=-T`ディレクトリ名

使用するディレクトリの優先順位については、「7.2.2(7) `-T` ディレクトリ名」を参照してください。

(d) **-T2** オプション (マージ用一時ファイルを割り当てるディレクトリ)

マージ用一時ファイルを割り当てるディレクトリを指定できます。この指定は、ソートの実行時だけ有効となります。指定方法を次に示します。

`SORTOPT=-T`ディレクトリ名, `-T2`ディレクトリ名2

使用するディレクトリの優先順位については、「7.2.2(8) `-T2` ディレクトリ名 2」を参照してください。

(e) **-g** オプション (日付 (英数字) 項目の基準年度)

日付 (英数字) 項目でソートするときに最小値とする基準年度を指定できます。基準年度には、西暦下 2 桁の数値を、2 桁以下の数値で指定します。基準年度に指定した年度の項目は、キーの比較時に先頭に位置づけられます。また、指定した年度の数値より小さい値には、100 を加えた値が設定されます。

指定方法を次に示します。

`SORTOPT=-g`基準年度

この指定は、キーに日付 (英数字) 項目を指定した場合だけ有効となり、省略した場合は、70 が設定されます。また、0 を指定した場合の比較結果は、文字項目で指定した結果と同じとなります。

## 14. 環境変数の設定

### (f) -E オプション (選択条件値の特殊記号「¥」の有効化)

選択機能の選択条件値に特殊記号「¥」を指定する場合「¥¥」と指定します。ただし、C シェル以外のシェルの場合、「¥¥」が認識できない場合があります。この場合は、-E オプションを指定してください。

-E オプションを指定すると、C シェル以外のシェルでも「¥¥」が認識できるようになります。

指定方法を次に示します。

```
SORTOPT=-E
```

この指定は、選択条件値に特殊記号「¥」を指定し、選択コマンドを C シェル以外のシェルで実行する場合に有効となります。選択コマンドを C シェルで実行する場合、または選択条件値に特殊記号「|」(クォーテーション)を指定し選択コマンドを実行する場合は、-E オプションを指定しないでください。

### (g) -SORT\_EE オプション (機能拡張オプションの有効)

SORT の機能拡張オプションを使用する場合、-SORT\_EE オプションを指定します。

SORT の機能拡張オプションについては、「2.21 SORT の機能拡張オプション (SORT の機能)」を参照してください。

指定方法を次に示します。

```
SORTOPT=-SORT_EE
```

## (2) SORT\_BOMLNGIN

環境変数 SORT\_BOMLNGIN を指定することによって、Unicode シグニチャの読み飛ばし機能を使用できます。

UTF-8 のテキスト編成ファイルの場合、Unicode シグニチャとして (X'EFBBBBF') を入力ファイルから読み飛ばさせるオプションに「UNICODE」を指定します。

指定方法を次に示します。

```
SORT_BOMLNGIN=UNICODE
```

## (3) SORT\_BOMLNGOUT

環境変数 SORT\_BOMLNGOUT を指定することによって、Unicode シグニチャの付加機能を使用できます。

UTF-8 のテキスト編成ファイルの場合、Unicode シグニチャとして (X'EFBBBBF') を出



カファイルに付加するオプションに「UTF8」を指定します。

指定方法を次に示します。

```
SORT_BOMLNGOUT=UTF8
```

#### (4) SORT\_STRINGDIR

環境変数 `SORT_STRINGDIR` を指定することによって、ソート機能が「KBLS290-E メモリが不足しました。」のエラーで異常終了しない機能を使用できます。

ソート用ワークバッファ内の `SORT` 情報を一時ファイルに格納するための一時ファイルディレクトリを指定します。

この指定は、ソート機能の実行だけ有効となります。メッセージが表示された場合の原因と対処方法の詳細は、「付録 K.4 SORT のメッセージ一覧」を参照してください。

指定方法を次に示します。

```
SORT_STRINGDIR= ストリング情報格納一時ファイルディレクトリ
```

#### (5) SORT\_DMPDIR

`SORT` のコマンド処理が異常終了した場合に作成されるデバッグ情報ファイルの出力先ディレクトリを指定します。デバッグ情報ファイルの詳細は、「19.2.2 デバッグ情報ファイル」を参照してください。

指定方法を次に示します。

```
SORT_DMPDIR= デバッグ情報ファイルの出力先ディレクトリ名
```



# 15 環境ファイルの設定

この章では、SORT EE および SORT の環境ファイルの設定について説明します。

---

15.1 SORT EE の環境ファイルの設定

---

15.2 SORT の環境ファイルの設定（Windows の場合）

---

## 15.1 SORT EE の環境ファイルの設定

SORT EE 実行時、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」の内容に従い SORT EE の実行環境を初期化します。

### 15.1.1 環境ファイルに指定できる項目

環境ファイルに指定できる項目を次の表に示します。

表 15-1 Exsortconf.ini ファイルに設定できる項目

項番	項目	意味
1	EXS_LOGPATH	exsort コマンドのログファイル「ExSORT.log」の出力先ディレクトリ名を指定します。ログファイルについては、「19.1.3 ログファイル」を参照してください。
2	EXS_ALTSEQPATH	altseq 変換ファイル「exsort_altseq.txt ファイル (64 ビット版は「exsort64_altseq.txt ファイル)」が格納されているディレクトリ名を指定します。altseq 変換ファイルについては「5.15 altseq パラメタ」、または「付録 G altseq 変換ファイル (SORT EE の機能)」を参照してください。
3	EXS_COMPLETE_MSGOPT	完了メッセージオプションを指定します。完了メッセージオプションについては「19.1.4 完了メッセージオプション」を参照してください。
4	EXS_RECSEPA	入力レコードの改行文字を指定します。テキストファイルまたは CSV ファイルを指定した場合だけ有効です。
5	EXS_CELLSEPA	入力レコードのセルのデリミタ (区切り文字) を指定します。CSV ファイルを指定した場合だけ有効です。
6	EXS_QUOTTYPE	セルの引用符文字を指定します。CSV ファイルを指定した場合だけ有効です。
7	EXS_CURRENCY	固定小数点の数値データに付ける通貨記号として、任意の 1 文字を指定します。CSV ファイルを指定した場合だけ有効です。
8	EXS_ROUND	固定小数点の数値データの小数部の丸め方を指定します。CSV ファイルを指定した場合だけ有効です。
9	EXS_NULLSEQ	ナル文字列が含まれる入力レコードを、先頭または末尾に配置するオプションを指定します。CSV ファイルを指定した場合だけ有効です。
10	EXS_OUTRECSEPA	出力レコードの改行文字を指定します。テキストファイルまたは CSV ファイルを指定した場合だけ有効です。
11	EXS_PADDING	出力レコード長をそろえるためのパディング文字を指定します。
12	EXS_OUTCELLSEPA	出力レコードのセルのデリミタ (区切り文字) を指定します。CSV ファイルを指定した場合だけ有効です。

項番	項目	意味
13	EXS_EXSORTDMP_OPT	exsortrepo.log ファイルに出力するレポートログに関するオプションです。レポートログのオプションについては、「19.1.3 ログファイル」を参照してください。
14	EXS_KEYNUMMAX	-key パラメタに指定できるキーの最大個数と、合計長の倍数を指定します。
15	EXS_SUMNUMMAX	-sum パラメタに指定できる集約項目の最大個数と、合計長の倍数を指定します。
16	EXS_SELNUMMAX	include と omit オペランドに指定できる選択条件式の最大個数と、比較項目の合計長の倍数を指定します。
17	EXS_CSVKEYNUMMAX	-csvkey パラメタに指定できる CSV キーの最大個数と、合計長の倍数を指定します。
18	EXS_CSVSUMNUMMAX	-csvsum パラメタに指定できる CSV 集約項目の最大個数と、合計長の倍数を指定します。
19	EXS_CSVSELNUMMAX	csvinclude と csvomit オペランドに指定できる CSV 選択条件式の最大個数と、比較項目の合計長の倍数を指定します。
20	EXS_REPOLOG_FILESIZE	exsort コマンドのレポートログファイル「exsortrepo.log」のファイルサイズを MB 単位で指定します。レポートログファイルについては、「19.1.3 ログファイル」を参照してください。
21	EXS_CODETYPE	入力データのエンコードを指定します。テキストファイル、または CSV ファイルを指定した場合だけ有効です。
22	EXS_PACK_SIGNTYPE	内部 10 進数の符号タイプを指定します。
23	EXS_UNPACK_DATATYPE	外部 10 進数のデータ形式を指定します。
24	EXS_OUTFILENUMUNLIMIT	1 コマンドで指定できる出力ファイル (-outfile) パラメタの個数制限をなしにするオプションです。
25	EXS_PRMSIZEUNLIMIT	exsort パラメタファイルのサイズを制限なしにするオプションです。
26	EXS_DUMPLOG_FILESIZE	exsort コマンドのダンプログファイル「exsortdmp」のファイルサイズを MB 単位で指定します。ダンプログファイルについては、「19.1.2 デバッグ情報ファイル」を参照してください。

また、環境ファイルを作成するディレクトリを次の表に示します。

表 15-2 Exsortconf.ini ファイル作成ディレクトリ

OS	Exsortconf.ini ファイル作成ディレクトリ
Windows	SORT EE インストール先ディレクトリ下の Conf ディレクトリ 【例】 C:\Program Files\HITACHI\ExSORT\Conf (32 ビット版) C:\Program Files\HITACHI\ExSORT64\Conf (64 ビット版)
UNIX	/opt/HIEXSORT/conf (32 ビット版) /opt/HIEXSORT64/conf (64 ビット版)

## 15. 環境ファイルの設定

SORT EE インストール時の環境ファイルの初期値は次のとおりです。上書きインストールの場合は、インストール前の内容が残ります。また、アンインストールでは環境ファイルは削除されません。

```
EXS_LOGPATH=  
EXS_ALTSEQPATH=  
EXS_COMPLETE_MSGOPT=  
EXS_RECSEPA=  
EXS_CELLSEPA=  
EXS_QUOTTYPE=  
EXS_CURRENCY=  
EXS_ROUND=  
EXS_NULLSEQ=  
EXS_OUTRECSEPA=  
EXS_PADDING=  
EXS_OUTCELLSEPA=  
EXS_EXSORTDMP_OPT=  
EXS_KEYNUMMAX=  
EXS_SUMNUMMAX=  
EXS_SELNUMMAX=  
EXS_CSVKEYNUMMAX=  
EXS_CSVSUMNUMMAX=  
EXS_CSVSELNUMMAX=  
EXS_REPOLOG_FILESIZE=  
EXS_CODETYPE=  
EXS_PACK_SIGNTYPE=  
EXS_UNPACK_DATATYPE=  
EXS_OUTFILENUMUNLIMIT=  
EXS_PRMSIZEUNLIMIT=  
EXS_DUMPLOG_FILESIZE=
```

項目名の指定に誤りがある場合でも、エラーになりません。無視されます。項目を指定するときの注意事項を次に示します。

- 項目は必ず行の先頭から記述してください。また、項目の前に1個以上の空白またはタブを指定しないでください。
- 行が複数にわたる記述はできません。
- 項目とイコール (=) の間に1個以上の空白またはタブを指定しないでください。
- イコールの次から改行文字の前までを、指定した値として扱われます。改行文字の前に空白またはタブを入れると、空白またはタブまで指定値に含まれます。例えば、改行文字の前にタブを入れると、指定誤りと解釈され、標準の値が仮定されます。また、パス名の後ろから改行文字の間に空白を入れると、空白までパス名と認識されます。改行文字の前に空白またはタブを指定した場合の指定例を示します。

```
EXS_RECSEPA=LF→↓……………指定誤りと解釈され、標準の値が仮定される。  
EXS_LOGPATH=d:¥sortlog△↓…ログパス名は「d:¥sortlog△」となる。
```

(凡例)

→: タブ    △: 空白    ↓: 改行

- 1行に記述できる項目は一つだけです。
- 誤った指定をした場合、標準の値が仮定されます。この場合、エラーメッセージは出力されません。
- 同じ項目を複数指定した場合、最後に指定した正しい項目が有効となります。同じ項目を複数指定した場合の指定例を示します。

```
EXS_RECSEPA=LF…………… (1) ○ (無効)
EXS_RECSEPA=CRLF…………… (2) ○ (有効)
EXS_RECSEPA=CR…………… (3) ×
```

(解説)

後方 (3) の指定が誤っている場合、前方の指定が有効となります。この例では、前方に (1), (2) が指定されており、どちらも正しい指定です。この場合、最後に指定された正しい指定である (2) が有効となります。

### (1) EXS\_LOGPATH

exsort コマンドのログファイル「ExSORT.log」の出力先ディレクトリ名を指定します。

指定方法については、「19.1.3 ログファイル」を参照してください。

### (2) EXS\_ALTSEQPATH

altseq 変換ファイル「exsort\_altseq.txt ファイル (64 ビット版は「exsort64\_altseq.txt ファイル)」) が格納されているディレクトリ名を指定します。

指定方法については「5.15 -altseq パラメタ」、または「付録 G altseq 変換ファイル (SORT EE の機能)」を参照してください。

### (3) EXS\_COMPLETE\_MSGOPT

完了メッセージオプションを指定します。

指定方法については、「19.1.4 完了メッセージオプション」を参照してください。

### (4) EXS\_RECSEPA

入力レコードの改行文字を指定します。テキストファイルまたは CSV ファイルを指定した場合だけ有効です。指定方法を次に示します。

```
EXS_RECSEPA=オプションコード
```

指定できるオプションコードを次に示します。

オプションコード	意味
CRLF	CRLF を改行とします。
LF	LF を改行とします。
LF CRLF	LF と CRLF の両方を改行として扱います。

EXS\_RECSEPA を省略した場合の初期値は、LF です。

注意事項については、「5.8.7 recsepa オペランド」を参照してください。

## 15. 環境ファイルの設定

### (5) EXS\_CELLSEPA

入力レコードのセルのデリミタ（区切り文字）を指定します。CSV ファイルを指定した場合だけ有効です。指定方法を次に示します。

```
EXS_CELLSEPA=cell-separator [ $\Delta$ cell-separator ... ]
```

*cell-separator* に指定する値を次に示します。

指定値	意味
" <i>cell-separator</i> "	セルのデリミタとして、タブを除いた任意の 1 文字（半角英数字）を、ダブルクォーテーションで囲んで指定します。 【指定例】 空白を指定： EXS_CELLSEPA=" $\Delta$ " タブ、セミコロン、コンマ、空白、コロンの 5 個のデリミタを指定： EXS_CELLSEPA=TAB "; " ", " " $\Delta$ " ":"
TAB	セルのデリミタとして、タブを指定します。 【指定例】 EXS_CELLSEPA=TAB

(凡例)

$\Delta$  : スペース（空白）1 文字を示します。

最大 5 個のデリミタを指定できます。

引用符（シングルクォーテーション、ダブルクォーテーション）、改行文字、通貨記号、およびマルチバイト文字（日本語）はデリミタとして指定できません。

EXS\_CELLSEPA を省略した場合の初期値は、「EXS\_CELLSEPA=";"」です。

指定例については、「5.8.8 cellsepa オペランド」を参照してください

#### 注意事項

- EXS\_CELLSEPA の指定を誤ると、初期値のコンマが仮定されます。指定を誤っても構文エラーとなりません。このため、意図しない実行結果となるおそれがあるので注意してください。
- cell-separator は、必ずダブルクォーテーション (") で囲んで指定してください。
- exsort コマンド (-infile, -catinp, および -splitinp パラメタの cellsepa オペランド) に指定する場合は、cell-separator をシングルクォーテーション (') で囲みます。EXS\_CELLSEPA と指定方法が異なるので注意してください。

### (6) EXS\_QUOTTYPE

セルの引用符文字を指定します。CSV ファイルを指定した場合だけ有効です。指定方法を次に示します。



```
EXS_QUOTTYPE=オプションコード
```

指定できるオプションコードを次に示します。

オプションコード	意味
SQUOTE	シングルクォーテーション (') を引用符とします。
DQUOTE	ダブルクォーテーション (") を引用符とします。

EXS\_QUOTTYPE を省略した場合の初期値は、「DQUOTE」です。

## (7) EXS\_CURRENCY

固定小数点の数値データに付ける通貨記号として、任意の 1 文字を指定します。CSV ファイルを指定した場合だけ有効です。指定方法を次に示します。

```
EXS_CURRENCY=通貨記号
```

- 通貨記号として、任意の 1 文字（半角文字）を指定します。
- デリミタ、引用符として指定した文字、0～9 の数字、およびマルチバイト文字（日本語）は通貨記号として指定できません。
- EXS\_CURRENCY を省略した場合の初期値は、円記号「¥」です。

### 注意事項

EXS\_CURRENCY の指定を誤ると、初期値の円記号が仮定されます。指定を誤っても構文エラーとなりません。このため、意図しない実行結果となるおそれがあるので注意してください。

## (8) EXS\_ROUND

数値データの小数部の丸め方を指定します。CSV ファイルを指定した場合だけ有効です。この項目を省略した場合は、オプションコード「0」の切り捨てが仮定されます。指定方法を次に示します。

```
EXS_ROUND=オプションコード
```

指定できるオプションコードを次に示します。

オプションコード	意味
0	小数部の右側のあふれ桁を切り捨てます。
1	小数部の右側のあふれ桁を切り上げます。
2	小数部の右側のあふれ桁を四捨五入します。

## 15. 環境ファイルの設定

注意事項については、「5.8.11 round オペランド」を参照してください。

### (9) EXS\_NULLSEQ

CSV ファイルのナル文字列が含まれる入力レコードを、先頭または末尾に配置するオプションを指定します。

CSV ファイルのキー「-csvkey」を指定した場合だけ有効です。

指定方法を次に示します。

EXS\_NULLSEQ=オプションコード

指定できるオプションコードを次に示します。

オプションコード	意味
0	ナル文字列 (null キー) が含まれるレコードを、末尾に配置します。
1	ナル文字列 (null キー) が含まれるレコードを、先頭に配置します。

EXS\_NULLSEQ を省略した場合の初期値は、「0」です。

指定例については、「5.8.12 nullseq オペランド」を参照してください。

### (10) EXS\_OUTRECSEPA

出力レコードの改行文字を指定します。

テキストファイルまたは CSV ファイルを指定した場合だけ有効です。

指定方法を次に示します。

EXS\_OUTRECSEPA=オプションコード

指定できるオプションコードを次に示します。

オプションコード	意味
CRLF	CRLF を改行とします。
LF	LF を改行とします。

EXS\_OUTRECSEPA を省略した場合、入力ファイルの改行文字を適用します。

注意事項については、「5.10.5 recsepa オペランド」を参照してください。

### (11) EXS\_PADDING

出力レコード長をそろえるためのパディング文字を指定します。

-outfile, -catout, および -splitout パラメタの reclen または minlen オペランドを指定した場合だけ有効です。

指定方法を次に示します。

```
EXS_PADDING=パディング文字
```

指定できるパディング文字を次に示します。

パディング文字	意味
X	パディング文字として、空白を指定します。
Z	パディング文字として、ゼロを指定します。codetype で指定したエンコードに対応した値を次に示します。 <ul style="list-style-type: none"> <li>• ASCII, SJIS, EUC-JP, EUC-HJ, または UTF-8 : 0x00 (1 バイト)</li> <li>• UCS-2LE, UCS-2BE, UTF-16LE, または UTF-16BE : 0x0000 (2 バイト)</li> <li>• UCS-4LE, UCS-4BE, UTF-32LE, または UTF-32BE : 0x00000000 (4 バイト)</li> </ul>
C' <i>padding-character</i> '	パディング文字として、任意の 1 文字を指定します。 【指定例】 EXS_PADDING=C'*'
X' <i>16 進数</i> '	パディング文字として、任意の 1 バイトのデータを指定します。 【指定例】 EXS_PADDING=X'ff'

#### 注意事項

- 16 進数として指定できるのは「0～9」および「A～F」だけです。
- その他の注意事項については、「5.10.9 padding オペランド」を参照してください。

## (12) EXS\_OUTCELLSEPA

出力レコードのセルのデリミタ（区切り文字）を指定します。

入力レコードのデリミタを、EXS\_OUTCELLSEPA に指定したデリミタに変更して出力します。この項目の指定がない場合は、デリミタの変更は行いません。この項目は、CSV ファイルを指定した場合だけ有効です。

指定方法を次に示します。

```
EXS_OUTCELLSEPA= {cell-separator |
                  cell-separator-in_1△cell-separator-out_1
                  [△cell-separator-in_2△cell-separator-out_2 ... ]}
```

デリミタを 1 個だけ、または 2 個 1 対で 5 対までのどちらかの形式で指定します。

*cell-separator* に指定する値を次に示します。

## 15. 環境ファイルの設定

指定値	意味
"cell-separator"	セルのデリミタとして、タブを除いた任意の 1 文字（半角英数字）を、ダブルクォーテーションで囲んで指定します。 【指定例】 空白を指定：EXS_OUTCELLSEPA="△"（△は空白 1 文字を表す） コンマとセミコロンの 2 個のデリミタを、それぞれ空白とコロンに変更する指定： EXS_OUTCELLSEPA="," "△" ";" ":"
TAB	セルのデリミタとして、タブを指定します。 【指定例】 EXS_OUTCELLSEPA=TAB

(凡例) △：スペース（空白）1 文字を示します。

### 注意事項

- 引用符（シングルクォーテーション、ダブルクォーテーション）、改行文字、通貨記号、およびマルチバイト文字（日本語）をデリミタとして指定できません。
- cell-separator は、必ずダブルクォーテーション（"）で囲んで指定します。
- exsort コマンド（-outfile、-catout、および-splitout パラメタの cellsepa オペランド）に指定する場合は、cell-separator をシングルクォーテーション（'）で囲みます。EXS\_OUTCELLSEPA と指定方法が異なるので注意してください。
- 入力のデリミタに指定されていない文字を指定した場合、その指定は対で無効です。
- 3 個以上の奇数個を指定した場合、および 11 個以上指定した場合、右端の対になっていない指定は無効となります。例えば、「EXS\_OUTCELLSEPA=' TAB '△'」と 3 個指定した場合、右端の「'△'」は対で指定されていないので無効となります。有効な指定は「' TAB」だけです。
- そのほかの注意事項については、「5.10.6 cellsepa オペランド」を参照してください。

### (13) EXS\_EXSORTDMP\_OPT

exsortrepo.log ファイルに出力するレポートログに関するオプションです。

指定方法については、「19.1.3 ログファイル」を参照してください。

### (14) EXS\_KEYNUMMAX

-key パラメタに指定できるキーの最大個数の倍数と、キーの合計長（バイト）の上限の倍数を指定します。

「最大個数の倍数」の 128 倍が最大個数、「合計長の上限の倍数」の 4,096 ~ 16,384 倍※が合計長の上限となります。

指定方法を次に示します。

```
EXS_KEYNUMMAX=最大個数の倍数[,合計長の上限の倍数]
```

- 最大個数の倍数を、1 から 10 までの 10 進数で指定します。最大 1,280 個 (128 × 10) まで拡張できます。
- 合計長の上限の倍数を、1 から 16 までの 10 進数で指定します。
- この項目を指定しなかった場合、最大個数の倍数と合計長の上限の倍数に、1 を設定します。また、最大個数の倍数を指定し、合計長の上限の倍数を指定しなかった場合、合計長の上限の倍数に、1 を設定します。

注※

codetype オペランドで指定したエンコードによってキーの最大合計長が異なります。

エンコード種別	キーの合計長 (バイト)		
	指定範囲	最小値	最大値
ASCII SJIS EUC-JP EUC-HJ UTF-8	「合計長の上限の倍数」 × 4,096	4,096 (1 × 4,096)	65,536 (16 × 4,096)
UCS-2LE UCS-2BE UTF-16LE UTF-16BE	「合計長の上限の倍数」 × 8,192	8,192 (1 × 8,192)	131,072 (16 × 8,192)
UCS-4LE UCS-4BE UTF-32LE UTF-32BE	「合計長の上限の倍数」 × 16,384	16,384 (1 × 16,384)	262,144 (16 × 16,384)

### (15) EXS\_SUMNUMMAX

-sum パラメタに指定できる集約項目の最大個数の倍数と、集約項目の合計長 (バイト) の上限の倍数を指定します。

「最大個数の倍数」の 128 倍が最大個数、「合計長の上限の倍数」の 4,096 倍が合計長の上限となります。

指定方法を次に示します。

```
EXS_SUMNUMMAX=最大個数の倍数[,合計長の上限の倍数]
```

- 最大個数の倍数を、1 から 10 までの 10 進数で指定します。
- 合計長の上限の倍数を、1 から 16 までの 10 進数で指定します。
- この項目を指定しなかった場合、最大個数の倍数と合計長の上限の倍数に、1 を設定します。また、最大個数の倍数を指定し合計長の上限の倍数を指定しなかった場合、合計長の上限の倍数に、1 を設定します。

## (16) EXS\_SELNUMMAX

include, および omit オペランドに指定できる選択条件式の最大個数の倍数と、比較項目の合計長 (バイト) の上限の倍数を指定します。

「最大個数の倍数」の 128 倍が最大個数, 「合計長の上限の倍数」の 4,096 ~ 16,384 倍<sup>※</sup>が合計長の上限となります。

指定方法を次に示します。

```
EXS_SELNUMMAX=最大個数の倍数[,合計長の上限の倍数]
```

- 最大個数の倍数を, 1 から 10 までの 10 進数で指定します。最大 1,280 個 (128 × 10) まで拡張できます。
- 合計長の上限の倍数を, 1 から 16 までの 10 進数で指定します。
- この項目を指定しなかった場合, 最大個数の倍数と合計長の上限の倍数に, 1 を設定します。また, 最大個数の倍数を指定し合計長の上限の倍数を指定しなかった場合, 合計長の上限の倍数に, 1 を設定します。

注※

codetype オペランドで指定したエンコードによって比較項目の最大合計長が異なります。

エンコード種別	比較項目の合計長 (バイト)		
	指定範囲	最小値	最大値
ASCII SJIS EUC-JP EUC-HJ UTF-8	「合計長の上限の倍数」 × 4,096	4,096 (1 × 4,096)	65,536 (16 × 4,096)
UCS-2LE UCS-2BE UTF-16LE UTF-16BE	「合計長の上限の倍数」 × 8,192	8,192 (1 × 8,192)	131,072 (16 × 8,192)
UCS-4LE UCS-4BE UTF-32LE UTF-32BE	「合計長の上限の倍数」 × 16,384	16,384 (1 × 16,384)	262,144 (16 × 16,384)

## (17) EXS\_CSVKEYNUMMAX

-csvkey パラメタに指定できるキーの最大個数の倍数と, キーの合計長 (バイト) の上限の倍数を指定します。

「最大個数の倍数」の 128 倍が最大個数, 「合計長の上限の倍数」の 4,096 ~ 16,384 倍<sup>※</sup>が合計長の上限となります。

指定方法を次に示します。

**EXS\_CSVKEYNUMMAX=最大個数の倍数 [, 合計長の上限の倍数]**

- 最大個数の倍数を、1 から 10 までの 10 進数で指定します。最大 1,280 個 (128 × 10) まで拡張できます。
- 合計長の上限の倍数を、1 から 16 までの 10 進数で指定します。
- この項目を指定しなかった場合、最大個数の倍数と合計長の上限の倍数に、1 を設定します。また、最大個数の倍数を指定し合計長の上限の倍数を指定しなかった場合、合計長の上限の倍数に、1 を設定します。

注※

codetype オペランドで指定したエンコードによって、CSV キーの最大合計長が異なります。

エンコード種別	CSV キーの合計長 (バイト)		
	指定範囲	最小値	最大値
ASCII SJIS EUC-JP EUC-HJ UTF-8	「合計長の上限の倍数」 × 4,096	4,096 (1 × 4,096)	65,536 (16 × 4,096)
UCS-2LE UCS-2BE UTF-16LE UTF-16BE	「合計長の上限の倍数」 × 8,192	8,192 (1 × 8,192)	131,072 (16 × 8,192)
UCS-4LE UCS-4BE UTF-32LE UTF-32BE	「合計長の上限の倍数」 × 16,384	16,384 (1 × 16,384)	262,144 (16 × 16,384)

## (18) EXS\_CSVSUMNUMMAX

-csvsum パラメタに指定できる集約項目の最大個数の倍数と、集約項目の合計長 (バイト) の上限の倍数を指定します。

「最大個数の倍数」の 128 倍が最大個数、「合計長の上限の倍数」の 4,096 倍が合計長の上限となります。

指定方法を次に示します。

**EXS\_CSVSUMNUMMAX=最大個数の倍数 [, 合計長の上限の倍数]**

- 最大個数の倍数を、1 から 10 までの 10 進数で指定します。
- 合計長の上限の倍数を、1 から 16 までの 10 進数で指定します。
- この項目を指定しなかった場合、最大個数の倍数と合計長の上限の倍数に、1 を設定

## 15. 環境ファイルの設定

します。また、最大個数の倍数を指定し合計長の上限の倍数を指定しなかった場合、合計長の上限の倍数に、1を設定します。

### (19) EXS\_CSVSELNUMMAX

csvinclude, および csvomit オペランドに指定できる選択条件式の最大個数の倍数と、比較項目の合計長 (バイト) の上限の倍数を指定します。

「最大個数の倍数」の128倍が最大個数、「合計長の上限の倍数」の4,096～16,384倍※が合計長の上限となります。

指定方法を次に示します。

**EXS\_CSVSELNUMMAX=最大個数の倍数[,合計長の上限の倍数]**

- 最大個数の倍数を、1から10までの10進数で指定します。最大1,280個(128×10)まで拡張できます。
- 合計長の上限の倍数を、1から16までの10進数で指定します。
- この項目を指定しなかった場合、最大個数の倍数と合計長の上限の倍数に、1を設定します。また、最大個数の倍数を指定し合計長の上限の倍数を指定しなかった場合、合計長の上限の倍数に、1を設定します。

注※

codetype オペランドで指定したエンコードによって CSV 比較項目の最大合計長が異なります。

エンコード種別	CSV 比較項目の合計長 (バイト)		
	指定範囲	最小値	最大値
ASCII SJIS EUC-JP EUC-HJ UTF-8	「合計長の上限の倍数」×4,096	4,096 (1×4,096)	65,536 (16×4,096)
UCS-2LE UCS-2BE UTF-16LE UTF-16BE	「合計長の上限の倍数」×8,192	8,192 (1×8,192)	131,072 (16×8,192)
UCS-4LE UCS-4BE UTF-32LE UTF-32BE	「合計長の上限の倍数」×16,384	16,384 (1×16,384)	262,144 (16×16,384)

### (20) EXS\_REPOLOG\_FILESIZE

exsort コマンドのレポートログファイル「exsortrepo.log」のファイルサイズを指定します。



指定方法を次に示します。

**EXS\_REPOLOG\_FILESIZE=ファイルサイズ**

- ファイルサイズは、MB 単位の 1 から 3 桁の 10 進数で指定します。
- 指定できる範囲は、5 から 100 までです。
- 指定値を誤った場合、レポートログファイルのサイズは、初期値の 50MB が適用されます。

## (21) EXS\_CODETYPE

入力データのエンコード（文字コード）の種類を指定します。

この項目に指定したエンコード（文字コード）で、CSV ファイルの区切り文字（デリミタ）、引用符、文字属性コード「w」、「u」および「v」を解析します。

この項目は、テキストファイル、または CSV ファイルを指定した場合だけ有効です。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は無視されます。

指定方法を次に示します。

**EXS\_CODETYPE=エンコード**

指定できるエンコードを次に示します。

エンコード	意味
ASCII	入力データの文字コードが ASCII であることを示します。
SJIS	入力データの文字コードが Shift_JIS であることを示します。
EUC-JP	入力データの文字コードが日本語 EUC であることを示します。
EUC-HJ	入力データの文字コードが EUC-HJ であることを示します。
UTF-8	入力データのエンコードが UTF-8 であることを示します。
UCS-2LE	入力データの文字コードが UCS-2 little endian であることを示します。
UCS-2BE	入力データの文字コードが UCS-2 big endian であることを示します。
UCS-4LE	入力データの文字コードが UCS-4 little endian であることを示します。
UCS-4BE	入力データの文字コードが UCS-4 big endian であることを示します。
UTF-16LE	入力データのエンコードが UTF-16 little endian であることを示します。
UTF-16BE	入力データのエンコードが UTF-16 big endian であることを示します。
UTF-32LE	入力データのエンコードが UTF-32 little endian であることを示します。
UTF-32BE	入力データのエンコードが UTF-32 big endian であることを示します。

## 15. 環境ファイルの設定

エンコード	意味
UNICODE	入力ファイルの Unicode シグニチャ（以下、BOM）形式を適用します。 SORT EE で使用できる BOM の種類は次のとおりです。 <ul style="list-style-type: none"> <li>• UTF-8…0xEFBBBF（3 バイト）</li> <li>• UTF-16LE…0xFFFE（2 バイト）</li> <li>• UTF-16BE…0xFEFF（2 バイト）</li> <li>• UTF-32LE…0xFFFE0000（4 バイト）</li> <li>• UTF-32BE…0x0000FEFF（4 バイト）</li> </ul>

注意事項については、「5.8.13 codetype オペランド」を参照してください。

### (22) EXS\_PACK\_SIGNTYPE

内部 10 進数の符号の範囲を変更するオプションです。指定方法を次に示します。

```
EXS_PACK_SIGNTYPE=オプションコード
```

指定できるオプションコードを次に示します。

オプションコード	意味
SE	内部 10 進数の符号は C, D, F の 3 種類です。
MF	内部 10 進数の符号の範囲を 0 ~ F（集約項目は A ~ F）に拡張します。

- EXS\_PACK\_SIGNTYPE を省略した場合の初期値は「SE」です。
- 環境変数「EXSORTPACKSIGNTYPE（64 ビット版は「EXSORT64PACKSIGNTYPE）」の指定を優先します。
- 符号の範囲の詳細については、環境変数「14.1.1(30) EXSORTPACKSIGNTYPE」または「14.1.2(30) EXSORT64PACKSIGNTYPE」を参照してください。

### (23) EXS\_UNPACK\_DATATYPE

外部 10 進数のデータ形式を変更するオプションです。指定方法を次に示します。

```
EXS_UNPACK_DATATYPE=オプションコード
```

指定できるオプションコードを次に示します。

オプションコード	意味
ASCII-1	外部 10 進数のデータ形式は次のとおりです。 ゾーン部：3 数値部：0 ~ 9 符号部：(正) 3, (負) 7

オプション コード	意味
ASCII-2	外部 10 進数のデータ形式は次のとおりです。 数字：0x30 ～ 0x39 正值：0x7B, 0x41 ～ 0x49 負値：0x7D, 0x4A ～ 0x52
EBCDIK	外部 10 進数のデータ形式は次のとおりです。 ゾーン部：F 数値部：0 ～ 9 符号部：(正) C, (負) D

- EXS\_UNPACK\_DATATYPE を省略した場合の初期値は「ASCII-1」です。
- 環境変数「EXSORTUNPACKDATATYPE (64 ビット版は「EXSORT64UNPACKDATATYPE)」」の指定を優先します。
- データ形式の詳細については、環境変数「14.1.1(31) EXSORTUNPACKDATATYPE」または「14.1.2(31) EXSORT64UNPACKDATATYPE」を参照してください。

#### (24) EXS\_OUTFILENUMUNLIMIT 【08-50 以降】

1 コマンドに指定できる出力ファイル (-outfile) パラメタの個数制限をなしにするオプションです。指定方法を次に示します。

```
EXS_OUTFILENUMUNLIMIT=オプションコード
```

指定できるオプションコードを次に示します。

オプション コード	意味
0	指定できる -outfile パラメタの最大個数は 64 個です。
1	指定できる -outfile パラメタの個数に制限はありません。ただし、1 つの -outfile パラメタに指定できる出力ファイルの最大数は 4 個です。

EXS\_OUTFILENUMUNLIMIT を省略した場合の初期値は、「0」です。

パラメタの最小値および最大値については、「3.3.1 パラメタ」の表 3-5 を参照してください。

#### (25) EXS\_PRMSIZEUNLIMIT 【08-50 以降】

exsort パラメタファイルのサイズを制限なしにするオプションです。

指定方法を次に示します。

## 15. 環境ファイルの設定

**EXS\_PRMSIZEUNLIMIT=オプションコード**

指定できるオプションコードを次に示します。

オプションコード	意味
0	exsort パラメタファイルサイズは 256KB 以内です。
1	exsort パラメタファイルサイズに制限はありません。

EXS\_PRMSIZEUNLIMIT を省略した場合の初期値は、「0」です。

パラメタファイル形式については、「4.4.1 ファイル形式」を参照してください。

### (26) EXS\_DUMPLOG\_FILESIZE 【08-50 以降】

exsort コマンドのダンプログファイル「exsortdmp」のファイルサイズを指定します。

指定方法を次に示します。

**EXS\_DUMPLOG\_FILESIZE=ファイルサイズ**

- ファイルサイズは、MB 単位の 1 から 3 桁の 10 進数で指定します。
- 指定できる範囲は、5 から 100 までです。
- 指定値を誤った場合、ダンプログファイルのサイズは、初期値の 50MB が適用されます。

## 15.2 SORT の環境ファイルの設定（Windows の場合）

環境ファイルは、SORT の各コマンドの実行時に省略したオペランドの初期値を設定するファイルです。環境ファイルのファイル名および格納ディレクトリを表 15-3 に示します。また、環境ファイルのひな形のファイル名および格納ディレクトリを表 15-4 に示します。

表 15-3 環境ファイルのファイル名および格納ディレクトリ

ファイル名	格納ディレクトリ
wnsort.wse	SORT インストールディレクトリ

表 15-4 環境ファイルのひな形のファイル名および格納ディレクトリ

SORT の種別	ファイル名	ディレクトリ名
32 ビット版	wnsort_sample.wse	SORT インストールディレクトリ
64 ビット版	wnsort_sample.wse	SORT インストールディレクトリ ¥samples

環境ファイルは、改行文字を区切り記号として、パラメタとオペランドを生成します。環境ファイルの形式を次の図に示します。

図 15-1 環境ファイルの形式

```
[WNSORTEV]※
date_opt=日付項目の基準年度
msg_opt=メッセージ出力オプション
msg_file=メッセージの出力先ファイル名
temp_dir1=ソート用一時ファイルディレクトリ
temp_dir2=中間ファイルディレクトリ
temp_dir3=マージ用一時ファイルディレクトリ
work_size=ワークバッファサイズ
string_dir=ストリング情報格納一時ファイルディレクトリ名
bom_IngIn=Unicodeシグニチャの読み飛ばし機能オプション
bom_Ingout=Unicodeシグニチャの付加機能オプション
dmp_dir=デバッグ情報ファイルの出力先ディレクトリ名
SORT_EE
```

注※ [WNSORTEV]は必ず先頭に指定する。

### 15.2.1 SORT の環境ファイルのパラメタの意味

パラメタには、必ず指定するパラメタと任意に指定できるパラメタの 2 種類があります。

#### (1) [WNSORTEV]

wnsortev コマンドのパラメタファイルの先頭を意味します。この指定は省略できません。

## 15. 環境ファイルの設定

ん。

### (2) date\_opt= 日付項目の基準年度

日付（英数字）項目の基準年度を指定します。2桁以下の数値で指定します。

基準年度はキーの比較時に、指定した値を年度の先頭に位置づけ、それより小さい値には、100を加えた値を設定します。

この指定は、省略できます。ただし、SORTをインストールしたときに、初期値として「70」が設定されています。

指定を省略した場合、または「0」を指定した場合は、文字項目で指定した結果と同じです。

### (3) msg\_opt= メッセージ出力オプション

コマンド実行時のメッセージ出力オプションを指定します。

- 0: すべてのメッセージを表示します。
- 1: エラーメッセージだけを表示します。
- 2: すべてのメッセージを表示しません。
- 3: メッセージを画面に表示しないで、「msg\_file」パラメタで指定したファイルに、すべてのメッセージを出力します。
- 4: メッセージを画面に表示しないで、「msg\_file」パラメタで指定したファイルに、エラーメッセージだけを出力します。
- 5: メッセージを画面に表示しないで、イベントログファイルにすべてのメッセージを出力します。
- 6: メッセージを画面に表示しないで、イベントログファイルにエラーメッセージだけを出力します。
- 7: メッセージを画面に表示しないで、標準エラーファイルにすべてのメッセージを出力します。
- 8: メッセージを画面に表示しないで、標準エラーファイルにエラーメッセージだけを出力します。

この指定は、省略できます。省略した場合は「0」が設定されます。

### (4) msg\_file= メッセージの出力先ファイル名

コマンド実行時のメッセージの出力先ファイル名を指定します。

この指定は、「msg\_opt」パラメタで「3」または「4」を指定した場合にだけ有効となります。

### (5) temp\_dir1= ソート用一時ファイルディレクトリ

ソート用一時ファイルを割り当てるディレクトリを指定します。この指定は、ソートの

実行時だけ有効となります。

#### (6) temp\_dir2= 中間ファイルディレクトリ

中間ファイルを割り当てるディレクトリを指定します。この指定は、拡張 SORT コマンド (WNSORTEX または WNSORTEX64) 実行オプションで 2 個以上の指定をした場合だけ有効となります。

#### (7) temp\_dir3= マージ用一時ファイルディレクトリ

マージ用一時ファイルを割り当てるディレクトリを指定できます。この指定は、ソートの実行時だけ有効となります。

#### (8) work\_size= ワークバッファサイズ

ソート機能でワークバッファとして使用するメモリの容量を KB または MB 単位で指定します。ワークバッファサイズは、メモリ量と単位で指定します。

##### (a) メモリ量

メモリ量の指定範囲を次の表に示します。

表 15-5 メモリ量の指定範囲 (環境ファイル)

SORT の種別	レコード長	最小値	最大値
32 ビット版	32,767 以下	128KB	2,000MB
	32,768 以上	256KB	2,000MB
64 ビット版	32,767 以下	256KB	2,000MB
	32,768 以上	384KB	2,000MB

##### (b) 単位

メモリ量の単位を、KB にするか MB にするか指定します。

- k : 単位を KB にします。
- m : 単位を MB にします。

また、単位を省略した場合は KB (k) が設定されます。

指定例を次に示します。

(例 1) 1,024KB → 「work\_size=1024k」または「work\_size=1024」

(例 2) 2MB → 「work\_size=2m」

#### (9) string\_dir= スtring情報格納一時ファイルディレクトリ名

「KBLS290-E メモリが不足しました。」が発生しないように、ワークバッファ内の

## 15. 環境ファイルの設定

SORT 情報を一時ファイルに格納するための一時ファイルディレクトリを指定します。この指定は、ソートの実行時だけ有効となります。メッセージが表示されたときの原因と対処方法の詳細は、「付録 K メッセージ」を参照してください。

### (10) bom\_Ingin=Unicode シグニチャの読み飛ばし機能オプション

UTF-8 のテキスト編成ファイルの場合、Unicode シグニチャとして (X'EFBBBF') を入力ファイルから読み飛ばさせるオプションに「unicode」を指定します。

### (11) bom\_Ingout=Unicode シグニチャの付加機能オプション

UTF-8 のテキスト編成ファイルの場合、Unicode シグニチャとして (X'EFBBBF') を出力ファイルに付加するオプションに「utf8」を指定します。

### (12) dmp\_dir= デバッグ情報ファイルの出力先ディレクトリ

SORT のコマンド処理が異常終了した場合に作成されるデバッグ情報ファイルの出力先ディレクトリを指定します。デバッグ情報ファイルの詳細は、「19.2.2 デバッグ情報ファイル」を参照してください。

### (13) SORT\_EE

SORT の機能拡張オプションを使用する場合、SORT\_EE オプションを指定します。

SORT の機能拡張オプションについては、「2.21 SORT の機能拡張オプション (SORT の機能)」を参照してください。

## 15.2.2 SORT の環境ファイルの有効範囲

環境ファイルで設定したパラメタの値が有効となる範囲を、機能別に次の表に示します。

表 15-6 環境ファイルの有効範囲

機能		date _opt	m sg _o pt	m sg _fi le	te mp _di r1	te mp _di r2	te mp _di r3	wo rk_ siz e	stri ng_ di r	bo m_ Ingi n	bo m_ Ing out	dm p_ dir	SO RT _E E
コマンド	ソート	○	○	○	○	×	○	○	○	○	○	○	○
	マージ	○	○	○	×	×	×	×	×	×	○	○	○
	選択	○	○	○	×	×	×	×	×	×	○	○	○
	集約	○	○	○	×	×	×	×	×	×	○	○	○
	拡張 SORT	○	○	○	○	○	○	○	○	○	○	○	○



機能		date _opt	m sg _o pt	m sg _fi le	te mp _di r1	te mp _di r2	te mp _di r3	wo rk_ siz e	stri ng _ di r	bo m_ l ngi n	bo m_ l ng out	dm p_ dir	SO RT _ E E
C 言語	ソート	×	×	×	×	×	×	×	×	×	×	○	×
	選択	×	×	×	×	×	×	×	×	×	×	○	×
	集約	×	×	×	×	×	×	×	×	×	×	○	×
	拡張 SORT	×	×	×	×	×	×	×	×	×	×	○	×

(凡例)

○：実行時に有効

×

### 15.2.3 SORT の環境ファイル指定時の注意事項

環境ファイルを作成するときの注意事項を次に示します。

#### (1) パラメタの指定順序

各パラメタの指定順序は任意です。ただし、「[WNSORTEV]」は、必ず先頭に指定してください。

#### (2) 同一パラメタの複数指定

同じパラメタを複数指定した場合は、最初に指定したオペランドが有効になります。

#### (3) スペースの扱い

各パラメタとオペランドの間に、スペースは入力しないでください。

#### (4) 英文字の扱い

各パラメタを大文字で入力するか、小文字で入力するかは任意です。また、大文字と小文字が混在してもかまいません。

#### (5) 改行文字

パラメタファイルを作成する場合、各パラメタの最後に改行文字を付けてください。ただし、最後のパラメタには、改行文字の代わりに EOF (**End Of File**) コードを付けることもできます。

#### (6) エラーメッセージ

SORT の実行時、環境ファイルの指定方法が誤っている場合、エラーメッセージは表示されません。



# 16 一時ファイルの作成条件と ファイル容量の算出方法

この章では、ソート機能の一時ファイルの作成条件および一時ファイルのファイル容量の算出方法を SORT EE と SORT に分けて説明します。

---

16.1 一時ファイルの種類

---

16.2 SORT EE の一時ファイルの作成条件

---

16.3 SORT EE の一時ファイル容量の算出方法

---

16.4 SORT EE のファイル分割用一時ファイルの作成条件

---

16.5 SORT EE のファイル分割用一時ファイル容量の算出方法

---

16.6 SORT の一時ファイルの作成条件

---

16.7 SORT の一時ファイル容量の算出方法

---

## 16.1 一時ファイルの種類

SORT EE および SORT が作成する一時ファイルの種類と目的を次に示します。

表 16-1 一時ファイルの種類

一時ファイルの種類	目的	一時ファイルを使用する機能およびコマンド		
		SORT EE	SORT	
			UNIX	Windows
ソート用一時ファイル	ソートフェーズの結果を格納する。 <sup>※1</sup>	ソート機能 (-function sort)	<ul style="list-style-type: none"> <li>• rsort コマンド</li> <li>• rsortsum シェルスクリプト</li> </ul>	<ul style="list-style-type: none"> <li>• wnsort コマンド</li> <li>• 拡張 SORT コマンド<sup>※2</sup></li> <li>• C 言語ライブラリ関数<sup>※3</sup></li> </ul>
マージ用一時ファイル	中間マージフェーズの結果を格納する。 <sup>※1</sup>	ソート機能 (-function sort)	<ul style="list-style-type: none"> <li>• rsort コマンド</li> <li>• rsortsum シェルスクリプト</li> </ul>	<ul style="list-style-type: none"> <li>• wnsort コマンド</li> <li>• 拡張 SORT コマンド<sup>※2</sup></li> <li>• C 言語ライブラリ関数<sup>※3</sup></li> </ul>
中間ファイル <sup>※4</sup>	<ul style="list-style-type: none"> <li>• UNIX ソートの実行結果を格納する。</li> <li>• Windows 選択またはソートの実行結果を格納する。</li> </ul>	—	<ul style="list-style-type: none"> <li>• rsortsum シェルスクリプト</li> </ul>	<ul style="list-style-type: none"> <li>• 拡張 SORT コマンド<sup>※5</sup></li> </ul>
分割用一時ファイル	集約、選択、またはレコード編集の結果を格納する。	ファイル分割機能 (-splitfile)	—	—

(凡例)

— : 作成しない

注※1

ソートフェーズおよび中間マージフェーズについては、「1.11.1 ソート機能の処理の流れ」を参照してください。

注※2

拡張 SORT コマンドの実行オプションにソート機能 (sort) を指定した場合が該当します。拡張 SORT コマンドの詳細については、「11. 拡張 SORT コマンドの使い方 (Windows の場

合)」を参照してください。

注※ 3

C 言語ライブラリ関数でソート機能を実行する場合は該当します。C 言語ライブラリ関数の詳細については、「13. C 言語プログラムでの使い方 (Windows の場合)」を参照してください。

注※ 4

中間ファイルの容量については、「12.3.3 中間ファイルの容量」を参照してください。

注※ 5

拡張 SORT コマンドの実行オプションを 2 個以上指定した場合は該当します。

## 16.2 SORT EE の一時ファイルの作成条件

ワークバッファサイズ、入力ファイルのレコード件数、レコード長およびソートキーのキー長と、一時ファイルの作成条件の関係を次に示します。

### 16.2.1 ソート用一時ファイルおよびマージ用一時ファイルを作成する場合のワークバッファサイズ

ワークバッファサイズが「中間マージフェーズが動作する場合のワークバッファサイズ見積もり式」を満たす場合に、ソート用一時ファイルおよびマージ用一時ファイルを作成します。

「中間マージフェーズが動作する場合のワークバッファサイズ見積もり式」については、「17.1.1(2) アウトコアソート」の「図 17-4 中間マージフェーズが動作する場合のワークバッファサイズ見積もり式」を参照してください。

### 16.2.2 ソート用一時ファイルだけを作成する場合のワークバッファサイズ

ワークバッファサイズが次に示す条件を満たす場合は、ソート用一時ファイルだけを作成します。

「一時ファイルを作成する（アウトコアソート）場合のワークバッファサイズ見積もり式」  
 $\leq$ ワークバッファサイズ  
「一時ファイルを作成しない（インコアソート）場合のワークバッファサイズ見積もり式」

「一時ファイルを作成する（アウトコアソート）場合のワークバッファサイズ見積もり式」については、「17.1.1(2) アウトコアソート」の「図 17-3 一時ファイルを作成する（アウトコアソート）場合のワークバッファサイズ見積もり式」を参照してください。

また、「一時ファイルを作成しない（インコアソート）場合のワークバッファサイズ見積もり式」については、「17.1.1(1) インコアソート」の「図 17-2 一時ファイルを作成しない（インコアソート）場合のワークバッファサイズ見積もり式」を参照してください。

### 16.2.3 一時ファイルを作成しない場合のワークバッファサイズ

ワークバッファサイズが「一時ファイルを作成しない（インコアソート）場合のワークバッファサイズ見積もり式」を満たす場合は、一時ファイルを作成しません。

「一時ファイルを作成しない（インコアソート）場合のワークバッファサイズ見積もり式」については、「17.1.1(1) インコアソート」の「図 17-2 一時ファイルを作成しない

## 16. 一時ファイルの作成条件とファイル容量の算出方法

(インコアソート) 場合のワークバッファサイズ見積もり式」を参照してください。

## 16.3 SORT EE の一時ファイル容量の算出方法

---

一時ファイル作成ディレクトリには、計算式で算出した一時ファイル容量以上の空き容量が必要です。ソート用一時ファイル作成ディレクトリとマージ用一時ファイル作成ディレクトリに同じディレクトリを使用する場合は、ソート用一時ファイルのファイル容量とマージ用一時ファイルのファイル容量を合計した容量以上の空き容量が必要です。

なお、マージ用一時ファイルのファイル容量は、ソート用一時ファイルのファイル容量と同じです。

「-cmdopt DUP」（重複レコードの入力順序保存機能）指定がない場合

$$\text{一時ファイル容量} = \text{入力ファイルサイズ}^{\ast 1}$$

「-cmdopt DUP」（重複レコードの入力順序保存機能）指定がある場合

- 32 ビット版の場合

$$\text{一時ファイル容量} = \text{入力ファイルサイズ}^{\ast 1} + 4 \times \text{入力レコード件数}^{\ast 2}$$

- 64 ビット版の場合

$$\text{一時ファイル容量} = \text{入力ファイルサイズ}^{\ast 1} + 8 \times \text{入力レコード件数}^{\ast 2}$$

注※ 1

入力ファイルを複数指定した場合は、すべてのファイルのサイズを合計した値が入力ファイルサイズとなります。

注※ 2

入力ファイルを複数指定した場合は、すべてのファイルのレコード件数を合計した値が入力ファイルのレコード件数となります。



## 16.4 SORT EE のファイル分割用一時ファイルの作成条件

均等分割情報 (-splitfile パラメタ指定) によるファイル分割機能と、次の表に示す機能を組み合わせて使用した場合に、ファイル分割用一時ファイルを作成します。

表 16-2 ファイル分割用一時ファイルを作成する機能の組み合わせ

指定値	基本機能	レコード処理機能
method=recent	ソート	集約または出力レコード選択
	コピーまたはマージ	入力レコード選択, 集約, または出力レコード選択
method=filesize	ソート	集約, 中間レコード編集, 出力レコード選択, 出力レコード編集, CSV レコード変更 <sup>※1</sup> , またはファイル編成変更機能 <sup>※2</sup>
	コピーまたはマージ	入力レコード選択, 入力レコード編集, 集約, 中間レコード編集, 出力レコード選択, 出力レコード編集, CSV レコード変更 <sup>※1</sup> , またはファイル編成変更機能 <sup>※2</sup>

### 注※1

出力ファイル別に出力 CSV レコード形式を変更する機能で、次の機能を使用した場合に分割用一時ファイルが作成されます。

- ・連続区切り文字オプション (OUTCONSEPA オプション)
- ・出力ファイルの引用符変更 (quottype オペランド)
- ・出力ファイルのセル数調整 (cellnum オペランド)

### 注※2

出力ファイルへ出力する際に、出力ファイル別にファイル編成を変更する機能で、次の機能を使用した場合に分割用一時ファイルが作成されます。

- ・出力レコード長変更 (reclen, minlen オペランド)

## 16.5 SORT EE のファイル分割用一時ファイル容量の算出方法

---

ファイル分割用一時ファイルには、選択および集約の結果を出力します。そのため、ファイル容量は不定となります。なお、ファイル容量の最大値は、次の計算式で求めた値となります。

<ソート機能使用時>

1. 「-cmdopt DUP」（重複レコードの入力順序保存機能）指定がない場合

一時ファイル容量 = 入力ファイルサイズ※<sup>1</sup>

2. 「-cmdopt DUP」（重複レコードの入力順序保存機能）指定がある場合

・32ビット版の場合

一時ファイル容量 = 入力ファイルサイズ※<sup>1</sup> + 4 × 入力ファイルのレコード件数  
※<sup>2</sup>

・64ビット版の場合

一時ファイル容量 = 入力ファイルサイズ※<sup>1</sup> + 8 × 入力ファイルのレコード件数  
※<sup>2</sup>

<コピーまたはマージ機能使用時>

一時ファイル容量 = 入力ファイルサイズ※<sup>1</sup>

注※<sup>1</sup>

入力ファイルを複数指定した場合は、すべてのファイルのサイズを合計した値が入力ファイルサイズとなります。

注※<sup>2</sup>

入力ファイルを複数指定した場合は、すべてのファイルのレコード件数を合計した値が入力ファイルのレコード件数となります。

## 16.6 SORT の一時ファイルの作成条件

ワークバッファサイズと入力ファイルのレコード件数、レコード長およびソートキーのキー長による、一時ファイルの作成条件の関係を次に示します。

### 16.6.1 ソート用一時ファイルおよびマージ用一時ファイルを作成する場合のワークバッファサイズ

ワークバッファサイズが次に示す条件を満たす場合は、ソート用一時ファイルとマージ用一時ファイルを作成します。

$$\text{ワークバッファサイズ} < \sqrt{\text{レコード件数} \times (\text{レコード長} * + \text{キー長} + 40)} \text{ バイト}$$

注※

「/d」（重複レコードの入力順序保存機能）を指定した場合は、次のように対応すること。

32ビット版SORTの場合：レコード長に4バイトを追加する。

64ビット版SORTの場合：レコード長に8バイトを追加する。

可変長ファイル、またはテキストファイルの場合：最大レコード長を使用する。

### 16.6.2 ソート用一時ファイルだけを作成する場合のワークバッファサイズ

ワークバッファサイズが次に示す条件を満たす場合は、ソート用一時ファイルだけを作成します。

$$\sqrt{\text{レコード件数} \times (\text{レコード長} * + \text{キー長} + 40)} \text{ バイト} \leq \text{ワークバッファサイズ} < \text{レコード件数} \times (\text{レコード長} * + \text{キー長} + 16) \text{ バイト}$$

注※

「/d」（重複レコードの入力順序保存機能）を指定した場合は、次のように対応すること。

32ビット版SORTの場合：レコード長に4バイトを追加する。

64ビット版SORTの場合：レコード長に8バイトを追加する。

可変長ファイル、またはテキストファイルの場合：最大レコード長を使用する。

### 16.6.3 一時ファイルを作成しない場合のワークバッファサイズ

ワークバッファサイズが次に示す条件を満たす場合は、一時ファイルを作成しません。

16. 一時ファイルの作成条件とファイル容量の算出方法

ワークバッファサイズ $\geq$ レコード件数 $\times$  (レコード長<sup>※</sup>+キー長+16) バイト

注※

「/d」(重複レコードの入力順序保存機能)を指定した場合は、次のように対応すること。

32ビット版SORTの場合：レコード長に4バイトを追加する。

64ビット版SORTの場合：レコード長に8バイトを追加する。

可変長ファイル、またはテキストファイルの場合：最大レコード長を使用する。

## 16.7 SORT の一時ファイル容量の算出方法

一時ファイル作成ディレクトリには、計算式で算出した一時ファイル容量以上の空き容量が必要です。また、ソート用一時ファイル作成ディレクトリとマージ用一時ファイル作成ディレクトリに同じディレクトリを使用する場合は、ソート用一時ファイルのファイル容量とマージ用一時ファイルのファイル容量を合計した容量以上の空き容量が必要です。

ソート用一時ファイルのファイル容量を求める計算式を、次に示します。なお、マージ用一時ファイルのファイル容量は、ソート用一時ファイルのファイル容量と同じです。

「/d」（重複レコードの入力順序保存機能）指定がない場合

$$\text{一時ファイル容量} = \text{入力ファイルサイズ}^{\ast 1}$$

「/d」（重複レコードの入力順序保存機能）指定がある場合

- 32 ビット版の場合

$$\text{一時ファイル容量} = \text{入力ファイルサイズ}^{\ast 1} + 4 \times \text{入力ファイルのレコード件数}^{\ast 2}$$

- 64 ビット版の場合

$$\text{一時ファイル容量} = \text{入力ファイルサイズ}^{\ast 1} + 8 \times \text{入力ファイルのレコード件数}^{\ast 2}$$

注※1

入力ファイルを複数指定した場合は、すべてのファイルのサイズを合計した値が入力ファイルサイズとなります。

注※2

入力ファイルを複数指定した場合は、すべてのファイルのレコード件数を合計した値が入力ファイルのレコード件数となります。



# 17 性能チューニング方法

この章では、性能チューニング方法について説明します。

---

17.1 ソート機能の処理時間を短くするためには

---

17.2 ソート機能以外の機能の処理時間を短くするためには

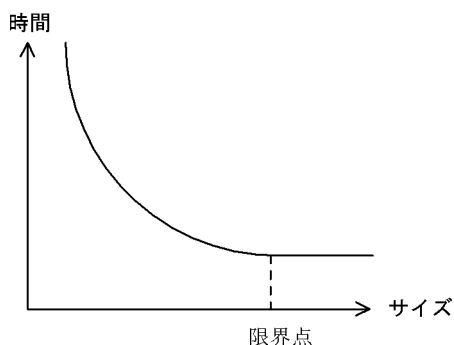
---

## 17.1 ソート機能の処理時間を短くするためには

### 17.1.1 ワークバッファサイズと処理時間

一般的に、ワークバッファサイズが大きいほど処理時間は短くなります。しかし、ワークバッファサイズの増加量に対する処理時間の減少量は一定ではなく、ワークバッファサイズが増えるに従って処理時間の減少量は小さくなっていきます。そして、ワークバッファがあるサイズに達すると、それ以上処理時間は減少しなくなります。ワークバッファサイズに対する処理時間の減少について、一般的な傾向を次の図に示します。

図 17-1 ワークバッファサイズと処理時間の関係



上記の図の限界点は、入力データ量が多いほど右側へ移動します。大量のデータをソートする場合は、ワークバッファのサイズを大きく指定すると効果があります。

ただし、極端に大きなワークバッファを指定すると、ソート処理だけでなく、システム全体のスループットが低下する可能性があります。これは、実メモリ量に対する仮想メモリの比率が大きいほど、ページングやスワッピングが多く発生してしまうためです。同時に実行するソート処理の多重度を上げた場合も同様です。

システム全体の運用を考慮した上で、ワークバッファサイズを指定してください。

`exsort` コマンドには、ワークバッファサイズを自動で見積もる機能を用意しています。ただし、この機能はワークバッファサイズの指定がない場合に有効となります。この機能を使用すれば、少量のメモリ所要量で、比較的速いソートが期待できます。この機能で見積もるワークバッファサイズについては、「17.1.1(2) アウトコアソート」の「一時ファイルを作成する(アウトコアソート)場合のワークバッファサイズ見積もり式」を参照してください。

#### (1) インコアソート

一時ファイルを使用しないでソートすることをインコアソートといいます。入力データ



量に対し十分なワークバッファサイズを指定した場合は、インコアソートとなります。

一般的に、インコアソートは一時ファイルに対する入出力が不要となるため、処理時間が短くなります。ただし、大きなワークバッファサイズを指定すると、ページングやスワッピングが多く発生し、処理時間が増加する可能性があります。また、ワークバッファが大きくなった分、データのソートが多く実行されることになり、CPU 時間が増加する可能性もあります。

一時ファイルを作成しない（インコアソート）場合のワークバッファサイズ見積もり式を次に示します。

図 17-2 一時ファイルを作成しない（インコアソート）場合のワークバッファサイズ見積もり式

$$\text{ワークバッファサイズ} \geq \text{入力データサイズ} + \text{レコード件数} \times (n1 + n2 + n3 + n4) \text{ バイト}$$

SORT EE の計算式の変数値を次の表 17-1 に示します。

また、SORT の計算式の変数の値を次の表 17-2 に示します。

表 17-1 ワークバッファサイズ見積もり式の変数値（SORT EE の場合）

変数	SORT EE の条件	変数の値	
		32 ビット版	64 ビット版
n1	重複レコードの入力順序保存機能（-cmdopt DUP オプション）指定なし	0	0
	重複レコードの入力順序保存機能（-cmdopt DUP オプション）指定あり	4	8
n2	入力ファイルは、固定長順編成ファイル (fileorg=S) または可変長順編成ファイル (fileorg=V)	0	0
	入力ファイルは、テキストファイル (fileorg=A) または CSV ファイル (fileorg=C)	4 - 改行文字の長さ※1	4 - 改行文字の長さ※1
n3	(可変サイズ)	ソートキーの合計長※2	ソートキーの合計長※2
n4	(固定サイズ)	12	24
n5	(固定サイズ)	20	32
n6	(固定サイズ)	40	64

注※1

codetype オペランドに指定した入力データのエンコードの種別によって、改行文字の長さは異なります。例えば、エンコードが「UCS-4LE」で、改行文字が「CRLF」の場合は、改行文字の長さは 8 バイトなので、n3 の値は「4・8=4」バイトとなります。

## 17. 性能チューニング方法

エンコードの種類	改行文字の長さ	
	LF	CRLF
ASCII, SJIS, EUC-JP, EUC-HJ, または UTF-8	1	2
UCS-2LE, UCS-2BE, UTF-16LE, または UTF-16BE	2	4
UCS-4LE, UCS-4BE, UTF-32LE, または UTF-32BE	4	8

### 注※2

マルチバイト文字の属性コード「u」または「v」で、キー長に「文字数」を指定している場合、1文字の長さ（実際の入力データの文字のバイト長ではなく、正規化したデータ長）は次のとおりです。

エンコードの種類	1文字の長さ（正規化長）	
	属性コード「u」	属性コード「v」
ASCII	1	1
SJIS EUC-HJ UCS-2LE UCS-2BE	2	2
EUC-JP	3	3
UTF-8 UTF-16LE UTF-16BE UCS-4LE UCS-4BE UTF-32LE UTF-32BE	4	8

例えば、エンコード「UTF-8」、属性コード「v」、キー長「6文字 (v+0\*6)」と指定した場合、1文字の長さは8バイトのため、キーの長さは「6文字×8バイト＝48」バイトとなります。

表 17-2 ワークバッファサイズ見積もり式の変数値（SORT の場合）

変数	SORT の条件	変数の値	
		32ビット版	64ビット版
n1	重複レコードの入力順序保存機能の指定なし	0	0
	重複レコードの入力順序保存機能の指定あり	4	8
n2	入力ファイルは、固定長順編成ファイル、または可変長順編成ファイル	0	0
	入力ファイルは、テキストファイル、または CSV ファイル	3	3
n3	(可変サイズ)	ソートキーの合計長	ソートキーの合計長

変数	SORT の条件	変数の値	
		32 ビット版	64 ビット版
n4	(固定サイズ)	12	24
n5	(固定サイズ)	20	32
n6	(固定サイズ)	40	64

なお、この概算式は、`exsort` コマンドの `-worksize` パラメタに「`incore`」を指定した場合の自動見積もりで使用されます。

## (2) アウトコアソート

一時ファイルを使用してソートすることをアウトコアソートといいます。使用メモリ量を小さくして、CPU 時間を短くできます。ただし、インコアソートと比べて入出力が増えます。

一時ファイルを作成する（アウトコアソート）場合のワークバッファサイズ見積もり式を次に示します。

図 17-3 一時ファイルを作成する（アウトコアソート）場合のワークバッファサイズ見積もり式

$$\text{ワークバッファサイズ} = \sqrt{\text{レコード件数} \times 66000 \times (\text{レコード長} * n1 + n2 + n3 + n5)} \text{ バイト}$$

注※

可変長ファイル、テキストファイル、または CSV ファイルの場合は、最大レコード長を使用する。

SORT EE の計算式の変数値については、「17.1.1(1) インコアソート」の「表 17-1 ワークバッファサイズ見積もり式の変数値 (SORT EE の場合)」を参照してください。

また、SORT の計算式の変数の値については、「表 17-2 ワークバッファサイズ見積もり式の変数値 (SORT の場合)」を参照してください。

この概算式は、`exsort` コマンドの `-worksize` パラメタに「`auto`」を指定した場合の自動見積もりで使用されます。

この概算式の値は目安であり、必ずしも十分な性能が得られるとは限りません。メモリの使用状況、ソート処理の多重度、ファイルの配置ボリュームなどにより、概算式よりも大きなサイズを指定した方が処理時間が短くなる可能性もあります。このため、チューニングの実施をお勧めします。

ワークバッファが小さすぎると、一時ファイルを 2 個作成する「中間マージフェーズ」が動作し、処理時間が増加することになるので注意してください。

中間マージフェーズが動作する場合のワークバッファサイズ見積もり式を次に示します。

図 17-4 中間マージフェーズが動作する場合のワークバッファサイズ見積もり式

$$\text{ワークバッファサイズ} < \sqrt{\text{レコード件数}} \times (\text{レコード長} * n1 + n2 + n3 + n6) \text{ バイト}$$

注※

可変長ファイル、テキストファイル、またはCSVファイルの場合は、最大レコード長を使用する。

SORT EE の計算式の変数値については、「17.1.1(1) インコアソート」の「表 17-1 ワークバッファサイズ見積もり式の変数値 (SORT EE の場合)」を参照してください。

また、SORT の計算式の変数の値については、「表 17-2 ワークバッファサイズ見積もり式の変数値 (SORT の場合)」を参照してください。

なお、`exsort` コマンドでは、中間マージフェーズが動作した場合に `KBLS1007-I` メッセージに中間マージフェーズが動作したことを示す「`WORKSIZE=outcore(PREM)`」が追加出力されます。この際は、ワークバッファサイズに指定した値を見直してください。`KBLS1007-I` メッセージの詳細については、「付録 K.2 SORT EE のメッセージ一覧」を参照してください。

## 17.1.2 SORT EE のワークバッファサイズ自動見積もり機能

`exsort` コマンドには、ワークバッファサイズを自動で見積もる機能があります。この機能を使用すると、少量のメモリで比較的速いソートを提供するアウトコアソート、または高速なインコアソートのどちらかのワークバッファサイズを見積もる指定ができます。ワークバッファサイズの自動見積もり機能については、「5.13 `-worksize` パラメタ」を参照してください。

**!** 注意事項

- システム環境によっては、自動見積もり機能で見積もったサイズを割り当てることできない場合があります。この場合は、見積もりサイズよりも少ないサイズでソート処理を実行します。このときのソート完了を示す **KBLS1007-I** メッセージには、見積もりサイズおよび実際に割り当てることのできたサイズが出力されます。
- テキストファイルおよび CSV ファイルの場合は、入力データの一部をサンプリングした仮定のレコード長を使ってサイズを見積もります。このため、正確なサイズを見積もることができない場合があります。
- レコード長が一定していない入力データの場合、仮定されるレコード長によって、見積もりサイズと実際のサイズの差が大きくなる可能性があります。例えば、インコアソートを指定しても、必要なサイズよりも小さく見積もられ、結果的にアウトコアソートで実行されることがあります。必要なサイズを見積もることができなかつたとき、または割り当てられなかつたときは、ソートの実行結果から必要なワークバッファサイズを再計算した見積もりサイズ（必要なワークバッファサイズ）が **KBLS1007-I** メッセージに出力されます。**KBLS1007-I** メッセージに「yy..y（必要なワークバッファサイズ）」が出力されたときは、性能チューニングの効果がでない可能性があります。

**KBLS1007-I** メッセージの詳細については、「付録 K.2 SORT EE のメッセージ一覧」を参照してください。

- インコアソートの見積もりサイズが上限値を超えたとき、アウトコアソート（**auto** 指定を仮定）で見積もり直します。**KBLS1007-I** メッセージの「ww..w（自動見積もりしたワークバッファサイズ）」には、アウトコアソートで見積もり直したサイズが出力されます。また、「yy..y（必要なワークバッファサイズ）」に、ソートの実行結果から必要なワークバッファサイズを再計算したインコアソートの見積もりサイズを出力します。**KBLS1007-I** メッセージの詳細については、「付録 K.2 SORT EE のメッセージ一覧」を参照してください。

### 17.1.3 ファイルの配置

ソート機能では、入力ファイル、出力ファイル、および一時ファイルを次の表に示す組み合わせで使用します。

表 17-3 使用ファイルの組み合わせ

ソート方式	フェーズ	使用ファイル			
		入力ファイル	ソート用一時ファイル	マージ用一時ファイル	出力ファイル
インコア	ソート	○ (IN)	—	—	—
	最終マージ	—	—	—	○ (OUT)
アウトコア (中間マージ無し)	ソート	○ (IN)	○ (OUT)	—	—
	最終マージ	—	○ (IN)	—	○ (OUT)

## 17. 性能チューニング方法

ソート方式	フェーズ	使用ファイル			
		入力ファイル	ソート用一時ファイル	マージ用一時ファイル	出力ファイル
アウトコア (中間マージあり)	ソート	○ (IN)	○ (OUT)	—	—
	中間マージ	—	○ (IN)	○ (OUT)	—
	最終マージ	—	—	○ (IN)	○ (OUT)

(凡例)

○ : 組み合わせ対象    — : 該当しない

同時に使用するファイルを、それぞれ別々のボリュームに配置すると、入出力効率が向上します。

また、他プロセスと競合の少ないボリュームを選択してください。

なお、入力ファイルと出力ファイルを同時に使用することはないため、同一ボリューム上に配置しても問題ありません。また、指定したワークバッファサイズが小さいと、中間マージが動作することがあります。この場合は、ソート用一時ファイルとマージ用一時ファイルを同時に使用するため、ソート用一時ファイルと別のボリュームにマージ用一時ファイルを配置してください。

各ファイルは、最低でも 3 ボリュームに分けて配置することを推奨します。

なお、入力ファイルとは別のボリュームに出力ファイルを配置する場合は、ソート用／マージ用一時ファイルの配置ボリュームと分けて配置することを推奨します。

**SORT EE** では、出力ファイルを複数個指定できます。このため、できるだけ別々のボリュームに分けて配置してください。

### 17.1.4 I/O サイズ

ディスク装置の I/O ブロック長の整数倍となるように I/O サイズを設定してください。これは、ソート機能の読み込み／書き出しサイズが、ディスク装置の I/O ブロック長よりも小さい場合に、その差分だけデータの読み込み／書き出しが余計に動作することを防ぐためです。

**SORT EE** では、ユーザが読み込み／書き出しサイズを任意のサイズに指定できます。また、I/O サイズの指定がない場合、デフォルト値の「1,024KB」を仮定します。

### 17.1.5 レコード長の指定

テキストファイル、および CSV ファイルをソートする場合、レコード長の指定を省略できます。しかし、レコード長を指定するとソート時間を短縮できます。レコード長を指定するときは、入力ファイル内の最大レコード長より大きな値を指定してください。

### 17.1.6 キー属性の指定

ソート機能では、キー属性が文字の場合には、数字に比べて、処理効率が向上します。したがって、ファイル設計時にはキーとなる部分の属性を文字にしておくことでソート処理の時間を短くできます。

### 17.1.7 コマンドオプションの指定

「重複レコードの入力順序保存」機能を使用した場合に比べ、使用しない方が処理時間が短く、一時ファイルの容量も小さくなります。

また、UNIX 版 SORT で入力ファイルか索引ファイルの場合は、「ファイル排他機能」を使用した方が処理時間は短くなります。

### 17.1.8 マルチバイト文字データ

テキストファイルおよび CSV ファイルでは、マルチバイト文字に対応した文字属性コード「w」、「u」および「v」を指定できます。しかし、文字属性コード「w」、「u」および「v」では、エンコードによる入力データの文字解析を実施するため、文字属性コード「c」よりも処理効率が低下します。

なお、キー・項目範囲の入力データがすべて ASCII コード（英数字）である場合は、文字属性コード「c」を指定すると処理効率が向上します。文字属性コード「c」は 1 文字 1 バイトで扱うため、エンコードに対応したバイト長を指定してください。

ASCII コードについては、「付録 H Unicode 操作文字コード一覧 (SORT EE)」を参照してください。また、1 文字の長さについては、「付録 I.2(1) 文字の長さ」を参照してください。

### 17.1.9 最適化したソートライブラリの使用【Linux(x64) SORT EE(64) 08-41 以降】

64 ビット版 Linux(x64) SORT EE(64) 08-41 以降、最適化したソートライブラリを使用できます。

最適化したソートライブラリを使用すれば、ソート機能の処理時間が短縮できる可能性があります。ただし、処理時間の短縮効果は、入力ファイル（レコード件数）や指定されたソートキーの値によって変動します。したがって、適用する際は、事前に処理時間を測定し問題ないかどうかを検証することを推奨します。

指定方法については、環境変数「14.1.2(37) EXSORT64SORTLIBOPTION【Linux(x64) SORT EE(64) 08-41 以降】」を参照してください。

## 17.2 ソート機能以外の機能の処理時間を短くするためには

---

### 17.2.1 ファイルの配置

ソート機能以外の機能では、入力ファイルと出力ファイルを同時に使用します。それぞれ別々のボリュームに配置すると、入出力効率が向上します。入力ファイルと出力ファイルは、他プロセスと競合の少ないボリュームを選択してください。

なお、マージ機能の場合は、すべての入力ファイルを同時に使用します。このため、個々の入力ファイルについても別々のボリュームに配置することを推奨します。

また、SORT EE では、出力ファイルを複数個指定できます。これについても、できるだけ別々のボリュームに分けて配置してください。

### 17.2.2 I/O サイズ

ディスク装置の I/O ブロック長の整数倍となるように I/O サイズを設定してください。これは、ソート機能の読み込み／書き出しサイズが、ディスク装置の I/O ブロック長よりも小さい場合に、その差分だけデータの読み込み／書き出しが余計に動作することを防ぐためです。

SORT EE では、読み込み／書き出しサイズを、任意のサイズに指定できます。また、I/O サイズの指定がない場合、デフォルト値の「1,024KB」を仮定します。

### 17.2.3 マルチバイト文字データ

テキストファイルおよび CSV ファイルでは、マルチバイト文字に対応した文字属性コード「w」、「u」および「v」を指定できます。しかし、文字属性コード「w」、「u」および「v」では、エンコードによる入力データの文字解析を実施するため、文字属性コード「c」よりも処理効率が低下します。

なお、キー・項目範囲の入力データがすべて ASCII コード（英数字）である場合は、文字属性コード「c」を指定すると処理効率が向上します。文字属性コード「c」は1文字1バイトで扱うため、エンコードに対応したバイト長を指定してください。

ASCII コードについては、「付録 H Unicode 操作文字コード一覧 (SORT EE)」を参照してください。また、1文字の長さについては、「付録 I.2(1) 文字の長さ」を参照してください。



## 17.2.4 最適化したソートライブラリの使用【Linux(x64) SORT EE(64) 08-41 以降】

64ビット版 Linux(x64) SORT EE(64) 08-41 以降、最適化したソートライブラリを使用できます。

最適化したソートライブラリを使用すれば、マージ機能、選択機能、集約機能、およびキーによる分割機能の処理時間が短縮できる可能性があります。

ただし、処理時間の短縮効果は、入力ファイル（レコード件数）、指定されたキー（マージキー、集約キー、分割キー、非分割データ）、または比較項目の値などによって変動します。したがって、適用する際は、事前に処理時間を測定し問題ないかどうかを検証することを推奨します。

指定方法については、環境変数「14.1.2(37) EXSORT64SORTLIBOPTION【Linux(x64) SORT EE(64) 08-41 以降】」を参照してください。



# 18 SORTの機能拡張オプションの使い方

この章では、SORT EE 固有の機能を SORT で使用できるようにするオプションについて説明します。

---

18.1 SORTの機能拡張オプションの環境設定

---

18.2 SORTの機能拡張オプションの注意事項と見直しポイント

---

## 18.1 SORT の機能拡張オプションの環境設定

---

機能拡張オプションを使用する場合は、前提 PP として SORT EE が必要となります。また、機能拡張オプションは、索引編成ファイル、相対編成ファイルには使用できません。

SORT EE の機能を使用するには、次の手順で環境設定を行います。

1. 機能拡張オプションを有効にする
2. 使用する SORT EE 機能を有効にする

### 18.1.1 機能拡張オプションを有効にする

#### ● Windows の場合

SORT の環境ファイル「wmsort.wse」に、「SORT\_EE」を指定します。環境ファイルの指定方法については、「15.2 SORT の環境ファイルの設定 (Windows の場合)」を参照してください。

#### ● UNIX の場合

SORT の環境変数 SORTOPT に、「-SORT\_EE」を指定します。環境変数の指定方法については、「14.2 SORT の環境変数の設定 (UNIX の場合)」を参照してください。

### 18.1.2 使用する SORT EE 機能を有効にする

使用する機能に対応した SORT EE の環境変数を指定します。環境変数の指定方法については、「14.1 SORT EE の環境変数の設定」を参照してください。

#### (1) ファイル排他機能の無効化 (UNIX の場合だけ)

SORT EE では、ファイル排他を無効にできます。

ファイル排他機能を無効にする場合は、SORT EE の環境変数「EXSORTMGTOPT<sup>※</sup>」に、「LOCK\_INVALID」を指定します。

注<sup>※</sup>

64 ビット版の環境変数名は「EXSORT64MGTOPT」です。

#### (2) チェック緩和機能

SORT EE では、キー、比較項目、または集約項目の入力データ形式が不正であってもエラーとしないで処理を続行する、チェック緩和オプションがあります。

#### ● 10 進キーのキーチェック緩和機能

SORT EE の環境変数「EXSORTMGTOPT<sup>※</sup>」に、「KEY\_INVALID\_NOCHK」を指定します。

- 10 進集約項目のチェック緩和機能

SORT EE の環境変数「EXSORTMGTOPT<sup>※</sup>」に、「SUM\_INVALID\_NOCHK」を指定します。

- キー位置のチェック緩和機能

SORT EE の環境変数「EXSORTMGTOPT<sup>※</sup>」に、「VKEY\_POS\_NOCHK」を指定します。

- 日付データのチェック緩和機能

SORT EE の環境変数「EXSORTMGTOPT<sup>※</sup>」に、「DATE\_INVALID\_NOCHK」を指定します。

注※

64 ビット版の環境変数名は「EXSORT64MGTOPT」です。

### (3) I/O サイズ変更機能

SORT EE では、入力ファイル、出力ファイル、ソート用一時ファイル、およびマージ用一時ファイルに対する 1 回のデータ読み込みサイズ、および書き出しサイズを変更できます。

適切な I/O サイズを指定することで、実行時の性能を向上できます。I/O サイズのチューニングについては、「17. 性能チューニング方法」を参照してください。

入力ファイル、出力ファイル、ソート用一時ファイル、およびマージ用一時ファイルに対して、すべて同じ I/O サイズを指定する場合は、SORT EE の環境変数「EXSORTIOSIZE<sup>※1</sup>」を指定します。

ファイルごとに個別に指定する場合は、それぞれに対応した環境変数を指定します。<sup>※2</sup> 環境変数を指定していない場合、1,024KB が設定されます。

入力ファイル：EXSORTINPIOSIZE

出力ファイル：EXSORTOUTIOSIZE

ソート用一時ファイル：EXSORTSORTTMPIOSIZE

マージ用一時ファイル：EXSORTMERGETMPIOSIZE

注※1

64 ビット版の環境変数名は「EXSORT64IOSIZE」です。

注※2

64 ビット版の環境変数名はつぎのとおりです。

入力ファイル：EXSORT64INPIOSIZE

出力ファイル：EXSORT64OUTIOSIZE

ソート用一時ファイル：EXSORT64SORTTMPIOSIZE

マージ用一時ファイル : EXSORT64MERGETMPIOSIZE

#### (4) ワークバッファサイズの自動見積もり機能

SORT EE では、ソート機能で使用するワークバッファのサイズを自動で見積もることができます。ワークバッファサイズを指定していない場合、あるいはインコアソートで実行時間を短縮させたい場合などに効果的な機能です。ワークバッファのチューニングについては、「17. 性能チューニング方法」を参照してください。

- ワークバッファサイズの指定がない場合  
exsort コマンドの `-worksize auto` オペランドを設定します (SORT EE の省略値で、自動見積もり機能が有効になります)。
- 自動見積もり機能を使用する場合  
SORT EE の環境変数「EXSORTWORKSIZE<sup>※1</sup>」に、「AUTO」を指定します。
- 自動見積もり機能で確保するワークバッファサイズの上限值を設定したい場合  
SORT EE の環境変数「EXSORTWORKSIZE<sup>※1</sup>」に、「AUTO, 最大ワークバッファサイズ」を指定します。
- インコアソートで実行したい場合  
SORT EE の環境変数「EXSORTWORKSIZE<sup>※1</sup>」に、「INCORE」を指定します。
- コマンド、および SORT の環境変数 (Windows は環境ファイル) に指定しているワークバッファサイズを無視し、強制的に自動見積もり機能を使用させる場合  
SORT EE の環境変数「EXSORT\_HI\_WORKSIZE<sup>※2</sup>」を指定します。指定値は「EXSORTWORKSIZE<sup>※1</sup>」と同じで、「AUTO」、「AUTO, 最大ワークバッファサイズ」、「INCORE」を指定できます。

ワークバッファサイズの指定値は、次の優先順位で決定します。

Windows の場合

1. SORT EE の環境変数「EXSORT\_HI\_WORKSIZE<sup>※2</sup>」
2. wnsort コマンドの `/y` パラメタまたは wnsortex コマンドの `work_size` パラメタ
3. 環境ファイル「wnsort.wse」の `work_size` オプション
4. SORT EE の環境変数「EXSORTWORKSIZE<sup>※1</sup>」
5. exsort コマンドの `-worksize auto` オペランドを設定 (SORT EE の省略値)

UNIX の場合

1. SORT EE の環境変数「EXSORT\_HI\_WORKSIZE<sup>※2</sup>」
2. rsort コマンドの `-y` パラメタ
3. 環境変数「SORTOPT」の `-y` オプション
4. SORT EE の環境変数「EXSORTWORKSIZE<sup>※1</sup>」
5. exsort コマンドの `-worksize auto` オペランドを設定 (SORT EE の省略値)

## 注※ 1

64 ビット版の環境変数名は「EXSORT64WORKSIZE」です。

## 注※ 2

64 ビット版の環境変数名は「EXSORT64\_HI\_WORKSIZE」です。

## (5) 2GB 超ワークバッファサイズの指定 (64 ビット版の場合)

64 ビット版の SORT EE では、ソート機能で使用するワークバッファのサイズに、2,000GB まで指定できます。インコアソートで、大容量ファイルのソート実行時間を短縮させたい場合に効果的な機能です。

## ● インコアソートで実行したい場合

SORT EE の環境変数「EXSORT64WORKSIZE」に、「INCORE」を指定します。

- コマンド、および SORT の環境変数 (Windows は環境ファイル) に指定しているワークバッファサイズを無視し、強制的にインコアソートで実行させる場合  
SORT EE の環境変数「EXSORT64\_HI\_WORKSIZE」に、「INCORE」を指定します。

## 18.2 SORT の機能拡張オプションの注意事項と見直しポイント

SORT EE の機能を使用する上で、次のコマンド仕様が変更となります。

- メモリ所要量
- メッセージ
- ディスク使用量
- 選択条件式の文字条件値の円記号 (¥) の扱い

上記の変更に伴い、メモリ所要量、およびディスク使用量の再見積もりが必要となります。

### 18.2.1 メモリ所要量

SORT EE のメモリ所要量については、SORT EE のリリースノートをご確認ください。

SORT と SORT EE では、ワークバッファサイズの省略値と、I/O バッファサイズが異なりますので、注意してください。

バッファ種別	バッファサイズの省略値	
	SORT	SORT EE
ワークバッファ	1,024KB	-worksize auto (ワークバッファサイズ自動見積もり機能が有効になります。入力データ量によってサイズは異なります。最小サイズは1,024KB です)
I/O バッファ	省略値なし	1,024KB (入力ファイル、出力ファイルおよび一時ファイルの I/O バッファサイズ)

SORT と SORT EE では、指定できるワークバッファサイズの単位が異なります。このため、SORT でワークバッファサイズ<sup>※</sup>を KB 単位で指定した場合、SORT EE では MB 単位に切り上げて確保されるので、注意してください。例えば、SORT のワークバッファサイズに 2,000KB と指定された場合、SORT EE では 2MB (2,048KB) に切り上げて確保されます。

注※

SORT で指定するワークバッファサイズパラメタおよびオプション

Windows の場合

- wnsort コマンドの /y パラメタ、または wnsortex コマンドの work\_size パラメタ



- 環境ファイル「wmsort.wse」の work\_size オプション

UNIX の場合

- rsort コマンドの -y パラメタ
- 環境変数「SORTOPT」の -y オプション

ワークバッファサイズに指定できる最小値および最大値については、「3.3.2 メモリ量」を参照してください。

## 18.2.2 メッセージ

SORT のメッセージは KBLSn<sub>nnn</sub>-x 形式 (nnn : 3 桁の数字)、SORT EE のメッセージは KBLSn<sub>nnnn</sub>-x 形式 (nnnn : 4 桁の数字) で出力されます。

メッセージの詳細は、「付録 K.1 SORT EE メッセージ」および「付録 K.2 SORT EE のメッセージ一覧」を参照してください。

## 18.2.3 ディスク使用量

SORT EE では、コマンドログファイル (ExSORT.log)、レポートログファイル (exsortrepo.log)、およびデバッグ情報ファイル (exsortdmp) を、追加書きのテキストデータ形式で出力します。コマンドログファイルのサイズは、バックアップを含め最大 40MB です。レポートログファイルのサイズは、バックアップを含め最大 10 ~ 200MB です (レポートログファイルは、ファイルの最大サイズを変更できます)。詳細は、「19.1.2 デバッグ情報ファイル」を参照してください。

## 18.2.4 選択条件式の文字条件値の円記号 (¥) の扱い

選択条件式の文字条件値に円記号 (¥) を一つ指定し、このあとに、次の三つの記号以外を指定した場合、SORT と SORT EE では円記号 (¥) の扱いが異なりますので、注意してください。

- シングルクォーテーション (')
- ダブルクォーテーション (")
- 円記号 (¥)

SORT での円記号 (¥) の扱い

文字条件値の文字として扱われなくて、エスケープ文字として扱われます。

SORT EE での円記号 (¥) の扱い

文字条件値の文字として扱われ、円記号 (¥) を含むレコードが選択されます。



# 19 障害時の対処方法

この章では、障害時の対処方法を SORT EE と SORT に分けて説明します。

---

19.1 SORT EE の障害情報

---

19.2 SORT の障害情報

---

## 19.1 SORT EE の障害情報

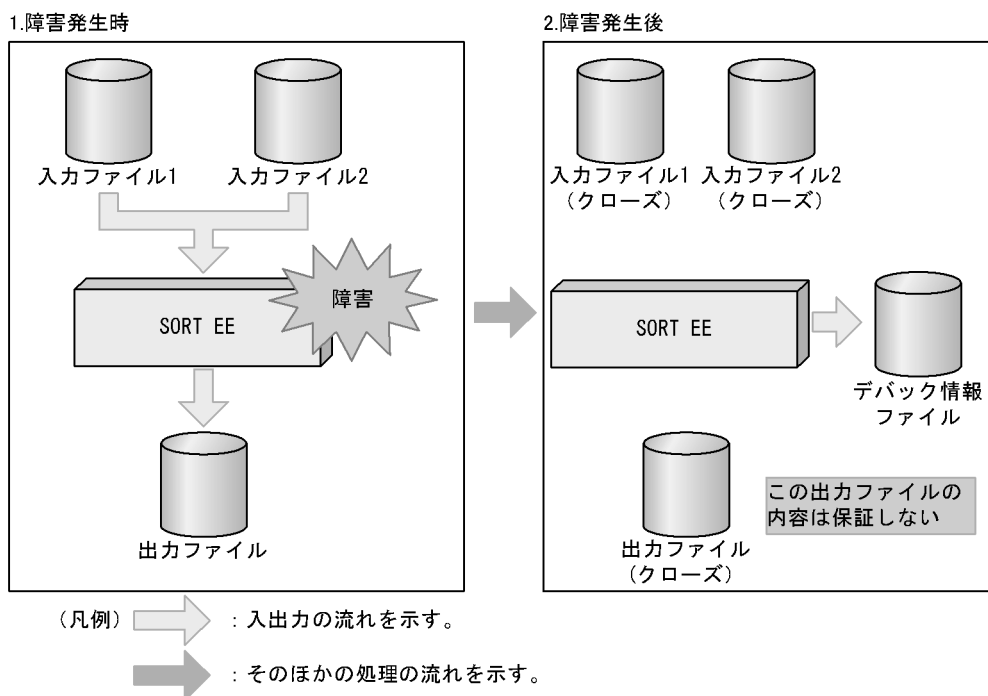
### 19.1.1 SORT EE の入出力障害時のファイルの扱い

コマンド処理中に入出力障害が発生した場合は、処理を中止し、すべてのファイルをクローズします。また、そのときの出力ファイルの内容は保証しません。

SORT EE のエラーメッセージから障害の原因を特定できない場合は、デバッグ情報ファイルを取得してください。

入出力障害時のファイルの扱いを、次の図に示します。

図 19-1 入出力障害時のファイルの扱い



### 19.1.2 デバッグ情報ファイル

SORT EE では、`exsort` コマンドが異常終了した場合のトラブルシューティング情報をデバッグ情報ファイルに出力します。

デバッグ情報ファイルの種類を次の表に示します。なお、括弧内のファイル名は、64ビット版のデバッグ情報ファイル名を示します。

表 19-1 デバッグ情報ファイルの種類

デバッグ情報ファイル名	ファイルの説明	出力方式
exsortdmp (exsortdmp64)	<p>(ダンプログファイル)</p> <p>exsort コマンドが異常終了した場合のトラブルシュート情報をテキスト形式で出力します。</p> <p>SORT EE 08-50 より前のファイルの最大サイズは 5MB です。</p> <p>SORT EE 08-50 以降のファイル最大サイズは、初期値で 50MB です。ファイル最大サイズは、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」の「EXS_DUMPLOG_FILESIZE」項目で、5 ~ 100 (単位 : MB) の範囲で変更できます。環境ファイルの「EXS_DUMPLOG_FILESIZE」項目については、「15.1.1(26) EXS_DUMPLOG_FILESIZE 【08-50 以降】」を参照してください。</p>	追加書き※ 2
exsortdmp1 (exsortdmp641)	<p>(exsortdmp のバックアップファイル)</p> <p>exsortdmp のファイルサイズが最大サイズに達すると、「exsortdmp1」のファイル名称でバックアップを作成します。すでにバックアップが存在する場合、既存のバックアップファイルを破棄し、新たにバックアップを作成します。</p>	上書き※ 3
exsortdmp.conf (exsortdmp64.conf)	<p>(exsortdmp のコントロールファイル)</p> <p>exsortdmp を出力するためのコントロールファイルです。</p>	—
exsortdmp_Pid_YYMMDDhhmmss (exsortdmp64_Pid_YYMMDDhhmmss)	<p>(ダンプファイル)</p> <p>追加書きのダンプログファイル</p> <p>「exsortdmp」の出力に失敗した場合、ダンプログファイルの代わりに出力されるデバッグ情報ファイルです。</p> <p>exsort コマンドの入力情報をテキスト形式で出力します。</p> <p>ダンプログファイルの出力に成功した場合は出力しません。</p> <p>ファイル名には、Pid (プロセス id) と、YYMMDDhhmmss (ダンプを出力した年月日時分秒) を付加します。</p>	上書き※ 3

## 19. 障害時の対処方法

デバッグ情報ファイル名	ファイルの説明	出力方式
EXSORTIODMP_Pid_YMMDdhmmss_x (EXSORTIODMP64_Pid_YMMDdhmmss_x)	(ダンプファイル) 追加書きのダンプログファイル 「exsortdmp」の出力に失敗した場合、 ダンプログファイルの代わりに出力される デバッグ情報ファイルです。 I/O エラーに関する情報をテキスト形式 で出力します。 ダンプログファイルの出力に成功した 場合は出力しません。 ファイル名には、Pid (プロセス id)、 YMMDdhmmss (ダンプを出力した 年月日時分秒)、および x (出力番号) を付加します。	上書き※ <sup>3</sup>
exsortrdmp_Pid_YMMDdhmmss (exsortrdmp64_Pid_YMMDdhmmss)	(不正レコード情報ファイル) 次のエラーメッセージで <b>exsort</b> コマンド が異常終了した場合に、不正なキー または項目を含んだ入力データを 1 レ コード分出力します。ファイルサイズ は、レコード長となります。 ファイル名には、Pid (プロセス id) と YMMDdhmmss (ダンプを出力した 年月日時分秒) を付加します。 <ul style="list-style-type: none"> <li>• <b>KBLS1211-E</b> キーにエラーデータを検出しました。 ※1</li> <li>• <b>KBLS1320-E</b> 比較項目にエラーデータを検出しま した。</li> <li>• <b>KBLS1411-E</b> 集約項目にエラーデータを検出しま した。</li> <li>• <b>KBLS2042-E</b> 分割キーにエラーデータを検出しま した。</li> <li>• <b>KBLS2060-E</b> 非分割データにエラーデータを検出 しました。</li> </ul>	上書き※ <sup>3</sup>

(凡例)

— : なし

注※1

-function copy 指定時は出力されません。

注※2

同名のファイルがすでに存在する場合は追加書き出力されます。同名のファイルが存在しない場合は新規に作成されます。

注※3

同名のファイルがすでに存在する場合は上書き出力されます。同名のファイルが存在しない場合は新規に作成されます。

**SORT EE** のデバッグ情報ファイルは、次の表に示すディレクトリ下に作成されます。

表 19-2 デバッグ情報ファイルを出力するディレクトリ

OS	作成ディレクトリ
Windows	<p>次の優先順位で使用するディレクトリが決まります。</p> <ol style="list-style-type: none"> <li>1. 環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル」)」の「EXS_LOGPATH」項目で指定した、ログ出力先ディレクトリ</li> <li>2. SORT EE インストール先ディレクトリ下の Log ディレクトリ</li> </ol> <p>【例】 C:\Program Files\HITACHI\ExSORT\Log (32 ビット版) C:\Program Files\HITACHI\ExSORT64\Log (64 ビット版)</p> <p>ただし、ディレクトリが存在しない、書き込み権限がない、ディスクの空きがないなどの理由により、上記ディレクトリに出力できない場合は、次の優先順位で示すディレクトリとなります。</p> <ol style="list-style-type: none"> <li>1. 環境変数「TMP」で指定されているディレクトリ</li> <li>2. 環境変数「TEMP」で指定されているディレクトリ</li> <li>3. Windows のディレクトリ</li> </ol> <p>【例】 C:\WINDOWS</p>
UNIX	<p>次の優先順位で使用するディレクトリが決まります。</p> <ol style="list-style-type: none"> <li>1. 環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル」)」の「EXS_LOGPATH」項目で指定した、ログ出力先ディレクトリ</li> <li>2. /opt/HIEXSORT/log (32 ビット版) /opt/HIEXSORT64/log (64 ビット版)</li> </ol>

なお、追加書きのダンプログファイル「exsortdmp」の出力に失敗した場合に出力される「exsortdmp\_Pid\_YYMMDDhhmmss」ファイルおよび「EXSORTIODMP\_Pid\_YYMMDDhhmmss\_x」ファイルは、次の表で示すディレクトリ下に作成されます。

表 19-3 「exsortdmp\_Pid\_YYMMDDhhmmss」ファイルおよび「EXSORTIODMP\_Pid\_YYMMDDhhmmss\_x」ファイルの出力ディレクトリ

OS	作成ディレクトリ
Windows	<p>次の優先順位で使用するディレクトリが決まります。</p> <ol style="list-style-type: none"> <li>1. 環境変数「TMP」で指定されているディレクトリ</li> <li>2. 環境変数「TEMP」で指定されているディレクトリ</li> <li>3. カレントディレクトリ</li> </ol>
UNIX	<p>次の優先順位で使用するディレクトリが決まります。</p> <ol style="list-style-type: none"> <li>1. /usr/tmp</li> <li>2. /tmp</li> <li>3. カレントディレクトリ</li> </ol>

### デバッグ情報ファイルを出力するメッセージ ID

デバッグ情報ファイルを出力するメッセージ ID を次の表に示します。丸括弧 ( ) 内のファイル名は、64 ビット版のデバッグ情報ファイル名を示します。

表 19-4 デバッグ情報ファイルを出力するメッセージ

メッセージ ID	exsortdmp (exsortdmp64)	exsortrdmp_Pid_YYMMDDhhmm mss ※ (exsortrdmp64_Pid_YYMMDD hhmmss)
KBLS1208-E	○	×
KBLS1211-E	○	○
KBLS1212-E	○	×
KBLS1309-E	○	×
KBLS1320-E	○	○
KBLS1407-E	○	×
KBLS1411-E	○	○
KBLS1507-E	○	×
KBLS1633-E	○	×
KBLS1634-E	○	×
KBLS1636-E	○	×
KBLS1655-E	○	×
KBLS1656-E	○	×
KBLS1672-E	○	×
KBLS1700-E	○	×
KBLS1720-E	○	×
KBLS1721-E	○	×
KBLS1750-E	○	×
KBLS2042-E	○	○
KBLS2060-E	○	○

(凡例)

- : 出力する。
- × : 出力しない。

注※

ファイル名には、Pid (プロセス id) と YYMMDDhhmmss (ダンプを出力した年月日時分秒) が付加されます。

### 19.1.3 ログファイル

SORT EE では、exsort コマンドのレポート情報をログファイルに出力します。ログファイルの種類を次の表に示します。括弧内のファイル名は、64 ビット版のログファイル名を示します。



表 19-5 ログファイルの種類

ログファイル名	ファイルの説明	出力方式
ExSORT.log(ExSORT64.log)	(exsort コマンドログファイル) exsort コマンドのコマンドログ情報をテキスト形式で出力します。 SORT EE 08-50 より前のファイル最大サイズは 20MB です。 SORT EE 08-50 以降のファイル最大サイズは 50MB です。	追加書き
ExSORT1.log(ExSORT641.log)	(ExSORT.log のバックアップファイル) ExSORT.log の出力量が最大サイズに達すると、「ExSORT1.log」のファイル名称でバックアップファイルを作成します。すでにバックアップファイルが存在する場合、既存のバックアップファイルを破棄し、新たにバックアップファイルを作成します。必要な場合は、定期的にバックアップファイルの内容を保存してください。 ファイルのバックアップローテーションについては、「図 19-2 ファイルのバックアップローテーション (ExSORT1.log の場合の例)」を参照してください。	上書き
ExSORT.conf(ExSORT64.conf)	(ExSORT.log のコントロールファイル) ExSORT.log を出力するためのコントロールファイルです。	—
exsortrepo.log(exsortrepo64.log)	(レポートログファイル) exsort コマンド実行時のリソース使用量などのレポート情報をテキスト形式で出力します。ファイルの最大サイズの初期値は 50MB です。 ファイルの最大サイズは、環境ファイル「EXS_REPOLOG_FILESIZE」で、5～100 (単位: MB) の範囲で変更できます。環境ファイル「EXS_REPOLOG_FILESIZE」については、「15.1.1(20) EXS_REPOLOG_FILESIZE」を参照してください。	追加書き
exsortrepo1.log(exsortrepo641.log)	(exsortrepo.log のバックアップファイル) exsortrepo.log の出力量が最大サイズに達すると、「exsortrepo1.log」のファイル名称でバックアップファイルを作成します。すでにバックアップファイルが存在する場合、既存のバックアップファイルを破棄し、新たにバックアップファイルを作成します。必要な場合は、定期的にバックアップファイルの内容を保存してください。 ファイルのバックアップローテーションについては、「図 19-2 ファイルのバックアップローテーション (ExSORT1.log の場合の例)」を参照してください。	上書き
exsortrepo.conf(exsortrepo64.conf)	(exsortrepo.log のコントロールファイル) exsortrepo.log を出力するためのコントロールファイルです。	—

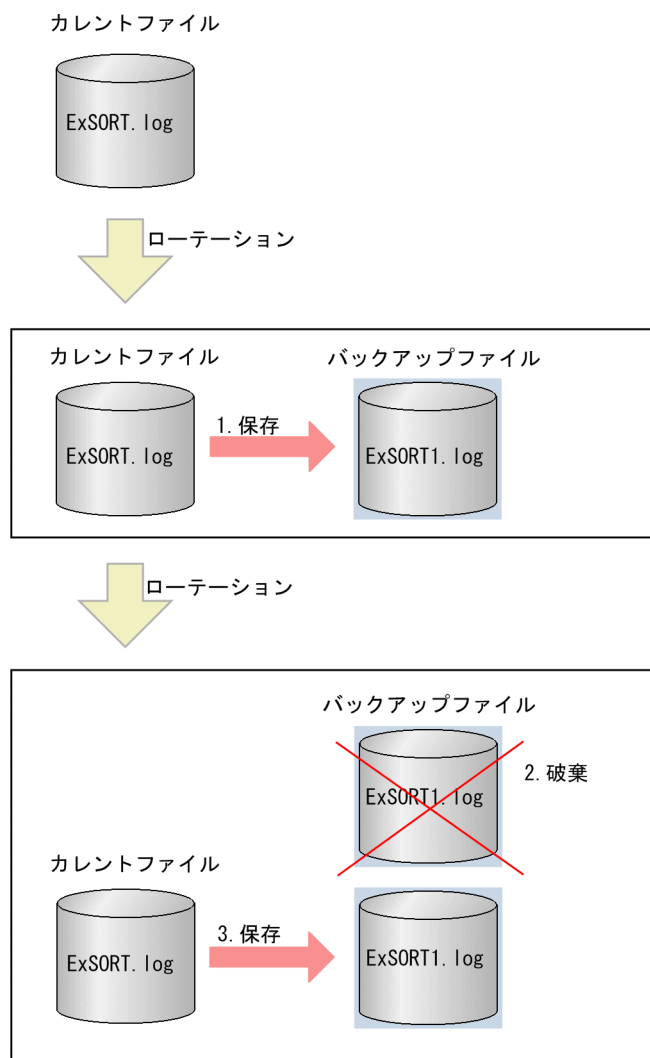
(凡例)

— : なし

## 19. 障害時の対処方法

ファイルのバックアップローテーションを次の図に示します。

図 19-2 ファイルのバックアップローテーション (ExSORT1.log の場合の例)



カレントファイルとバックアップファイルの最大2ファイル作成されます。

カレントファイルの出力量が最大サイズに達すると、ファイル名称を変更してバックアップファイルとして保存されます (1.)。カレントファイルには、改めて先頭から追加書きで出力されるため、ファイルサイズはいったん小さくなります。

2回目以降のローテーションでは、古いバックアップは破棄し (2.)、ファイル名称を変更して新たなバックアップファイルとして保存されます (3.)。

## (1) ログファイルの出力先ディレクトリ

SORT EE のログファイルは、次の表で示すディレクトリ下に作成されます。

表 19-6 ログファイルを出力するディレクトリ

OS	ログファイル出力先ディレクトリ
Windows	<p>次の優先順位で使用するディレクトリが決まります。</p> <ol style="list-style-type: none"> <li>1. 環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル」)」の「EXS_LOGPATH」項目で指定した、ログ出力先ディレクトリ</li> <li>2. SORT EE インストール先ディレクトリ下の Log ディレクトリ (例) C:\Program Files\HITACHI\ExSORT\Log (32 ビット版) C:\Program Files\HITACHI\ExSORT64\Log (64 ビット版)</li> </ol> <p>ただし、エラーによっては、上記ディレクトリに出力しない場合があります。この場合、次の優先順位で示すディレクトリとなります。</p> <ol style="list-style-type: none"> <li>1. 環境変数「TMP」で指定されているディレクトリ</li> <li>2. 環境変数「TEMP」で指定されているディレクトリ</li> <li>3. Windows のディレクトリ (例えば、C:\WINDOWS)</li> </ol>
UNIX	<p>次の優先順位で使用するディレクトリが決まります。</p> <ol style="list-style-type: none"> <li>1. 環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル」)」の「EXS_LOGPATH」項目で指定した、ログ出力先ディレクトリ</li> <li>2. /opt/HIEXSORT/log (32 ビット版) /opt/HIEXSORT64/log (64 ビット版)</li> </ol>

ログファイルの出力先ディレクトリは、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル」)」で指定します。環境ファイルには、次のとおり指定してください。なお、ログファイル出力先ディレクトリに書き込み権限がない場合、ログファイルは出力されません。

EXS\_LOGPATH= ログファイル出力先ディレクトリ名

環境ファイルの作成先ディレクトリについては、「15.1 SORT EE の環境ファイルの設定」を参照してください。

## (2) ExSORT.log および exsortrepo.log ファイルの出力オプション

exsort コマンドの正常終了・異常終了に関係なく、exsort コマンドのコマンドログ情報が ExSORT.log に、exsort コマンド実行時のレポート情報が exsortrepo.log ファイルへ出力されます。

これらのログを出力しない、または一部情報の出力を抑止できます。

### (a) レポートログの出力抑止

レポートログの出力を抑止するには、環境ファイル「Exsortconf.ini ファイル (64 ビット

## 19. 障害時の対処方法

ト版は「Exsortconf64.ini ファイル」の「EXS\_EXSORTDMP\_OPT」項目に「1」を指定してください。この項目を省略した場合、初期値に「EXS\_EXSORTDMP\_OPT=0」が仮定されます。

EXS\_EXSORTDMP\_OPT= オプション番号

表 19-7 レポートログオプション

オプション番号	意味
0	exsort コマンド実行時のレポートログが出力されます。
1	exsort コマンド実行時のレポートログは出力されません。

環境ファイルの作成先ディレクトリについては、「15.1 SORT EE の環境ファイルの設定」を参照してください。

### (b) コマンドログ、レポートログの出力抑止（一部）

コマンドログ、およびレポートログ（一部）の出力を抑止するには、環境変数「EXSORTLOGSUPPRESS（64 ビット版は「EXSORT64LOGSUPPRESS）」に「YES」を指定してください。

なお、環境ファイル「Exsortconf.ini ファイル（64 ビット版は「Exsortconf64.ini ファイル）」の「EXS\_EXSORTDMP\_OPT」項目に「1」を指定した場合、レポートログは出力されないため、環境変数「EXSORTLOGSUPPRESS（64 ビット版は「EXSORT64LOGSUPPRESS）」の指定は無効となります。

出力を抑止する情報は、指定されたパラメタファイル、出力ファイル情報、および完了メッセージの一部です。

環境変数の指定方法については、環境変数「14.1.1(36) EXSORTLOGSUPPRESS【08-50以降】」、「14.1.2(36) EXSORT64LOGSUPPRESS【08-50以降】」を参照してください。

## 19.1.4 完了メッセージオプション

SORT Version7 - Extended Edition 07-60 以降から、処理完了時に出力するメッセージのメッセージ ID と形式を変更しています。

SORT Version7 - Extended Edition 07-50 以前の完了メッセージを出力したい場合は、SORT EE の Exsortconf.ini ファイルに、次のオプションを指定してください。

オプション番号	意味
1	SORT Version7 - Extended Edition 07-50 以前の完了メッセージ (KBLS1001-I ~ KBLS1006-I) を出力

オプション番号	意 味
1 以外または指定なし	SORT Version7 - Extended Edition 07-60 以降の完了メッセージ (KBLS1007-I ~ KBLS1009-I) を出力

Exsortconf.ini ファイルに、次のとおりに指定してください。

```
EXS_COMPLETE_MSGOPT=1
```

Exsortconf.ini ファイルの作成先ディレクトリについては、「15.1 SORT EE の環境ファイルの設定」を参照してください。

### 19.1.5 キーまたは項目の不正による障害の対処方法

コマンド処理中にキーまたは項目の不正による障害が発生した場合は、処理を中止し、すべてのファイルをクローズします。また、次のメッセージを表示します。

- KBLS1211-E キーにエラーデータを検出しました。
- KBLS1320-E 比較項目にエラーデータを検出しました。
- KBLS1411-E 集約項目にエラーデータを検出しました。

これらのメッセージが表示された場合に、障害の詳しい原因を知る方法を次に示します。

#### (1) メッセージを確認する

19.1.5 項のリード文に記述したメッセージには、次のような情報が表示されます。

- 障害の原因となったレコードを含むファイルのファイル名
- 障害の原因となったレコードのレコード番号
- 不正なキーまたは項目の番号

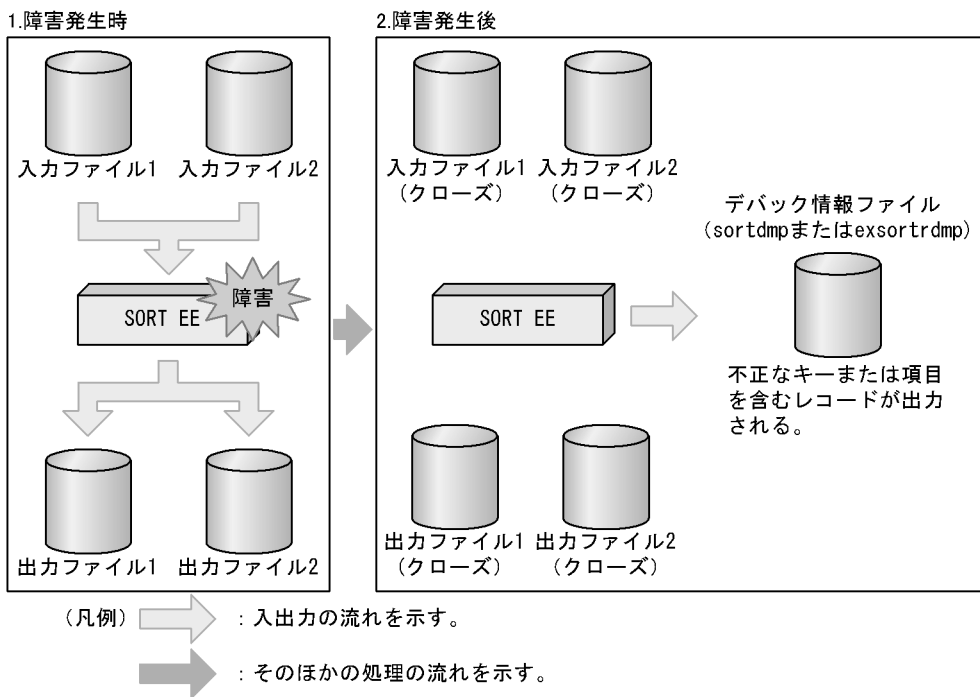
#### (2) sortrdmp の内容を確認する

キーまたは項目の不正による障害が発生すると、デバッグ情報ファイル「sortrdmp」または「exsortrdmp」に、不正なキーまたは項目を含んだレコードが出力されます。sortrdmp または exsortrdmp の内容を確認することによって、どのような不正があったかを知ることができます (sortrdmp および exsortrdmp の内容はテキストエディタで確認することができます)。

なお、sortrdmp および exsortrdmp に出力されたレコードは、実際のレコードとは長さ異なる場合がありますが、キーまたは項目はすべて含んでいます。

キーまたは項目の不正による障害時の処理を次の図に示します。

図 19-3 キーまたは項目の不正による障害時の処理

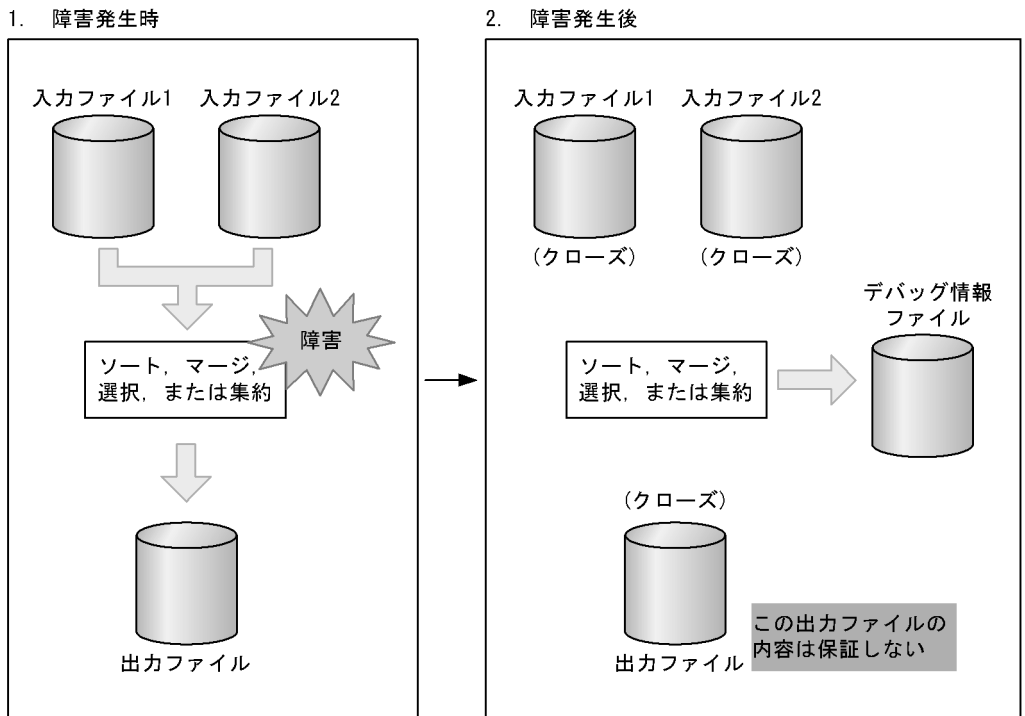


## 19.2 SORT の障害情報

### 19.2.1 SORT の入出力障害時のファイルの扱い

ソート、マージ、選択または集約処理中に入出力障害が発生した場合は、処理を中止してすべてのファイルをクローズし、メッセージを表示します。また、そのときの出力ファイルの内容は保証しません。入出力障害時のファイルの扱いを次の図に示します。

図 19-4 入出力障害時のファイルの扱い



### 19.2.2 デバッグ情報ファイル

コマンド処理が異常終了した場合は、デバッグ情報ファイルを作成します。デバッグ情報ファイルは、次の表で示すディレクトリ下にテキスト形式で作成されます。すでに同名のファイルがある場合は上書きされます。デバッグ情報の出力先ディレクトリに書き込み権限がない場合、デバッグ情報は出力されません。

デバッグ情報ファイルの出力先ディレクトリとファイル名を次に示します。

表 19-8 デバッグ情報ファイルを出力するディレクトリ

OS	作成ディレクトリ
Windows	<p>次の優先順位で使用するディレクトリが決まります。</p> <ul style="list-style-type: none"> <li>• sortdmp/sortrdmp/SORTDMP2/SORTIODMP <ol style="list-style-type: none"> <li>1. wnsort.wse ファイルの dmp_dir オプションで指定したディレクトリ</li> <li>2. 環境変数「TMP」または「TEMP」で指定したディレクトリ</li> <li>3. 環境変数「USERPROFILE」または「SYSTEMROOT」で指定したディレクトリ</li> <li>4. カレントディレクトリ</li> </ol> </li> <li>• sortidmp <ol style="list-style-type: none"> <li>1. 環境変数「TMP」または「TEMP」で指定したディレクトリ</li> <li>2. 環境変数「USERPROFILE」または「SYSTEMROOT」で指定したディレクトリ</li> <li>3. カレントディレクトリ</li> </ol> </li> </ul>
UNIX	<p>次の優先順位で使用するディレクトリが決まります。</p> <ul style="list-style-type: none"> <li>• sortdmp/sortrdmp <ol style="list-style-type: none"> <li>1. 環境変数「SORT_DMPDIR」で指定したディレクトリ</li> <li>2. /usr/tmp</li> <li>3. /tmp</li> <li>4. カレントディレクトリ</li> </ol> </li> <li>• SORTDMP2/SORTIODMP <ol style="list-style-type: none"> <li>1. 環境変数「SORT_DMPDIR」で指定したディレクトリ</li> <li>2. /usr/tmp</li> </ol> </li> </ul>

表 19-9 デバッグ情報ファイル名

OS	SORT の種別	デバッグ情報ファイル名
Windows	32 ビット版	sortdmp sortidmp sortrdmp SORTDMP2 SORTIODMP
	64 ビット版	sortdmp64 sortidmp sortrdmp64 SORTDMP264 SORTIODMP64
UNIX	—	sortdmp sortrdmp SORTDMP2 SORTIODMP

(凡例)

— : 該当しない。

## デバッグ情報ファイルを出力するメッセージ ID

デバッグ情報ファイルを出力するメッセージ ID を次の表に示します。丸括弧 ( ) 内のファイル名は、64 ビット版のデバッグ情報ファイル名を示します。



表 19-10 デバッグ情報ファイルを出力するメッセージ ID

メッセージ ID	デバッグ情報ファイル				
	sortdmp (sortdmp64)	sortidmp※	sortrdmp (sortrdmp64)	SORTDMP2 (SORTDM P264)	SORTIODMP (SORTIOD MP64)
KBLS201-E	○	○	○	○	×
KBLS202-E	○	○	×	○	×
KBLS207-E	○	○	×	○	×
KBLS209-E	○	○	×	○	×
KBLS210-E	○	○	×	○	○
KBLS211-E	○	○	×	○	×
KBLS212-E	○	○	×	○	×
KBLS213-E	○	○	×	○	×
KBLS217-E	○	○	×	○	×
KBLS218-E	○	○	×	○	×
KBLS290-E	○	○	×	○	×
KBLS300-E	○	○	×	○	×
KBLS401-E	○	○	×	○	×
KBLS402-E	○	○	×	○	×
KBLS501-E	○	○	○	×	×
KBLS601-E	○	○	○	×	×
KBLS602-E	○	○	○	×	×
KBLS605-E	○	○	×	×	×
KBLS606-E	○	○	×	×	×
KBLS607-E	○	○	×	×	×

(凡例)

○ : 出力する。

× : 出力しない。

注※

Windows 版の C 言語の SORT 関数でエラーとなった場合に出力するファイルです。

### 19.2.3 キーまたは項目の不正による障害の場合

ソート、マージ、選択または集約処理中にキーの不正による障害が発生した場合は、処理を中止してすべてのファイルをクローズし、メッセージを表示します。このメッセージが表示された場合に、障害の詳しい原因を知る方法を次に示します。

### (1) メッセージを確認する

メッセージには次のような情報が表示されます。

- 障害の原因となったレコードを含むファイルのファイル名
- 障害の原因となったレコードのレコード番号
- 不正なキーまたは項目のキー番号

コマンドプロンプトで実行した場合に、コマンドプロンプトにメッセージが表示されま  
す。表示されるメッセージのメッセージ ID は次のとおりです。

- KBLS201-E
- KBLS501-E
- KBLS601-E
- KBLS602-E

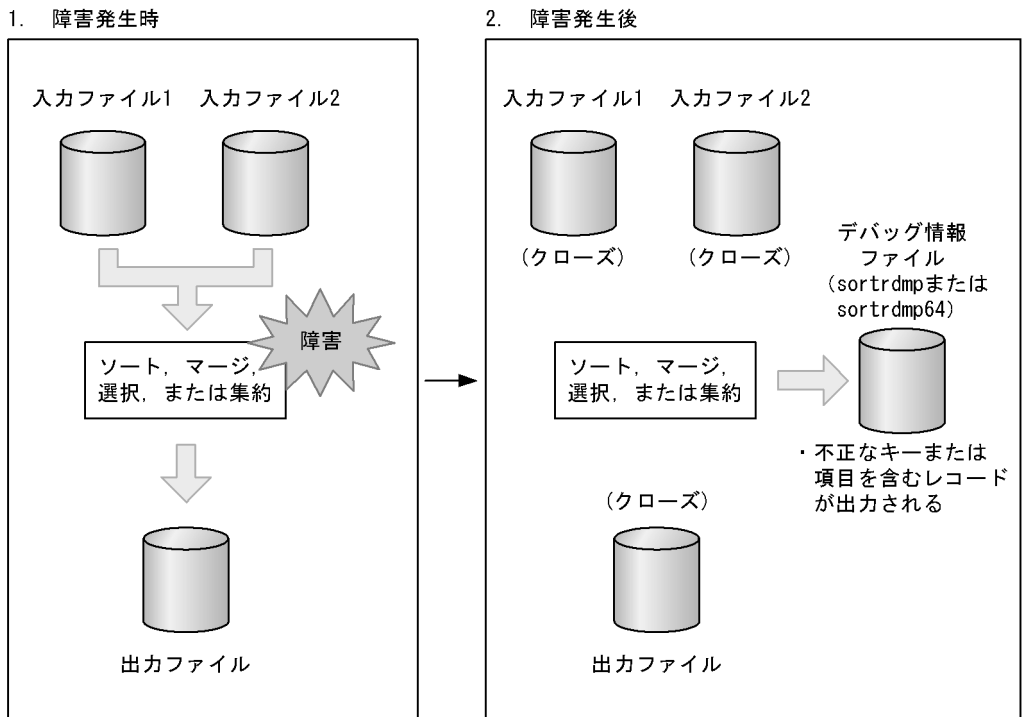
### (2) sortrdmp または sortrdmp64 の内容を確認する

キーまたは項目の不正による障害が発生すると、デバッグ情報ファイル「`sortrdmp`」ま  
たは「`sortrdmp64`」に、不正なキーまたは項目を含んだレコードが出力されます。  
`sortrdmp` または `sortrdmp64` の内容を見ることによって、どのような不正があったかを  
知ることができます。

なお、`sortrdmp` または `sortrdmp64` に出力されたレコードは、実際のレコードとは長さ  
が異なることがありますが、キーまたは項目をすべて含んでいます。

キーまたは項目の不正による障害時の処理を次の図に示します。

図 19-5 キーまたは項目の不正による障害時の処理



### 19.2.4 C 言語で作成したプログラムの不正による障害の場合 (Windows の場合)

C 言語で作成したプログラムの不正による障害が発生した場合、その詳しい原因を知る方法を示します。

#### (1) sortidmp の内容を確認する

障害が発生すると、インタフェース情報ファイル「sortidmp」に、不正な情報を含んだテーブルが出力されます。sortidmp の内容を見ることで、どのような不正があったかを知ることができます。

また、メッセージ ID 「KBLS901-E」のメッセージは、C 言語で作成したプログラムの不正による障害を示します。



# 20 SORT から SORT EE への移行ガイド

この章では、SORT から SORT EE に移行する場合のコマンド、およびワークバッファサイズの移行内容について説明します。

---

20.1 SORT と SORT EE の処理比較

---

20.2 UNIX 版 SORT コマンドの移行

---

20.3 Windows 版 SORT コマンドの移行

---

20.4 ワークバッファサイズ

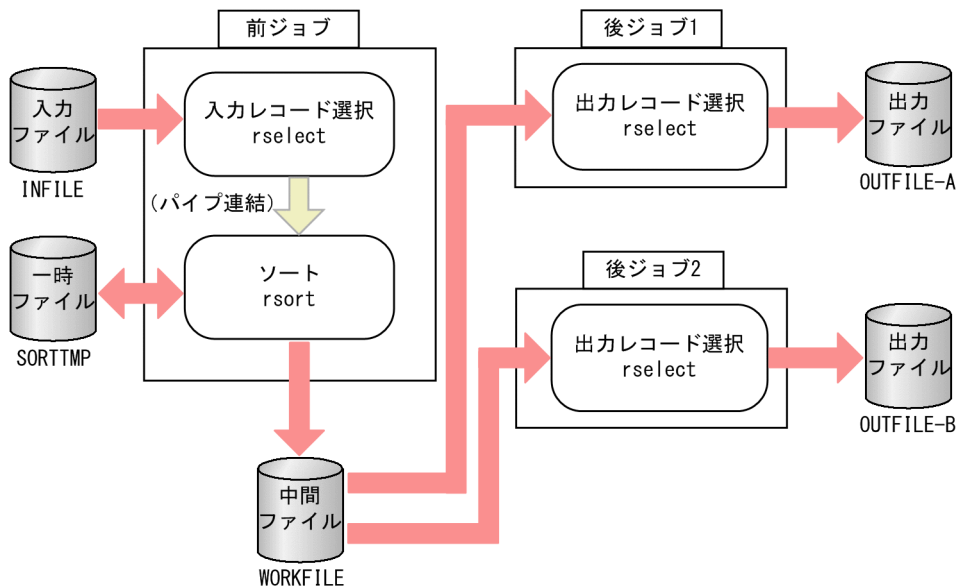
---

## 20.1 SORT と SORT EE の処理比較

SORT は、機能ごとにコマンドがあり、各コマンドを組み合わせて利用します。

例えば、入力ファイルから条件に一致するレコードを入力し、ソートした後、条件ごとに 2 つの出力ファイルへ出力するには、次の図に示すとおり、複数のコマンドを実行します。この場合、各コマンドの実行結果を保存する中間ファイルが必要になります。

図 20-1 SORT のコマンド処理



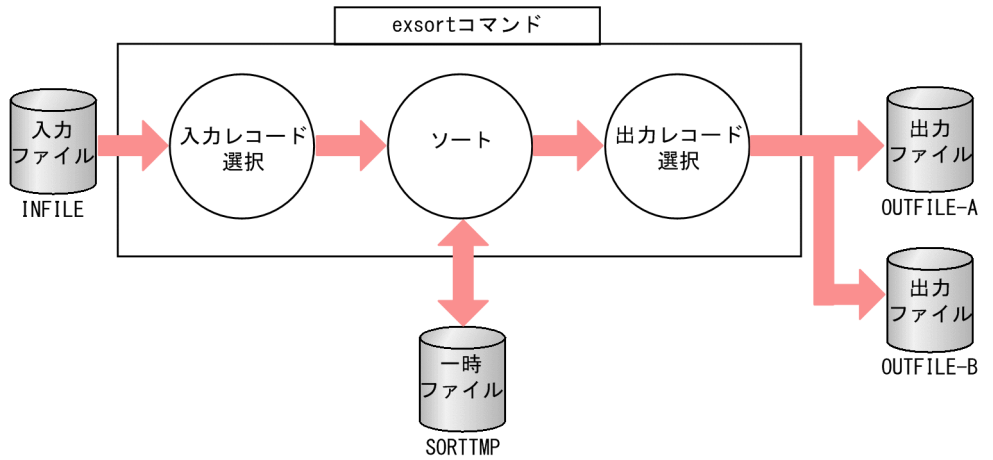
```
前ジョブ
rselect -A -sl"-z +0-4 < N' 5000'" INFILE | rsort -A -z +0-4 -o WORKFILE -T SORTTMP
```

```
後ジョブ1
rselect -A -sl"-z +0-4 < N' 2000'" WORKFILE -o OUTFILE-A
```

```
後ジョブ2
rselect -A -sl"-z +0-4 >= N' 2000'" WORKFILE -o OUTFILE-B
```

SORT EE では、次の図に示すとおり、複数の処理を 1 コマンドで実現できます。また、中間ファイルも必要としません。

図 20-2 SORT EE のコマンド処理



```

exsort
-function sort
-key z+0-4
-inpfile fileorg=A file=INFILE include="z+0-4 < N' 5000' "
-outfile file=OUTFILE-A include="z+0-4 < N' 2000' "
-outfile file=OUTFILE-B include="z+0-4 >= N' 2000' "
-sorttmpdir SORTTMP

```

## 20.2 UNIX 版 SORT コマンドの移行

### 20.2.1 rsort コマンド

rsort コマンドの移行内容を次の表に示します。

- 項番 2～4 は、rsort コマンドと exsort コマンドで指定値の形式が異なります。各オペランドの詳細については、次の表に示す該当する章を参照してください。
- 項番 5～10 は、rsort コマンドの指定値を、exsort コマンドのパラメタ、またはオペランド値としてそのまま転記します。

表 20-1 rsort コマンド移行

項番	rsort コマンドのオペランド	exsort コマンドのパラメタおよびオペランド	移行内容
1	—	-function △ sort	転記しません。
2	-ソートオプション	-cmdopt △コマンドオプション	20.2.6(1)を参照してください。
3	-入力ファイルのファイル編成： 出力ファイルのファイル編成	-infile △ fileorg= 入力ファイル編成 -outfile △ fileorg= 出力ファイル編成	20.2.6(2)を参照してください。
4	-キー属性△+キー位置・ キー長	-key △キー属性+キー位置・ キー長	20.2.6(3)を参照してください。
5	-L レコード長	-infile △ reclen= レコード長	指定値を転記します。
6	-o 出力ファイル名	-outfile △ file= 出力ファイル名	指定値を転記します。
7	-y ワークバッファサイズ	-worksize △ワークバッファサイズ※1	指定値を転記します。
8	-T ディレクトリ名	-sorttmpdir △ディレクトリ名※2	指定値を転記します。
9	-T2 ディレクトリ名 2	-mergetmpdir △ディレクトリ名 2	指定値を転記します。
10	入力ファイル名	-infile △ file= 入力ファイル名	指定値を転記します。

(凡例)

— : 該当しません。

△ : 1 文字以上の空白を示します。

注※ 1

KB 単位指定の場合は、MB 単位に変換した値を指定してください。



## 注※2

`exsort` コマンドのソート機能を使用する場合、一時ファイル作成ディレクトリ名を指定する「`-sorttmpdir` パラメタ」または環境変数「`EXSORTSORTTMPDIR`」の指定が必須です。どちらも指定しない場合は、`SORT` の環境変数「`SORTOPT`」の「`-T`」の指定を仮定します。すべての指定がなかった場合は、`exsort` コマンドは「`KBLS1674-E` 一時ファイルのディレクトリが指定されていません。」のエラーを出力して失敗します。  
`SORT EE` の環境変数の設定については、「20.2.7 `SORT` 環境変数の移行」を参照してください。

20.2.2 `rmerge` コマンド

`rmerge` コマンドの移行内容を次の表に示します。

- 項番 2～4 は、`rmerge` コマンドと `exsort` コマンドで指定形式が異なります。各オペランドの詳細については、次の表に示す該当する章を参照してください。
- 項番 5～7 は、`rmerge` コマンドの指定値を、`exsort` コマンドのパラメタまたはオペランド値としてそのまま転記します。

表 20-2 `rmerge` コマンド移行

項番	<code>rmerge</code> コマンドのオペランド	<code>exsort</code> コマンドのパラメタおよびオペランド	移行内容
1	—	<code>-function Δ merge</code>	転記しません。
2	<code>- マージオプション</code>	<code>-cmdopt Δ オプション</code>	20.2.6(1) を参照してください。
3	<code>- 入力ファイルのファイル編成： 出力ファイルのファイル編成</code>	<code>-infile Δ fileorg= 入力ファイル編成</code> <code>-outfile Δ fileorg= 出力ファイル編成</code>	20.2.6(2) を参照してください。
4	<code>- キー属性Δ + キー位置 - キー長</code>	<code>-key Δ キー属性 + キー位置 - キー長</code>	20.2.6(3) を参照してください。
5	<code>-L</code> レコード長	<code>-infile Δ reclen=</code> レコード長	指定値を転記します。
6	<code>-o</code> 出力ファイル名	<code>-outfile Δ file=</code> 出力ファイル名	指定値を転記します。
7	入力ファイル名	<code>-infile Δ file=</code> 入力ファイル名	指定値を転記します。

(凡例)

— : 該当しません。

Δ : 1 文字以上の空白を示します。

20.2.3 `rselect` コマンド

`rselect` コマンドの移行内容を次の表に示します。

## 20. SORT から SORT EE への移行ガイド

- 項番 2～4 は、`rselect` コマンドと `exsort` コマンドで指定形式が異なります。各オペランドの詳細については、次の表に示す該当する章を参照してください。
- 項番 5～7 は、`rselect` コマンドの指定値を、`exsort` コマンドのパラメタまたはオペランド値としてそのまま転記します。

表 20-3 rselect コマンドの移行

項番	rselect コマンドのオペランド	exsort コマンドのパラメタおよびオペランド	移行内容
1	—	-function △ copy	転記しません。
2	-選択オプション	-cmdopt △オプション	20.2.6(1) を参照してください。
3	-入力ファイルのファイル編成: 出力ファイルのファイル編成	-infile △ fileorg=入力ファイル編成 -outfile △ fileorg=出力ファイル編成	20.2.6(2) を参照してください。
4	-s1" 選択条件式 "	-infile △ include=" 選択条件式 "	20.2.3(1) を参照してください。
5	-L レコード長	-infile △ reclen= レコード長	指定値を転記します。
6	-o 出力ファイル名	-outfile △ file= 出力ファイル名	指定値を転記します。
7	入力ファイル名	-infile △ file= 入力ファイル名	指定値を転記します。

(凡例)

- : 該当しません。
- △ : 1 文字以上の空白を示します。

### (1) 選択条件式の移行内容

#### ● rselect コマンドの選択条件式

```
-s1"-比較項目属性△+項目位置-項目長△比較演算子△条件値  
[△論理演算子△-比較項目属性△+項目位置-項目長△比較演算子△条件値…]"
```

#### ● exsort コマンドの選択条件式

```
-infile△ {include | omit} ="比較項目属性+項目位置-項目長△比較演算子△条件値  
[△論理演算子△比較項目属性+項目位置-項目長△比較演算子△条件値…]"
```

- 項番 2 の比較項目属性は、`exsort` コマンドではハイフン (-) を付けません。
- 項番 3～6 は、`rselect` コマンドの指定値を、`exsort` コマンドのオペランド値としてそのまま転記します。
- 項番 7 の論理演算子は、`exsort` コマンドでは英大文字です。

表 20-4 選択条件式の移行

項番	rselect コマンド	意味	移行内容	
1	コマンドオプション	-u <sup>※</sup>	指定なし：条件を満たす入力レコードを選択	include=
			指定あり：条件を満たさない入力レコードを選択	omit=
2	比較項目属性	-j	文字 (JIS8 単位コード)	j
		-c	文字	c
		-n	符号付き固定小数点 2 進数	n
		-m	符号なし 2 進数	m
		-p	内部 10 進数	p
		-z	外部 10 進数	z
		-x	左符号付き外部 10 進数	x
		-s	右分離符号付き外部 10 進数	s
		-t	左分離符号付き外部 10 進数	t
		-g	日付	g
3	+ 項目位置	比較項目の開始バイト位置	+ 項目位置	
4	- 項目長	比較項目のバイト長	- 項目長	
5	比較演算子	=	条件値と等しい値の入力レコードを選択	=
		!=	条件値と異なる値の入力レコードを選択	!=
		>	条件値より大きい値の入力レコードを選択	>
		>=	条件値より大きいかまたは等しい値の入力レコードを選択	>=
		<	条件値より小さい値の入力レコードを選択	<
		<=	条件値より小さいかまたは等しい値の入力レコードを選択	<=
6	条件値	C' 文字条件値 '	文字属性の比較項目に対する条件値	C' 文字条件値 '
		N' 数値条件値 '	数値属性の比較項目に対する条件値	N' 数値条件値 '
		D' 日付条件値 '	日付属性の比較項目に対する条件値	D' 日付条件値 '

項番	rselect コマンド	意味	移行内容	
7	論理演算子	and	複数の条件式を論理積で結合する	AND
		or	複数の条件式を論理和で結合する	OR

## 注※

exsort コマンドには、「-u」オプションに相当するコマンドオプションはありません。  
条件を満たさないレコード選択を指定する場合は、「omit オペランド」で指定します。

## 20.2.4 rsum コマンド

rsum コマンドの移行内容を次の表に示します。

- 項番 2 ～ 5 は、rsum コマンドと exsort コマンドで指定形式が異なります。各オペランドの詳細については、次の表に示す該当する章を参照してください。
- 項番 6 ～ 8 は、rsum コマンドの指定値を、exsort コマンドのパラメタまたはオペランド値としてそのまま転記します。

表 20-5 rsum コマンドの移行

項番	rsum コマンドのオペランド	exsort コマンドのパラメタおよびオペランド	移行内容
1	—	-function △ copy	転記しません。
2	-集約オプション	-cmdopt △オプション	20.2.6(1)を参照してください。
3	-入力ファイルのファイル編成： 出力ファイルのファイル編成	-inpfile △ fileorg=入力ファイル編成 -outfile △ fileorg=出力ファイル編成	20.2.6(2)を参照してください。
4	-キー属性△+キー位置- キー長	-key △キー属性+キー位置- キー長	20.2.6(3)を参照してください。
5	-a 集約項目属性△+項目位置- 項目長	-sum △集約項目属性+項目位置- 項目長	20.2.6(4)を参照してください。
6	-L レコード長	-inpfile △ reclen=レコード長	指定値を転記します。
7	-o 出力ファイル名	-outfile △ file=出力ファイル名	指定値を転記します。
8	入力ファイル名	-inpfile △ file=入力ファイル名	指定値を転記します。

## (凡例)

—：該当しません。  
△：1文字以上の空白を示します。

## 20.2.5 rsortsum コマンド

rsortsum コマンドの移行内容を次の表に示します。

- 項番 2 ～ 5 は、rsortsum コマンドと exsort コマンドで指定形式が異なります。各オペランドの詳細については、次の表に示す該当する章を参照してください。
- 項番 6 ～ 10 および項番 12 は、rsortsum コマンドの指定値を、exsort コマンドのパラメタまたはオペランド値としてそのまま転記します。

表 20-6 rsortsum コマンドの移行

項番	rsortsum コマンドのオペランド	exsort コマンドのパラメタおよびオペランド	移行内容
1	—	-function △ sort	転記しません。
2	-機能オプション	-cmdopt △オプション	20.2.6(1) を参照してください。
3	-入力ファイルのファイル編成： 出力ファイルのファイル編成	-infile △ fileorg= 入力ファイル編成 -outfile △ fileorg= 出力ファイル編成	20.2.6(2) を参照してください。
4	-キー属性△ + キー位置・キー長	-key △キー属性 + キー位置・キー長	20.2.6(3) を参照してください。
5	-a 集約項目属性△ + 項目位置・項目長	-sum △集約項目属性 + 項目位置・項目長	20.2.6(4) を参照してください。
6	-L レコード長	-infile △ reclen= レコード長	指定値を転記します。
7	-o 出力ファイル名	-outfile △ file= 出力ファイル名	指定値を転記します。
8	-y ワークバッファサイズ	-worksize △ワークバッファサイズ※1	指定値を転記します。
9	-T ディレクトリ名	-sorttmpdir △ディレクトリ名※2	指定値を転記します。
10	-T2 ディレクトリ名 2	-mergetmpdir △ディレクトリ名 2	指定値を転記します。
11	-W 中間ファイルディレクトリ名	—	—
12	入力ファイル名	-infile △ file= 入力ファイル名	指定値を転記します。

(凡例)

— : 該当しません。

△ : 1 文字以上の空白を示します。

注※ 1

KB 単位指定の場合は、MB 単位に変換した値を指定してください。

## 注※ 2

exsort コマンドのソート機能を使用する場合、一時ファイル作成ディレクトリ名を指定する「-sorttmpdir パラメタ」または環境変数「EXSORTSORTTMPDIR」の指定は必須です。どちらも指定しなかった場合、SORT の環境変数「SORTOPT」の「-T」の指定を仮定します。すべての指定がなかった場合、exsort コマンドは「KBLS1674-E 一時ファイルのディレクトリが指定されていません。」のエラーを出力して失敗します。

SORT EE の環境変数の設定については、「20.2.7 SORT 環境変数の移行」を参照してください。

## 20.2.6 コマンド共通項目

## (1) コマンドオプション

rsort, rmerge, rselect, rsum および rsortsum コマンドのコマンドオプションの移行内容を表 20-7 に示します。

## ● rsort/rmerge/rselect/rsum/rsortsum コマンドのオプション

```
{-オプション[△-オプション...] | -オプション[オプション...]}
```

オプションを複数指定する場合、次の 2 とおりの方法で指定できます。

- 個々のオプションを空白で分けて指定する場合、オプションの前にハイフン (-) を付けて指定します。
- つなげて指定する場合、2 個目以降のオプションにはハイフン (-) を付けません。

## ● exsort コマンドのコマンドオプション

```
-cmdopt△オプション[:オプション...]
```

オプションを複数指定する場合、コロン (:) でオプションを区切ります。

表 20-7 コマンドオプションの移行

オプション	意味	rsort	rmerge	rselect	rsum	rsortsum	exsort -cmdopt
-d	重複レコードの入力順序保存	○	—	—	—	○	DUP
-e	入出力ファイル排除	○	○	○	○	○	EXC
-i	既存ファイルの書き換え防止	○	○	○	○	○	_※1
-v	EBCDIK コード比較	○	○	○	○	○	EBCJ
-v1	EBCDIK コード比較 (入力 JIS8)	—	—	○	—	—	EBCJ
-v2	EBCDIK コード比較 (入力 EBCDIK)	—	—	○	—	—	EBCE

オプション	意味	rsort	rmerge	rselect	rsum	rsortsum	exsort -cmdopt
-q	キャッシュファイルの同期出力	○	○	○	○	○	SYNC
-B	big endian 比較	○	○	○	○	○	BIG※2
-k	強制マージ	—	○	—	—	—	FMRG
-u	条件を満たさないレコード選択	—	—	○	—	—	_※3
-b	10 進数オーバーフロー判定	—	—	—	○	○	OVFLW
-if	内部 10 進数符号保障オプション	—	—	—	○	○	ABS

(凡例)

○ : オプションあり

— : オプションなし

注※1

「-i」オプション指定の有無によって、-cmdopt の OVWRT の指定が異なります。

「-i」オプション	OVWRT の指定	OVWRT オプション有無による exsort コマンド実行結果
指定あり	指定しません	同一の出力ファイル名がある場合、exsort コマンドは「KBL1657-E 出力ファイルがすでに存在するため処理を中止しました。」のエラーを出力して失敗します。
指定なし	指定します	同一の出力ファイル名がある場合、上書きします。exsort コマンドは正常終了します。

注※2

「-B」オプションは、Linux 版 SORT で有効なオプションです。

注※3

条件を満たさないレコード選択は、次に示す「omit」オペランドに移行します。

```
-inpfile△omit="選択条件式"
```

選択条件式は、rselect コマンドの -sl" 選択条件式 " から転記します。

詳細については、「表 20-23 選択条件式の移行」を参照してください。

## (2) ファイル編成

rsort, rmerge, rselect, rsum および rsortsum コマンドの入力ファイル編成の移行内容を次の表に、出力ファイル編成の移行内容を表 20-9 に示します。

表 20-8 入力ファイル編成の移行

入力ファイル編成	意味	exsort コマンド <code>-inpfile</code> パラメタ
S	固定長順編成ファイル	fileorg=S
V	可変長順編成ファイル	fileorg=V
R	固定長相対編成ファイル	対応していません。
W	可変長相対編成ファイル	対応していません。
I	固定長索引編成ファイル	対応していません。※
X	可変長索引編成ファイル	対応していません。※
A	テキストファイル	fileorg=A

注※

SORT EE では索引編成ファイルに対応していませんが、次の手順で扱うことができます。

1. ISAM のユーティリティ `isconv` コマンド（ファイル変換）で順編成ファイルに変換します。
2. `exsort` コマンドの入力ファイルに、変換した順編成ファイルを指定します。

表 20-9 出力ファイル編成の移行

出力ファイル編成	意味	exsort コマンド <code>-outfile</code> パラメタ
S	固定長順編成ファイル	fileorg=S
V	可変長順編成ファイル	fileorg=V
R	固定長相対編成ファイル	対応していません。
W	可変長相対編成ファイル	対応していません。
A	テキストファイル	fileorg=A

### (3) キー属性 (`rsort`, `rmerge`, `rsum`, `rsortsum` コマンド共通)

`rsort`, `rmerge`, `rsum` および `rsortsum` コマンドのキー属性の移行内容を表 20-10 に示します。

#### ● `rsort/rmerge/rsum/rsortsum` コマンドのキー書式

```
-キー属性△+キー位置-キー長 [△-キー属性△+キー位置-キー長…]
```

#### ● `exsort` コマンドの `-key` パラメタ書式

```
-key△ [r] キー属性+キー位置-キー長 [△ [r] キー属性+キー位置-キー長…]
```

- キー属性にハイフン (-) は付けません。
- [r] キー属性 + キー位置 - キー長はつなげて指定します。
- 複数のキーを指定する場合、1 個以上の空白またはタブで区切って指定します。
- キー位置とキー長は、`rsort`, `rmerge`, `rsum` および `rsortsum` コマンドの指定内容



をそのまま転記します。

表 20-10 キー属性の移行

キー属性	意味	exsort コマンド -key パラメタ
-r	指定なし：昇順ソート 指定あり：降順ソート	r キー属性 (例：-rc → rc)
-j	文字 (JIS8 単位コード)	j
-c	文字	c
-n	符号付き固定小数点 2 進数	n
-m	符号なし 2 進数	m
-p	内部 10 進数	p
-z	外部 10 進数	z
-x	左符号付き外部 10 進数	x
-s	右分離符号付き外部 10 進数	s
-t	左分離符号付き外部 10 進数	t
-f	内部浮動小数点	f
-g	日付 (英数字)	g

#### (4) 集約項目属性 (rsum, rsortsum コマンド共通)

rsum および rsortsum コマンドの集約項目属性の移行内容を表 20-11 に示します。

##### ● rsum/rsortsum コマンドのキー書式

-a集約項目属性△+項目位置-項目長 [△集約項目属性△+項目位置-項目長…]

##### ● exsort コマンドの -sum パラメタ書式

-sum△集約項目属性+項目位置-項目長 [△集約項目属性+項目位置-項目長…]

- 集約項目属性 + 項目位置 - 項目長はつなげて指定します。
- 複数の項目を指定する場合、1 個以上の空白またはタブで区切って指定します。
- rsum および rsortsum コマンドの集約項目属性、項目位置および項目長の指定内容をそのまま転記します。

表 20-11 集約項目属性の移行

集約項目属性	意味	exsort コマンド -sum パラメタ
n	符号付き固定小数点 2 進数	n
m	符号なし 2 進数	m
p	内部 10 進数	p

## 20. SORT から SORT EE への移行ガイド

集約項目属性	意味	exsort コマンド -sum パラメタ
z	外部 10 進数	z
x	左符号付き外部 10 進数	x
s	右分離符号付き外部 10 進数	s
t	左分離符号付き外部 10 進数	t

### 20.2.7 SORT 環境変数の移行

SORT の環境変数の移行内容を次の表に示します。

表 20-12 SORT 環境変数の移行

SORT		SORT EE	
環境変数名	値	環境変数名 (括弧内は 64 ビット版)	値
SORTOPT	-y ワークバッファ サイズ	EXSORTWORKSIZE (EXSORT64WORKSIZE)	ワークバッファサ イズを転記※ <sup>1</sup>
	-T ディレクトリ 名	EXSORTSORTTMPDIR (EXSORT64SORTTMPDI R)	ディレクトリ名を 転記
	-T2 ディレクトリ 名 2	EXSORTMERGETMPDIR (EXSORT64MERGETMP DIR)	ディレクトリ名 2 を転記
	-g 日付項目の基準 年度	EXSORTDATEOPT (EXSORT64DATEOPT)	日付項目の基準年 度を転記
	-E ※ <sup>2</sup>	—	—
	-SORT_EE ※ <sup>3</sup>	—	—
SORT_BOMLNGIN	UNICODE	EXSORTBOMLNGIN (EXSORT64BOMLNGIN)	UNICODE
SORT_BOMLNGOUT	UTF8	EXSORTBOMLNGOUT (EXSORT64BOMLNGOU T)	UTF8
SORT_STRINGDIR	ディレクトリ名	EXSORTSTRINGDIR (EXSORT64STRINGDIR)	ディレクトリ名を 転記
SORT_DMPDIR	ディレクトリ名	—	—

(凡例)

— : 対応する環境変数なし

注※ 1

KB 単位指定の場合は、MB 単位に変換した値を指定してください。

なお、ワークバッファサイズの自動見積もり機能を使用する場合は、環境変数

「EXSORTWORKSIZE (64 ビット版は「EXSORT64WORKSIZE)」の指定は不要です。

注※ 2

指定値「**E**」は、選択条件値の特殊記号「**¥**」の有効化を意味します。

**SORT EE** では、選択条件式の書き方で「**¥**」の有効化が図れます。詳細については、「4.3.6 選択条件式の書き方」を参照してください。

注※ 3

**SORT** コマンドで **SORT EE** が提供する機能を利用する場合に指定します。移行対象の環境変数ではありません。

## 20.3 Windows 版 SORT コマンドの移行

### 20.3.1 wnsort コマンド

wnsort コマンドの移行内容を次の表に示します。

- 項番 2～4 は、wnsort コマンドと exsort コマンドで指定値の形式が異なります。各オペランドの詳細については、次の表に示す該当する章を参照してください。
- 項番 5～10 は、wnsort コマンドの指定値を、exsort コマンドのパラメタまたはオペランド値としてそのまま転記します。

表 20-13 wnsort コマンド移行

項番	wnsort コマンドのオペランド	exsort コマンドのパラメタおよびオペランド	移行内容
1	—	-function △ sort	転記しません。
2	/ソートオプション	-cmdopt △コマンドオプション	20.3.6(1)を参照してください。
3	/g 入力ファイルのファイル編成： 出力ファイルのファイル編成	-infile △ fileorg= 入力ファイル編成 -outfile △ fileorg= 出力ファイル編成	20.3.6(2)を参照してください。
4	/キー属性△+キー位置・キー長	-key △キー属性+キー位置・キー長	20.3.6(3)を参照してください。
5	/l レコード長	-infile △ reclen= レコード長	指定値を転記します。
6	/o 出力ファイル名	-outfile △ file= 出力ファイル名	指定値を転記します。
7	/y ワークバッファサイズ	-worksize △ワークバッファサイズ※1	指定値を転記します。
8	/w ディレクトリ名	-sorttmpdir △ディレクトリ名※2	指定値を転記します。
9	/w2 ディレクトリ名 2	-mergetmpdir △ディレクトリ名 2	指定値を転記します。
10	入力ファイル名	-infile △ file= 入力ファイル名	指定値を転記します。

(凡例)

— : 該当しません。

△ : 1 文字以上の空白を示します。

注※ 1

KB 単位指定の場合は、MB 単位に変換した値を指定してください。

## 注※2

`exsort` コマンドのソート機能を使用する場合、一時ファイル作成ディレクトリ名を指定する「`-sorttmpdir` パラメタ」または環境変数「`EXSORTSORTTMPDIR`」の指定は必須です。どちらも指定しなかった場合、`SORT` の環境ファイルの「`temp_dir1`」の指定を仮定します。すべての指定がなかった場合、`exsort` コマンドは「`KBLS1674-E` 一時ファイルのディレクトリが指定されていません。」のエラーを出力して失敗します。`SORT EE` の環境変数の設定については、「20.3.7 `SORT` 環境ファイルの移行」を参照してください。

## 20.3.2 `wnmerge` コマンド

`wnmerge` コマンドの移行内容を次の表に示します。

- 項番 2～4 は、`wnmerge` コマンドと `exsort` コマンドで指定形式が異なります。各オペランドの詳細については、次の表に示す該当する章を参照してください。
- 項番 5～7 は、`wnmerge` コマンドの指定値を、`exsort` コマンドのパラメタまたはオペランド値としてそのまま転記します。

表 20-14 `wnmerge` コマンド移行

項番	<code>wnmerge</code> コマンドのオペランド	<code>exsort</code> コマンドのパラメタおよびオペランド	移行内容
1	—	<code>-function Δ merge</code>	転記しません。
2	<code>/</code> マージオプション	<code>-cmdopt Δ オプション</code>	20.3.6(1) を参照してください。
3	<code>/g</code> 入力ファイルのファイル編成： 出力ファイルのファイル編成	<code>-infile Δ fileorg=</code> 入力ファイル編成 <code>-outfile Δ fileorg=</code> 出力ファイル編成	20.3.6(2) を参照してください。
4	<code>/</code> キー属性 <code>Δ</code> + キー位置 - キー長	<code>-key Δ</code> キー属性 + キー位置 - キー長	20.3.6(3) を参照してください。
5	<code>/l</code> レコード長	<code>-infile Δ reclen=</code> レコード長	指定値を転記します。
6	<code>/o</code> 出力ファイル名	<code>-outfile Δ file=</code> 出力ファイル名	指定値を転記します。
7	入力ファイル名	<code>-infile Δ file=</code> 入力ファイル名	指定値を転記します。

(凡例)

— : 該当しません。

Δ : 1文字以上の空白を示します。

## 20.3.3 `wnselect` コマンド

`wnselect` コマンドの移行内容を次の表に示します。

## 20. SORT から SORT EE への移行ガイド

- 項番 2～4 は、`wselect` コマンドと `exsort` コマンドで指定形式が異なります。各オペランドの詳細については、次の表に示す該当する章を参照してください。
- 項番 5～7 は、`wselect` コマンドの指定値を、`exsort` コマンドのパラメタまたはオペランド値としてそのまま転記します。

表 20-15 `wselect` コマンドの移行

項番	<code>wselect</code> コマンドのオペランド	<code>exsort</code> コマンドのパラメタおよびオペランド	移行内容
1	—	<code>-function Δ copy</code>	転記しません。
2	<code>/</code> 選択オプション	<code>-cmdopt Δ</code> オプション	20.3.6(1) を参照してください。
3	<code>/g</code> 入力ファイルのファイル編成： 出力ファイルのファイル編成	<code>-inpfile Δ fileorg=</code> 入力ファイル編成 <code>-outfile Δ fileorg=</code> 出力ファイル編成	20.3.6(2) を参照してください。
4	<code>/s!</code> 選択条件式 "	<code>-inpfile Δ include="</code> 選択条件式 "	20.3.6(5) を参照してください。
5	<code>/l</code> レコード長	<code>-inpfile Δ reclen=</code> レコード長	指定値を転記します。
6	<code>/o</code> 出力ファイル名	<code>-outfile Δ file=</code> 出力ファイル名	指定値を転記します。
7	入力ファイル名	<code>-inpfile Δ file=</code> 入力ファイル名	指定値を転記します。

(凡例)

- : 該当しません。
- Δ : 1 文字以上の空白を示します。

### 20.3.4 `wnsum` コマンド

`wnsum` コマンドの移行内容を次の表に示します。

- 項番 2～5 は、`wnsum` コマンドと `exsort` コマンドで指定形式が異なります。各オペランドの詳細については、次の表に示す該当する章を参照してください。
- 項番 6～8 は、`wnsum` コマンドの指定値を、`exsort` コマンドのパラメタまたはオペランド値としてそのまま転記します。

表 20-16 `wnsum` コマンドの移行

項番	<code>wnsum</code> コマンドのオペランド	<code>exsort</code> コマンドのパラメタおよびオペランド	移行内容
1	—	<code>-function Δ copy</code>	転記しません。
2	<code>/</code> 集約オプション	<code>-cmdopt Δ</code> オプション	20.3.6(1) を参照してください。

項番	wncsum コマンドのオペランド	exsort コマンドのパラメタおよびオペランド	移行内容
3	/g 入力ファイルのファイル編成： 出力ファイルのファイル編成	-infile △ fileorg= 入力ファイル編成 -outfile △ fileorg= 出力ファイル編成	20.3.6(2) を参照してください。
4	/キー属性△ + キー位置 - キー長	-key △ キー属性 + キー位置 - キー長	20.3.6(3) を参照してください。
5	/a" 集約項目属性△ + 項目位置 - 項目長 "	-sum △ 集約項目属性 + 項目位置 - 項目長	20.3.6(4) を参照してください。
6	/l レコード長	-infile △ reclen= レコード長	指定値を転記します。
7	/o 出力ファイル名	-outfile △ file= 出力ファイル名	指定値を転記します。
8	入力ファイル名	-infile △ file= 入力ファイル名	指定値を転記します。

(凡例)

— : 該当しません。

△ : 1 文字以上の空白を示します。

### 20.3.5 拡張 SORT コマンド (wncsort コマンド)

wncsort コマンドの移行内容を次の表に示します。

- 項番 1 ~ 7 は、wncsort コマンドと exsort コマンドで指定形式が異なります。各オペランドの詳細については、次の表に示す該当する章を参照してください。
- 項番 8 ~ 13 は、wncsort コマンドの指定値を、exsort コマンドのパラメタまたはオペランド値としてそのまま転記します。

表 20-17 wncsort コマンドの移行

項番	wncsort コマンド (パラメタファイル指定値)	exsort コマンドのパラメタおよびオペランド	移行内容
1	exec_opt= {select[:sort][:sum]   sort[:sum]   sum}	sort 指定がある場合： -function △ sort  sort 指定がない場合： -function △ copy	転記しません。
2	sort_opt= 機能オプション	-cmdopt △ オプション	20.3.6(1) を参照してください。
3	file_org= 入力ファイルの ファイル編成： 出力ファイルのファイル編成	-infile △ fileorg= 入力ファイル編成 -outfile △ fileorg= 出力ファイル編成	20.3.6(2) を参照してください。

## 20. SORT から SORT EE への移行ガイド

項番	wnortsum コマンド (パラメータファイル指定値)	exsort コマンドのパラメータおよびオペランド	移行内容
4	select_exp= 選択条件式	-inpfile △ include=" 選択条件式 "	20.3.6(5)を参照してください。
5	sort_key= キー属性△ + キー位置 - キー長	-key △ キー属性 + キー位置 - キー長	20.3.6(3)を参照してください。
6	sum_key= キー属性△ + キー位置 - キー長	-key △ キー属性 + キー位置 - キー長	20.3.6(3)を参照してください。
7	sum_field= 集約項目属性△ + 項目位置 - 項目長	-sum △ 集約項目属性 + 項目位置 - 項目長	20.3.6(4)を参照してください。
8	record_leng= レコード長	-inpfile △ reclen= レコード長	指定値を転記します。
9	output_file= 出力ファイル名	-outfile △ file= 出力ファイル名	指定値を転記します。
10	input_file= 入力ファイル名	-inpfile △ file= 入力ファイル名	指定値を転記します。
11	work_size= ワークバッファサイズ	-worksize △ ワークバッファサイズ※1	指定値を転記します。
12	temp_dir1= ソート用一時ファイルディレクトリ名	-sorttmpdir △ ソート用一時ファイルディレクトリ名※2	指定値を転記します。
13	temp_dir3= マージ用一時ファイルディレクトリ名	-mergetmpdir △ マージ用一時ファイルディレクトリ名	指定値を転記します。
14	temp_dir2= 中間ファイルディレクトリ名	—	転記しません。

(凡例)

— : 該当しません。

△ : 1文字以上の空白を示します。

注※1

KB単位指定の場合は、MB単位に変換した値を指定してください。

注※2

exsort コマンドのソート機能を使用する場合、一時ファイル作成ディレクトリ名を指定する「-sorttmpdir パラメータ」または環境変数「EXSORTSORTTMPDIR」の指定は必須です。どちらも指定しなかった場合、SORT の環境ファイルの「temp\_dir1」の指定を仮定します。すべての指定がなかった場合、exsort コマンドは「KBLS1674-E 一時ファイルのディレクトリが指定されていません。」のエラーを出力して失敗します。

SORT EE の環境変数の設定については、「20.3.7 SORT 環境ファイルの移行」を参照してください。



## 20.3.6 コマンド共通項目

### (1) コマンドオプション

wnsort, wnmerge, wnselect, wnsum および wnsortex コマンドのコマンドオプションの移行内容を表 20-18 に示します。

● wnsort/wnmerge/wnselect/wnsum コマンドのオプション

```
{/オプション[△/オプション…] | /オプション[オプション…]}
```

オプションを複数指定する場合、次の 2 とおりの方法で指定できます。

- 個々のオプションを空白で分けて指定する場合、オプションの前にスラント (/) を付けて指定します。
- つなげて指定する場合、2 個目以降のオプションにはスラント (/) を付けません。

● wnsortex コマンドのオプション (パラメタファイル指定)

```
sort_opt=オプション[オプション…]
```

オプションを複数指定する場合、オプションをつなげて指定します。

● exsort コマンドのコマンドオプション

```
-cmdopt△オプション[:オプション…]
```

オプションを複数指定する場合、コロン (:) でオプションを区切ります。

表 20-18 コマンドオプションの移行

オプション ※1	意味	wnsort	wnmerge	wnselect	wnsum	wnsortex	exsort-cmdopt
/d	重複レコードの入力順序保存	○	—	—	—	○	DUP
/i	既存ファイルの書き換え防止	○	○	○	○	○	_※2
/v	EBCDIK コード比較	○	○	○	○	○	EBCJ
/v1	EBCDIK コード比較 (入力 JIS8)	—	—	○	—	—	EBCJ
/v2	EBCDIK コード比較 (入力 EBCDIK)	—	—	○	—	—	EBCE
/b	big endian 比較	○	○	○	○	○	BIG
/k	強制マージ	—	○	—	—	—	FMRG
/u	条件を満たさないレコード選択	—	—	○	—	○	_※3

## 20. SORT から SORT EE への移行ガイド

オプション ※1	意味	wnsort	wnmerge	wnselect	wnsum	wnsort ex	exsort -cmdopt
/h	10進数オーバーフロー判定	—	—	—	○	○	OVFLW
/if	内部10進数符号保障オプション	—	—	—	○	—	ABS

(凡例)

○ : オプションあり

— : オプションなし

注※1

wnsortex コマンドの場合、オプションにスラント (/) は付きません。

注※2

「/i」オプション指定の有無によって、-cmdopt の OVWRT の指定が異なります。

「/i」オプション	OVWRT の指定	OVWRT オプション有無による exsort コマンド 実行結果
指定あり	指定しません	同一の出力ファイル名がある場合、exsort コマンドは「KBLS1657-E 出力ファイルがすでに存在するため処理を中止しました。」のエラーを出力して失敗します。
指定なし	指定します	同一の出力ファイル名がある場合、上書きします。exsort コマンドは正常終了します。

注※3

条件を満たさないレコード選択は、次に示す「omit」オペランドで指定します。

```
-infile△omit="選択条件式"
```

選択条件式は、wnselect コマンドの /sl" 選択条件式" から転記します。

詳細については、「表 20-23 選択条件式の移行」を参照してください。

### (2) ファイル編成

wnsort, wnmerge, wnselect, wnsum, および wnsortex コマンドの入力ファイル編成の移行内容を次の表に、出力ファイル編成の移行内容を表 20-20 に示します。

表 20-19 入力ファイル編成の移行

入力ファイル編成	意味	exsort コマンド -infile パラメタ
s	固定長順編成ファイル	fileorg=S
v	可変長順編成ファイル	fileorg=V
r	固定長相対編成ファイル	対応していません。
w	可変長相対編成ファイル	対応していません。

入力ファイル編成	意味	exsort コマンド -infile パラメタ
i	固定長索引編成ファイル	対応していません。※
x	可変長索引編成ファイル	対応していません。※
a	テキストファイル	fileorg=A

注※

SORT EE では索引編成ファイルに対応していませんが、次の手順で扱うことができます。

1. ISAM のユーティリティ isconv コマンド (ファイル変換) で順編成ファイルに変換します。
2. exsort コマンドの入力ファイルに、変換した順編成ファイルを指定します。

表 20-20 出力ファイル編成の移行

出力ファイル編成	意味	exsort コマンド -outfile パラメタ
s	固定長順編成ファイル	fileorg=S
v	可変長順編成ファイル	fileorg=V
r	固定長相対編成ファイル	対応していません。
w	可変長相対編成ファイル	対応していません。
a	テキストファイル	fileorg=A

### (3) キー属性 (wnsort, wnmerge, wnsun, wnsortex コマンド共通)

wnsort, wnmerge, wnsun および wnsortex コマンドのキー属性の移行内容を表 20-21 に示します。

#### ● wnsort/wnmerge/wnsun コマンドのキー書式

```
/キー属性△+キー位置-キー長 [△/キー属性△+キー位置-キー長…]
```

#### ● wnsortex コマンドのソートキー書式 (パラメタファイル指定)

```
sort_key=キー属性+キー位置-キー長 [, キー属性+キー位置-キー長…]
```

#### ● wnsortex コマンドの集約キー書式 (パラメタファイル指定)

```
sum_key=キー属性+キー位置-キー長 [, キー属性+キー位置-キー長…]
```

#### ● exsort コマンドの -key パラメタ書式

```
-key△ [r] キー属性+キー位置-キー長 [△ [r] キー属性+キー位置-キー長…]
```

- キー属性にスラント (/) は付けません。
- [r] キー属性 + キー位置 - キー長はつなげて指定します。
- 複数のキーを指定する場合、1 個以上の空白またはタブで区切って指定します。

## 20. SORT から SORT EE への移行ガイド

- キー位置とキー長は、`wnsort`、`wnmerge`、`wnsum` および `wnsortex` コマンドの指定内容をそのまま転記します。

表 20-21 キー属性の移行

キー属性※	意味	exsort コマンド <code>-key</code> パラメタ
<code>/r</code>	指定なし：昇順ソート 指定あり：降順ソート	<code>r</code> キー属性 (例： <code>/rc</code> → <code>rc</code> )
<code>/j</code>	文字 (JIS8 単位コード)	<code>j</code>
<code>/c</code>	文字	<code>c</code>
<code>/n</code>	符号付き固定小数点 2 進数	<code>n</code>
<code>/m</code>	符号なし 2 進数	<code>m</code>
<code>/p</code>	内部 10 進数	<code>p</code>
<code>/z</code>	外部 10 進数	<code>z</code>
<code>/x</code>	左符号付き外部 10 進数	<code>x</code>
<code>/s</code>	右分離符号付き外部 10 進数	<code>s</code>
<code>/t</code>	左分離符号付き外部 10 進数	<code>t</code>
<code>/f</code>	内部浮動小数点	<code>f</code>
<code>/e</code>	日付 (英数字)	<code>g</code>

注※

`wnsortex` コマンドの場合、キー属性にスラント (`/`) は付きません。

### (4) 集約項目属性 (`wnsum`、`wnsortex` コマンド共通)

`wnsum` および `wnsortex` コマンドの集約項目属性の移行内容を表 20-22 に示します。

#### ● `wnsum` コマンドのキー書式

```
/a"集約項目属性△+項目位置-項目長[△集約項目属性△+項目位置-項目長…]"
```

#### ● `wnsortex` コマンドのキー書式 (パラメタファイル指定)

```
sum_field=集約項目属性+項目位置-項目長[,集約項目属性+項目位置-項目長…]
```

#### ● `exsort` コマンドの `-sum` パラメタ書式

```
-sum△集約項目属性+項目位置-項目長[△集約項目属性+項目位置-項目長…]
```

- 集約項目属性 + 項目位置 - 項目長はつなげて指定します。
- 複数の項目を指定する場合、1 個以上の空白またはタブで区切って指定します。
- `wnsum` および `wnsortex` コマンドの集約項目属性、項目位置および項目長の指定内容をそのまま転記します。

表 20-22 集約項目属性の移行

集約項目属性	意味	exsort コマンド -sum パラメタ
n	符号付き固定小数点 2 進数	n
m	符号なし 2 進数	m
p	内部 10 進数	p
z	外部 10 進数	z
x	左符号付き外部 10 進数	x
s	右分離符号付き外部 10 進数	s
t	左分離符号付き外部 10 進数	t

## (5) 選択条件式の移行内容

wnselect および wnsortex コマンドの選択条件式の移行内容を表 20-23 に示します。

## ● wnselect コマンドの選択条件式

```
/s1"-比較項目属性△+項目位置-項目長△比較演算子△条件値  
[△論理演算子△-比較項目属性△+項目位置-項目長△比較演算子△条件値…]"
```

## ● wnsortex コマンドの選択条件式 (パラメタファイル指定)

```
select_exp=比較項目属性+項目位置-項目長△比較演算子△条件値  
[△論理演算子△比較項目属性+項目位置-項目長△比較演算子△条件値…]
```

## ● exsort コマンドの選択条件式

```
-infile△ {include | omit} ="比較項目属性+項目位置-項目長△比較演算子△条件値  
[△論理演算子△比較項目属性+項目位置-項目長△比較演算子△条件値…]"
```

- ・ 項番 2 の比較項目属性のうち、日付だけ形式が異なります。
- ・ 項番 3～7 は、wnselect および wnsortex コマンドの指定値を、exsort コマンドのオペランド値としてそのまま転記します。

表 20-23 選択条件式の移行

項番	wnselect, wnsortex コマンド	意味	移行内容
1	コマンド オプション	/u ※	指定なし：条件を満たす入力レコードを選択
			指定あり：条件を満たさない入力レコードを選択
			include= omit=

20. SORT から SORT EE への移行ガイド

項番	wnselect,wnsortex コマンド	意味	移行内容	
2	比較項目 属性	j	文字 (JIS8 単位コード)	j
		c	文字	c
		n	符号付き固定小数点 2 進数	n
		m	符号なし 2 進数	m
		p	内部 10 進数	p
		z	外部 10 進数	z
		x	左符号付き外部 10 進数	x
		s	右分離符号付き外部 10 進数	s
		t	左分離符号付き外部 10 進数	t
		f	内部浮動小数点	f
e	日付	g		
3	+ 項目位置	比較項目の開始バイト位置	+ 項目位置	
4	- 項目長	比較項目のバイト長	- 項目長	
5	比較演算 子	=	条件値と等しい値の入力レコードを選択	=
		!=	条件値と異なる値の入力レコードを選択	!=
		>	条件値より大きい値の入力レコードを選択	>
		>=	条件値より大きいかまたは等しい値の入力レコードを選択	>=
		<	条件値より小さい値の入力レコードを選択	<
		<=	条件値より小さいかまたは等しい値の入力レコードを選択	<=
6	条件値	c' 文字 条件値 ,	文字属性の比較項目に対する条件値	C' 文字条件値 '
		n' 数値 条件値 ,	数値属性の比較項目に対する条件値	N' 数値条件値 '
		d' 日付 条件値 ,	日付属性の比較項目に対する条件値	D' 日付条件値 '
		x'16 進 条件値 ,	文字または数値属性の比較項目に対する条件値	X'16 進条件値 '
7	論理演算 子	AND	複数の条件式を論理積で結合する	AND
		OR	複数の条件式を論理和で結合する	OR

注※

esort コマンドには, 「-u」 オプション (wnsortex コマンドの場合は sort\_opt=u) に相当する

コマンドオプションはありません。

条件を満たさないレコード選択を指定する場合は、「omit オペランド」で指定します。

## 20.3.7 SORT 環境ファイルの移行

SORT の環境ファイル (wnsort.wse ファイル) の移行内容を次の表に示します。

表 20-24 SORT 環境ファイルの移行

SORT 環境ファイルの項目	SORT EE	
	環境変数名 (括弧内は 64 ビット版)	値
date_opt= 日付項目の基準年度	EXSORTDATEOPT (EXSORT64DATEOPT)	日付項目の基準年度を転記
msg_opt= メッセージ出力オプション	EXSORTMSGOPT (EXSORT64MSGOPT)	メッセージ出力オプションを転記
msg_file= メッセージの出力先ファイル名	EXSORTMSGFILE (EXSORT64MSGFILE)	メッセージの出力先ファイル名を転記
temp_dir1= ソート用一時ファイルディレクトリ名	EXSORTSORTTMPDIR (EXSORT64SORTTMPDIR)	ソート用一時ファイルディレクトリ名を転記
temp_dir2= 中間ファイルディレクトリ	—	—
temp_dir3= マージ用一時ファイルディレクトリ名	EXSORTMERGETMPDIR (EXSORT64MERGETMPDIR)	マージ用一時ファイルディレクトリ名を転記
work_size= ワークバッファサイズ	EXSORTWORKSIZE (EXSORT64WORKSIZE)	ワークバッファサイズを転記※ <sup>1</sup>
string_dir= スtring情報格納一時ファイルディレクトリ名	EXSORTSTRINGDIR (EXSORT64STRINGDIR)	String情報格納一時ファイルディレクトリ名を転記
bom_lngin=unicode	EXSORTBOMLINGIN (EXSORT64BOMLINGIN)	UNICODE
bom_lngout=utf8	EXSORTBOMLINGOUT (EXSORT64BOMLINGOUT)	UTF8
dmp_dir= デバッグ情報ファイルの出力先ディレクトリ	—	—
SORT_EE ※ <sup>2</sup>	—	—

## 20. SORT から SORT EE への移行ガイド

(凡例) - : 該当しません。

### 注※ 1

KB 単位指定の場合は、MB 単位に変換した値を指定してください。

なお、ワークバッファサイズの自動見積もり機能を使用する場合は、環境変数「EXSORTWORKSIZE (64 ビット版は「EXSORT64WORKSIZE))」の指定は不要です。

### 注※ 2

SORT コマンドで SORT EE が提供する機能を利用する場合に指定します。移行対象の環境変数ではありません。



## 20.4 ワークバッファサイズ

---

SORT コマンドのワークバッファサイズの移行内容を示します。

- SORT からの移行ポイント

SORT EE は、ソート機能で使用するワークバッファを、入力データから適切なサイズを自動的に確保し、ソート処理を行います。これをワークバッファサイズ自動見積もり機能といいます。

SORT EE は、このワークバッファサイズ自動見積もり機能がデフォルトとなります。

- SORT EE の自動算出ワークバッファサイズの上限值の設定

ワークバッファサイズ自動見積もり機能で算出したサイズに上限値を設定できます。

**【例】** `-worksize auto,100M` (100MB を上限値としてワークバッファを確保する)

- インコアソートによるソート処理の高速化

ワークバッファサイズ自動見積もり機能では、一時ファイルを作成するソート処理 (アウトコアソート) を行います。

物理メモリが豊富な場合、一時ファイルを作成しないで、メモリだけでソート処理を行うインコアソートを指定することもできます。

**【例】** `-worksize incore,2000M` (2000MB を上限値としてワークバッファを確保する)



# 付録

---

付録 A	JIS8 単位コード体系と EBCDIK コード体系の変換表
付録 B	CSV ファイルのデータの扱い (SORT EE の機能)
付録 C	CSV ファイル形式と CSV データ形式 (SORT EE の機能)
付録 D	CSV ファイル内のナル文字列の扱い (SORT EE の機能)
付録 E	左分離符号付き数字のデータ形式 (SORT EE の機能)
付録 F	レコード編集処理で基になるレコード (SORT EE の機能)
付録 G	altseq 変換ファイル (SORT EE の機能)
付録 H	Unicode 操作文字コード一覧 (SORT EE)
付録 I	マルチバイト文字データの扱い
付録 J	日付時刻データの形式 (SORT EE の機能)
付録 K	メッセージ
付録 L	各バージョンの変更内容
付録 M	このマニュアルの参考情報
付録 N	用語解説

---

# 付録 A JIS8 単位コード体系と EBCDIK コード体系 の変換表

図 A-1 JIS8 単位コード体系と EBCDIK コード体系の変換表

		上位4ビット															
		0x		1x		2x		3x		4x		5x		6x		7x	
下 位 4 ビ ツ ト	x0	00 NUL	00 NUL	10 DLE	10 DLE	20 SP	40 SP	30 0	f0 0	40 @	7c @	50 P	d7 P	60 ,	79 ,	70 p	76 p
	x1	01 SOH	01 SOH	11 DC1	11 DC1	21 !	4f !	31 1	f1 1	41 A	c1 A	51 Q	d8 Q	61 a	59 a	71 q	77 q
	x2	02 STX	02 STX	12 DC2	12 DC2	22 "	7f "	32 2	f2 2	42 B	c2 B	52 R	d9 R	62 b	62 b	72 r	78 r
	x3	03 ETX	03 ETX	13 DC3	13 DC3	23 #	7b #	33 3	f3 3	43 C	c3 C	53 S	e2 S	63 c	63 c	73 s	80 s
	x4	04 EOT	37 EOT	14 DC4	3c DC4	24 \$	e0 \$	34 4	f4 4	44 D	c4 D	54 T	e3 T	64 d	64 d	74 t	8b t
	x5	05 ENQ	2d ENQ	15 NAK	3d NAK	25 %	6c %	35 5	f5 5	45 E	c5 E	55 U	e4 U	65 e	65 e	75 u	9b u
	x6	06 ACK	2e ACK	16 SYN	32 SYN	26 &	50 &	36 6	f6 6	46 F	c6 F	56 V	e5 V	66 f	66 f	76 v	9c v
	x7	07 BEL	2f BEL	17 ETB	26 ETB	27 '	7d '	37 7	f7 7	47 G	c7 G	57 W	e6 W	67 g	67 g	77 w	a0 w
	x8	08 BS	16 BS	18 CAN	18 CAN	28 (	4d (	38 8	f8 8	48 H	c8 H	58 X	e7 X	68 h	68 h	78 x	ab x
	x9	09 HT	05 HT	19 EM	19 EM	29 )	5d )	39 9	f9 9	49 I	c9 I	59 Y	e8 Y	69 l	69 l	79 y	b0 y
	xA	0a LF	15 NL	1a SUB	3f SUB	2a *	5c *	3a :	7a :	4a J	d1 J	5a Z	e9 Z	6a j	70 j	7a z	b1 z
	xB	0b VT	0b VT	1b ESC	27 ESC	2b +	4e +	3b ;	5e ;	4b K	d2 K	5b [	4a [	6b k	71 k	7b {	c0 {
	xC	0c FF	0c FF	1c FS	1c FS	2c ,	6b ,	3c <	4c <	4c L	d3 L	5c ¥	5b ¥	6c ll	72 ll	7c 	6a 
	xD	0d CR	0d CR	1d GS	1d GS	2d -	60 -	3d =	7e =	4d M	d4 M	5d ]	5a ]	6d m	73 m	7d }	d0 }
	xE	0e SO	0e SO	1e RS	1e RS	2e .	4b .	3e >	6e >	4e N	d5 N	5e ^	5f ^	6e n	74 n	7e _	a1 _
	xF	0f SI	0f SI	1f US	1f US	2f /	61 /	3f ?	6f ?	4f 0	d6 0	5f -	6d -	6f o	75 o	7f DEL	07 DEL

		上位4ビット															
		8x		9x		Ax		Bx		Cx		Dx		Ex		Fx	
下 位 4 ビ ット	x0	80	00 NUL	90	ff	a0	57	b0	58	c0	91	d0	a5	e0	ff	f0	ff
	x1	81	ff	91	ff	a1	41 。	b1	81 ア	c1	92 チ	d1	a6 ム	e1	ff	f1	ff
	x2	82	ff	92	ff	a2	42 Γ	b2	82 イ	c2	93 ツ	d2	a7 メ	e2	ff	f2	ff
	x3	83	ff	93	ff	a3	43 J	b3	83 ウ	c3	94 テ	d3	a8 モ	e3	ff	f3	ff
	x4	84	ff	94	ff	a4	44 、	b4	84 エ	c4	95 ト	d4	a9 ヤ	e4	ff	f4	ff
	x5	85	ff	95	ff	a5	45 ・	b5	85 オ	c5	96 ナ	d5	aa ユ	e5	ff	f5	ff
	x6	86	ff	96	ff	a6	46 ヲ	b6	86 カ	c6	97 ニ	d6	ac ヨ	e6	ff	f6	ff
	x7	87	ff	97	ff	a7	47 ア	b7	87 キ	c7	98 ヌ	d7	ad ラ	e7	ff	f7	ff
	x8	88	ff	98	ff	a8	48 イ	b8	88 ク	c8	99 ネ	d8	ae リ	e8	ff	f8	ff
	x9	89	ff	99	ff	a9	49 ウ	b9	89 ケ	c9	9a ノ	d9	af ル	e9	ff	f9	ff
	xA	8a	ff	9a	ff	aa	51 エ	ba	8a コ	ca	9d ハ	da	ba レ	ea	ff	fa	ff
	xB	8b	ff	9b	ff	ab	52 オ	bb	8c サ	cb	9e ヒ	db	bb ロ	eb	ff	fb	ff
	xC	8c	ff	9c	ff	ac	53 ヤ	bc	8d シ	cc	9f フ	dc	bc ワ	ec	ff	fc	ff
	xD	8d	ff	9d	ff	ad	54 ユ	bd	8e ス	cd	a2 ヘ	dd	bd ン	ed	ff	fd	ff
	xE	8e	ff	9e	ff	ae	55 ヨ	be	8f セ	ce	a3 ホ	de	be 。	ee	ff	fe	ff
	xF	8f	ff	9f	ff	af	56 ッ	bf	90 ソ	cf	a4 マ	df	bf 。	ef	ff	ff	ff

(凡例)

- |     |     |
|-----|-----|
| (1) | (3) |
| (2) | (4) |
- (1) : JIS8単位コード体系の文字コードを示す。  
 (2) : (1) のコードに対応する文字を示す (空白は、対応する文字がない)。  
 (3) : (1) に対するEBCDIKコード体系の文字コードを示す。  
 (4) : (3) のコードに対応する文字を示す (空白は、対応する文字がない)。

注

■ は、次のとおりにコード順序が変換される。

- ・ JISコード「0x0A」は、「0x15」のコード順に変換される。
- ・ JISコード「0x80」は、「0x00」のコード順に変換される。
- ・ JISコード「0x81~0x9F」は、「0xFF」のコード順に変換される。
- ・ JISコード「0xA0」は、「0x57」のコード順に変換される。
- ・ JISコード「0xE0~0xFF」は、「0xFF」のコード順に変換される。

## 付録 B CSV ファイルのデータの扱い (SORT EE の機能)

### 付録 B.1 数値 (固定小数点数) データの端数処理

ここでは、入力データが数値 (固定小数点数) の場合の端数処理について説明します。

入力データの数値が小数の場合、小数点位置を合わせて入力します。入力データ中に小数点がない場合、データの末尾に小数点を仮定して入力します。入力データの桁に過不足がある場合は、桁調整を行った内部データ形式に変換されたあと、比較されます。

入力データの数値の扱いを次に示します。

#### 整数部の桁調整

- 有効桁数に満たないデータを入力した場合、左側に 0 (ゼロ) を加えて、有効桁に調整されます。
- 左側が空白 (ゼロ抑止) のデータを入力した場合、空白を 0 (ゼロ) と見なして有効桁に調整されます。
- 有効桁数を超えるデータを入力した場合、左側の桁あふれ部分が無視されます。
- 丸め方法は、環境ファイル (EXS\_ROUND)、または round オプションで指定された方法が適用されます。詳細は「5.8.11 round オペランド」を参照してください。

#### 小数部の桁調整

- 有効桁数に満たないデータを入力した場合、右側に 0 (ゼロ) を加えて、有効桁数に調整されます。
- 有効桁数を超えるデータを入力した場合、右側の桁あふれ部分を丸めて、有効桁数に調整されます。ただし、CSV 数値データ (固定小数点数) の形式チェックオプション (CSVFRMCHK オプション) を指定した場合は、数字以外を丸めに使いません。有効桁の右隣りから数字以外の文字が出てくるまでが「あふれ部分」と見なされます。

CSVFRMCHK オプション指定時の例を次の表に示します。

入力データ	小数部あふれ部分 (p = 4, s = 1)	解説
999	なし	小数点以下なし
999.	なし	小数点以下なし
999.0	なし	小数点以下 2 桁目以降なし
999.12	2 (1 桁)	小数点以下 2 桁目以降があふれ部分
999.123	23 (2 桁)	小数点以下 2 桁目以降があふれ部分
999.12A	2 (1 桁)	英字の前までがあふれ部分
999.1A3	なし	英字の前まで (数字なし)

- 丸め方法は、環境ファイル (EXS\_ROUND)、または round オプションで指定された方法が適用されます。詳細は「5.8.11 round オペランド」を参照してください。

#### ゼロ値の扱い

- 入力データがすべて空白文字である場合は、0 (ゼロ) として扱われます。
- 符号付きゼロ (+0, -0) と、符号なしゼロ (0) は同じ値として扱われます。

#### 小数点位置の合わせ方

- データ中に小数点がある場合は、小数点位置を合わせて入力されます。
- データ中に小数点がない場合は、データの末尾に小数点を仮定して入力されます。

入力データ	小数点位置	説明
小数点あり .123 1.23 12.3 123.	.123 1.23 12.3 123.3	小数点位置を合わせて入力します。
小数点なし 1 12 123 1234	1. 12. 123. 1234.	データ末尾に小数点を仮定して入力します。

入力データが数値の場合の、指定できる有効桁数の指定例を次の表に示します。なお、表中の p および s は、-csvkey のパラメタに対応しています。

入力データ	有効桁数の指定			内部データ形式	説明
	総桁数 (p)	整数部 (p-s)	小数部 (s)		
1.234	8	5	3	00001.234	<ul style="list-style-type: none"> <li>整数部の桁が不足している場合 整数部の左側に 0 を加えられて、5 桁に調整されます。</li> </ul>
△△ 12.34	6	4	2	0012.34	<ul style="list-style-type: none"> <li>整数部の先頭が空白の場合 整数部の空白は 0 とされます。</li> </ul>
12345.67	5	3	2	345.67	<ul style="list-style-type: none"> <li>整数部の桁があふれている場合 整数部の左側 2 桁は入力されません。</li> </ul>
1234.5678	4	0	4	0.5678	<ul style="list-style-type: none"> <li>整数部なしの場合 整数部の有効桁が 0 のため、整数部はすべて切り捨てられます。</li> </ul>

入力データ	有効桁数の指定			内部データ形式	説明
	総桁数 (p)	整数部 (p-s)	小数部 (s)		
1.23	5	1	4	1.2300	<ul style="list-style-type: none"> <li>小数部の桁が不足している場合 小数部の右側に 0 を加えられて、4 桁に調整されます。</li> </ul>
12.3456	3	2	1	12.3	<ul style="list-style-type: none"> <li>小数部の桁があふれている場合 小数部の 2 桁目を丸めます。この例では、丸め方法に四捨五入を適用しています。</li> </ul>
1234.56	4	4	0	1235	<ul style="list-style-type: none"> <li>小数部なしの場合 小数部の有効桁が 0 のため、整数の値に丸めます。この例では、丸め方法に四捨五入を適用しています。</li> </ul>
△△△△△	4	3	1	000.0	<ul style="list-style-type: none"> <li>すべて 1 バイトの空白文字の場合 すべての空白文字は 0 とされます。</li> </ul>

(凡例)

△ : 半角スペース (空白) を示します。

## 付録 B.2 文字列データの端数処理

ここでは、入力データが文字列の場合の端数処理について説明します。

入力データの文字列の扱いを次に示します。

- 入力データの文字列は、左端をそろえて入力します。
- 指定した最大バイト長に満たないデータを入力した場合、右側に空白を加えて最大バイト長に調整します。
- 指定した最大バイト長を超えるデータを入力した場合、右側のあふれ部分は入力しません。

指定例を次に示します。

入力データ	最大バイト長	内部データ形式	説明
ABC	5	ABC △△	右側に空白を加えて 5 バイトに調整します。
ABCDE	3	ABC	右側のあふれ部分は入力しません。



(凡例)

△：半角スペース (空白) を示します。

また、マルチバイト文字の途中で有効桁を超えた場合、超えた部分の途中の文字を空白文字に置き換える端数処理を実行します。なお、端数処理は属性コード「c」の場合だけ実行します。属性コード「j」と「q」は、シングルバイト文字として扱うので、端数処理は実行しません。

端数処理については、「付録 I.2(4)(c) バイト単位比較の端数処理」を参照してください。

## 付録 B.3 日付データの端数処理

ここでは、入力データが日付の場合の端数処理について説明します。

入力時の注意事項を次に示します。

- 入力データの文字列は、左端をそろえて入力します。
- 書式で指定した、桁 (桁修飾文字ハイフン、スラント、およびピリオド※は含まない) を超えるデータが入力された場合、指定桁数を超えた部分のデータが無視されます。
- 有効桁数に満たない場合は、0 (ゼロ) を加えられて、有効桁数に調整されます。
- 日付として正しくないデータの場合も、エラーにしないで正常に扱います。例えば、「2008-11-31」、および「2008-01-00」などの存在しない日付、または計算機やほかのアプリケーションで表現ができないとされている年号など。
- 桁修飾文字 (ハイフン、スラント、およびピリオド※) で年月日を区切っている場合、MM (月) 部分、および DD (日) 部分のデータは 1 桁でもかまいません。1 桁データは 2 桁に調整されます。例えば、「2008-7-5」と「2008-07-05」は同じ値として扱われます。
- 桁修飾文字 (ハイフン、スラント、およびピリオド※) で区切られた年月日のそれぞれの桁が指定された書式と合致しない場合、それぞれの桁ごとに桁調整されます。

注※

環境変数「EXSORTGOPTION (64 ビット版は「EXSORT64GOPTION)」」の指定がある場合、ピリオドを桁修飾文字として扱います。

指定例を次に示します。なお、d は日付データの有効桁を示します。

入力データ	書式の設定	有効桁の指定 (d)	内部データ形式	説明
2008-05-09 2008-9-3	YYYY-MM-DD	6	200805 200809	右側の桁あふれ部分 (DD) は入力されません。
080509	YYMMDD	2	08	右側の桁あふれ部分 (MMDD) は入力されません。

入力データ	書式の設定	有効桁の指定 (d)	内部データ形式	説明
05/09/08 9/3/08	MM/DD/YY	4	0509 0903	右側の桁あふれ部分 (YY) は入力されません。
123456-7890-123	YYYY-MM-DD	8	12347812	年月日のそれぞれの桁あふれ部分 (下線部分) は入力されません。 YYYY : 123456 MM : 7890 DD : 123
2008 2008-9	YYYY-MM-DD	8	20080000 20080900	右側 (MMDD, または DD) に 0 (ゼロ) を加えられて、8 桁に調整されます。
200805	YYYYMMDD	8	20080500	右側 (DD) に 0 (ゼロ) を加えられて、8 桁に調整されます。
05/09 9/3	MM/DD/YY	6	000509 000903	右側 (YY) に 0 (ゼロ) を加えられて、6 桁に調整されます。
1//2	MM/DD/YY	6	200100	次の下線部分のように、年月日それぞれの右側に 0 (ゼロ) を加えられます。 MM : 1 (1 桁の月は 01 と見なされます) DD : 00 YY : 20
08.6.24 08.10.8	YY.MM.DD	4	0806 0810	右側の桁あふれ部分 (DD) は入力されません。

**注意事項**

年月日の数値部分だけが有効桁数の対象です。ハイフン (-), スラント (/), およびピリオド (.) は、有効桁数に含まれません。

**付録 B.4 引用符のエスケープ**

ここでは、入力データ中の引用符の扱いについて説明します。

入力データの先頭が引用符の場合だけ、対の引用符で囲んだ範囲がエスケープされます。入力データと引用符の対応を次の表に示します。

入力データの先頭	入力データ	エスケープ範囲	エスケープ結果	説明
引用符の場合	'aa'bb' 'aa'""bb' ""aabb' 'aabb'"" 'aa'bb'cc' 'aa'bb'	'aa'bb' 'aa'""bb' ""aabb' 'aabb'"" 'aa'bb'cc' 'aa'bb'	aa'bb aa'""bb 'aabb' aabb' aa'bbcc' aabb'	引用符で囲んだ範囲 (下線部分) がエスケープされます。エスケープ範囲内の 2 個並んだ引用符が 1 個で扱われます。
引用符ではない場合	aa"bb aa""bb aa'bb'cc	aa"bb aa""bb aa'bb'cc	aa"bb aa""bb aa'bb'cc	エスケープされません。

## 付録 B.5 CSV 数値データ (固定小数点数) の除外文字の位置規則

ここでは、数値データ (固定小数点数) 中の除外文字の位置規則について説明します。

- 数値データ中には、0～9の数字以外に、データを囲んでいる引用符、符号、通貨記号、および桁修飾文字 (コンマ、小数点、空白) を含むことができます。これらの文字を除いた内部データ形式に変換して処理されます。それぞれの除外文字には位置規則があります。位置規則の対応を次の表に示します。

表 B-1 除外文字の位置規則

除外文字の種類	例	正常なデータ形式として扱う除外文字の位置	不正なデータ形式として扱う除外文字の位置	備考
引用符	"1234"	データの先頭および末尾	データの途中	—
符号※	+1234 1234-	数字の前、またはあとに 1 個だけ	<ul style="list-style-type: none"> <li>数字の途中</li> <li>2 個以上</li> </ul>	<ul style="list-style-type: none"> <li>符号がない場合は正値を仮定します。</li> <li>+0, 0, -0 は、同じ値として処理されます。</li> </ul>
通貨記号※	¥1234	数字の前に 1 個だけ	<ul style="list-style-type: none"> <li>数字の途中</li> <li>数字のあと</li> <li>2 個以上</li> </ul>	—
コンマ	1,234,567	数字の途中に 1 個ずつ	<ul style="list-style-type: none"> <li>数字の前</li> <li>数字のあと</li> <li>続けて 2 個以上</li> </ul>	—
小数点	.1234 12.34 1234.	数字の前、途中またはあとに 1 個だけ	2 個以上	—
空白※	△△△△△ △ 1234 △△	数字の前後、またはすべての空白	数字の途中	すべて空白の場合は 0 で処理されます。

(凡例)

- : なし

注※

空白、符号、通貨記号を数字の前に指定する場合、これらの並び順は任意です。  
例を次に示します。

△・△ ¥△ 1234 → -1234

¥△△・△△ 56 → -56

- CSV 数値データ (固定小数点数) の形式チェックオプション (CSVFRMCHK オプション) を指定する場合、引用符を除いた除外文字は、有効桁内の数字に付帯していなければなりません。付帯していない場合、除外文字として解釈されません (除外文字が付帯していなくてもエラーにはなりません、符号や小数点などとして扱われません)。

(例) 有効桁 4 桁 (下線) の符号の場合

1234.5#AA → -1234 ... 数字に付いているハイフンは「符号」として扱う

DFX#2101 → -2101 ... 数字に付いているハイフンは「符号」として扱う

#1234567 → -4567 ... 数字に付いているハイフンは「符号」として扱う

#BBB1234 → +1234 ... 数字に付いていないハイフンは「符号」として扱わない

#12E3456 → +3456 ... 有効桁の数字に付いていないハイフンは「符号」として扱わない

- CSVFRMCHK オプションを指定しても、有効桁内の数字とその数字に付帯する除外文字が、表 B-1 に示す位置規則に違反している場合は、KBLS1211-E または KBLS1320-E エラーとなります。

## 付録 C CSV ファイル形式と CSV データ形式 (SORT EE の機能)

CSV ファイル形式および CSV データ形式について説明します。

### 付録 C.1 CSV ファイル形式

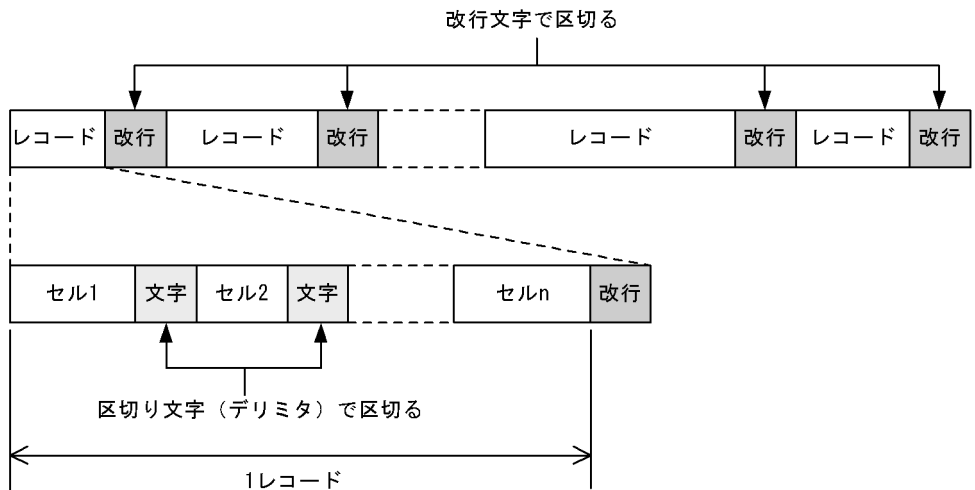
可変長データの CSV ファイルのソート、マージ、およびコピーができます。

CSV ファイルとは、幾つかのデータをコンマで区切ったテキストデータ、またはテキストファイルのことを指しますが、SORT EE では、コンマ以外の任意の文字をデータの区切りに指定することもできます。

SORT EE の CSV ファイルは、改行文字で複数のレコードに区切られ、さらに各レコードは任意の文字によって複数のデータ項目「セル」に区切られます。

SORT EE で扱う CSV ファイルの形式を次の図に示します。

図 C-1 SORT EE で扱う CSV ファイルの形式



SORT EE では、CSV ファイルの仕様を次のとおりに規定しています。

- 改行文字をレコードの区切りとします (最終レコードの改行文字はなくてもかまいません)。
- レコードは、区切り文字である「デリミタ」によって、複数のデータ項目「セル」に区切られます。
- レコード長は可変長となります。
- 実データのないセルのことを、「ナル文字列 (データなしセル)」と言います。
- セル数 0 個は、改行文字だけの空レコードを意味します。この場合は、すべてのセル

がナル文字列であると仮定してソートの対象にします。

- セルデータに、改行文字またはデリミタを含む場合、セルデータを引用符で囲んでください。
- セルデータに引用符を含む場合、引用符を二つ並べてエスケープしてください (エスケープされていない引用符の直前までがセルデータと見なされます)。

CSV ファイルのサイズを、次の表に示します。

表 C-1 CSV ファイルのサイズ

項目	範囲	説明
レコード数	1 ~ 2,147,483,647 件 (1 ~ 2G-1 件)	レコード件数が 0 件の場合は処理されません (空のファイルが出力されます)。
レコード長	1 ~ 65,535 バイト	<ul style="list-style-type: none"> <li>改行文字を CRLF とした場合、最大長は 65,534 バイトとなります。</li> <li>改行文字を LF とした場合、最大長は 65,535 バイトとなります。</li> <li>改行文字、およびすべてのデリミタおよびセルデータを含めた長さが、65,536 バイトを超えてはなりません。</li> <li>入力レコード長を指定する <code>-inpfile</code> パラメタの <code>reclen</code> オペランドは省略できます。省略すると、ソート機能で一時ファイルを使用する場合があるので注意してください。reclen オペランドの詳細は、「5.8.3 reclen オペランド」を参照してください。</li> </ul>
ファイル容量	0 バイト以上 (OS の制限に準拠)	4GB を超えるファイルの指定ができます。
セル数	0 ~ 16,384 個 / レコード	<ul style="list-style-type: none"> <li>ナル文字列 (データなしセル) の個数も含まれます。</li> <li>キーに指定できるセルの個数は最大 64 個です。</li> </ul>
セル長	最小 : 0 バイト 最大 : 65,535 バイト	<ul style="list-style-type: none"> <li>セル長が 0 バイトとは、ナル文字列 (データなしセル) のことを表します。</li> <li>改行文字を CRLF とした場合、最大長は 65,534 バイトとなります。</li> <li>改行文字を LF とした場合、最大長は 65,535 バイトとなります。</li> <li>マルチバイト文字 (日本語) の途中で最大長を超えた場合、超えた部分のマルチバイト文字以降は無視されます。</li> </ul>

## 付録 C.2 CSV データ形式

CSV ファイルで扱えるデータの種類の、指定できる値の範囲、およびデータ形式について説明します。

### (1) データの種類

CSV ファイルとして指定できるデータの種類の、および指定できる最小値および最大値を、

次の表に示します。

表 C-2 データの種類

種別		指定できる値の範囲		備考
		最小	最大	
数値		1 桁	38 桁	10 進数の固定小数点
文字	ASCII SJIS EUC-JP EUC-HJ UTF-8	1 バイト	4,096 バイト	codetype の指定値によつて、指定できる値の範囲が異なります。
	UCS-2LE UCS-2BE UTF-16LE UTF-16BE	2 バイト	8,192 バイト	
	UCS-4LE UCS-4BE UTF-32LE UTF-32BE	4 バイト	16,384 バイト	
左分離符号付き数字		1 桁	256 桁	—
日付		2 桁	8 桁	—
日付時刻	日付	0 桁	8 桁	日付は 0, 4, 6, 8 桁, 時刻は 0, 2, 4, 6, 9 桁以外、指定できません。日付、時刻ともに 0 桁は指定できません。
	時刻	0 桁	9 桁	

(凡例)

— : なし

## (2) データ形式

CSV ファイルとして指定できるデータの形式を、次の表に示します。

表 C-3 データ形式

種別	形式
数値 (固定小数点数)	0 ~ 9 の数字以外に、次の文字をデータに含むことができます。 <ul style="list-style-type: none"> <li>引用符 (データの先頭および末尾の引用符)</li> <li>符号 (+, または -)</li> <li>通貨記号</li> <li>コンマ</li> <li>小数点</li> <li>空白 (データの前後の空白)</li> </ul>
文字	データに、デリミタまたは改行文字を含む場合、データの先頭、および末尾を引用符で囲んでください。

種別	形式
左分離符号付き数字	0～9の数字と符号 (+ または -) 以外に、引用符でデータを囲むことができます。 形式の詳細については、「付録 E 左分離符号付き数字のデータ形式 (SORT EE の機能)」を参照してください。
日付	<p>年号 (YYYY, または YY) の桁数と位置によって、次の 4 種類の形式に分かれます。</p> <ul style="list-style-type: none"> <li>前置 4 桁年号 YYYY/MM/DD YYYY-MM-DD YYYY.MM.DD ※ YYYYMMDD YYYY</li> <li>後置 4 桁年号 MM/DD/YYYY MM-DD-YYYY MM.DD.YYYY ※ MMDDYYYY</li> <li>前置 2 桁年号 YY/MM/DD YY-MM-DD YY.MM.DD ※ YYMMDD YY</li> <li>後置 2 桁年号 MM/DD/YY MM-DD-YY MM.DD.YY ※ MMDDYY</li> </ul> <p>また、次の文字をデータに含むことができます。</p> <ul style="list-style-type: none"> <li>引用符 (データの先頭、および末尾の引用符)</li> <li>スラント (年月日の区切り)</li> <li>ハイフン (年月日の区切り)</li> <li>ピリオド (年月日の区切り) ※</li> </ul>
日付時刻	0～9の数字と 12 時間制表記以外に、引用符、区切り文字および桁修飾文字を含むことができます。 形式の詳細については、「3.2 キー属性と形式」を参照してください。

(凡例)

YYYY : 西暦 4 桁年号  
 YY : 西暦下 2 桁年号  
 MM : 月  
 DD : 日

桁修飾文字で年月日を区切っている場合は、MM および DD 部分のデータは 1 桁でもかまいません。

注※

環境変数「EXSORTGOPTION (64 ビット版は「EXSORT64GOPTION)」」の指定がある場合、ピリオドを桁修飾文字として扱います。



各データ形式には、詳細な規則があります。CSV ファイルの指定方法、および注意事項については、「5.16 `-csvkey` パラメタ」を参照してください。

---

## 付録 D CSV ファイル内のナル文字列の扱い (SORT EE の機能)

SORT EE で扱える CSV ファイル内のナル文字列について説明します。

### 付録 D.1 ソートキー・マージキー

ソート機能およびマージ機能では、ナル文字列 (null キー) が含まれるレコードは、ソート対象外のレコードとして末尾に配置されます。

null キーが含まれるレコードを先頭に配置したい場合は、環境ファイルの「EXS\_NULLSEQ」または `-inpf`、`-catinp`、`-splitinp` パラメタの `nullseq` オペランドを指定します。

`nullseq` オペランドの詳細は「5.8.12 `nullseq` オペランド」を参照してください。

図 D-1 null キーのソート例 (EXS\_NULLSEQ=0 を指定した場合)

- セル2をキーに昇順ソートを行う例

【入力データ】				【昇順ソート結果】				
	セル1	セル2	セル3		セル1	セル2	セル3	
レコード1	ST80	210	5		PN21	-110	4	レコード4
レコード2	FL15	150	2		FL15	150	2	レコード2
レコード3	CL10	null	8		ST80	210	5	レコード1
レコード4	PN21	-110	4		CL10	null	8	レコード3

nullキーが含まれるレコードは、末尾に配置します。

- セル1とセル2をキーにソートを行う例

【入力データ】				【昇順ソート結果】				
	セル1	セル2	セル3		セル1	セル2	セル3	
レコード1	CL10	0	1		CL10	-130	5	レコード5
レコード2	ST80	210	2		CL10	0	1	レコード1
レコード3	CL10	null	3		CL10	null	3	レコード3
レコード4	FL15	150	4		FL15	150	4	レコード4
レコード5	CL10	-130	5		ST80	210	2	レコード2

【降順ソート結果】				
	セル1	セル2	セル3	
	ST80	210	2	レコード2
	FL15	150	4	レコード4
	CL10	0	1	レコード1
	CL10	-130	5	レコード5
	CL10	null	3	レコード3

第2キー (セル2) にnullキーが含まれるレコードは、同一の第1キー (セル1) 内の末尾に配置します。

図 D-2 null キーのソート例 (EXS\_NULLSEQ=1 を指定した場合)

●セル2をキーに昇順ソートを行う例

【入力データ】				【昇順ソート結果】			
	セル1	セル2	セル3	セル1	セル2	セル3	
レコード1	ST80	210	5	CL10	null	8	レコード3
レコード2	FL15	150	2	PN21	-110	4	レコード4
レコード3	CL10	null	8	FL15	150	2	レコード2
レコード4	PN21	-110	4	ST80	210	5	レコード1

nullキーが含まれるレコードは、先頭に配置します。

●セル1とセル2をキーにソートを行う例

【入力データ】				【昇順ソート結果】			
	セル1	セル2	セル3	セル1	セル2	セル3	
レコード1	CL10	0	1	CL10	null	3	レコード3
レコード2	ST80	210	2	CL10	-130	5	レコード5
レコード3	CL10	null	3	CL10	0	1	レコード1
レコード4	FL15	150	4	FL15	150	4	レコード4
レコード5	CL10	-130	5	ST80	210	2	レコード2

【降順ソート結果】				
	セル1	セル2	セル3	
	ST80	210	2	レコード2
	FL15	150	4	レコード4
	CL10	null	3	レコード3
	CL10	0	1	レコード1
	CL10	-130	5	レコード5

第2キー (セル2) にnullキーが含まれるレコードは、同一の第1キー (セル1) 内の先頭に配置します。

## 付録 D.2 集約キー

集約機能では、ナル文字列 (null キー) 同士は同値として扱い、null キーのレコードを集約します。

null キーを一意的なキーと解釈し、集約対象外にしたい場合は、コマンドオプションの SUMNULKEY を指定します。

SUMNULKEY オプションの詳細は「5.5.19 ナル文字列の集約オプション (SUMNULKEY, SUMNONULL オプション)」を参照してください。

図 D-3 null キーの集約例 (SUMNULKEY オプション指定なし)

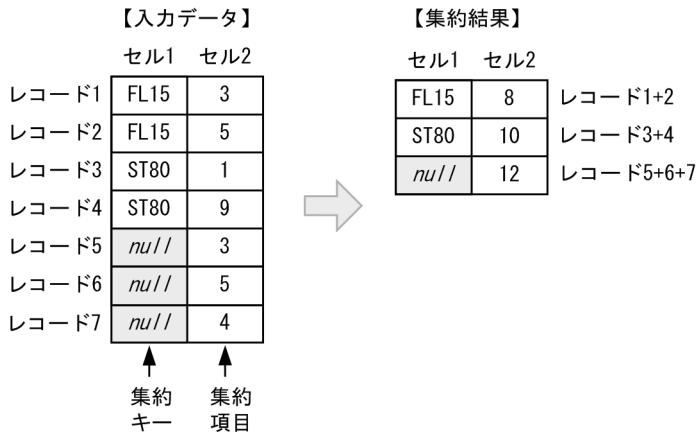
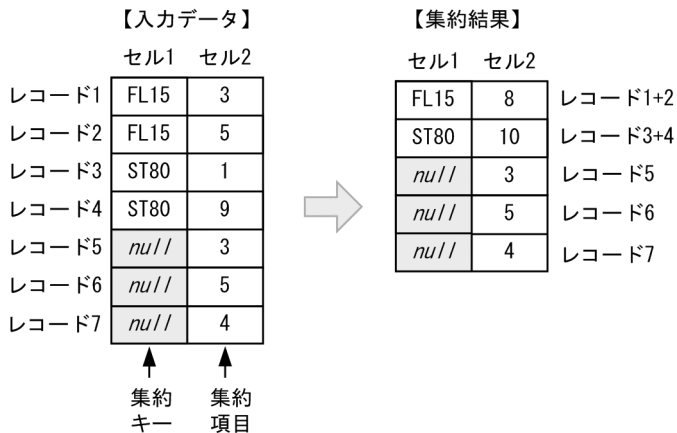


図 D-4 null キーの集約例 (SUMNULKEY オプション指定あり)



### 付録 D.3 比較項目

ナル文字列 (null 項目) は, ナル文字列であるかどうか (ナル文字列の抽出) の判定だけを行うことができます。

null 項目は, 条件値との大小を比較できません。大小比較 (>または<) を指定した場合, include, omit の指定に関わらず, 比較結果は常に「偽 (選択せず)」と判定されます。

図 D-5 null 項目の選択例 (比較項目同士の比較)

【入力データ】		【セル1とセル2の比較結果】						
	セル1	セル2	(1) =	(2) !=	(3) >	(4) >=	(5) <	(6) <=
レコード1	null	0	×	○	×	×	×	×
レコード2	0	null	×	○	×	×	×	×
レコード3	null	null	○	×	×	×	×	×
レコード4	0	0	○	×	×	○	×	○
レコード5	1	0	×	○	○	○	×	×

(凡例) ○ : 選択する。 × : 選択しない。

- (1), (2) は, ナル文字列と比較項目の大小比較ではないので, レコード1~3も選択されます。
  - (1) csvinclude=c+1-1 = c+2-1
  - (2) csvinclude=c+1-1 != c+2-1
- (3)~(6) は, ナル文字列と比較項目の大小を比較することになるため, レコード1~3が選択されることはありません。
  - (3) csvinclude=c+1-1 > c+2-1
  - (4) csvinclude=c+1-1 >= c+2-1
  - (5) csvinclude=c+1-1 < c+2-1
  - (6) csvinclude=c+1-1 <= c+2-1

図 D-6 null 項目の選択例 (比較項目と条件値 (NULL) との比較)

【入力データ】		【比較結果】					
	セル1	セル2	(1)	(2)	(3)	(4)	(5)
レコード1	null	0	×	○	○	○	○
レコード2	0	null	○	×	×	×	×
レコード3	null	null	○	×	○	×	○
レコード4	0	0	×	○	×	○	×
レコード5	1	0	×	○	○	○	○

(凡例) ○ : 選択する。 × : 選択しない。

- (1)~(5)の選択条件は次のとおりです。
- 比較項目 (セル2) がナル文字列であるレコードを選択します。
    - (1) csvinclude=c+2-1 = NULL
  - 比較項目 (セル2) がナル文字列であるレコードを選択しません。
    - (2) csvomit=c+2-1 = NULL
  - 最初の選択条件式では, ナル文字列と比較項目の大小を比較することになるため, レ

コード 1 ~ 3 は選択されません。2 つ目の選択条件式では、比較項目 (セル 1) がナル文字列であるレコードを選択します。

(3) `csvinclude=c+1-1 > c+2-1 OR c+1-1 = NULL`

- 比較項目 (セル 2) が「0」であるレコードを選択します。

(4) `csvinclude=c+2-1 = C'0'`

- 比較項目 (セル 1) が「0」でないレコードを選択します。

(5) `csvinclude=c+1-1 != C'0'`

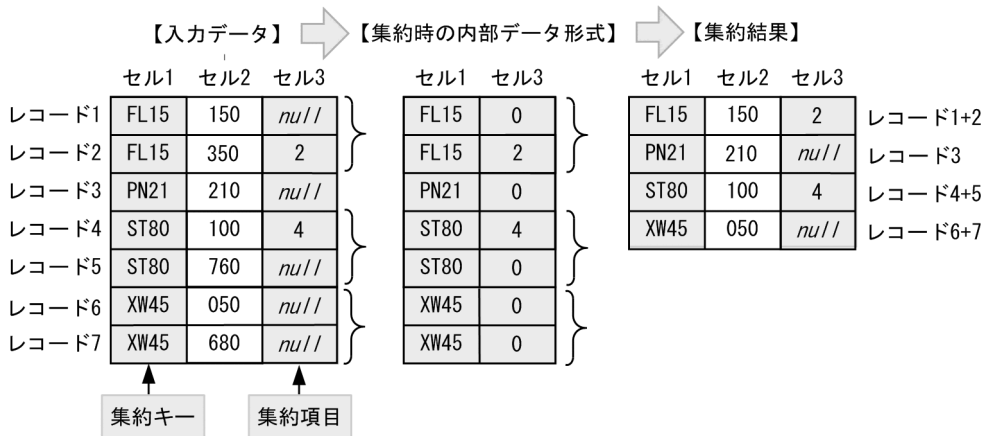
## 付録 D.4 集約項目

ナル文字列 (null 項目) はゼロ値を仮定して加算します。ただし、同一キーのレコードがなく、集約項目の加算が発生しなかった場合、データが変更されることはありません。

null 項目をデータがない項目と解釈し、集約対象外にしたい場合は、コマンドオプションの `SUMNONULL` を指定します。

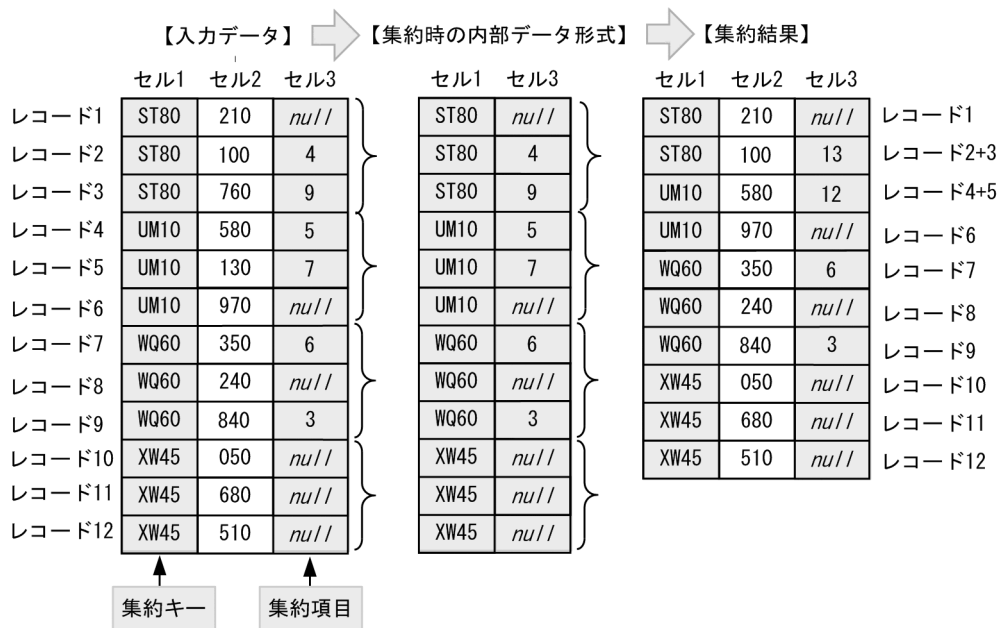
`SUMNONULL` オプションの詳細は「5.5.19 ナル文字列の集約オプション (`SUMNULKEY`, `SUMNONULL` オプション)」を参照してください。

図 D-7 null 項目の集約例 (`SUMNONULL` オプション指定なし)



- レコード 1 と 2, レコード 4 と 5, およびレコード 6 と 7 は, null 項目をゼロ値と仮定して集約します。
- レコード 3 は, 加算する同一キーのレコードがないので, 集約項目は変更しないでナル文字列 (null) のまま出力します。
- レコード 6 と 7 は, どちらも null 項目なので, ナル文字列 (null) で出力します。

図 D-8 null 項目の集約例 (SUMNONULL オプション指定あり)



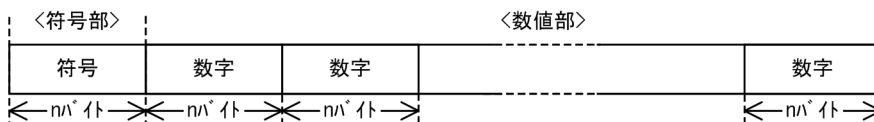
- レコード 1, 6, 8 および 10 ~ 12 は, null 項目を集約対象外とします。
- レコード 1 ~ 3 は同一キー (ST80) ですが, null 項目のレコード 1 は加算されません。レコード 1 はそのまま出力します。次に, レコード 2 と 3 を集約した結果を出力します。
- レコード 4 ~ 6 は同一キー (UM10) ですが, null 項目のレコード 6 は加算されません。レコード 4 と 5 を集約した結果を出力します。次に, レコード 6 をそのまま出力します。
- レコード 7 ~ 9 は同一キー (WQ60) ですが, null 項目のレコード 8 は加算されません。結果的にレコード 6 ~ 8 は集約されずに, そのまま出力します。
- レコード 10 ~ 12 は同一キー (XW45) ですが, すべて null 項目なので加算されずに, そのまま出力します。



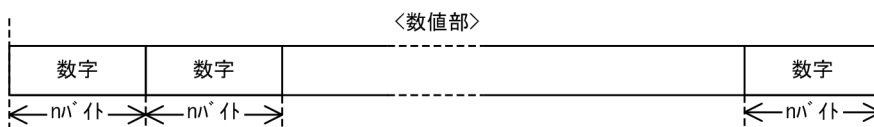
## 付録 E 左分離符号付き数字のデータ形式 (SORT EE の機能)

左分離符号付き数字 (文字) について説明します。キー・項目属性「d」の左分離符号付き数字 (文字) は、1 ~ 256 桁の数字データです。データ形式を次に示します。

### ●符号あり



### ●符号なし



- 符号部は、+ (正) または - (負) のどちらかの文字を指定します。
- 数値部は、0 ~ 9 の文字を指定します。
- 符号ありの場合、データは符号と数値部で構成します。
- 符号なしの場合、データは数値部だけで構成します。
- 符号なしのデータは、正の値として扱います。
- 1 桁の長さ (n バイト) は入力ファイルのエンコードによって異なります。  
ASCII, SJIS, EUC-JP, EUC-HJ, または UTF-8 : 1 バイト  
UCS-2LE, UCS-2BE, UTF-16LE, または UTF-16BE : 2 バイト  
UCS-4LE, UCS-4BE, UTF-32LE, または UTF-32BE : 4 バイト
- 符号の前、および数値部の前後に指定された空白、およびタブは無視します。ただし、数字部の途中に空白、およびタブが存在した場合は、空白、およびタブから後ろはすべて無視します。
- 符号なしゼロ、正符号付きゼロ (+0)、および負符号付きゼロ (-0) は同じ値として扱います。
- 数値部に数字と数字以外の文字が混在した場合、先頭から連続した数字だけを比較の対象とします。数字以外の文字より後ろは無視します。
- 符号だけのデータは、正符号付きゼロ値 (+0) として扱います。
- 先頭が符号、数字、空白およびタブでない場合、正符号付きゼロ値 (+0) として扱います。

### 【指定例】

●例1

バイト位置	0	1	2	3	4	5
	+	0	1	2	3	4

符号指定ありの正数値と扱います。データ値は、+1234です。

●例2

バイト位置	0	1	2	3	4	5
	5	6	7	8	9	0

符号指定省略の正数値と扱います。データ値は、+567890です。

●例3

バイト位置	0	1	2	3	4	5
	△	-	△	1	2	△

符号部の前、および数値部の前後に存在した空白を無視します。データ値は、-12です。

●例4

バイト位置	0	1	2	3	4	5
	1	△	2	△	3	△

数値部の途中(1バイト目)に空白が存在したため、空白から後ろはすべて無視します。データ値は、+1です。

●例5

バイト位置	0	1	2	3	4	5
	+	0	1	△	2	3

数値部の途中(3バイト目)に空白が存在したため、空白から後ろはすべて無視します。データ値は、+1です。

●例6

バイト位置	0	1	2	3	4	5
	△	△	△	△	-	0

負符号付きゼロ値として扱います。-cmdoptコマンドのDIFFZEROオペランドが指定された場合「+0」より小さい値として扱います。データ値は、-0です。

●例7

バイト位置	0	1	2	3	4	5
	-	0	1	2	a	3

先頭から連続した数字を、負数値として扱います。文字「a(4バイト目)」から後ろは無視します。データ値は、-12です。

## ●例8

バイト位置	0	1	2	3	4	5
	△	△	△	△	△	+

バイト位置	0	1	2	3	4	5
	△	△	△	△	△	-

符号だけのデータのため、正符号付きゼロ値として扱います。  
データ値は、+0です。

## ●例9

バイト位置	0	1	2	3	4	5
	/	1	3	5	7	9

先頭文字が、符号、数字、空白、およびタブ以外だったため、正符号付きゼロ値として扱います。データ値は、+0です。

## ●例10

バイト位置	0	1	2	3	4	5
	あ		い		う	

先頭文字が、符号、数字、空白、およびタブ以外だったため、正符号付きゼロ値として扱います。データ値は、+0です。

## ●例11

バイト位置	0	1	2	3	4	5
	9	8	7	6	5	t

先頭から連続した数字を、正数値として扱います。文字「t(5バイト目)」は無視します。データ値は、+98765です。

注 左分離符号付き数字(文字)として指定された、負値の外部10進数データは、負値として認識せず、正値として扱います。

## 付録 F レコード編集処理で基になるレコード (SORT EE の機能)

レコード編集によって、例えば、入力レコードと異なった位置にキーや項目を再編成する場合は注意が必要となります。ソートキー、マージキー、集約キー、比較項目、集約項目、分割キー、および非分割データの範囲は、最後に実行したレコード編集後のレコードを基に指定してください。レコード編集処理で基になるレコードについて次の表に示します。

表 F-1 レコード編集処理で基になるレコード

実行順序※1			実行処理	指定パラメータ	指定オペランド	レコード編集処理で基になるレコード
ファイル分割・連結なし	ファイル分割	ファイル連結				
1	1	1	レコード入力	·cmdopt	INPBIG, INPLTL INPCONSEPA	—
				·inpfile ·catinp ·splitinp	fileorg reclen recsepa	—
2	2	2	入力レコード読み飛ばし	·inpfile ·catinp ·splitinp	skiprec startrec	—
—	—	3	ファイル連結	·catinp	—	—
3	3	4	入力レコード選択処理	·inpfile ·catinp ·splitinp	{include   omit} {csvinclude   csvomit}	入力レコードを基に指定する。
4	4	5	入力レコード打ち切り処理	·inpfile ·catinp ·splitinp	stopaft	—
5	5	6	入力レコード編集処理	·inpfile ·catinp ·splitinp	reorg csvreorg	入力レコードを基に指定する。

実行順序※1			実行処理	指定パラメータ	指定オペランド	レコード編集処理で基になるレコード
ファイル分割・連結なし	ファイル分割	ファイル連結				
6	6	7	ソート処理 マージ処理 コピー処理	-key -csvkey	—	<ul style="list-style-type: none"> <li>入力レコード編集処理を実行しない場合は、入力レコードを基に指定する。</li> <li>入力レコード編集処理を実行する場合は、編集後のレコードを基に指定する。</li> </ul>
7	7	8	集約処理	-sum -csvsum	—	
8	8	9	中間レコード編集処理	-reorg -csvreorg	—	
9	9	10	出力レコード選択処理	-outifle -catout -splitout	{include   omit} {csvinclude   csvomit}	<ul style="list-style-type: none"> <li>中間レコード編集処理を実行しない場合は、集約処理の時点のレコードを基に指定する。</li> <li>中間レコード編集処理を実行する場合は、編集後のレコードを基に指定する。</li> </ul>
10	10	11	出力レコード編集処理	-outifle -catout -splitout	reorg csvreorg	
			CSVレコード変更処理	-cmdopt -outifle -catout -splitout	OUTCONSEPA cellsepa quottype cellnum	—
—	11	—	ファイル分割処理	-splitfile -splitdata -splitkey	— nosplitdata nosplitcsvdata key csvkey	<ul style="list-style-type: none"> <li>出力レコード編集処理、および CSV レコード変更処理の cellnum (セル数調整) オプションを実行しない場合は、出力レコード選択処理時点のレコードを基に指定する。</li> <li>出力レコード編集処理、または CSV レコード変更処理の cellnum (セル数調整) オプションを実行する場合は、編集後のレコードを基に指定する。</li> </ul>

実行順序※1			実行処理	指定パラメータ	指定オペランド	レコード編集処理で基になるレコード
ファイル分割・連結なし	ファイル分割	ファイル連結				
11	12	12	ファイル編成変更※2	-cmdopt	OUTBIG, OUTLTL	—
				-outifle -catout -splitout	fileorg reclen minlen recsepa	<ul style="list-style-type: none"> <li>出力レコード編集処理、および CSV レコード変更処理の <b>cellnum</b> (セル数調整) オプションを実行しない場合は、出力レコード選択処理時点のレコードを基に、出力レコードを指定する。</li> <li>出力レコード編集処理、または CSV レコード変更処理の <b>cellnum</b> (セル数調整) オプションを実行する場合は、編集後のレコード長を基に、出力レコード長を指定する。</li> </ul>

(凡例)

— : なし

注※1

「4.1.1 機能の実行順序」の「表 4-1」の実行順序を示します。

注※2

出力ファイルへ出力する直前で、ファイル編成とレコード形式を変更します。実行順序 1 (ファイル連結) ~ 11 (ファイル分割) は、入力ファイルのファイル編成とレコード形式で動作します。

## 付録 G altseq 変換ファイル (SORT EE の機能)

文字順序変更機能での文字の順は、`-altseq` パラメタを指定する以外に `altseq` 変換ファイルを使用することでも変更できます。

`altseq` 変換ファイルについて説明します。

### 付録 G.1 altseq 変換ファイルの詳細

256 個の変更後の順序 `yy` (1 バイトの 16 進文字) を格納します。

なお、`yy` 以外に空白、タブ、改行文字、およびコメントを書くことができます。書くことができる個所はファイル内のすべての個所です。例えば、コード `yy` の `y` と `y` の間に書くこともできます。

- ファイルの名称は、32 ビット版では「`exsort_altseq.txt`」、64 ビット版では「`exsort64_altseq.txt`」という固定の名称です。ディレクトリ名は、環境ファイル「`Exsortconf.ini` ファイル (64 ビット版は「`Exsortconf64.ini` ファイル)」に指定します。
- ファイルはテキスト形式です。
- ファイルの大きさは、最大 4,096 バイトです。ファイルの大きさが制限以内であれば、1 行当たりの文字数の制限はありません。
- `/*` から `*/` までは、コメントと見なします。コメントには、ファイルに格納できるすべての文字を使用することができます。例えば、ひらがなや漢字などを使用できます。
- 先頭から `xx` 個目 (`x` は 16 進数) の `yy` は、JIS8 単位コードの変更したいコード `xx` に対し、それをどの位置に置くかを `yy` で表します。変更のないコードの `xx` と `yy` は同じです。

### 付録 G.2 altseq 変換ファイルの指定方法

`altseq` 変換ファイル格納ディレクトリを、`Exsortconf.ini` ファイルに指定します。

`Exsortconf.ini` ファイルに `altseq` 変換ファイルの指定がない場合は、JIS8 単位コード体系で比較します。

#### (1) `Exsortconf.ini` ファイルの指定

環境ファイル「`Exsortconf.ini` ファイル (64 ビット版は「`Exsortconf64.ini` ファイル)」には、次に示すようにディレクトリ名を指定します。

`EXS_ALTSEQPATH=altseq` 変換ファイル格納ディレクトリ名

- `altseq` 変換ファイルを格納したディレクトリと `altseq` 変換ファイル

「exsort\_altseq.txt ファイル (64 ビット版は「exsort64\_altseq.txt ファイル)」) に対し、読み込みの権限が必要です。

- Exsortconf.ini ファイルの作成先ディレクトリについては、「15.1 SORT EE の環境ファイルの設定」を参照してください。

## (2) altseq 変換ファイルの指定例

文字コード 0x30 ~ 0x39 (数字の 0 ~ 9) を、0xF0 ~ 0xF9 の順に変更する指定例を次の図に示します。

図 G-1 altseq 変換ファイルの指定例

/*△altseq△data△*/↓
0001020304△05060708△090A0B0C△0D0E0F△/* 00 - 0F */↓
1011121314△15161718△191A1B1C△1D1E1F△/* 10 - 1F */↓
2021222324△25262728△292A2B2C△2D2E2F△/* 20 - 2F */↓
F0F1F2F3F4△F5F6F7F8△F93A3B3C△3D3E3F△/* 30 - 39 -> F0 - F9 */↓
4041424344△45464748△494A4B4C△4D4E4F△/* 40 - 4F */↓
5051525354△55565758△595A5B5C△5D5E5F△/* 50 - 5F */↓
6061626364△65666768△696A6B6C△6D6E6F△/* 60 - 6F */↓
7071727374△75767778△797A7B7C△7D7E7F△/* 70 - 7F */↓
8081828384△85868788△898A8B8C△8D8E8F△/* 80 - 8F */↓
9091929394△95969798△999A9B9C△9D9E9F△/* 90 - 9F */↓
A0A1A2A3A4△A5A6A7A8△A9AAABAC△ADAEAF△/* A0 - AF */↓
B0B1B2B3B4△B5B6B7B8△B9BABBBC△BDBEBF△/* B0 - BF */↓
C0C1C2C3C4△C5C6C7C8△C9CACBCC△CDCECF△/* C0 - CF */↓
D0D1D2D3D4△D5D6D7D8△D9DADBDC△DDDEDF△/* D0 - DF */↓
E0E1E2E3E4△E5E6E7E8△E9EAEBEC△EDEEEF△/* E0 - EF */↓
F0F1F2F3F4△F5F6F7F8△F9FAFBFC△FDFEFF△/* F0 - FF */↓

(凡例)

- ↓ : 改行文字を示す。
- △ : 空白を示す。



## 付録 H Unicode 操作文字コード一覧 (SORT EE)

表 H-1 ASCII 対応 UCS-2・UTF-16 コード一覧

文字	ASCII	UCS-2LE UTF-16LE	UCS-2BE UTF-16BE
SP	0x20	0x2000	0x0020
!	0x21	0x2100	0x0021
"	0x22	0x2200	0x0022
#	0x23	0x2300	0x0023
\$	0x24	0x2400	0x0024
%	0x25	0x2500	0x0025
&	0x26	0x2600	0x0026
'	0x27	0x2700	0x0027
(	0x28	0x2800	0x0028
)	0x29	0x2900	0x0029
*	0x2A	0x2A00	0x002A
+	0x2B	0x2B00	0x002B
,	0x2C	0x2C00	0x002C
-	0x2D	0x2D00	0x002D
.	0x2E	0x2E00	0x002E
/	0x2F	0x2F00	0x002F
0	0x30	0x3000	0x0030
1	0x31	0x3100	0x0031
2	0x32	0x3200	0x0032
3	0x33	0x3300	0x0033
4	0x34	0x3400	0x0034
5	0x35	0x3500	0x0035
6	0x36	0x3600	0x0036
7	0x37	0x3700	0x0037
8	0x38	0x3800	0x0038
9	0x39	0x3900	0x0039
:	0x3A	0x3A00	0x003A
;	0x3B	0x3B00	0x003B
<	0x3C	0x3C00	0x003C
=	0x3D	0x3D00	0x003D
>	0x3E	0x3E00	0x003E

文字	ASCII	UCS-2LE UTF-16LE	UCS-2BE UTF-16BE
?	0x3F	0x3F00	0x003F
@	0x40	0x4000	0x0040
A	0x41	0x4100	0x0041
B	0x42	0x4200	0x0042
C	0x43	0x4300	0x0043
D	0x44	0x4400	0x0044
E	0x45	0x4500	0x0045
F	0x46	0x4600	0x0046
G	0x47	0x4700	0x0047
H	0x48	0x4800	0x0048
I	0x49	0x4900	0x0049
J	0x4A	0x4A00	0x004A
K	0x4B	0x4B00	0x004B
L	0x4C	0x4C00	0x004C
M	0x4D	0x4D00	0x004D
N	0x4E	0x4E00	0x004E
O	0x4F	0x4F00	0x004F
P	0x50	0x5000	0x0050
Q	0x51	0x5100	0x0051
R	0x52	0x5200	0x0052
S	0x53	0x5300	0x0053
T	0x54	0x5400	0x0054
U	0x55	0x5500	0x0055
V	0x56	0x5600	0x0056
W	0x57	0x5700	0x0057
X	0x58	0x5800	0x0058
Y	0x59	0x5900	0x0059
Z	0x5A	0x5A00	0x005A
[	0x5B	0x5B00	0x005B
¥	0x5C	0x5C00	0x005C
]	0x5D	0x5D00	0x005D
^	0x5E	0x5E00	0x005E
_	0x5F	0x5F00	0x005F
`	0x60	0x6000	0x0060

文字	ASCII	UCS-2LE UTF-16LE	UCS-2BE UTF-16BE
a	0x61	0x6100	0x0061
b	0x62	0x6200	0x0062
c	0x63	0x6300	0x0063
d	0x64	0x6400	0x0064
e	0x65	0x6500	0x0065
f	0x66	0x6600	0x0066
g	0x67	0x6700	0x0067
h	0x68	0x6800	0x0068
i	0x69	0x6900	0x0069
j	0x6A	0x6A00	0x006A
k	0x6B	0x6B00	0x006B
l	0x6C	0x6C00	0x006C
m	0x6D	0x6D00	0x006D
n	0x6E	0x6E00	0x006E
o	0x6F	0x6F00	0x006F
p	0x70	0x7000	0x0070
q	0x71	0x7100	0x0071
r	0x72	0x7200	0x0072
s	0x73	0x7300	0x0073
t	0x74	0x7400	0x0074
u	0x75	0x7500	0x0075
v	0x76	0x7600	0x0076
w	0x77	0x7700	0x0077
x	0x78	0x7800	0x0078
y	0x79	0x7900	0x0079
z	0x7A	0x7A00	0x007A
{	0x7B	0x7B00	0x007B
	0x7C	0x7C00	0x007C
}	0x7D	0x7D00	0x007D
~	0x7E	0x7E00	0x007E

表 H-2 ASCII 対応 UCS-4・UTF-32 コード一覧

文字	ASCII	UCS-4LE UTF-32LE	UCS-4BE UTF-32BE
SP	0x20	0x20000000	0x00000020

付録 H Unicode 操作文字コード一覧 (SORT EE)

文字	ASCII	UCS-4LE UTF-32LE	UCS-4BE UTF-32BE
!	0x21	0x21000000	0x00000021
"	0x22	0x22000000	0x00000022
#	0x23	0x23000000	0x00000023
\$	0x24	0x24000000	0x00000024
%	0x25	0x25000000	0x00000025
&	0x26	0x26000000	0x00000026
'	0x27	0x27000000	0x00000027
(	0x28	0x28000000	0x00000028
)	0x29	0x29000000	0x00000029
*	0x2A	0x2A000000	0x0000002A
+	0x2B	0x2B000000	0x0000002B
,	0x2C	0x2C000000	0x0000002C
-	0x2D	0x2D000000	0x0000002D
.	0x2E	0x2E000000	0x0000002E
/	0x2F	0x2F000000	0x0000002F
0	0x30	0x30000000	0x00000030
1	0x31	0x31000000	0x00000031
2	0x32	0x32000000	0x00000032
3	0x33	0x33000000	0x00000033
4	0x34	0x34000000	0x00000034
5	0x35	0x35000000	0x00000035
6	0x36	0x36000000	0x00000036
7	0x37	0x37000000	0x00000037
8	0x38	0x38000000	0x00000038
9	0x39	0x39000000	0x00000039
:	0x3A	0x3A000000	0x0000003A
;	0x3B	0x3B000000	0x0000003B
<	0x3C	0x3C000000	0x0000003C
=	0x3D	0x3D000000	0x0000003D
>	0x3E	0x3E000000	0x0000003E
?	0x3F	0x3F000000	0x0000003F
@	0x40	0x40000000	0x00000040
A	0x41	0x41000000	0x00000041
B	0x42	0x42000000	0x00000042

文字	ASCII	UCS-4LE UTF-32LE	UCS-4BE UTF-32BE
C	0x43	0x43000000	0x00000043
D	0x44	0x44000000	0x00000044
E	0x45	0x45000000	0x00000045
F	0x46	0x46000000	0x00000046
G	0x47	0x47000000	0x00000047
H	0x48	0x48000000	0x00000048
I	0x49	0x49000000	0x00000049
J	0x4A	0x4A000000	0x0000004A
K	0x4B	0x4B000000	0x0000004B
L	0x4C	0x4C000000	0x0000004C
M	0x4D	0x4D000000	0x0000004D
N	0x4E	0x4E000000	0x0000004E
O	0x4F	0x4F000000	0x0000004F
P	0x50	0x50000000	0x00000050
Q	0x51	0x51000000	0x00000051
R	0x52	0x52000000	0x00000052
S	0x53	0x53000000	0x00000053
T	0x54	0x54000000	0x00000054
U	0x55	0x55000000	0x00000055
V	0x56	0x56000000	0x00000056
W	0x57	0x57000000	0x00000057
X	0x58	0x58000000	0x00000058
Y	0x59	0x59000000	0x00000059
Z	0x5A	0x5A000000	0x0000005A
[	0x5B	0x5B000000	0x0000005B
¥	0x5C	0x5C000000	0x0000005C
]	0x5D	0x5D000000	0x0000005D
^	0x5E	0x5E000000	0x0000005E
_	0x5F	0x5F000000	0x0000005F
`	0x60	0x60000000	0x00000060
a	0x61	0x61000000	0x00000061
b	0x62	0x62000000	0x00000062
c	0x63	0x63000000	0x00000063
d	0x64	0x64000000	0x00000064

文字	ASCII	UCS-4LE UTF-32LE	UCS-4BE UTF-32BE
e	0x65	0x65000000	0x00000065
f	0x66	0x66000000	0x00000066
g	0x67	0x67000000	0x00000067
h	0x68	0x68000000	0x00000068
i	0x69	0x69000000	0x00000069
j	0x6A	0x6A000000	0x0000006A
k	0x6B	0x6B000000	0x0000006B
l	0x6C	0x6C000000	0x0000006C
m	0x6D	0x6D000000	0x0000006D
n	0x6E	0x6E000000	0x0000006E
o	0x6F	0x6F000000	0x0000006F
p	0x70	0x70000000	0x00000070
q	0x71	0x71000000	0x00000071
r	0x72	0x72000000	0x00000072
s	0x73	0x73000000	0x00000073
t	0x74	0x74000000	0x00000074
u	0x75	0x75000000	0x00000075
v	0x76	0x76000000	0x00000076
w	0x77	0x77000000	0x00000077
x	0x78	0x78000000	0x00000078
y	0x79	0x79000000	0x00000079
z	0x7A	0x7A000000	0x0000007A
{	0x7B	0x7B000000	0x0000007B
	0x7C	0x7C000000	0x0000007C
}	0x7D	0x7D000000	0x0000007D
~	0x7E	0x7E000000	0x0000007E

表 H-3 ASCII 対応予約文字一覧

項目	指定値 (予約文字)	文字コード				
		ASCII 対応文 字	UCS-2 LE UTF-1 6LE	UCS-2 BE UTF-1 6BE	UCS-4 LE UTF-3 2LE	UCS-4 BE UTF-3 2BE
改行文字	CRLF	0x0d0 a	0x0d0 00a00	0x000 d000a	0x0d0 00000 0a000 000	0x0000 000d0 00000 00000 0a
	LF	0x0a	0x0a00	0x000a	0x0a00 0000	0x0000 000a
区切り文字	TAB	0x09	0x0900	0x0009	0x0900 0000	0x0000 0009
	'△' (スペース)	0x20	0x2000	0x0020	0x2000 0000	0x0000 0020
	';' (コンマ)	0x2c	0x2c00	0x002c	0x2c00 0000	0x0000 002c
	'上記以外の任意の1文字'※1	0xXX	0xXX0 0	0x00X X	0xXX0 00000	0x0000 00XX
引用符	SQUOT (引用符)	0x27	0x2700	0x0027	0x2700 0000	0x0000 0027
	DQUOT (二重引用符)	0x22	0x2200	0x0022	0x2200 0000	0x0000 0022
通貨記号	¥ (円記号)	0x5c	0x5c00	0x005c	0x5c00 0000	0x0000 005c
	円記号以外の任意の1文字※1	0xXX	0xXX0 0	0x00X X	0xXX0 00000	0x0000 00XX
挿入文字	X (空白)	0x20	0x2000	0x0020	0x2000 0000	0x0000 0020
	Z (ゼロ)	0x00	0x0000	0x0000	0x0000 0000	0x0000 0000
	C' 文字列'※2	0xXX	0xXX0 0	0x00X X	0xXX0 00000	0x0000 00XX
	X'16 進数'※3	0xXXX X	変換なし※5	変換なし※5	変換なし※5	変換なし※5
	NULL (CSV ファイルだけ)	—	—	—	—	—

項目	指定値 (予約文字)	文字コード				
		ASCII 対応文字	UCS-2 LE UTF-16 LE	UCS-2 BE UTF-16 BE	UCS-4 LE UTF-32 LE	UCS-4 BE UTF-32 BE
比較条件値	C' 文字列', ※2	0xXX	0xXX00	0x00XX	0xXX0000	0x000000XX
	N'10 進数', ※4	0xXX	0xXX00	0x00XX	0xXX0000	0x000000XX
	X'16 進数', ※3	0xXXX X	変換なし ※5	変換なし ※5	変換なし ※5	変換なし ※5
	D' 日付', ※6	0xXX	0xXX00	0x00XX	0xXX0000	0x000000XX
	H' 日付 T 時刻', ※7	0xXX	0xXX00	0x00XX	0xXX0000	0x000000XX
埋め字	空白	0x20	0x2000	0x0020	0x20000000	0x00000020
	NULL 値	0x00	0x0000	0x0000	0x00000000	0x00000000
	数字「0」	0x30	0x3000	0x0030	0x30000000	0x00000030

注※ 1

「表 H-1 ASCII 対応 UCS-2・UTF-16 コード一覧」および「表 H-2 ASCII 対応 UCS-4・UTF-32 コード一覧」で示す範囲の文字を 1 文字指定してください。

注※ 2

「表 H-1 ASCII 対応 UCS-2・UTF-16 コード一覧」および「表 H-2 ASCII 対応 UCS-4・UTF-32 コード一覧」で示す範囲の文字の文字列を指定してください。

日本語などの「表 H-1 ASCII 対応 UCS-2 コード・UTF-16 コード一覧」および「表 H-2 ASCII 対応 UCS-4 コード一覧・UTF-32 コード」の範囲外の文字については「X'16 進数」の形式で指定してください。

注※ 3

UCS-2 または UCS-4 形式の 2 バイトコードを「0」～「9」, 「A」～「F」で指定してください。

【例】

(UCS-2LE の場合) 「あいう」 → X'423044304630'

(UCS-4LE の場合) 「あいう」 → X'423000004430000046300000'

注※ 4

文字の「0」～「9」, および符号(「+」または「-」)を指定してください。

【例】

N'+123'

注※ 5

指定した値を, そのまま 16 進数値として使用します。



**【例】**

(UCS-2LE の場合) X'30003100' → 「0x3000」, 「0x3100」の 4 バイトの指定値

注※ 6

CSV ファイルの 4 桁年号だけ対応します。

**【例】**

D'20130823' → 2013 年 8 月 23 日

注※ 7

CSV ファイルだけ対応します。

**【例】**

H'20101223T110629498' → 2010 年 12 月 23 日 11 時 6 分 29 秒 498'

## 付録I マルチバイト文字データの扱い

### 付録I.1 文字コード体系とエンコード

入力レコードの文字列データを表現するコード体系とエンコード種別を、次の表に示します。

表 I-1 SORT EE と SORT で対応している文字コード体系

コード体系		文字種別			SORT EE	SORT
		英数字	カタカナ	漢字・外字 など		
ASCII		ASCII	—	—	○※4	○※2
JIS8		ASCII	JIS8	—	○※4	○※2
JIS X 0201	Shift_JIS	JIS8	JIS8	2バイト符号化	○※1	○※2
JIS X 0208	EUC	G0	G2	G1, G3, EUC-HJ 領域	○※1	—
				G1, G3	—	○※2
Unicode	UCS-2	基本ラテン文字 U+0020 ～ U+007F	半角形 / 全角形 U+FF00 ～ FFFD	CJK 統合漢字など	○※3	×
	UCS-4	基本ラテン文字 U+0020 ～ U+007F	半角形 / 全角形 U+FF00 ～ FFFD	CJK 統合漢字, CJK 統合漢字拡張など	○※3	×

(凡例)

- ：対応している
- ×：対応していない
- ：該当しない

注※1

属性コード「c」、「j」および「q」で、1文字1バイトとして扱います。

属性コード「u」および「v」で、1文字nバイトのマルチバイト文字として扱います。

注※2

属性コード「c」、「j」で、1文字1バイトとして扱います。

注※3

属性コード「w」で、1文字2または4バイトのワイド文字として扱います。

属性コード「u」および「v」で、1文字 n バイトのマルチバイト文字として扱います。

注※4

すべての文字属性コードで、1文字1バイトとして扱います。

表 I-2 SORT EE で指定できる文字コードおよびエンコードの種別

文字コード / エンコード 種別※1	説明	コード範囲
ASCII	ASCII コード + JIS8 (JIS 8 ビット単位コードを含みます)	<ul style="list-style-type: none"> <li>1 バイト長 0x00 ~ 0xFF</li> </ul>
SJIS	Shift-JIS (Windows は「Microsoft コード ページ 932」が該当します)	<ul style="list-style-type: none"> <li>1 バイト長 0x00 ~ 0x80, 0xA0 ~ 0xDF, 0xFD ~ 0xFF</li> <li>2 バイト長 第 1 バイト: 0x81 ~ 0x9F, 0xE0 ~ 0xFC 第 2 バイト: 0x40 ~ 0x7E, 0x80 ~ 0xFC</li> </ul>
EUC-JP	日本語 EUC G0: ASCII G1: JIS X 0213 第 1 面 G2: JIS X 0201 カタカナ G3: JIS X 0213 第 2 面 (Windows は「Microsoft コード ページ 51932」が該当します)	<ul style="list-style-type: none"> <li>1 バイト長 (G0) 0x00 ~ 0x7F</li> <li>2 バイト長 (G1) 第 1, 2 バイト: 0xA1 ~ 0xFE</li> <li>2 バイト長 (G2) 第 1 バイト: 0x8E 第 2 バイト: 0xA1 ~ 0xFE</li> <li>3 バイト長 (G3) 第 1 バイト: 0x8F 第 2, 3 バイト: 0xA1 ~ 0xFE</li> </ul>
EUC-HJ※3	日本語 EUC (EUC-HJ) G0: ASCII G1: JIS X 0213 第 1 面 G2: JIS X 0201 カタカナ EUC-HJ 領域	<ul style="list-style-type: none"> <li>1 バイト長 (G0) 0x00 ~ 0x7F</li> <li>2 バイト長 (G1) 第 1, 2 バイト: 0xA1 ~ 0xFE</li> <li>2 バイト長 (G2) 第 1 バイト: 0x8E 第 2 バイト: 0xA1 ~ 0xFE</li> <li>2 バイト長 (EUC-HJ 領域) 第 1 バイト: 0xA1 ~ 0xFE 第 2 バイト: 0x21 ~ 0x7E</li> </ul>

文字コード/ エンコード 種別※1	説明	コード範囲
UTF-8	8ビット単位の Unicode 符号化方式 (Windows は「Microsoft コードページ 65001」が該当します)	<ul style="list-style-type: none"> <li>• 1 バイト長 (U+0000 ~ U+007F) 0x00 ~ 0x7F</li> <li>• 2 バイト長 (U+0080 ~ U+07FF) 第1バイト: 0xC0 ~ 0xDF 第2バイト: 0x80 ~ 0xBF</li> <li>• 3 バイト長 (U+0800 ~ U+D7FF, U+E000 ~ U+FFFF) 第1バイト: 0xE0 ~ 0xEF 第2, 3バイト: 0x80 ~ 0xBF</li> <li>• 4 バイト長 (U+10000 ~ U+10FFFF) 第1バイト: 0xF0 ~ 0xF7 第2 ~ 4バイト: 0x80 ~ 0xBF</li> <li>• 5 バイト長 (未定義) 第1バイト: 0xF8 ~ 0xFB 第2 ~ 5バイト: 0x80 ~ 0xBF</li> <li>• 6 バイト長 (未定義) 第1バイト: 0xFC ~ 0xFD 第2 ~ 6バイト: 0x80 ~ 0xBF</li> </ul>
UCS-2LE	Little endian の 2 オクテット (16bit) 文字コード	<ul style="list-style-type: none"> <li>• 1 文字長 (2 バイト) U+0000 ~ U+D7FF U+E000 ~ U+FFFF</li> </ul>
UCS-2BE	Big endian の 2 オクテット (16bit) 文字コード	<ul style="list-style-type: none"> <li>• サロゲートペア文字は 2 文字として扱う※2</li> </ul>
UCS-4LE	Little endian の 4 オクテット (32bit) 文字コード	<ul style="list-style-type: none"> <li>• 1 文字長 (4 バイト) U+0000 ~ U+D7FF U+E000 ~ U+10FFFF</li> </ul>
UCS-4BE	Big endian の 4 オクテット (32bit) 文字コード	
UTF-16LE	little endian の 16 ビット単位の Unicode 符号化方式 (Windows は「Microsoft コードページ 1200」が該当します)	<ul style="list-style-type: none"> <li>• 1 文字長 (2 バイト) U+0000 ~ U+D7FF U+E000 ~ U+FFFF</li> <li>• 1 文字長 (サロゲートペア 4 バイト) ※2 U+10000 ~ U+10FFFF</li> </ul>
UTF-16BE	big endian の 16 ビット単位の Unicode 符号化方式 (Windows は「Microsoft コードページ 1201」が該当します)	
UTF-32LE	little endian の 32 ビット単位の Unicode 符号化方式 (Windows は「Microsoft コードページ 12000」が該当します)	<ul style="list-style-type: none"> <li>• 1 文字長 (4 バイト) U+0000 ~ U+D7FF U+E000 ~ U+10FFFF</li> </ul>
UTF-32BE	big endian の 32 ビット単位の Unicode 符号化方式 (Windows は「Microsoft コードページ 12001」が該当します)	

注※1

入力データのエンコードは、`codetype` オペランドで指定します。オペランドについては、「5.8.13 `codetype` オペランド」を参照してください。

注※2

サロゲートペア文字は、上位サロゲート「0xD800～0xDBFF」と下位サロゲート「0xDC00～0xDFFF」の範囲を使用し、ペアで1つの字を表現します。

注※3

EUC-HJ は、EUC で外字を2バイトの範囲で使用するために、日立が独自に拡張したコードセットです。

## 付録 I.2 文字列データの扱い

### (1) 文字の長さ

入力データの文字コード（エンコード）の種別によって、マルチバイト文字の1文字の長さは異なります。キー・項目の属性ごとの1文字の長さは、次の表のとおりです。

表 I-3 エンコードと属性コードの関係

エンコード種別	1文字の長さ（単位：バイト）			
	c, q	j	w	u, v
ASCII	1	1	1	1
SJIS	1	1	1	1, 2
EUC-JP	1	1	1	1～3
EUC-HJ	1	1	1	1～2
UTF-8	1	1	1	1～4
UCS-2LE	1	1	2	2
UCS-2BE	1	1	2	2
UTF-16LE	1	1	2	2, 4
UTF-16BE	1	1	2	2, 4
UCS-4LE	1	1	4	4
UCS-4BE	1	1	4	4
UTF-32LE	1	1	4	4
UTF-32BE	1	1	4	4

- 属性コード「w（ワイド文字）」、「u（マルチバイト文字）」および「v（マルチバイト文字）」は、テキストファイルおよびCSVファイルだけで指定できます。
- 属性コード「w」は、UCS-2およびUTF-16を1文字2バイト、UCS-4、およびUTF-32を1文字4バイトで扱う属性です。UCS-2、UCS-4、UTF-16、およびUTF-32以外のエンコードはワイド文字ではないので、文字の最小構成バイトの1文字1バイトとして扱います。

- 属性コード「w」は、サロゲートペア文字を2文字（2文字×2バイト）で扱います。
- 属性コード「u」および「v」は、入力ファイルのエンコード方式に従って、1文字nバイトの可変長で扱います。ただし、ASCII、UCS-2、UCS-4、およびUTF-32は固定長で扱います。
- 属性コード「u」および「v」は、UTF-16のサロゲートペア文字を1文字（4バイト）で扱います。
- 属性コード「u」および「v」は、UnicodeのVS（Variation Selector、異体字セレクタ）を文字数に数えません。
- 属性コード「u」は、UnicodeのIVS（Ideographic Variation Sequence）の異体字を同じ文字として扱います。属性コード「v」は、UnicodeのIVSの異体字を異なる文字として扱います。

## (2) 文字属性コードの特徴

文字データには、1バイトで比較する「c, j, q」、ワイド文字で比較する「w」、マルチバイトの文字単位の比較をする「u」および「v」の3種類の属性コードを用意しています。

「c」が最も処理効率が高く、「j, q」→「w」→「u, v」の順に処理効率が低下します。キー・項目データを構成する文字コードを確認し、最適な属性コードを指定してください。

表 I-4 文字属性コードの特徴

処理速度	属性コード	特徴	最適エンコード
速	c	<ul style="list-style-type: none"> <li>1文字1バイトの最も単純な比較を実施します。</li> <li>マルチバイト文字の文字構成や、エンディアンは考慮しません。</li> </ul>	<ul style="list-style-type: none"> <li>ASCIIコード</li> <li>JIS8単位コード</li> <li>UCS-2BE</li> <li>UCS-4BE</li> <li>UTF-32BE</li> </ul>
↑	j, q	<ul style="list-style-type: none"> <li>1文字1バイトで並び順序を変換して比較します。</li> <li>マルチバイト文字の文字構成や、エンディアンは考慮しません。</li> </ul>	<ul style="list-style-type: none"> <li>ASCIIコード</li> <li>JIS8単位コード</li> </ul>
↓	w	<ul style="list-style-type: none"> <li>UCS-2, UCS-4, UTF-16, および UTF-32 は、2バイトおよび4バイトのワイド文字として比較します。また、little endian のエンコードは、big endian に正規化して比較します。</li> <li>UCS-2, UCS-4, UTF-16 および UTF-32 以外は、1文字1バイトで比較します。属性コード「c」と同じです。</li> </ul>	<ul style="list-style-type: none"> <li>UCS-2LE</li> <li>UCS-4LE</li> <li>UTF-32LE</li> </ul>
遅	u, v	<ul style="list-style-type: none"> <li>ASCII は、1文字1バイトで比較します。属性コード「c」と同じです。</li> <li>SJIS は、1バイト文字を2バイトに引き伸ばし、文字単位（1文字2バイト）の比較を行います。</li> <li>EUC-JP は、1バイト文字および2バイト文字を3バイトに引き伸ばし、文字単位（1文字3バイト）の比較を行います。</li> <li>EUC-HJ は、1バイト文字を2バイトに引き伸ばし、文字単位（1文字2バイト）の比較を行います。</li> <li>UCS-2 と UCS-4 は、2バイトおよび4バイトのワイド文字として比較します。また、little endian のエンコードは、big endian に正規化して比較します。属性コード「w」と同じです。</li> <li>UTF-16 は、Unicode (U+0000 ~ U+10FFFF の値) にデコードし、文字単位（1文字4バイト）の比較を行います。</li> </ul>	<ul style="list-style-type: none"> <li>SJIS</li> <li>EUC-JP</li> <li>EUC-HJ</li> <li>UTF-8</li> <li>UTF-16LE</li> <li>UTF-16BE</li> </ul>

- 属性コード「c」は、文字構成やエンディアンを考慮しないので、ASCII, JIS8 単位コード, UCS-2, UCS-4 などの固定長の文字コードの入力データに最適です。big endian の UCS-2, little endian の UCS-2 と UCS-4 の場合は、キー・項目範囲の入力データが ASCII 対応コード（英数字）だけで構成されていることが条件です。
- 属性コード「j」は、EBCDIK コードの体系に従った並び順にしたい場合に、EBCDIK コード比較オプション機能（-emdopt EBCJ, EBCE）と一緒に使用します。

EBCDIK コード比較オプション機能を指定していない場合は、属性「c」と同じです。

- 属性コード「q」は、JIS8 単位コード体系以外の並び順にしたい場合に、文字順序変更機能 (-altseq パラメタ) と一緒に使用します。
- 属性コード「w」は、ワイド文字のエンディアン正規化を実施するので、little endian の UCS-2, UCS-4, UTF-32 の入力データに最適です。特に、キー・項目範囲の入力データが日本語などの ASCII 対応コード以外で構成されている場合、正しく比較するためには「w」を使用しなければなりません。
- 属性コード「u」と「v」は、文字構成を考慮した文字単位の比較を行うため、SJIS, EUC-JP, EUC-HJ, UTF-8, UTF-16 などの、可変長の文字コードの入力データに最適です。特に UTF-16 のサロゲートペア文字を 1 文字として扱いたい場合は「u」または「v」を使用しなければなりません。
- 属性コード「u」は、Unicode の IVS (Ideographic Variation Sequence) の異体字を同じ文字として扱います。逆に、属性コード「v」は、Unicode の IVS の異体字を異なる文字として扱います。Unicode 以外のエンコードの場合、または Unicode の IVS がない場合、属性コード「u」と「v」の違いはありません。IVS については、「付録 I.2(6) IVS 対応」を参照してください。

### (3) キー・項目の開始位置と長さの指定

#### (a) 開始バイト位置

キー・項目に指定した「+開始バイト位置」から、エンコードに従って文字解析します。

- エンコードの種類が「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」の場合は「2の倍数」を開始バイト位置に指定してください。
- エンコードの種類が「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」の場合は「4の倍数」を開始バイト位置に指定してください。
- エンコードの種類が上記以外の場合は、開始バイト位置の制約はありません。

マルチバイト文字の途中を「+開始バイト位置」と指定した場合、そのマルチバイト文字を空白に置き換える端数処理を実行します。端数処理の詳細については、「付録 I.2(4) マルチバイト文字の端数処理」を参照してください。また、エンコードが「SJIS」, 「EUC-JP」および「EUC-HJ」の場合は、正しく文字解析ができない場合があるので注意してください。

#### (b) バイト長

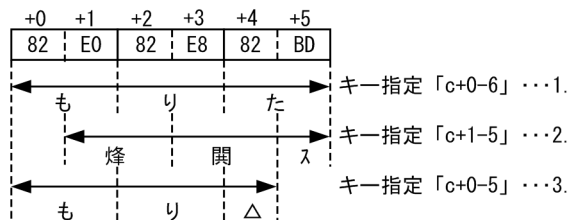
指定された「-バイト長」をエンコードに従って文字解析します。

- エンコードの種類が「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」または「UTF-16BE」の場合は「2の倍数」をバイト長に指定してください。
- エンコードの種類が「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」または「UTF-32BE」の場合は「4の倍数」をバイト長に指定してください。
- エンコードの種類が上記以外の場合は、バイト長の制約はありません。



マルチバイト文字の途中までを「 $n$ バイト長」で範囲指定した場合、そのマルチバイト文字を空白に置き換える端数処理を実行します。端数処理の詳細については、「付録 I.2(4) マルチバイト文字の端数処理」を参照してください。また、エンコードが「SJIS」、「EUC-JP」および「EUC-HJ」の場合は、正しく文字解析ができない場合があるので注意してください。

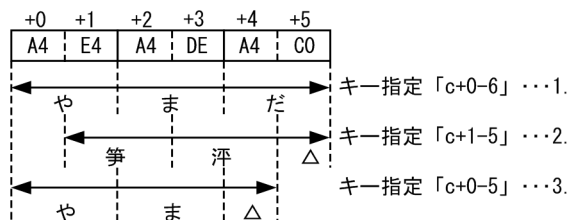
図 I-1 開始バイト位置によって文字の解釈が異なる例 (Shift\_JIS)



(凡例) △ : 空白を示す。

1. 文字「も (0x82E0)」の先頭からキーとして指定。
2. 文字「も」の途中からキーとして指定。第2バイトの「0xE0」を Shift\_JIS の第1バイトと解釈し、1バイトずつずれて解析されるので、正しく解釈できない。
3. 文字「た (0x82BD)」の途中までをキーとして指定。第1バイトの「0x82」は ASCII 文字以外なので空白 (0x20) を仮定する。

図 I-2 開始バイト位置によって文字の解釈が異なる例 (EUC-JP)



(凡例) △ : 空白を示す。

1. 文字「や (0xA4E4)」の先頭からキーとして指定。
2. 文字「や」の途中からキーとして指定。第2バイトの「0xE4」を EUC-JP の第1バイトと解釈し、1バイトずつずれて解析されるので、正しく解釈できない。5バイト目の「0xC0」は ASCII 文字以外なので空白 (0x20) を仮定する。
3. 文字「だ (0xA4C0)」の途中までをキーとして指定。第1バイトの「0xA4」は ASCII 文字以外なので空白 (0x20) を仮定する。

(c) 文字数

テキストファイルと CSV ファイルにだけ指定できる属性コード「u」および「v」では、キー・項目の長さを「文字数」で指定します。文字数の最小値は「1」文字で、最大「4,096」文字を指定できます。

なお、属性コード「u」および「v」では、キー・項目の長さを「バイト長」で指定できません。

(4) マルチバイト文字の端数処理

(a) 文字の比較単位

属性コード「c, j, q」は、指定したエンコードに関係なく1バイト単位で比較します。

ASCII, SJIS, EUC-JP, EUC-HJ および UTF-8 に指定した属性コード「w」は、1バイト単位で比較します。

属性コード「u, v」と、UCS-2, UCS-4, UTF-16 および UTF-32 に指定した属性コード「w」は、文字単位で比較します。

表 I-5 文字の比較単位

属性コード	エンコード種別	比較単位	備考
c, j, q	すべてのエンコード	バイト	—
w	ASCII, SJIS, EUC-JP, EUC-HJ, または UTF-8	バイト	—
	UCS-2LE, UCS-2BE, UTF-16LE, または UTF-16BE	文字	UTF-16 のサロゲートペア文字は 2 文字として扱う
	UCS-4LE, UCS-4BE, UTF-32LE, または UTF-32BE	文字	—
u, v	すべてのエンコード	文字	UTF-16 のサロゲートペア文字は 1 文字として扱う

(凡例)

— : 該当なし

(b) 文字単位比較の端数処理

文字単位の比較では、次の表に示す最小構成バイトから比較文字数が決定されます。

表 I-6 1 文字の最小構成バイト

エンコード種別	最小構成バイト
ASCII, SJIS, EUC-JP, EUC-HJ または UTF-8	1

エンコード種別	最小構成バイト
UCS-2LE, UCS-2BE, UTF-16LE または UTF-16BE	2
UCS-4LE, UCS-4BE, UTF-32LE または UTF-32BE	4

例えば、エンコード「UTF-8」でキー長 5 バイトの属性コード「w」を指定した場合、UTF-8 の最小構成バイトは「1 バイト」なので「 $5 \div 1 = 5$  文字」の比較となります。

なお、「UCS-2」、「UCS-4」、「UTF-16」および「UTF-32」では、最小構成バイトの倍数でキー・項目の長さを指定しなければなりません。

エンコードが「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」のときは、「2 の倍数」の長さを指定します。

エンコードが「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」のときは、「4 の倍数」の長さを指定します。

例えば、エンコード「UCS-2LE」でキー長 8 バイトの属性コード「w」を指定した場合、UCS-2LE の最小構成バイトは「2 バイト」なので「 $8 \div 2 = 4$  文字」の比較となります。

文字単位の比較では、比較文字数に満たないデータを入力した場合、右側に空白を加えて比較文字数に調整します。また、比較文字数を超えるデータを入力した場合、右側のあふれ部分を入力しません。

### (c) バイト単位比較の端数処理

バイト単位の比較では、キー・項目の範囲がマルチバイト文字の途中であった場合、途中の文字を空白文字に置き換える端数処理を実行します。なお、この端数処理によって入力データが変更されることはありません。

属性コードによる端数処理の有無を「表 I-7」に、また仮定する文字の種類を「表 I-8」に示します。

表 I-7 属性コードによる端数処理の有無

属性コード	途中文字の端数処理の有無	
	テキストファイル	CSV ファイル※
c	×	○
q	×	×
j	×	×
w	×	○

(凡例)

- ：端数処理を実行する
- ×：端数処理を実行しない

注※

CSV ファイルの端数処理については、「付録 B.2 文字列データの端数処理」もあわせて参照してください。

表 I-8 仮定する文字の種類

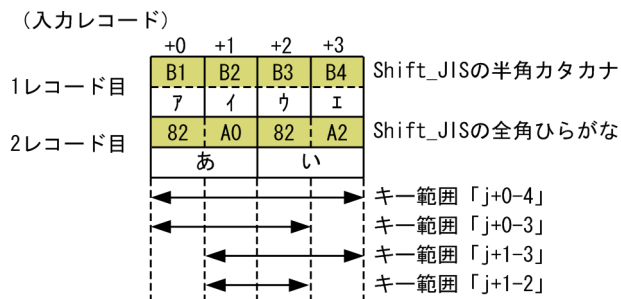
エンコード種別	属性コード		
	q, j	c, w	
		CSV ファイル	CSV ファイル以外
ASCII SJIS EUC-JP EUC-HJ UTF-8	文字の置き換えなし	空白 (0x20)	文字の置き換えなし
UCS-2LE UCS-2BE UCS-4LE UCS-4BE UTF-16LE UTF-16BE UTF-32LE UTF-32BE	—	—	—

(凡例)

—：最小構成バイト単位 (2 または 4 バイト) の比較となるので該当しません

端数処理の例を次に示します。

図 I-3 バイト単位比較の端数処理の例 (codetype=SJIS の属性コード「j」指定)



(端数処理)

項番	キー指定値	1レコード目				2レコード目			
1	j+0-4	B1	B2	B3	B4	82	A0	82	A2

項番	キー指定値	1レコード目				2レコード目			
2	j+0-3	B1	B2	B3	—	82	A0	82	—
3	j+1-3	B2	B3	B4	—	A0	82	A2	—
4	j+1-2	B2	B3	—	—	A0	82	—	—

(凡例)

—：該当なし

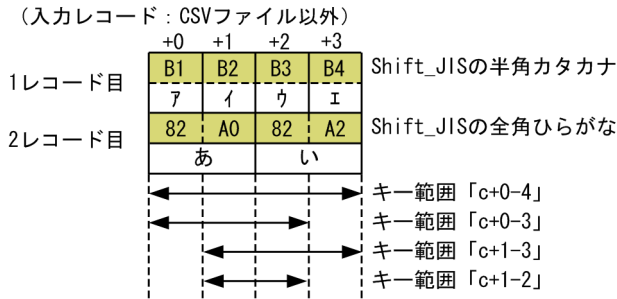
属性コード「j」は端数処理を実行しません。

項番2の2レコード目のキー末尾は上位バイトの「0x82」が残ります。

項番3の2レコード目のキー先頭は下位バイトの「0xA0」が残ります。

項番4の2レコード目のキー先頭は下位バイトの「0xA0」が、キー末尾は上位バイトの「0x82」が残ります。

図 I-4 バイト単位比較の端数処理の例 (CSV ファイル以外の codetype=SJIS の属性コード「c」指定)



(端数処理)

項番	キー指定値	1レコード目				2レコード目			
1	c+0-4	B1	B2	B3	B4	82	A0	82	A2
2	c+0-3	B1	B2	B3	—	82	A0	82	—
3	c+1-3	B2	B3	B4	—	A0	82	A2	—
4	c+1-2	B2	B3	—	—	A0	82	—	—

(凡例)

—：該当なし

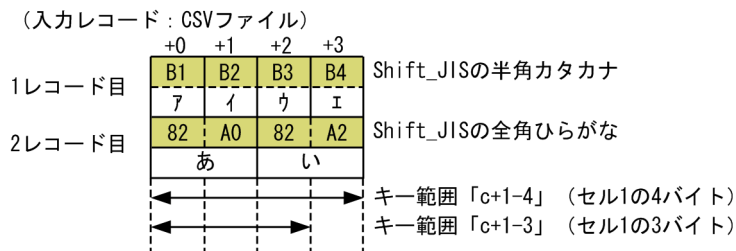
CSV ファイル以外の属性コード「c」は端数処理を実行しません。属性コード「j」と同じ結果となります。詳細については、「図 I-1 開始バイト位置によって文字の解釈が異なる例 (Shift\_JIS)」を参照してください。

項番2の2レコード目のキー末尾は上位バイトの「0x82」が残ります。

項番3の2レコード目のキー先頭は下位バイトの「0xA0」が残ります。

項番4の2レコード目のキー先頭は下位バイトの「0xA0」が、キー末尾は上位バイトの「0x82」が残ります。

図 I-5 バイト単位比較の端数処理の例 (CSV ファイルの codetype=SJIS の属性コード「c」指定)



(端数処理)

項番	キー指定値	1レコード目				2レコード目			
1	c+0-4	B1	B2	B3	B4	82	A0	82	A2
2	c+0-3	B1	B2	B3	—	82	A0	(20)	—

(凡例)

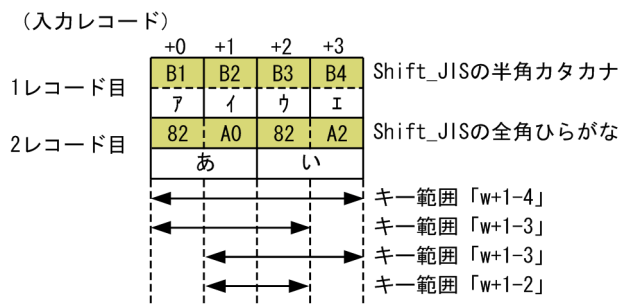
—: 該当なし

CSV ファイルの属性コード「c」は端数処理を実行します。途中文字を空白 (0x20) に置き換えます。

なお、CSV のセルデータは、左端をそろえて入力するので、CSV キー・項目範囲の左側が途中文字となることはありません。

項番 2 の 2レコード目のキー末尾は 2文字目の途中なので、2文字目を空白に置き換えます。

図 I-6 バイト単位比較の端数処理の例 (codetype=SJIS の属性コード「w」指定)



(端数処理)

項番	キー指定値	1レコード目				2レコード目			
1	w+0-4	B1	B2	B3	B4	82	A0	82	A2
2	w+0-3	B1	B2	B3	—	82	A0	(20)	—
3	w+1-3	B2	B3	B4	—	(20)	82	A2	—

項番	キー指定値	1レコード目				2レコード目			
		B2	B3	—	—	(20)	(20)	—	—
4	w+1~2	B2	B3	—	—	(20)	(20)	—	—

(凡例)

— : 該当なし

SJIS に対する属性コード「w」は端数処理を実行します。途中文字を空白 (0x20) に置き換えます。

項番 2 の 2レコード目のキー末尾は 2文字目の途中なので、2文字目を空白に置き換えます。

項番 3 の 2レコード目のキー先頭は 1文字目の途中なので、1文字目を空白に置き換えます。

項番 4 の 2レコード目のキー先頭と末尾はどちらも文字の途中なので、1文字目と 2文字目の両方を空白に置き換えます。

## (5) ソート結果例

入力データのエンコードと属性コードによって、ソート結果が異なる例を次に示します。

図 I-7 UTF-16 little endian の例

(入力レコード)	+0	+2	+4	+6	+8	+12	
1レコード目	7400	6F00	6200	6900	7500	6F00	英小文字 (ASCII対応文字)
	t	o	b	i	u	o	
2レコード目	84FF	8BFF	9EFF	73FF	75FF		半角カタカナ
	ト	ヒ	ウ	オ			
3レコード目	DB98	5A9B					第一水準漢字
	飛	魚					
4レコード目	67DB	49DE					第四水準漢字 (サロゲートペア文字)
	鰯						

属性コード「c, j, w」のソートキー範囲

(ソート結果)

【例1】属性コード「c」のソート結果

入力レコード番号	入力キーデータ				ソート結果
	+0	+1	+2	+3	
1	74	00	6F	00	2
2	84	FF	8B	FF	3
3	DB	98	5A	9B	4
4	67	D8	49	DE	1

- キー「c+0~4 (キー位置 0 バイト目, キー長 4 バイト)」を指定した場合のソート例です。
- 属性コード「c」はバイト単位で比較します。
- 入力データのエンコードに「UCS-2LE」または「UTF-16LE」のどちらを指定しても、ソート結果は変わりません。

【例2】属性コード「j」のソート結果

入力レコード番号	入力キーデータ				EBCDIK 変換※				ソート結果
	+0	+1	+2	+3	+0	+1	+2	+3	
1	74	00	6F	00	8B	00	75	00	2
2	84	FF	8B	FF	FF	FF	FF	FF	4
3	DB	98	5A	9B	BB	FF	E9	FF	3
4	67	D8	49	DE	67	AE	C9	BE	1

注※

EBCDIK コード比較機能「-cmdopt EBCJ」オプションの指定あり。

- キー「j+0-4（キー位置0バイト目、キー長4バイト）」を指定した場合のソート例です。
- 属性コード「j」はバイト単位で比較します。
- 正しく EBCDIK 変換できるのは、ASCII 対応文字の1番レコードだけです。
- 入力データのエンコードに「UCS-2LE」または「UTF-16LE」のどちらを指定しても、ソート結果は変わりません。

【例3】属性コード「w」のソート結果

入力レコード番号	入力キーデータ		Unicode 比較		ソート結果
	+0 ~ 1	+2 ~ 3	1 文字	2 文字	
1	7400	6F00	U+0074	U+006F	1
2	84FF	8BFF	U+FF84	U+FF8B	4
3	DB98	5A9B	U+98DB	U+9B5A	2
4	67D8	49DE	U+D867	U+DE49	3

- キー「w+0-4（キー位置0バイト目、キー長4バイト）」を指定した場合のソート例です。
- 属性コード「w」は文字単位で比較します。
- UCS-2 の最小構成バイトは2バイトなので、比較文字数は「4 ÷ 2 = 2 文字」です。
- UCS-2 では、4 番レコードのサロゲートペア文字「鯨」を2文字として扱います。
- 入力データのエンコードに「UCS-2LE」または「UTF-16LE」のどちらを指定しても、ソート結果は変わりません。

【例4】属性コード「u」のソート結果 (codetype=UCS-2LE 指定)

入力レコード番号	入力キーデータ				Unicode 比較				ソート結果
	1 文字	2 文字	3 文字	4 文字	1 文字	2 文字	3 文字	4 文字	
1	7400	6F00	6200	6900	U+0074	U+006F	U+0062	U+0069	1
2	84FF	8BFF	9EFF	73FF	U+FF84	U+FF8B	U+FF9E	U+FF73	4



入力レコード番号	入力キーデータ				Unicode 比較				ソート結果
	1文字	2文字	3文字	4文字	1文字	2文字	3文字	4文字	
3	DB98	5A9B	—	—	U+98D B	U+9B5 A	(U+002 0)	(U+002 0)	2
4	67D8	49DE	—	—	U+D86 7	U+DE4 9	(U+002 0)	(U+002 0)	3

(凡例)

—：該当しません。

- キー「u+0-4 (キー位置0バイト目, キー長4文字)」を指定した場合のソート例です。
- 属性コード「u」は文字単位で比較します。
- UCS-2 では、4番レコードのサロゲートペア文字「𪗇」を2文字として扱います。
- 3番と4番レコードのキーデータは2文字分のデータが入力されたと見なし、不足分に空白(U+0020)を仮定します。
- 属性コード「w」と同じソート結果となります。

【例5】属性コード「u」のソート結果 (codetype=UTF-16LE 指定)

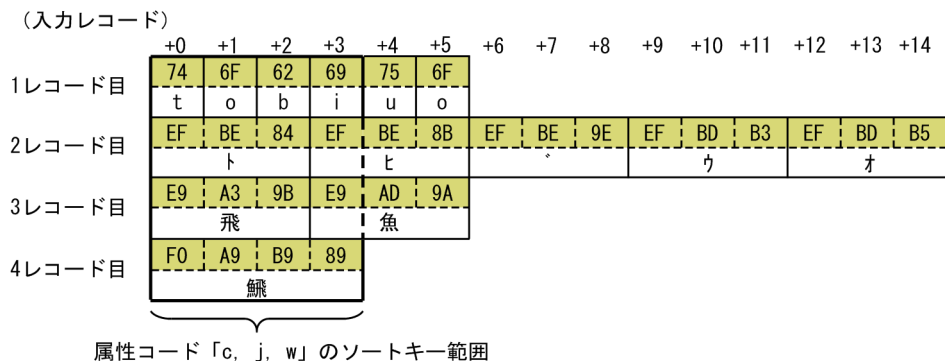
入力レコード番号	入力キーデータ				Unicode 比較				ソート結果
	1文字	2文字	3文字	4文字	1文字	2文字	3文字	4文字	
1	7400	6F00	6200	6900	U+0074	U+006F	U+0062	U+0069	1
2	84FF	8BFF	9EFF	73FF	U+FF8 4	U+FF8 B	U+FF9 E	U+FF7 3	3
3	DB98	5A9B	—	—	U+98D B	U+9B5 A	(U+002 0)	(U+002 0)	2
4	67D8 49DE	—	—	—	U+29E 49	(U+002 0)	(U+002 0)	(U+002 0)	4

(凡例)

—：該当しません。

- キー「u+0-4 (キー位置0バイト目, キー長4文字)」を指定した場合のソート例です。
- 属性コード「u」は文字単位で比較します。
- UTF-16 では、4番レコードのサロゲートペア文字「𪗇」を1文字として扱います。
- 3番レコードのキーデータは2文字、4番レコードのキーデータは1文字分のデータが入力されたと見なし、不足分に空白(0x20)を仮定します。

図 I-8 UTF-8 文字 (codetype=UTF-8) の例



(ソート結果)

【例1】属性コード「c」のソート結果

入力レコード番号	入力キーデータ				ソート結果
	+0	+1	+2	+3	
1	74	6F	62	69	1
2	EF	BE	84	EF	3
3	E9	A3	9B	E9	2
4	F0	A9	B9	89	4

- キー「c+0-4 (キー位置0バイト目, キー長4バイト)」を指定した場合のソート例です。
- 属性コード「c」はバイト単位で比較します。

【例2】属性コード「j」のソート結果

入力レコード番号	入力キーデータ				EBCDIK 変換※				ソート結果
	+0	+1	+2	+3	+0	+1	+2	+3	
1	74	6F	62	69	8B	75	62	69	1
2	EF	BE	84	EF	FF	8F	FF	FF	4
3	E9	A3	9B	E9	FF	43	FF	FF	2
4	F0	A9	B9	89	FF	49	89	FF	3

注※

EBCDIK コード比較機能「-cmdopt EBCJ」オプションの指定あり。

- キー「j+0-4 (キー位置0バイト目, キー長4バイト)」を指定した場合のソート例です。
- 属性コード「j」はバイト単位で比較します。
- 正しく EBCDIK 変換できるのは, ASCII 対応文字の1番レコードだけです。

【例3】属性コード「w」のソート結果

入力レコード番号	入力キーデータ				ソート結果
	+0	+1	+2	+3	
1	74	6F	62	69	1
2	EF	BE	84	(20)	3
3	E9	A3	9B	(20)	2
4	F0	A9	B9	89	4

- キー「w+0-4 (キー位置 0 バイト目, キー長 4 バイト)」を指定した場合のソート例です。
- UTF-8 に指定した属性コード「w」はバイト単位で比較します。また、途中文字を空白 (0x20) に置き換える端数処理を実行します。
- 2 番レコードおよび 3 番レコードのキーデータは 2 文字目の途中までなので、2 文字目を空白に置き換えます。

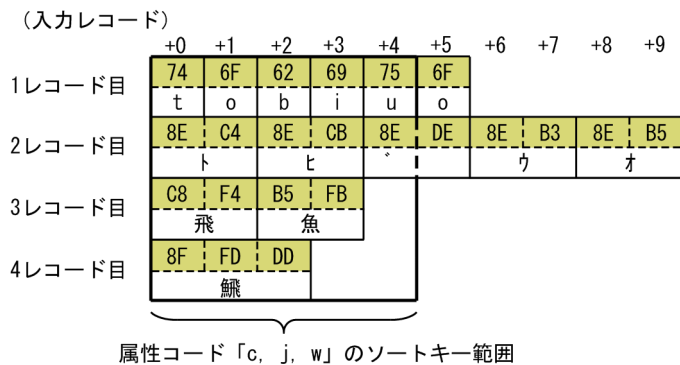
【例 4】属性コード「u」のソート結果

入力レコード番号	入力キーデータ				Unicode 比較				ソート結果
	1 文字	2 文字	3 文字	4 文字	1 文字	2 文字	3 文字	4 文字	
1	74	6F	62	69	U+0074	U+006F	U+0062	U+0069	1
2	EFB E84	EFB E8B	EFB E9E	EFBDB3	U+FF84	U+FF8B	U+FF9E	U+FF73	3
3	E9A3 9B	E9A D9A	—	—	U+98DB	U+9B5A	(U+0020)	(U+0020)	2
4	F0A9 B989	—	—	—	U+29E49	(U+0020)	(U+0020)	(U+0020)	4

(凡例) — : 該当しません。

- キー「u+0-4 (キー位置 0 バイト目, キー長 4 文字)」を指定した場合のソート例です。
- 3 番レコードのキーデータは 2 文字, 4 番レコードのキーデータは 1 文字分のデータが入力された見なし, 不足分に空白 (U+0020) を仮定します。

図 I-9 日本語 EUC 文字 (codetype=EUC-JP) の例



(ソート結果)

【例1】属性コード「c」のソート結果

入力レコード番号	入力キーデータ					ソート結果
	+0	+1	+2	+3	+4	
1	74	6F	62	69	75	1
2	8E	C4	8E	CB	8E	2
3	C8	F4	B5	FB	(00)※	4
4	8F	FD	DD	(00)※	(00)※	3

注※

キー位置チェック緩和機能「-cmdopt POSNOCHK」オプションの指定あり。

- キー「c+0-5 (キー位置0バイト目, キー長5バイト)」を指定した場合のソート例です。
- 属性コード「c」はバイト単位で比較します。
- キー位置チェック緩和機能の指定があると、キーデータが不足していてもエラーにしません。3番レコードおよび4番レコードの不足バイトに NULL 値 (0x00) を仮定します。

【例2】属性コード「j」のソート結果

入力レコード番号	入力キーデータ					EBCDIK 変換※2					ソート結果
	+0	+1	+2	+3	+4	+0	+1	+2	+3	+4	
1	74	6F	62	69	75	8B	75	62	69	9B	1
2	8E	C4	8E	CB	8E	FF	95	FF	9E	FF	3
3	C8	F4	B5	FB	(00)※1	99	FF	85	FF	00	2
4	8F	FD	DD	(00)※1	(00)※1	FF	FF	BD	00	00	4

## 注※1

キー位置チェック緩和機能「-cmdopt POSNOCHK」オプションの指定あり。

## 注※2

EBCDIK コード比較機能「-cmdopt EBCJ」オプションの指定あり。

- キー「j+0-5（キー位置0バイト目，キー長5バイト）」を指定した場合のソート例です。
- 属性コード「j」はバイト単位で比較します。
- キー位置チェック緩和機能の指定があると，キーデータが不足していてもエラーにしません。3番レコードおよび4番レコードの不足分に NULL 値（0x00）を仮定します。
- 正しく EBCDIK 変換できるのは，ASCII 対応文字の1番レコードだけです。

## 【例3】属性コード「w」のソート結果

入力レコード番号	入力キーデータ					ソート結果
	+0	+1	+2	+3	+4	
1	74	6F	62	69	75	1
2	8E	C4	8E	CB	(20)	2
3	C8	F4	B5	FB	(00)※	4
4	8F	FD	DD	(00)※	(00)※	3

## 注※

キー位置チェック緩和機能「-cmdopt POSNOCHK」オプションの指定あり。

- キー「w+0-5（キー位置0バイト目，キー長5バイト）」を指定した場合のソート例です。
- EUC-JP に指定した属性コード「w」はバイト単位で比較します。また，途中文字を空白（0x20）に置き換える端数処理を実行します。
- 2番レコードのキーデータは3文字目の途中までなので，3文字目を空白に置き換えます。
- キー位置チェック緩和機能の指定があると，キーデータが不足していてもエラーにしません。3番レコードおよび4番レコードの不足分に NULL 値（0x00）を仮定します。

## 【例4】属性コード「u」のソート結果

入力レコード番号	入力キーデータ					EUC-JP 比較					ソート結果
	1文字	2文字	3文字	4文字	5文字	1文字	2文字	3文字	4文字	5文字	
1	74	6F	62	69	75	74	6F	62	69	75	1
2	8EC4	8ECB	8EDE	8EB3	8EB5	8EC4	8ECB	8EDE	8EB3	8EB5	2
3	C8F4	B5FB	—	—	—	C8F4	B5FB	(20)	(20)	(20)	3
4	8FFDDD	—	—	—	—	8FFDD	(20)	(20)	(20)	(20)	4

（凡例）—：該当しません。

- キー「u+0-5 (キー位置 0 バイト目, キー長 5 文字)」を指定した場合のソート例です。
- 属性コード「u」は文字単位で比較します。
- 3 番レコードのキーデータは 2 文字, 4 番レコードのキーデータは 1 文字分のデータを入力されたと思なし, 不足分に空白 (0x20) を仮定します。

## (6) IVS 対応

### (a) IVS とは

同じ意味の漢字文字を複数の字形で表現できる仕組みで, ISO/IEC 10646 で規定されています。漢字を表すコードの直後に VS (Variation Selector, 字形選択子) と呼ばれるコードを付加することで, その漢字の「異体字」を表現する方法です。

VS の範囲としてサロゲートエリア内の「U+E0100 ~ U+E01EF」が割り当てられており, UTF-16 ではサロゲートペアで VS が表現されます。漢字を表すコードを「基底文字 (Base Character)」と呼び, 基底文字に付随する VS とあわせて「U+hhhhh;U+E01hh」と表現します。

同一文字コードの漢字でも, VS によってグリフ (印刷・画面表示イメージ) が異なる例を次の図に示します。

図 I-10 VS (Variation Selector) なし「かつしかく」

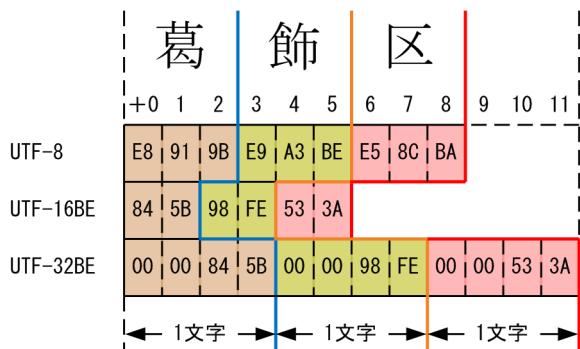


図 I-11 VS (Variation Selector) あり「かつらぎし」

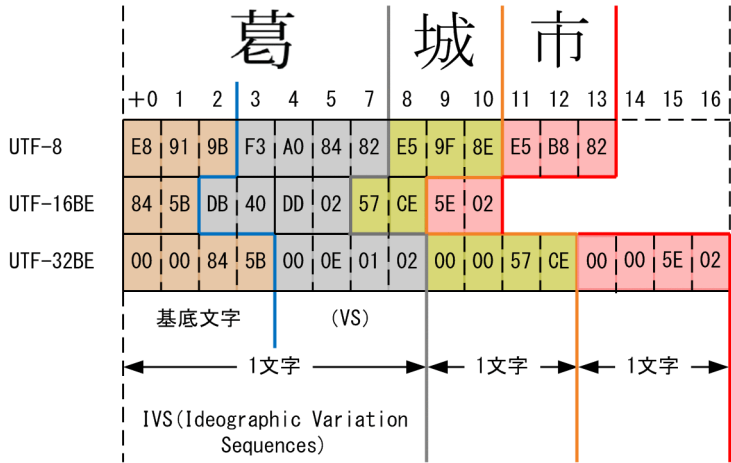


図 I-11 にある「葛」 + VS (U+E0102) を並べた文字符号列を「IVS」と呼びます。この文字は「葛」と同一文字であり、ほかのグリフも存在します (図 I-12 参照)。

図 I-12 U+845B の VS 例 (IVD 登録字形)

基底文字
U+845B
葛

グリフ種別	グリフID	VS	字形
Adobe-Japan1	CID+1481 (Adobe-Japan1)	U+E0100	葛
	CID+7652 (Adobe-Japan1)	U+E0101	葛
Hanyo-Denshi	JA1975 (JIS X 0208:1990)	U+E0102	葛
	FT1769 (FDPC追加, 78JIS字形追加)	U+E0103	葛
	JTB9D7 (住基統一文字)	U+E0104	葛
	JTB9D8 (住基統一文字)	U+E0105	葛
	JTB9D6 (住基統一文字)	U+E0106	葛
	KS352870 (戸籍統一文字)	U+E0107	葛
	JTC0F7 (住基統一文字)	U+E0108	葛
	TK01079040 (登記固有文字)	U+E0109	葛

(b) SORT EE での IVS の扱い

VS (Variation Selector) は、字体に“バリエーション”をつけるために付随するコードであり、文字データの本質部分ではありません。

文字の意味で分けた場合、前述の図 I-10 「葛」と図 I-11 「葛」は同じ文字を指します。また、文字の外観で分けた場合、「葛」と「葛」は異なる文字となります。



異体字を同字と扱う場合は属性コード「u」を指定します。また、異字と扱う場合は属性コード「v」を指定します。次の図に、属性コードによる比較結果の違いを示します。

図 I-13 異体字の比較例

1文字目	2文字目	3文字目	並び (昇順)	
			属性「u」	属性「v」
葛 U+845B (VSなし)	飾 U+98FE (VSなし)	区 U+533A (VSなし)	2	1
葛 U+845B;U+E0102	城 U+570E (VSなし)	市 U+5E02 (VSなし)	1	2
葛 U+845B;U+E0103	飾 U+98FE (VSなし)	区 U+533A (VSなし)	3	3

## 付録 I.3 数値データの扱い

### (1) ワイド文字対応の属性コード

UCS-2, UCS-4, UTF-16 および UTF-32 では、10 進数字の「0」～「9」、外部 10 進数で使用するゾーン部の英小文字「p」～「y」、符号文字の「-」と「+」、および CSV の固定小数点数で使用する通貨記号、小数点、コンマなどの桁修飾文字が、それぞれ 2 バイトまたは 4 バイトのワイド文字で表現されます。

図 I-14 UCS-2LE で表現した外部 10 進数のデータ例

+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
31	00	32	00	33	00	34	00	75	00
1	2	3	4	u					

値: -12345

外部10進数のデータ形式: '1234u'

ワイド文字表現の数字に対応している数値属性コードを、次の表に示します。

表 I-9 ワイド文字対応の数値属性コード

エンコード 種別	1文字の長さ (単位: バイト)	属性コード										
		2進数					10進数					数字
		n	m	a	b	f	p	z	x	s	t	d
UCS-2LE	2	×	×	×	×	×	×	○	○	○	○	○
UCS-2BE	2	×	×	×	×	×	×	○	○	○	○	○
UCS-4LE	4	×	×	×	×	×	×	○	○	○	○	○

エンコード種別	1文字の長さ (単位: バイト)	属性コード										
		2進数					10進数					数字
		n	m	a	b	f	p	z	x	s	t	d
UCS-4BE	4	×	×	×	×	×	×	○	○	○	○	○
UTF-16LE	2	×	×	×	×	×	×	○	○	○	○	○
UTF-16BE	2	×	×	×	×	×	×	○	○	○	○	○
UTF-32LE	4	×	×	×	×	×	×	○	○	○	○	○
UTF-32BE	4	×	×	×	×	×	×	○	○	○	○	○

(凡例)

○: ワイド文字に対応している

×: ワイド文字で対応していない (ワイド文字で表現できない数値)

10進数のデータ形式の詳細については、「3.2 キー属性と形式」の「SORT EE 固有のデータ形式」を参照してください。

## (2) キー・項目の開始位置と長さの指定

### (a) 開始バイト位置

キー・項目に指定した「+開始バイト位置」から、エンコードに従って文字 (数字) 解析します。

エンコードの種類が「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、または「UTF-16BE」の場合は「2の倍数」を開始バイト位置に指定してください。

エンコードの種類が「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、または「UTF-32BE」の場合は「4の倍数」を開始バイト位置に指定してください。

### (b) 桁数

キー・項目に指定した「-桁数」をエンコードに従って文字 (数字) 解析します。指定する値はバイト長ではなく、文字 (数字や符号) の桁数です。

SORT EE では、指定された桁数を次の表に示すバイト長に換算して操作します。

表 I-10 エンコードと桁数の関係 (10進数)

エンコード種別	1桁のバイト長	備考 (例)
ASCII	1	キー「z+0-4」の長さは、 $4 \times 1 = 4$ バイトです。
SJIS		
EUC-JP		
EUC-HJ		
UTF-8		

エンコード種別	1桁のバイト長	備考(例)
UCS-2LE	2	キー「z+0-4」の長さは、 $4 \times 2 = 8$ バイトです。
UCS-2BE		
UTF-16LE		
UTF-16BE		
UCS-4LE	4	キー「z+0-4」の長さは、 $4 \times 4 = 16$ バイトです。
UCS-4BE		
UTF-32LE		
UTF-32BE		

## 付録 I.4 日付データの扱い

### (1) ワイド文字対応

「UCS-2」、「UCS-4」、「UTF-16」および「UTF-32」では、10進数字の「0」～「9」、およびCSVの日付で使用するスラント、ハイフン、ピリオドなどの桁修飾文字が、それぞれ2バイトまたは4バイトのワイド文字で表現されます。

日付のデータ形式の詳細については、「3.2 キー属性と形式」の「SORT EE 固有のデータ形式」を参照してください。

### (2) キー・項目の開始位置と長さの指定

#### (a) 開始バイト位置

キー・項目に指定した「+開始バイト位置」から、エンコードに従って文字(数字)解析します。

エンコードの種類が「UCS-2LE」、「UCS-2BE」、「UTF-16LE」または「UTF-16BE」の場合は「2の倍数」を開始バイト位置に指定してください。

エンコードの種類が「UCS-4LE」、「UCS-4BE」、「UTF-32LE」または「UTF-32BE」の場合は「4の倍数」を開始バイト位置に指定してください。

#### (b) 桁数

キー・項目に指定した「-桁数」をエンコードに従って文字(数字)解析します。指定する値はバイト長ではなく、文字の桁数(CSVファイルの日付属性の場合は「年月日」の桁数)です。

SORT EE では、指定された桁数を次の表に示すバイト長に換算して操作します。

表 I-11 エンコードと桁数の関係 (日付)

エンコード種別	1 桁のバイト長	備考 (例)
ASCII	1	キー「g+0-4」の長さは、 $4 \times 1 = 4$ バイトです。
SJIS		
EUC-JP		
EUC-HJ		
UTF-8		
UCS-2LE	2	キー「g+0-4」の長さは、 $4 \times 2 = 8$ バイトです。
UCS-2BE		
UTF-16LE		
UTF-16BE		
UCS-4LE	4	キー「g+0-4」の長さは、 $4 \times 4 = 16$ バイトです。
UCS-4BE		
UTF-32LE		
UTF-32BE		

## 付録 J 日付時刻データの形式 (SORT EE の機能)

日付時刻データについて説明します。

キー・項目属性「h」の日付時刻は、日付と時刻を組み合わせた数字データです。

### データ形式

データ形式を次に示します。

日付	T   空白	時刻	空白	12時間制表記
4~8桁	1×n <sup>h</sup> 1t	2~9桁	1×n <sup>h</sup> 1t	(2~4)×n <sup>h</sup> 1t

(日付)

YYYY	MM	DD
4×n <sup>h</sup> 1t	2×n <sup>h</sup> 1t	2×n <sup>h</sup> 1t

(時刻)

hh	mm	ss	fff
2×n <sup>h</sup> 1t	2×n <sup>h</sup> 1t	2×n <sup>h</sup> 1t	3×n <sup>h</sup> 1t

- 日付時刻のデータ形式は、日付、時刻、12時間制表記の順になります。
- 日付の Y (年)、M (月)、D (日)、および時刻の h (時)、m (分)、s (秒)、f (ミリ秒) には、それぞれ 0～9 の数字を指定します。各項目の指定可能な桁数と値は次のとおりです。

書式	項目	桁	値	
日付	YYYY	年 (西暦年 4 桁)	4	<ul style="list-style-type: none"> <li>Windows 版, 64 ビット版 UNIX (AIX 5L V5.2 を除く) : 1970 ~ 2900</li> <li>32 ビット版 UNIX, 64 ビット版 AIX 5L V5.2 : 1970 ~ 2037</li> </ul>
	MM	月	2	1 ~ 99
	DD	日	2	1 ~ 99
時刻	hh	時	2	12 時間制表記なし : 0 ~ 99 12 時間制表記あり : 0 ~ 87
	mm	分	2	0 ~ 99
	ss	秒	2	0 ~ 99
	fff	秒の小数部 (ミリ秒)	3	0 ~ 999

書式		項目	桁	値
12 時間制表記※	AM am A.M. a.m.	午前を示す	—	—
	PM pm P.M. p.m.	午後を示す	—	—
T		日付と時刻の区切り	—	—
空白		日付, 時刻, 12 時間制表記の区切り	—	—

(凡例)

— : キーの桁数には含まれない

注※

入力データに 12 時間制表記を見つけた場合は, 時刻は 12 時間制で表記されると自動判定します。

- 日付時刻の指定可能な範囲は, 使用する SORT EE のコマンド, およびコマンド実行システムによって異なります。

OS		指定可能な範囲	
		exsort コマンド	exsort64 コマンド
Windows		19700101T000000000 (世界協定時 (UTC) の 1970 年 1 月 1 日 0 時 0 分 0.0 秒) ~ 29009999T999999999 (世界協定時 (UTC) の 2900 年 99 月 99 日 99 時 99 分 99.999 秒)	
Linux		19700101T000000000 (世界協定時 (UTC) の 1970 年 1 月 1 日 0 時 0 分 0.0 秒) ~ 20371231T235959999 (世界協定時 (UTC) の 2037 年 12 月 31 日 23 時 59 分 59.999 秒)	Windows と同じ
HP-UX		Linux の exsort コマンドと同じ	Windows と同じ
AIX	AIX 5L V5.2	Linux の exsort コマンドと同じ	Linux の exsort コマンドと同じ
	AIX 5L V5.3 以降	Linux の exsort コマンドと同じ	Windows と同じ

- 時を 12 時間制で表記した場合は, 12 時間制表記を設定します。
- 日付と時刻は, 「T」または空白で区切ります。
- 時刻と 12 時間制表記は, 空白で区切ります。
- 日付時刻は 0 ~ 9 の数字以外に, 次の文字をデータに含むことができます。

書式	項目
/ (スラント)	年月日の桁修飾
- (ハイフン)	年月日の桁修飾
. (ピリオド)	年月日, 時分秒の桁修飾
: (コロン)	時分秒の桁修飾
' (シングルクォーテーション)	引用符
" (ダブルクォーテーション)	引用符

- 1 桁の長さ (n バイト) は入力ファイルのエンコードによって異なります。

codetype 指定値	1 桁の長さ (n バイト)
ASCII, SJIS, EUC-JP, EUC-HJ, または UTF-8	1 バイト
UCS-2LE, UCS-2BE, UTF-16LE, または UTF-16BE	2 バイト
UCS-4LE, UCS-4BE, UTF-32LE, または UTF-32BE	4 バイト

#### 入力時の注意事項

入力時の注意事項を次に示します。

- 入力データのタイムゾーン指定子 (TZD) は参照しません。実行システムのローカルタイムとして扱います。
- タイムゾーン指定子 (TZD) までを日付時刻データとして扱います。入力データに「Z」, 「+」および時刻以降の「-」を検知した場合は、以降のデータは無視されます。
- 入力データは項目の区切り文字で区切り、それぞれのデータの左端をそろえて入力します。
- 指定された桁 (引用符, 項目区切り文字, および桁修飾文字は含まない) を超えるデータが入力された場合、指定された桁数を越えた部分 (後ろ) のデータは無視します。
- 指定された桁 (引用符, 項目区切り文字, および桁修飾文字は含まない) よりも少ない桁のデータが入力された場合は、埋め字を仮定し指定された桁数に調整します。調整する位置および埋め字は、データの有無および書式によって異なります。

#### 日付時刻データの端数処理

日付時刻データの端数処理を次の表に示します。

書式			桁不足		入力データなし※8
			埋め字の位置	埋め字データ	
日付	桁修飾文字あり	年	エラー※1※2	—	エラー※1※3
		月	前側※6	数字「0」	01 月を仮定
		日	前側※6	数字「0」	01 日を仮定
	桁修飾文字なし	入力データが 4 桁未満	エラー※1※4	—	エラー※1※5
		入力データが 4 桁以上	後ろ側※7	数字「0」	
時刻	桁修飾文字あり	時	前側	数字「0」	00 時を仮定
		分	前側	数字「0」	00 分を仮定
		秒	前側	数字「0」	00 秒を仮定
		ミリ秒	後ろ側	数字「0」	000 ミリ秒を仮定
	桁修飾文字なし	—	後ろ側	数字「0」	00 時 00 分 00 秒 000 ミリ秒を仮定

(凡例)

— : 該当なし

注※1

キー指定の場合は、KBLS1211-E メッセージが表示されてエラーになります。  
比較項目指定の場合は、KBLS1320-E メッセージが表示されてエラーになります。

注※2

コマンドオプション「-cmdopt DATETIMENOCK」が指定された場合は、後ろ側に数字「0」を仮定します。

注※3

コマンドオプション「-cmdopt DATETIMENOCK」が指定された場合は、「1900 (1900 年)」を仮定します。

注※4

コマンドオプション「-cmdopt DATETIMENOCK」が指定された場合、年は後ろ側に「0」を、月および日は「01 (01 月または 01 日)」を仮定します。

注※5

コマンドオプション「-cmdopt DATETIMENOCK」が指定された場合は、「19000101 (1900 年 01 月 01 日)」を仮定します。

注※6

月および日に 1 桁の「0 (0 月または 0 日)」が入力された場合は、注※1 に示



すエラーとなります。

例えば、「2013/1/1」が入力された場合は「20130101 (2013年01月01日)」を仮定しますが、「2013/0/0」が入力された場合はエラーとなります。

#### 注※7

後ろ側に数字「0」を仮定することで、月および日が「00」となった場合は、「01」を仮定します。ただし、月および日に1桁の「0 (0月または0日)」が入力された場合は、注※1に示すエラーとなります。例えば、「201301」が入力された場合は「20130101 (2013年01月01日)」を仮定しますが、「2013010」が入力された場合はエラーとなります。

#### 注※8

「入力データなし」とは、指定したセルの書式に示すデータの一部がなかった場合を示します。日付および時刻データともに入力されていない場合は、ナル文字列として処理します。

コマンドオプション「`-cmdopt DATETIMENOCHK`」の詳細については、「5.5.13 日付時刻データのチェックオプション (DATETIMENOCHK オプション)」を参照してください。

日付時刻データの端数処理についての注意事項を次に示します。

- 桁修飾文字 (スラント、ハイフン、コロン、またはピリオド) で年月日および時分秒ミリ秒を区切っている場合、日付の M (月)、D (日)、および時刻の h (時)、m (分)、s (秒)、f (ミリ秒) 部分のデータは1桁でもかまいません。1桁データは2桁または3桁に調整されます。例えば、「2013-7-7T8:3:5.1」と「2013-07-07T08:03:05.100」は同じ値として扱われます。
- 日付時刻データを桁修飾文字 (スラント、ハイフン、コロン、またはピリオド) で区切っている場合は、年月日時分秒ミリ秒のそれぞれの桁ごとに桁調整されます。
- 日付時刻として正しくないデータの場合も、扱える値の範囲内であればエラーにしないで日付、時刻またがりデータとして扱います。例えば、「2013-11-31」は「2013-12-01」、「20:59:99」は「21:00:39」として扱います。

#### 指定例

指定例を次に示します。

入力データ (書式)	有効桁の指定 (日付, 時刻)	内部データ形式	説明
2013-08-05T19:08:53.068 (YYYY-MM-DDThh:mm:ss.fff)	6,4	2013081908	有効桁以外の部分 (DD,ss,fff) は入力されません。
	0,9	190853068	時刻データだけ比較対象となり、日付 (YYYYMMDD) は入力されません。

付録 J 日付時刻データの形式 (SORT EE の機能)

入力データ (書式)	有効桁の指定 (日付, 時刻)	内部データ形式	説明
19876/054/0321 △ 098.076.0543.21098 (YYYY/MM/DD △ hh.mm.ss.fff)	8,9	19870503090705210	それぞれの桁あふれ部分 (下線部分) は入力されません。 YYYY : 1987 <u>6</u> MM : 05 <u>4</u> DD : 03 <u>21</u> hh : 09 <u>8</u> mm : 07 <u>6</u> ss : 05 <u>43</u> fff : 210 <u>98</u>
2013T201320 (YYYYThh)	6,6	201301201320	日付の月データがなかったため, 1月 (01) を仮定値とし, 日付を6桁に調整します。
20130805190853068 (YYYYMMDD)	6,4	2013080000	区切り文字がないため, 時刻データを検知できず, 0時0分 (0000) を仮定します。
2013/8/10 △ 7:27:53.250 (YYYY/MM/DD △ hh:mm:ss.fff)	8,9	20130810072753250	前側 (MM, hh) に0 (ゼロ) を仮定値とし, 日付時刻を17桁に調整します。
20130807T320808080 (YYYYMMDDThmmssfff)	8,9	20130808080808080	hh に24以上が入力されたため, 時刻またがりによって内部データを変更します。

(凡例)

△ : 空白を示す。

T : 区切り文字「T」を示す。

**!** 注意事項

- 日付時刻データの数値部分だけが有効桁数の対象です。データの先頭と末尾を囲んでいる引用符、項目区切り文字 (「T」または空白)、桁修飾文字 (スラント、ハイフン、コロン、またはピリオド)、および 12 時間制表記は、有効桁に含まれません。
- 日付時刻データの端数および日付時刻またがり処理によって、入力セルデータが変更されることはありません。
- 入力データは実行システムのローカルタイムとして解釈し、世界協定時 (UTC) に正規化します。正規化した時刻が、日付時刻の指定可能な範囲内でなければ不正なデータと見なします。
- 日付データの有効桁に「8」を、時刻データの有効桁に「2」以上を指定している場合、桁調整後のローカルタイムが世界協定時 (UTC) の範囲内に収まらない場合がありますので、注意してください。世界協定時 (UTC) の範囲外となる例を次に示します。

(例)

実行システムのローカルタイムのタイムゾーン : UTC+09:00

キーの指定「h+1-8,2」: 日付データの有効桁「8」、時刻データの有効桁「2」

入力データ	桁調整 (括弧は埋め字)				世界協定時 (UTC)
	年	月	日	時	
1970	1970	(01)	(01)	(00)	9 時間減算すると「1970 年 01 月 01 日 0 時 0 分 0.0 秒」以前となるため、不正データと見なされます。
1970-01	1970	01	(01)	(00)	
1970-01-01	1970	01	01	(00)	
1970-01-01T08	1970	01	01	08	
1970-01-01T09	1970	01	01	09	9 時間減算すると「1970 年 01 月 01 日 0 時 0 分 0.0 秒」となり、正常データとして処理されます。

---

## 付録 K メッセージ

SORT EE および SORT から出力されるメッセージについて説明します。

### 付録 K.1 SORT EE メッセージ

#### (1) SORT EE メッセージの種類

SORT EE のメッセージには、次の 3 種類があります。

- **正常終了時のメッセージ**  
コマンド処理が正常に終了した場合には表示されます。
- **異常終了時のメッセージ**  
コマンド処理中にエラーが発生した場合には表示されます。
- **システムエラー発生時のメッセージ**  
システムコールによるエラーが発生した場合には表示されます。

#### (2) SORT EE メッセージの記述形式

このマニュアルでは、次に示す形式でメッセージを記載します。

### メッセージ ID

---

メッセージテキスト（英語）

メッセージテキスト（日本語）

日本語文によるメッセージの説明を示す。

- (S) システムの処理を示す。
- (O) 使用者、または運用者の処置を示す。

#### メッセージ ID の説明

メッセージ ID には、メッセージを識別するコードを「KBLSnxxx-x」という形式で示します。

KBLS：プログラム識別コードを示す。

nnnn：メッセージ番号を示す。

x：メッセージの種類を示す。

- I：正常に終了した処理の結果を知らせるメッセージ
- E：エラーの発生を知らせるメッセージ

#### (3) SORT EE メッセージに出力されるデバッグ情報

SORT EE メッセージテキストに出力されるデバッグ情報の用語とその意味を、次に示し

ます。

表 K-1 デバッグ情報の説明

用語	意味
(zz...z)	SORT EE の内部エラーコード。
キー番号	エラーを検知したキーの番号を示します。キー番号を特定できる場合は、コマンドまたは <code>exsort</code> パラメタファイルの <code>-key</code> パラメタに指定した順番 (1 ~) が表示されます。キー番号を特定できない場合は 0 が表示されます。
比較項目番号	エラーを検知した比較項目の番号を示します。比較項目番号を特定できる場合は、コマンドまたは <code>exsort</code> パラメタファイルの <code>include   omit</code> オペラントに指定した順番 (1 ~) が表示されます。比較項目番号を特定できない場合は、0 が表示されます。
比較条件式の番号	エラーを検知した比較条件式の番号を示します。比較条件式の番号を特定できる場合は、コマンドまたは <code>exsort</code> パラメタファイルの <code>include   omit</code> オペラントに指定した順番 (1 ~) が表示されます。比較条件式の番号を特定できない場合は、0 が表示されます。
集約項目番号	エラーを検知した集約項目の番号を示します。集約項目番号を特定できる場合は、コマンドまたは <code>exsort</code> パラメタファイルの <code>-sum</code> パラメタに指定した順番 (1 ~) が表示されます。集約項目番号を特定できない場合は、0 が表示されます。
レコード番号	エラーを検知した入力レコードの番号を示します。レコード番号を特定できる場合は、入力順番 (1 ~) が表示されます。レコード番号を特定できない場合は、0 が表示されます。処理の中間結果を格納する中間ファイルでエラーを検知した場合は、中間ファイルで取得されたレコード番号が表示されます。
パラメタ番号	エラーを検知した <code>-inpfiler</code> 、 <code>-reorg</code> または <code>-outfile</code> パラメタの番号を示します。 <code>-inpfiler</code> パラメタにエラーがある場合は、パラメタ番号は必ず 1 が表示されます。 <code>-reorg</code> パラメタにエラーがある場合は、パラメタ番号は必ず 1 が表示されます。 <code>-outfile</code> パラメタにエラーがある場合は、コマンドまたは <code>exsort</code> パラメタファイルに指定した順番 (1 ~) が表示されます。
ファイル名およびディレクトリ名	ファイル名、およびディレクトリ名を特定できない場合は表示されません。 <ul style="list-style-type: none"> <li>表示するファイル名またはディレクトリ名の長さは <code>MAX_PATH</code> までです。</li> <li>環境変数に指定されたファイル名がエラーだった場合、環境変数が表示されます。</li> <li>標準入力ファイルを指定した場合、入力ファイル名は「<code>stdin</code>」と表示されます。</li> <li>標準出力ファイルを指定した場合、出力ファイル名は「<code>stdout</code>」と表示されます。</li> </ul>

#### (4) 終了コードの種類 (SORT EE の場合)

SORT EE の終了コードは次のとおりです。

メッセージのほかにコマンドの終了状態を表す 3 種類の終了コードがあります。終了コードの種類を次の表に示します。

表 K-2 終了コードの種類

終了状態		終了コード
正常終了		0
異常終了	KBLS1700-E 以外のメッセージ出力時	1
	KBLS1700-E メッセージ出力時、または SORT EE のメッセージが出力されないとき	システムの異常終了コードが設定される

## (5) SORT EE メッセージ出力オプション

SORT EE メッセージの出力先は、環境変数「EXSORTMSGOPT (64 ビット版は「EXSORT64MSGOPT)」」で指定できます。

### Windows

環境変数「EXSORTMSGOPT (64 ビット版は「EXSORT64MSGOPT)」」を省略した場合は、SORT 環境ファイルの msg\_opt オプションの指定を有効とします。環境変数および SORT 環境ファイルのどちらも省略した場合は、すべてのメッセージを画面に表示します。

### UNIX

環境変数「EXSORTMSGOPT (64 ビット版は「EXSORT64MSGOPT)」」を省略した場合は、標準エラーファイルにすべてのメッセージを出力します。

メッセージ出力オプションの詳細については、「2.27 メッセージ出力オプション (SORT EE の機能)」、および環境変数「14.1.1(5) EXSORTMSGOPT」, 「14.1.2(5) EXSORT64MSGOPT」を参照してください。

## 付録 K.2 SORT EE のメッセージ一覧

### KBLS1001-I

number of data items sorted: xx...x/yy...y (ff...f [ff...f] ...)

ソート済みデータ件数 xx...x/yy...y (ff...f [ff...f] ...)

完了メッセージオプションで「EXS\_COMPLETE\_MSGOPT=1」を指定したとき、最後に実行した機能がソート処理の場合に出力されます。

xx...x はソート済みデータ件数を、yy...y<sup>\*</sup>は入力データ件数を、ff...f は出力ファイル名を示します。

完了メッセージオプションについては、「19.1.4 完了メッセージオプション」を参照してください。

#### 注※

入力データ件数とは、入力ファイルから実際に入力したデータ件数です。「-inpfile」パラメタ、「-splitinp」パラメタ、および「-catinp」パラメタの「stopaft」オペラ

ンドを指定した場合、指定件数に達した時点で、レコードの入力を打ち切ります。  
このため、メッセージに表示されるデータ件数は、入力ファイルの総レコード件数  
ではなく、打ち切り時点の件数です。

### **KBLS1002-I**

---

number of data items merged: xx...x/yy...y (ff...f [ff...f] ...)

マージ済みデータ件数 xx...x/yy...y (ff...f [ff...f] ...)

完了メッセージオプションで「EXS\_COMPLETE\_MSGOPT=1」を指定したとき、最後  
に実行した機能がマージ処理の場合に出力されます。

xx...x はマージ済みデータ件数を、yy...y は入力データ件数を、ff...f は出力ファイル名  
を示します。

完了メッセージオプションについては、「19.1.4 完了メッセージオプション」を参照し  
てください。

### **KBLS1003-I**

---

number of data items selected: xx...x/yy...y (ff...f [ff...f] ...)

選択済みデータ件数 xx...x/yy...y (ff...f [ff...f] ...)

完了メッセージオプションで「EXS\_COMPLETE\_MSGOPT=1」を指定したとき、最後  
に実行した機能が選択処理の場合に出力されます。

xx...x は選択済みデータ件数を、yy...y<sup>※</sup>は入力データ件数を、ff...f は出力ファイル名を  
示します。

完了メッセージオプションについては、「19.1.4 完了メッセージオプション」を参照し  
てください。

注※

入力データ件数とは、入力ファイルから実際に入力したデータ件数です。「-inpfile」  
パラメタ、「-splitinp」パラメタ、および「-catinp」パラメタの「stopaft」オペラ  
ンドを指定した場合、指定件数に達した時点で、レコードの入力を打ち切ります。  
このため、メッセージに表示されるデータ件数は、入力ファイルの総レコード件数  
ではなく、打ち切り時点の件数です。

### **KBLS1004-I**

---

number of data items summation: xx...x/yy...y (ff...f [ff...f] ...)

集約済みデータ件数 xx...x/yy...y (ff...f [ff...f] ...)

完了メッセージオプションで「EXS\_COMPLETE\_MSGOPT=1」を指定したとき、最後  
に実行した機能が集約処理の場合に出力されます。

xx...x は集約済みデータ件数を、yy...y<sup>※</sup>は入力データ件数を、ff...f は出力ファイル名を  
示します。

完了メッセージオプションについては、「19.1.4 完了メッセージオプション」を参照し  
てください。

## 注※

入力データ件数とは、入力ファイルから実際に入力したデータ件数です。「-inpfile」パラメタ、「-splitinp」パラメタ、および「-catinp」パラメタの「stopaft」オペランドを指定した場合、指定件数に達した時点で、レコードの入力を打ち切ります。このため、メッセージに表示されるデータ件数は、入力ファイルの総レコード件数ではなく、打ち切り時点の件数です。

**KBLS1005-I**

number of data items record edited: xx...x/yy...y (ff...f [ff...f] ...)

レコード編集済みデータ件数 xx...x/yy...y (ff...f [ff...f] ...)

完了メッセージオプションで「EXS\_COMPLETE\_MSGOPT=1」を指定したとき、最後に実行した機能がレコード編集処理の場合に出力されます。

xx...x はレコード編集済みデータ件数を、yy...y<sup>\*</sup>は入力データ件数を、ff...f は出力ファイル名を示します。

完了メッセージオプションについては、「19.1.4 完了メッセージオプション」を参照してください。

## 注※

入力データ件数とは、入力ファイルから実際に入力したデータ件数です。「-inpfile」パラメタ、「-splitinp」パラメタ、および「-catinp」パラメタの「stopaft」オペランドを指定した場合、指定件数に達した時点で、レコードの入力を打ち切ります。このため、メッセージに表示されるデータ件数は、入力ファイルの総レコード件数ではなく、打ち切り時点の件数です。

**KBLS1006-I**

number of data items copied: xx...x/yy...y (ff...f [ff...f] ...)

コピー済みデータ件数 xx...x/yy...y (ff...f [ff...f] ...)

完了メッセージオプションで「EXS\_COMPLETE\_MSGOPT=1」を指定したとき、最後に実行した機能がコピー処理の場合に出力されます。

xx...x はコピー済みデータ件数を、yy...y<sup>\*</sup>は入力データ件数を、ff...f は出力ファイル名を示します。

完了メッセージオプションについては、「19.1.4 完了メッセージオプション」を参照してください。

## 注※

入力データ件数とは、入力ファイルから実際に入力したデータ件数です。「-inpfile」パラメタ、「-splitinp」パラメタ、および「-catinp」パラメタの「stopaft」オペランドを指定した場合、指定件数に達した時点で、レコードの入力を打ち切ります。このため、メッセージに表示されるデータ件数は、入力ファイルの総レコード件数ではなく、打ち切り時点の件数です。



**KBLS1007-I**

sort was completed: Pid=oo...o IN=pp...p OUT=qq...q [SKIP=rr...r] [INSEL=ss...s] [STOP=tt...t]  
[SUM=uu...u] [WORKSIZE=vv...v,ww...w,xx...x,yy...y] [DELBL=dd...d] (ff...f [ff...f] ...)

ソートが完了しました。Pid=oo...o IN=pp...p OUT=qq...q [SKIP=rr...r] [INSEL=ss...s] [STOP=tt...t]  
[SUM=uu...u] [WORKSIZE=vv...v,ww...w,xx...x,yy...y] [DELBL=dd...d] (ff...f [ff...f] ...)

ソート処理済みデータ件数が出力されます。[] で囲んだデータ件数は、対応する機能が指定されていない場合は表示されません ( [] の内容は 1 文字の空白に置き換えられます)。このメッセージは、「-outfile」パラメタ、「-splitout」パラメタ、および「-catout」パラメタの指定数だけ出力されます。

oo...o : プロセス id

pp...p : 入力データ件数 (1) ※<sup>1</sup>

qq...q : 出力データ件数 (14) ※<sup>2</sup>

rr...r : 読み飛ばし入力データ件数 (4) ※<sup>5</sup> ※<sup>6</sup>

ss...s : 選択済み入力データ件数 (5)

tt...t : 打ち切り入力データ件数 (6) ※<sup>1</sup>

uu...u : 集約済みデータ件数 (9)

vv...v : ソート処理方式 ※<sup>4</sup>

- outcore : アウトコアソート
- outcore(PREM) : 中間マージフェーズが動作したアウトコアソート
- incore : インコアソート

ww...w : 自動見積もりしたワークバッファサイズ (MB) ※<sup>4</sup>

xx...x : 確保したワークバッファサイズ (MB) ※<sup>4</sup>

yy...y : 必要なワークバッファサイズ (MB) ※<sup>3</sup> ※<sup>4</sup>

dd...d : 削除した空行件数 (3)

ff...f : 出力ファイル名

**注※ 1**

入力データ件数とは、入力ファイルから実際に入力したデータ件数です。「-inpfile」パラメタ、「-splitinp」パラメタ、および「-catinp」パラメタの「stopaft」オペランドを指定した場合、指定件数に達した時点で、レコードの入力を打ち切ります。このため、メッセージに表示されるデータ件数は、入力ファイルの総レコード件数ではなく、打ち切り時点の件数 (空行削除オプション (-cmdopt DELBLANKLINE) の指定がある場合、打ち切りレコード件数に空行件数は含まれません) です。

**注※ 2**

出力レコード選択機能が指定されている場合は (11) の実行時に出力されます。

**注※ 3**

「必要なワークバッファサイズ」とは、ソートの実行結果から見積もられた実際に必

要なワークバッファのサイズです。ワークバッファの見積もりサイズについては、「17.1 ソート機能の処理時間を短くするためには」を参照してください。

注※ 4

`-worksize` パラメタで `buffer-size` を指定した場合、「`WORKSIZE=vv…v,ww…w,xx…x,yy…y`」の情報は出力されません。

注※ 5

空行削除オプション (`-cmdopt DELBLANKLINE`) の指定がある場合、「`-inpfile`」パラメタ、「`-splitinp`」パラメタ、および「`-catinp`」パラメタの「`skiprec`」オペランドで指定した読み飛ばしレコード件数に空行件数は含まれません。

注※ 6

空行削除オプション (`-cmdopt DELBLANKLINE`) の指定がある場合、「`-inpfile`」パラメタ、「`-splitinp`」パラメタ、および「`-catinp`」パラメタの「`startrec`」オペランドで指定した入力開始レコード番号で読み飛ばしたレコード件数に空行件数は含まれません。

各機能の実行順序は次のとおりです。

- (1) 入力ファイルから実際に入力したデータ件数
- (2) ファイル連結機能
- (3) テキストファイル・CSV ファイルの空行削除 (`-cmdopt DELBLANKLINE` 指定あり)
- (4) 入力レコードの読み飛ばし (`-inpfile {skiprec | startrec}` 指定あり)
- (5) 入力レコード選択機能 (`-inpfile {include | omit}` 指定あり)
- (6) 入力レコードの打ち切り (`-inpfile stopaft` 指定あり)
- (7) 入力レコード編集機能 (`-inpfile reorg` 指定あり)
- (8) ソート機能
- (9) 集約機能 (`-sum` 指定あり)
- (10) 中間レコード編集機能 (`-reorg` 指定あり)
- (11) 出力レコード選択機能 (`-outfile {include | omit}` 指定あり)
- (12) 出力レコード編集機能 (`-outfile reorg` 指定あり)
- (13) ファイル分割機能
- (14) 出力ファイルへ実際に出力したデータ件数

## KBLS1008-I

---

merge was completed: Pid=ss…s IN=tt…t OUT=uu…u [SKIP=vv…v] [INSEL=ww…w] [SUM=yy…y] [DELBL=dd…d] (ff…f [ff…f] …)

マージが完了しました。Pid=ss…s IN=tt…t OUT=uu…u [SKIP=vv…v] [INSEL=ww…w] [SUM=yy…y] [DELBL=dd…d] (ff…f [ff…f] …)

マージ処理済みデータ件数が出力されます。[] で囲んだデータ件数は、対応する機能が指定されていない場合は表示されません ([ ] の内容は 1 文字の空白に置き換えられます)。このメッセージは、「`-outfile`」パラメタ、「`-splitout`」パラメタ、および「`-catout`」

パラメタの指定数だけ出力されます。

ss…s : プロセス id

tt…t : 入力データ件数 (1)

uu…u : 出力データ件数 (13) ※<sup>1</sup>

vv…v : 読み飛ばし入力データ件数 (4) ※<sup>2</sup>

ww…w : 選択済み入力データ件数 (5)

yy…y : 集約済みデータ件数 (8)

dd…d : 削除した空行件数 (3)

ff…f : 出力ファイル名

#### 注※ 1

出力レコード選択機能が指定されている場合は (10) の実行時に出力されます。

#### 注※ 2

空行削除オプション (-cmdopt DELBLANKLINE) の指定がある場合、「-inpfile」パラメタ、「-splitinp」パラメタ、および「-catinp」パラメタの「startrec」オペランドで指定した入力開始レコード番号で読み飛ばしたレコード件数に空行件数は含まれません。

各機能の実行順序は次のとおりです。

- (1) 入力ファイルから実際に入力したデータ件数
- (2) ファイル連結機能
- (3) テキストファイル・CSV ファイルの空行削除 (-cmdopt DELBLANKLINE 指定あり)
- (4) 入力レコードの読み飛ばし (-inpfile startrec 指定あり)
- (5) 入力レコード選択機能 (-inpfile {include | omit} 指定あり)
- (6) 入力レコード編集機能 (-inpfile reorg 指定あり)
- (7) マージ機能
- (8) 集約機能 (-sum 指定あり)
- (9) 中間レコード編集機能 (-reorg 指定あり)
- (10) 出力レコード選択機能 (-outfile {include | omit} 指定あり)
- (11) 出力レコード編集機能 (-outfile reorg 指定あり)
- (12) ファイル分割機能
- (13) 出力ファイルへ実際に出力したデータ件数

### **KBLS1009-I**

```
copy was completed: Pid=ss…s IN=tt…t OUT=uu…u [SKIP=vv…v] [INSEL=ww…w] [STOP=xx…x] [SUM=yy…y] [DELBL=dd…d] (ff…f [ff…f] …)
```

```
コピーが完了しました。Pid=ss…s IN=tt…t OUT=uu…u [SKIP=vv…v] [INSEL=ww…w] [STOP=xx…x] [SUM=yy…y] [DELBL=dd…d] (ff…f [ff…f] …)
```

コピー処理済みデータ件数が出力されます。[] で囲んだデータ件数は、対応する機能が

指定されていない場合は表示されません ([ ] の内容は 1 文字の空白に置き換えられます)。このメッセージは、「-outfile」パラメタ、「-splitout」パラメタ、および「-catout」パラメタの指定数だけ出力されます。

ss…s : プロセス id

tt…t : 入力データ件数 (1) ※<sup>1</sup>

uu…u : 出力データ件数 (14) ※<sup>2</sup>

vv…v : 読み飛ばし入力データ件数 (4) ※<sup>3</sup> ※<sup>4</sup>

ww…w : 選択済み入力データ件数 (5)

xx…x : 打ち切り入力データ件数 (6) ※<sup>1</sup>

yy…y : 集約済みデータ件数 (9)

dd…d : 削除した空行件数 (3)

ff…f : 出力ファイル名

#### 注※ 1

入力データ件数とは、入力ファイルから実際に入力したデータ件数です。「-inpfiler」パラメタ、「-splitinp」パラメタ、および「-catinp」パラメタの「stopaft」オペランドを指定した場合、指定件数に達した時点で、レコードの入力を打ち切ります。このため、メッセージに表示されるデータ件数は、入力ファイルの総レコード件数ではなく、打ち切り時点の件数（空行削除オプション（-cmdopt DELBLANKLINE）の指定がある場合、打ち切りレコード件数に空行件数は含まれません）です。

#### 注※ 2

出力レコード選択機能が指定されている場合は (11) の実行時に出力されます。

#### 注※ 3

空行削除オプション（-cmdopt DELBLANKLINE）の指定がある場合、「-inpfiler」パラメタ、「-splitinp」パラメタ、および「-catinp」パラメタの「skiprec」オペランドで指定した読み飛ばしレコード件数に空行件数は含まれません。

#### 注※ 4

空行削除オプション（-cmdopt DELBLANKLINE）の指定がある場合、「-inpfiler」パラメタ、「-splitinp」パラメタ、および「-catinp」パラメタの「startrec」オペランドで指定した入力開始レコード番号で読み飛ばしたレコード件数に空行件数は含まれません。

各機能の実行順序は次のとおりです。

- (1) 入力ファイルから実際に入力したデータ件数
- (2) ファイル連結機能
- (3) テキストファイル・CSV ファイルの空行削除（-cmdopt DELBLANKLINE 指定あり）
- (4) 入力レコードの読み飛ばし（-inpfiler {skiprec | startrec} 指定あり）
- (5) 入力レコード選択機能（-inpfiler {include | omit} 指定あり）
- (6) 入力レコードの打ち切り（-inpfiler stopaft 指定あり）

- (7) 入力レコード編集機能 (-infile reorg 指定あり)
- (8) コピー機能
- (9) 集約機能 (-sum 指定あり)
- (10) 中間レコード編集機能 (-reorg 指定あり)
- (11) 出力レコード選択機能 (-outfile {include | omit} 指定あり)
- (12) 出力レコード編集機能 (-outfile reorg 指定あり)
- (13) ファイル分割機能
- (14) 出力ファイルへ実際に出力したデータ件数

## **KBLS1010-I**

---

file split was completed. OUT=yy...y ("zz...z")

ファイル分割が完了しました。OUT=yy...y ("zz...z")

ファイル分割が完了しました。このメッセージは、分割した出力ファイルの数だけ出力されます。

このメッセージの出力を抑止したい場合は、環境変数名

「EXSORTSUPPRESSSPLITMSG (64 ビット版は

「EXSORT64SUPPRESSSPLITMSG)」に YES を指定してください。環境変数の詳細

については、「14.1.1(19) EXSORTSUPPRESSSPLITMSG」または「14.1.2(19)

EXSORT64SUPPRESSSPLITMSG」を参照してください。

yy...y は、zz...z で示すファイルに出力したレコード件数を示します。

zz...z は、分割した出力ファイル名を示します。

## **KBLS1011-I**

---

file cat was completed. IN=yy...y ("zz...z")

ファイル連結が完了しました。IN=yy...y ("zz...z")

ファイル連結が完了しました。このメッセージは、連結した入力ファイルの数だけ出力されます。

このメッセージの出力を抑止したい場合は、環境変数名

「EXSORTSUPPRESSCATMSG (64 ビット版は「EXSORT64SUPPRESSCATMSG)」

に YES を指定してください。環境変数の詳細については、「14.1.1(20)

EXSORTSUPPRESSCATMSG」または「14.1.2(20)

EXSORT64SUPPRESSCATMSG」を参照してください。

yy...y は、zz...z で示すファイルから入力したレコード件数を示します。

zz...z は、連結した入力ファイル名を示します。

## **KBLS1012-I**

---

input data overflowed: ww...w,xx...x,yy...y [ww...w,xx...x,yy...y...] (zz...z)

有効桁よりも大きい入力データを検出しました。ww…w,xx…x,yy…y [ww…w,xx…x,yy…y …] (zz…z)

「-csvsum」パラメタで指定した CSV ファイルの集約項目で、指定された有効桁よりも大きい入力データを検出しました。

このメッセージ 1 つにつき最大 32 個の集約項目の情報 (ww…w,xx…x,yy…y) を表示します。

集約項目の検出数によって、このメッセージの出力回数は異なります (集約項目は最大 1,280 個まで拡張できるので、最大 40 回出力されます)。

ww…w : 集約項目の番号

xx…x : 整数部の最大桁

yy…y : 小数部の最大桁

(O) オーバーフローが発生しない有効桁数を指定して再度実行してください。

## KBLS1013-I

---

rounding overflowed: xx…x […] (zz…z)

入力データの丸め処理でオーバーフローを検出しました。xx…x […] (zz…z)

「-csvsum」パラメタで指定した CSV ファイルの集約項目で、入力データの丸め処理でオーバーフローを検出しました。

このメッセージ 1 つにつき最大 128 個の集約項目の情報 (xx…x) を表示します。

集約項目の検出数によって、このメッセージの出力回数は異なります (集約項目は最大 1,280 個まで拡張できるので、最大 10 回出力されます)。

xx…x は、集約項目の番号を示します。

(O) オーバーフローが発生しない有効桁数、および丸め方式と丸め桁を指定して再度実行してください。

## KBLS1014-I

---

simple copy was completed: Pid=xx…x SIZE=yy…y (ff…f[ ff…f] …)

簡易コピーが完了しました。Pid=xx…x SIZE=yy…y (ff…f[ ff…f] …)

簡易コピーが完了しました。

xx…x : プロセス id

yy…y : 出力ファイルのサイズ (単位はバイト)

ff…f : 出力ファイル名

## KBLS1060-I

---

insufficient memory, executed in SORT xx…x yy…y (zz…z)

メモリ不足が発生したため、SORT でコマンドを実行しました。xx…x yy…y (zz…z)

SORT EE の起動時、メモリ不足が発生したため、SORT EE ではなく、SORT でコマンドを実行しました。

xx...x は確保しようとした領域のサイズを、yy...y はシステムエラーコードを示します。

**(S)** SORT でコマンドを実行する。

**(O)** 必要であれば、再度実行してください。

再度実行してもエラーになる場合は、ログファイルを採取の上、保守員に連絡してください。

## **KBLS1061-I**

---

system call error, executed in SORT xx...x yy...y (zz...z)

システムコールエラーが発生したため、SORT でコマンドを実行しました。xx...x yy...y (zz...z)

SORT EE の起動時、システムコールエラーが発生したため、SORT EE ではなく、SORT でコマンドを実行しました。

xx...x はシステムエラーコードを、yy...y は SORT EE 内部コードを示します。

**(S)** SORT でコマンドを実行する。

**(O)** 必要であれば、再度実行してください。

再度実行してもエラーになる場合は、ログファイルを採取の上、保守員に連絡してください。

## **KBLS1100-E**

---

invalid parameter specification: xx...x (zz...z)

不正なパラメタが指定されました。xx...x (zz...z)

パラメタの指定が間違っています。例えば、次のような場合が考えられます。

- パラメタ名称が誤っている (パラメタ名とは異なる、先頭にハイフンが付いたキーワードを指定した場合)
- オペランド名称が誤っている (オペランド名とは異なる、イコールを含んだキーワードを指定した場合)
- オペランド形式が誤っている (「オペランド=値」、または「オペランド=値△…△値」の形式ではない場合)
- 区切り文字の位置が誤っている
- パラメタファイルの指定で、改行の位置が誤っている
- パラメタファイルの指定で、行の先頭から指定していない
- パラメタファイルの指定で、1 行に複数のパラメタ、または複数のオペランドを指定している
- パラメタファイルの指定で、パラメタに対する値を指定していない
- パラメタファイルの指定で、選択条件式の条件値の指定が誤っている
- -inpf file の file オペランド値、または -outfile の file オペランド値を指定していない
- -catinp の file または {suffix | prefix} オペランド値、または -catout の file オペランド値を指定していない
- -splitinp の file オペランド値、または -splitout の {suffix | prefix} オペランド値を指定していない
- 文字条件値または挿入データの指定で、シングルクォーテーション、または円記号の

記述が間違っている（例えば、文字として指定するシングルクォーテーションの前に円記号を置いていない）

- 環境変数名「% ~ %」にファイル名が関連づけされていない
- 文字条件値または挿入データの文字列の指定で、文字が指定されていない
- コマンドまたはパラメタファイルで記述した文字コードが、コマンド実行時のロケールと異なっている
- `-cpyinp`, または `-cpyout` パラメタに指定できないオペランドを指定した

`xx...x` はパラメタ名を示します。

**(O)** 正しいパラメタを指定して再度実行してください。

### **KBLS1101-E**

---

two or more parameter are specified: `xx...x (zz...z)`

`xx...x` パラメタが複数個指定されています。(zz...z)

複数指定できないパラメタが、複数個指定されています。例えば、次のような場合が考えられます。

- 「`-outfile`」パラメタ、「`-splitout`」パラメタ、および「`-catout`」パラメタ以外のパラメタを複数個指定している
- 複数個の「`-outfile`」パラメタ、「`-splitout`」パラメタ、または「`-catout`」パラメタを指定した `exsort` パラメタファイルの内容を、`exsort` コマンドの指定で置き換えようとした
- `exsort` パラメタファイルの内容を、複数個の「`-outfile`」パラメタ、「`-splitout`」パラメタ、または「`-catout`」パラメタを指定した `exsort` コマンドの内容で置き換えようとした

`xx...x` はパラメタ名を示します。

**(O)** 重複しているパラメタを修正して再度実行してください。

### **KBLS1102-E**

---

two or more operand are specified: `xx...x yy...y (zz...z)`

`xx...x` パラメタの `yy...y` オペランドが複数個指定されています。(zz...z)

オペランドが複数個指定されています。

`xx...x` はパラメタ名を、`yy...y` はオペランド名を示します。

**(O)** 重複しているオペランドを修正して再度実行してください。

### **KBLS1103-E**

---

parameter has not parameter file: `xx...x (zz...z)`

`xx...x` パラメタはパラメタファイルに指定できません。(zz...z)

`exsort` パラメタファイルに指定できないパラメタが指定されています。



xx...x はパラメタ名を示します。

(O) 指定できないパラメタを削除して再度実行してください。

### **KBLS1104-E**

---

number of -outfile parameters exceeds limited: xx...x (zz...z)

-outfile パラメタの指定数が制限を超えています。xx...x (zz...z)

「-outfile」パラメタの個数が制限数を超えています。

xx...x は制限数を示します。

(O) 「-outfile」パラメタの個数を正しく指定して再度実行してください。「-outfile」パラメタの個数が制限を超える場合は、-outfile パラメタ数の制限解除オプション（環境変数「EXSORTOUTFILENUMUNLIMIT（64ビット版は「EXSORT64OUTFILENUMUNLIMIT）」、または環境ファイル「Exsortconf.ini ファイル（64ビット版は「Exsortconf64.ini ファイル）」の「EXS\_OUTFILENUMUNLIMIT）」の適用を検討してください。

### **KBLS1105-E**

---

cannot specify both ww...w and xx...x: yy...y (zz...z)

ww...w と xx...x は一緒に指定できません。yy...y (zz...z)

同時に指定できないオペランドを指定しています。

ww...w と xx...x はオペランド名を、yy...y はパラメタ名を示します。

(O) 正しく指定して再度実行してください。

### **KBLS1106-E**

---

cannot specify both xx...x and yy...y: (zz...z)

xx...x と yy...y は一緒に指定できません。(zz...z)

同時に指定できないパラメタを指定しています。

xx...x と yy...y はパラメタ名を示します。

(O) 正しく指定して再度実行してください。

### **KBLS1107-E**

---

cannot specify both ww...w of xx...x and yy...y. (zz...z)

xx...x パラメタの ww...w オペランドと環境変数 yy...y は一緒に指定できません。(zz...z)

同時に指定できないオペランドと環境変数を指定しています。

xx...x はパラメタ名、ww...w はオペランド名、yy...y は環境変数名を示します。

(O) 正しく指定して再度実行してください。

## **KBLS1108-E**

---

cannot specify both fileorg of output file and reorg. (zz...z)

ファイル編成の変更とレコード編集機能と一緒に指定できません。(zz...z)

入力ファイルと異なるファイル編成を指定（ファイル編成を変更）する場合、レコード編集機能と一緒に指定できません。

**(O)** 正しいパラメタを指定して再度実行してください。

## **KBLS1110-E**

---

must specify function option: (zz...z)

機能オプションが指定されていません。(zz...z)

「-function」パラメタが指定されていません。または、「-function」パラメタに機能オプション（sort | merge | copy）が指定されていません。

**(O)** 機能オプションを指定して再度実行してください。

## **KBLS1111-E**

---

invalid function option specification: xx...x (zz...z)

機能オプションの指定が正しくありません。xx...x (zz...z)

「-function」パラメタに機能オプション（sort | merge | copy）以外が指定されています。または、簡易コピー機能使用時、「-function」パラメタに機能オプション（copy）以外が指定されています。

xx...x はオプション指定値を示します。

**(O)** 機能オプションの指定値を修正して再度実行してください。

## **KBLS1120-E**

---

invalid command option specification: xx...x (zz...z)

コマンドオプションの指定が正しくありません。xx...x (zz...z)

「-cmdopt」パラメタに指定するコマンドオプションの指定が間違っています。例えば、次のような場合が考えられます。

- コマンドオプション以外の値を指定している
- コマンドオプションを小文字の英字で指定している
- 同じコマンドオプションを複数指定している
- コマンドオプションの区切り文字（コロン）の前後にスペースを指定している
- 同時に指定できないコマンドオプションを指定している

xx...x はコマンドオプション指定値を示します。

**(O)** コマンドオプションを修正して再度実行してください。

**KBLS1121-E**

---

command option must be specified: (zz...z)

コマンドオプションが指定されていません。(zz...z)

「-cmdopt」パラメタにコマンドオプションが 1 個も指定されていません。

(O) コマンドオプションを指定して再度実行してください。

**KBLS1122-E**

---

command option OVWRT must be specified: (zz...z)

既存ファイルの上書きオプションが指定されていません。(zz...z)

出力ファイル名重複チェック抑止オプションを使用する場合、既存ファイルの上書きオプションの指定が必要です。

(O) -cmdopt OVWRT (既存ファイルの上書きオプション) を指定してください。

**KBLS1130-E**

---

invalid buffer size specification: xx...x (zz...z)

バッファサイズの指定が正しくありません。xx...x (zz...z)

「-worksize」パラメタの指定が間違っています。例えば、次のような場合が考えられません。

32 ビット版の場合

- サイズを数値以外で指定している
- 単位の指定を、M 以外で指定している
- 指定できる桁数 (4 桁) を超えている
- サイズに 0 を指定している

64 ビット版の場合

- サイズを数値以外で指定している
- 単位の指定を、G または M 以外で指定している
- 指定できる桁数 (4 桁または 7 桁) を超えている
- サイズに 0 を指定している

xx...x はバッファサイズ指定値を示します。

(O) サイズを修正して再度実行してください。

**KBLS1131-E**

---

buffer size must be specified: (xx...x)

バッファサイズが指定されていません。(xx...x)

「-worksize」パラメタにバッファサイズ、またはソート処理で使用するワークバッファサイズの最大値が指定されていません。

(O) バッファサイズ, またはソート処理で使用するワークバッファサイズの最大値を指定して再度実行してください。

### **KBLS1132-E**

---

too many buffer size: xx...x (zz...z)

バッファサイズが制限を超えています。xx...x (zz...z)

「-worksize」パラメタに指定しているバッファサイズが制限を超えています。  
xx...x は制限サイズを示します。

(O) バッファサイズ, またはソート処理で使用するワークバッファサイズの最大値を修正して再度実行してください。

### **KBLS1133-I**

---

insufficient buffer to execute (zz...z)

ソート用ワークバッファサイズが不足しています。(zz...z)

(O) 適正なワークバッファサイズを指定してください。

### **KBLS1134-E**

---

too many size of string file (zz...z)

ストリング情報ファイルの制限を越えました。(zz...z)

(S) オープンしたファイルをすべてクローズし, 処理を中断します。  
(O) 適正なワークバッファサイズを指定してください。

### **KBLS1135-E**

---

I/O error on string file xx...x (zz...z)

ストリング情報ファイルに I-O エラーが発生しました。xx...x (zz...z)

ストリング情報ファイルに I/O エラーが発生しました。xx...x は, システムエラーコードを示します。

(S) オープンしたファイルをすべてクローズし, 処理を中断します。  
(O) 適正なワークバッファサイズを指定してください。

### **KBLS1140-E**

---

invalid base year specification: xx...x (zz...z)

基準年度の指定が正しくありません。xx...x (zz...z)

「-dateopt」パラメタの指定が間違っています。例えば, 次のような場合が考えられます。

- 年度を数値以外で指定している
- 3 桁以上を指定している

xx...x は基準年度指定値を示します。

- (O) 基準年度を修正して再度実行してください。

### **KBLS1141-E**

---

base year must be specified: (zz···z)

基準年度が指定されていません。(zz···z)

「-dateopt」パラメタに基準年度が指定されていません。

- (O) 基準年度を指定して再度実行してください。

### **KBLS1150-E**

---

invalid code specification: xx···x (zz···z)

コードの指定が正しくありません。xx···x (zz···z)

「-altseq」パラメタの「code」オペランドのコードの指定が間違っています。

例えば、次のような場合が考えられます。

- 4桁以外を指定した
- 16進数以外を指定した

xx···x はパラメタ名を示します。

- (O) コードを修正して再度実行してください。

### **KBLS1151-E**

---

code must be specified: xx···x (zz···z)

コードが指定されていません。xx···x (zz···z)

「-altseq」パラメタのコードが指定されていません。例えば、次のような場合が考えられます。

- 「code」オペランドが指定されていない
- 「code」オペランドにコードが指定されていない

xx···x はパラメタ名を示します。

- (O) コードを指定して再度実行してください。

### **KBLS1152-E**

---

duplicate character code: xx···x yy···y (zz···z)

文字コードが重複しています。xx···x yy···y (zz···z)

「-altseq」パラメタの「code」オペランドに、同じ文字コードを指定しています。

xx···x はパラメタ名を、yy···y は重複文字コードを示します。

- (O) 文字コードを正しく指定して再度実行してください。

## **KBLS1160-E**

---

invalid I/O size specification: xx...x (zz...z)

I/O サイズの指定が正しくありません。xx...x (zz...z)

「-iosize」パラメタの指定が間違っています。例えば、次のような場合が考えられます。

- 数値以外を指定している
- 6 桁以上を指定している
- 8 未満の値を指定している
- マージ機能で固定長順編成または可変長順編成ファイルの場合は、64 未満の値を指定している

xx...x は I/O サイズ指定値を示します。

**(O)** I/O サイズを修正して再度実行してください。

## **KBLS1161-E**

---

I/O size must be specified: (zz...z)

I/O サイズが指定されていません。(zz...z)

「-iosize」パラメタに I/O サイズが指定されていません。

**(O)** I/O サイズを指定して再度実行してください。

## **KBLS1162-E**

---

too many I/O size: xx...x (zz...z)

I/O サイズが制限を超えています。xx...x (zz...z)

「-iosize」パラメタに指定しているサイズが制限を超えています。

xx...x は制限サイズを示します。

**(O)** I/O サイズを修正して再度実行してください。

## **KBLS1200-E**

---

invalid number of keys: xx...x (zz...z)

キーの個数が制限を超えています。xx...x (zz...z)

キーの個数が 1 以上、制限数以下の範囲にありません。

xx...x はキー制限個数を示します。

**(O)** キーの個数を正しく指定して再度実行してください。または、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」のキーの最大個数の倍数を、設定または変更して再度実行してください。キーの最大個数の倍数については、「15.1.1(14)EXS\_KEYNUMMAX」または「15.1.1(17)EXS\_CSVKEYNUMMAX」を参照してください。

## KBLS1201-E

---

key must be specified: (xx...x)

キーが指定されていません。(xx...x)

「-key」または「-csvkey」パラメタが指定されていません。次に示す機能を使用する場合、「-key」オペランドを必ず指定してください（入力ファイルが CSV ファイルの場合は、「-csvkey」パラメタとなります）。

- 「-function」パラメタに sort を指定した
- 「-function」パラメタに merge を指定した
- 「-sum」パラメタを指定した（入力ファイルが CSV ファイルの場合は、「-csvsum」パラメタとなります）

(O) 「-key」パラメタまたは「-csvkey」パラメタを指定して再度実行してください。

## KBLS1202-E

---

key attribute must be specified: xx...x (zz...z)

キーの属性が指定されていません。xx...x (zz...z)

キーの属性が指定されていません。

xx...x はキー番号を示します。

(O) キーの属性を指定して再度実行してください。

## KBLS1203-E

---

invalid key attribute specification: xx...x (zz...z)

キーの属性の指定が正しくありません。xx...x (zz...z)

キーの属性の指定が間違っています。例えば、次のような場合が考えられます。

- 「r, j, c, n, m, a, b, p, z, x, s, t, f, g, q, d, w, u, v」以外の文字を指定した（CSV ファイルの場合は、「r, j, c, z, g, h, q, d, w, u, v」だけ指定できる）
- 大文字の英字で指定した
- 入力ファイル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に「w, u, v」を指定した

xx...x はキー番号を示します。

(O) キーの属性を修正して再度実行してください。

## KBLS1204-E

---

starting byte position of key must be specified: xx...x (zz...z)

キーの開始バイト位置が指定されていません。xx...x (zz...z)

キーの開始バイト位置が指定されていません。xx...x はキー番号を示します。

(O) キーの開始バイト位置を指定して再度実行してください。

## **KBLS1205-E**

---

invalid starting byte position of key: xx...x (zz...z)

キーの開始バイト位置の指定が正しくありません。xx...x (zz...z)

キーの開始バイト位置の指定が間違っています。例えば、次のような場合が考えられません。

- 6桁以上の数値を指定した
- 数値以外を指定した
- 入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定し、かつキーの開始バイト位置に 2 の倍数ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定し、かつキーの開始バイト位置に 4 の倍数ではない数値を指定した

xx...x はキー番号を示します。

**(O)** キーの開始バイト位置を修正して再度実行してください。

入力ファイルがテキストファイルまたは CSV ファイルの場合は、入力データのエンコードの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。入力データのエンコードは、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」の「EXS\_CODETYPE」または `-inpf`, `-catinp`, `-splitinp` パラメタの `codetype` オペランドで指定します。

## **KBLS1206-E**

---

key byte length must be specified: xx...x (zz...z)

キーのバイト長が指定されていません。xx...x (zz...z)

キーのバイト長、文字数またはマスクビットが指定されていません。

xx...x はキー番号を示します。

**(O)** キーのバイト長、文字数またはマスクビットを指定して再度実行してください。

## **KBLS1207-E**

---

invalid byte length for key: xx...x (zz...z)

キーのバイト長の指定が正しくありません。xx...x (zz...z)

キーのバイト長、文字数またはマスクビットの指定が間違っています。例えば、次のような場合が考えられます。

- 指定できる範囲以上の数値を指定した (キーの属性によって指定できるバイト長、文字数またはマスクビットの範囲は異なる)
- 数値以外を指定した



- 入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定し、かつキーのバイト長に 2 の倍数ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定し、かつキーのバイト長に 4 の倍数ではない数値を指定した

xx...x はキー番号を示します。

- (O) キーのバイト長、文字数またはマスクビットを修正して再度実行してください。入力ファイルがテキストファイルまたは CSV ファイルの場合は、入力データのエンコードの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。入力データのエンコードは、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル」)」の「EXS\_CODETYPE」または `-inpfiler`、`-catinp`、`-splitinp` パラメタの `codetype` オペランドで指定します。

## **KBLS1208-E**

---

sum of key position and length is greater than record length: xx...x (zz...z)

キー位置+キー長がレコード長を超えています。xx...x (zz...z)

キー位置+キー長が入力ファイルのレコード長を超えているか、可変長ファイルの場合は、最小レコード長を超えています。可変長順編成ファイルで、`-cmdopt POSNOCHK` の指定がある場合は、最大レコード長を超えています。または、入力ファイルに不当に短いデータがあります。

xx...x はキー番号を示します (可変長ファイル、またはテキストファイルの場合、0 が表示されます)。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。
- (O) キーの開始バイト位置、バイト長を正しく指定して再度実行してください。

## **KBLS1209-E**

---

sum of key position and length include record length area: xx...x (zz...z)

キー位置+キー長がレコード長領域を含んでいます。xx...x (zz...z)

キー位置+キー長が入力レコードのレコード領域長を含んでいます。次の条件を両方満たす場合に該当します。

- 「`-locatajst`」パラメタを指定している
- 可変長ファイルで、キー位置+キー長が 1 ~ 4 バイト目を含んでいる

xx...x はキー番号を示します。

- (O) レコード長領域を除いたキーの開始バイト位置、バイト長を指定して再度実行して

ください。

## KBLS1210-E

---

too many total key length : xx...x yy...y (zz...z)

キーの合計長が制限を超えています。xx...x yy...y (zz...z)

キーの合計長が制限サイズを超えています。

「-key」パラメタの場合は、「-len」に指定したバイト長または文字数の合計、または「-maskbit」に指定したマスクビットの合計となります。

「-csvkey」パラメタの場合は、「-len」および「-p」に指定した有効桁の合計となります。

xx...x はキー番号を、yy...y は制限サイズを示します。

- (O)** キーの個数、バイト長、文字数またはマスクビットを正しく指定して再度実行してください。または、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」のキーの合計長の制限サイズの倍数を、設定または変更して再度実行してください。キーの合計長の制限サイズの倍数については、「15.1.1(14)EXS\_KEYNUMMAX」または「15.1.1(17)EXS\_CSVKEYNUMMAX」を参照してください。

## KBLS1211-E

---

invalid key data: ww...w xx...x yy...y (zz...z)

キーにエラーデータを検出しました。ww...w xx...x yy...y (zz...z)

入力データ中の 10 進数、日付または日付時刻のキーとして指定した範囲の値に、不正な形式のデータが見つかりました (CSV ファイルの場合は、入力セルデータに引用符、通貨記号、符号、桁修飾文字 (コンマ、小数点、空白など) が含まれていてもエラーとしません。引用符、通貨記号、符号、桁修飾文字を除いた内部データ形式に正規化し、キー比較します)。

ww...w はファイル名を、xx...x はレコード番号を、yy...y はキー番号を示します。

- (S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

- (O)** キーの属性の指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。

あらかじめ 10 進数または日付のデータ形式と一致しないデータ (空白、Low\_value (X'0000'), High\_value (X'FFFF') など) が入力データに含まれることがわかっている場合は、10 進キーのチェックオプション (-cmdopt KEYNOCHK) または日付キーのチェックオプション (-cmdopt DATENOCHK) の適用を検討してください。CSV ファイルの場合は、次の確認を行ってください。

- 環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」に設定したレコードの改行文字「EXS\_RECSEPA」、セルのデリミタ「EXS\_CELLSEPA」、または -inpfile、-catinp、-splitinp パラメタの recsepa ま

- または `cellsepa` オペランドの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。
- 入力セルデータに数字および引用符、通貨記号、符号、桁修飾文字以外の文字が含まれていないか確認してください。CSV 固定小数点数の形式チェックオプション (`-cmdopt CSVFRMCHK`) が適用されている場合は、入力セルデータの有効範囲内を確認してください。数字および引用符、通貨記号、符号、桁修飾文字以外の文字が含まれることがわかっている場合は、10 進キーのチェックオプション (`-cmdopt KEYNOCHK`)、または日付キーのチェックオプション (`-cmdopt DATENOCHK`) の適用を検討してください。
  - 日付時刻データに、数字、引用符、項目区切り文字、および桁修飾文字以外の文字が含まれる、または日付時刻データが指定可能な範囲外であることがわかっている場合は、日付時刻キーのチェックオプション (`-cmdopt DATETIMENOCHK`) の適用を検討してください。日付時刻データの形式については「付録 J 日付時刻データの形式 (SORT EE の機能)」を参照してください。

## KBLS1212-E

---

improper order of merge keys, must be in ascending or descending order: `xx...x yy...y (zz...z)`

マージのキーが昇順または降順になっていません。 `xx...x yy...y (zz...z)`

強制マージオプション (`-cmdopt FMRG`) を指定しないで昇順でマージしたとき、入力ファイルが昇順ソートされていません。または、強制マージオプション (`-cmdopt FMRG`) を指定しないで降順でマージしたとき、入力ファイルが降順ソートされていません。

`xx...x` はファイル名を、`yy...y` はレコード番号を示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

**(O)** キー属性の降順指定 (`r`) が正しいか確認し、間違っているときは正しく指定して再度実行してください。

昇順になっていないファイルを昇順でマージしたい場合、または降順になっていないファイルを降順でマージしたい場合は、強制マージオプション (`-cmdopt FMRG`) を指定する、もしくはマージ前に入力ファイルをソートしてください。

## KBLS1213-E

---

invalid key-parameter specification: `xx...x (zz...z)`

キーパラメタの指定が正しくありません。 `xx...x (zz...z)`

キーパラメタの指定が間違っています。例えば、次のような場合が考えられます。

- CSV ファイルで「`-key`」パラメタを指定している
- CSV ファイル以外で「`-csvkey`」パラメタを指定している
- 「`-sum`」パラメタと「`-csvkey`」パラメタを同時に指定している
- 「`-csvsum`」パラメタと「`-key`」パラメタを同時に指定している

xx...x はパラメタ名を示します。

(O) 正しいパラメタを指定して再度実行してください。

### **KBLS1214-E**

---

cell-number must be specified: xx...x (zz...z)

キーのセル番号が指定されていません。xx...x (zz...z)

キーのセル番号が指定されていません。

xx...x はキー番号を示します。

(O) キーのセル番号を指定して再度実行してください。

### **KBLS1215-E**

---

invalid cell-number specification: xx...x (zz...z)

キーのセル番号の指定が正しくありません。xx...x (zz...z)

キーのセル番号の指定が間違っています。例えば、次のような場合が考えられます。

- 1～5桁の10進数で指定していない
- 1～16384の範囲で指定していない

xx...x はキー番号を示します。

(O) キーのセル番号を修正して再度実行してください。

### **KBLS1216-E**

---

key-precision must be specified: xx...x (zz...z)

キーの有効桁が指定されていません。xx...x (zz...z)

キーの有効桁が指定されていません。

xx...x はキー番号を示します。

(O) キーの有効桁を指定して再度実行してください。

### **KBLS1217-E**

---

invalid key-precision specification: xx...x (zz...z)

キーの有効桁の指定が正しくありません。xx...x (zz...z)

キーの有効桁の指定が間違っています。例えば、次のような場合が考えられます。

- 指定できる範囲以上の数値を指定している（キーの属性によって指定できる有効桁の範囲は異なる）
- 数値以外を指定している

xx...x はキー番号を示します。

(O) キーの有効桁を修正して再度実行してください。

**KBLS1218-E**

---

dateform must be specified: xx...x (zz...z)

日付キーの書式が指定されていません。xx...x (zz...z)

「-csvkey」パラメタに指定した日付キー（キー属性「g」）の書式番号が指定されていません。

xx...x はキー番号を示します。

(O) 日付キーの書式番号を指定して再度実行してください。

**KBLS1219-E**

---

invalid dateform specification: xx...x (zz...z)

日付キーの書式の指定が正しくありません。xx...x (zz...z)

「-csvkey」パラメタに指定した日付キー（キー属性「g」）の書式番号の指定が間違っています。例えば、次のような場合が考えられます。

- 指定できる範囲以上の数値を指定している
- 数値以外を指定している

xx...x はキー番号を示します。

(O) 日付キーの書式番号を修正して再度実行してください。

**KBLS1300-E**

---

number of comparison expression exceeds limited: ww...w xx...x yy...y (zz...z)

比較条件式の個数が制限を超えています。ww...w xx...x yy...y (zz...z)

比較条件式の個数が 1 以上、制限数以下の範囲にありません。

ww...w はパラメタ名を、xx...x はパラメタ番号を、yy...y は比較条件式の制限個数を示します。

(O) 比較条件式の個数を正しく指定して再度実行してください。または、環境ファイル「Exsortconf.ini ファイル（64 ビット版は「Exsortconf64.ini ファイル）」の比較条件式の最大個数の倍数を、設定または変更して再度実行してください。比較条件式の最大個数の倍数については、「15.1.1(16)EXS\_SELNUMMAX」または「15.1.1(19)EXS\_CSVSELNUMMAX」を参照してください。

**KBLS1301-E**

---

cannot specify both include and omit: xx...x yy...y (zz...z)

include と omit を両方指定することはできません。xx...x yy...y (zz...z)

-inpfiler, -catinpf, -splitinpf パラメタ内、または 1 個の -outfile, -catout, -splitout パラメタ内で、include および omit オペランド（CSV ファイルの場合は csvinclude および csvomit オペランド）の両方が指定されています。

xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) include または omit オペランド (CSV ファイルの場合は csvinclude または csvomit オペランド) のどちらかに指定し直して再度実行してください。

### **KBLS1302-E**

---

compare-field attribute must be specified: ww...w xx...x yy...y (zz...z)

比較項目の属性が指定されていません。ww...w xx...x yy...y (zz...z)

比較項目の属性が指定されていません。

ww...w は比較条件式の番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 比較項目を指定して再度実行してください。

### **KBLS1303-E**

---

invalid compare-field attribute specification: ww...w xx...x yy...y (zz...z)

比較項目の属性の指定が正しくありません。ww...w xx...x yy...y (zz...z)

比較項目の属性の指定が間違っています。例えば、次のような場合が考えられます。

- 「j, c, n, m, p, z, x, s, t, g, d, w, u, v」以外の文字を指定した (CSV ファイルの場合は、「j, c, z, g, h, d, w, u, v」だけ指定できる)
- 大文字の英字で指定した
- 入力ファイル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に「w, u, v」を指定した

ww...w は比較条件式の番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 比較項目の属性を修正して再度実行してください。

### **KBLS1304-E**

---

invalid combination of compare-field and compare-field: ww...w xx...x yy...y (zz...z)

比較項目と比較項目の組み合わせが正しくありません。ww...w xx...x yy...y (zz...z)

比較項目のデータ形式の組み合わせが間違っています。例えば、m と n の組み合わせを指定した場合などが考えられます。

ww...w は比較条件式の番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 比較項目のデータ形式を修正して再度実行してください。

### **KBLS1305-E**

---

starting byte position of compare-field must be specified: ww...w xx...x yy...y (zz...z)

比較項目の開始バイト位置が指定されていません。ww...w xx...x yy...y (zz...z)

比較項目の開始バイト位置が指定されていません。

ww…w は比較条件式の番号を、xx…x はパラメタ名を、yy…y はパラメタ番号を示します。

(O) 比較項目の開始バイト位置を指定して再度実行してください。

### **KBLS1306-E**

---

invalid starting byte position of compare-field: ww…w xx…x yy…y (zz…z)

比較項目の開始バイト位置が正しくありません。ww…w xx…x yy…y (zz…z)

比較項目の開始バイト位置の指定が間違っています。例えば、次のような場合が考えられます。

- 6桁以上の数値を指定した
- 数値以外を指定した
- 入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定し、かつ比較項目の開始バイト位置に 2 の倍数ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定し、かつ比較項目の開始バイト位置に 4 の倍数ではない数値を指定した

ww…w は比較条件式の番号を、xx…x はパラメタ名を、yy…y はパラメタ番号を示します。

(O) 比較項目の開始バイト位置を修正して再度実行してください。

入力ファイルがテキストファイルまたは CSV ファイルの場合は、入力データのエンコードの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。入力データのエンコードは、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル」)」の「EXS\_CODETYPE」または `-inpf`、`-catinp`、`-splitinp` パラメタの `codetype` オペランドで指定します。

### **KBLS1307-E**

---

compare-field byte length must be specified: ww…w xx…x yy…y (zz…z)

比較項目のバイト長が指定されていません。ww…w xx…x yy…y (zz…z)

比較項目のバイト長、または文字数が指定されていません。

ww…w は比較条件式の番号を、xx…x はパラメタ名を、yy…y はパラメタ番号を示します。

(O) 比較項目のバイト長、または文字数を指定して再度実行してください。

### **KBLS1308-E**

---

invalid byte length for compare-field: ww…w xx…x yy…y (zz…z)

**比較項目のバイト長の指定が正しくありません。ww…w xx…x yy…y (zz…z)**

比較項目のバイト長、または文字数の指定が間違っています。例えば、次のような場合が考えられます。

- 指定できる範囲以上の数値を指定した（比較項目の属性によって指定できるバイト長、または文字数の範囲は異なる）
- 数値以外を指定した
- 入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-16LE または UTF-16BE）」を指定し、かつ比較項目のバイト長に 2 の倍数ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-32LE または UTF-32BE）」を指定し、かつ比較項目のバイト長に 4 の倍数ではない数値を指定した

ww…w は比較条件式の番号を、xx…x はパラメタ名を、yy…y はパラメタ番号を示します。

**(O)** 比較項目のバイト長、または文字数を正しく指定して再度実行してください。

入力ファイルがテキストファイルまたは CSV ファイルの場合は、入力データのエンコードの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。入力データのエンコードは、環境ファイル「Exsortconf.ini ファイル（64 ビット版は「Exsortconf64.ini ファイル）」の「EXS\_CODETYPE」または `-inpf`、`-catin`、`-splitin` パラメタの `codetype` オペランドで指定します。

**KBLS1309-E**

sum of compare-field position and length is greater than record length: ww…w xx…x yy…y (zz…z)

**比較項目の項目位置+項目長がレコード長を超えています。ww…w xx…x yy…y (zz…z)**

比較項目の項目位置+項目長が入力ファイルのレコード長を超えています。可変長ファイルの場合、最小レコード長を超えています。または、入力ファイルに不当に短いデータがあります。

ww…w は比較条件式の番号を（可変長ファイルまたはテキストファイルの場合、0 が表示される）xx…x はパラメタ名を、yy…y はパラメタ番号を示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

**(O)** 比較項目の項目位置、項目長を正しく指定して再度実行してください。

**KBLS1310-E**

sum of compare-field position and length include record length area: ww…w xx…x yy…y (zz…z)

**比較項目の項目位置+項目長がレコード長領域を含んでいます。ww…w xx…x yy…y (zz…z)**

可変長レコードの場合、比較項目の項目位置+項目長が入力レコードのレコード長領域



を含んでいます。次の条件を両方満たす場合は該当します。

- `-locatajst` パラメタを指定している
- 可変長ファイルで、比較項目の項目位置+項目長が 1～4 バイト目を含んでいる

`ww...w` は比較条件式の番号を、`xx...x` はパラメタ名を、`yy...y` はパラメタ番号を示します。

- (O) レコード長領域を除いた比較項目の項目位置、項目長を指定して再度実行してください。

### **KBLS1311-E**

---

relative operator must be specified: `ww...w xx...x yy...y (zz...z)`

比較演算子が指定されていません。`ww...w xx...x yy...y (zz...z)`

比較演算子が指定されていません。

`ww...w` は比較条件式の番号を、`xx...x` はパラメタ名を、`yy...y` はパラメタ番号を示します。

- (O) 比較演算子を指定して再度実行してください。

### **KBLS1312-E**

---

invalid relative operator specification: `ww...w xx...x yy...y (zz...z)`

比較演算子の指定が正しくありません。`ww...w xx...x yy...y (zz...z)`

比較演算子の指定が間違っています。例えば、次のような場合が考えられます。

- 比較演算子が指定されていない
- 比較演算子に「=, !=, >, >=, <, <=」以外を指定した

`ww...w` は比較条件式の番号を、`xx...x` はパラメタ名を、`yy...y` はパラメタ番号を示します。

- (O) 比較演算子を正しく指定して再度実行してください。

### **KBLS1313-E**

---

qualification or compare-field must be specified: `ww...w xx...x yy...y (zz...z)`

条件値または比較項目が指定されていません。`ww...w xx...x yy...y (zz...z)`

条件値または比較項目が指定されていません。

`ww...w` は比較条件式の番号を、`xx...x` はパラメタ名を、`yy...y` はパラメタ番号を示します。

- (O) 条件値または比較項目を指定して再度実行してください。

### **KBLS1314-E**

---

invalid qualification: `ww...w xx...x yy...y (zz...z)`

### 条件値の指定が正しくありません。ww…w xx…x yy…y (zz…z)

条件値の指定が間違っています。例えば、次のような場合が考えられます。

- 比較項目の属性と条件値の組み合わせが間違っている（例えば、日付属性（g）以外の比較項目に対して、日付条件値を指定したなど）
- 16進数条件値の指定で奇数桁を指定した
- 16進数条件値の指定で「0～9」および「A～F」以外を指定した
- 10進数条件値の指定で符号を後ろに指定した
- 10進数条件値または日付条件値の指定で数値および符号以外を指定した
- CSV ファイルの場合、固定小数点数条件値の指定で、数値、小数点、および符号以外を指定した
- 「C' 文字列'」、「N' 固定小数点数'」、「X'16 進数'」、「D' 日付'」、「NULL (CSV ファイルだけ)」、または「H' 日付 T 時刻' (CSV ファイルだけ)」が指定されていない
- 条件値をシングルクォーテーションで囲んでいない (CSV ファイルの NULL 条件値を除く)
- 符号なし 2 進数の比較項目に対する 10 進数条件値に符号 (-,+ ) を指定している
- CSV ファイル以外の日付条件値の指定で、先頭の 1 文字目にスペースおよび「0～9」以外を指定した。または 2 文字目に「0～9」以外を指定した
- CSV ファイルの日付条件値の指定で「0～9」および「マスク文字 (アスタリスク)」以外を指定した
- CSV ファイルの日付時刻条件値の指定で「0～9」、「マスク文字 (アスタリスク)」および「T」以外を指定した
- CSV ファイルの日付時刻条件値またはマスク日付条件値で指定した桁が、比較項目の有効桁と一致していない
- CSV ファイルの日付時刻条件値に範囲外の値を指定した日付時刻条件値の指定可能な範囲については、「5.8 -inpfile パラメタ」の「5.8.17 {csvinclude | csvomit} オペランド」を参照してください。
- コマンドおよびパラメタファイルに記述した日本語 (マルチバイト文字) を含む文字条件値で、入力データのエンコードに変換できない文字がある
- 環境変数名「%～%」が、設定されていない
- 「C' 文字列'」以外の条件値に、環境変数名を「%%」だけで指定している（%～%で囲んだ変数名を指定する）
- exsort コマンドで、「C' 文字列'」以外の条件値に「%～%」で囲んだ環境変数名を指定している（環境変数名「%～%」はパラメタファイルだけ指定できる。UNIX ではコマンドで指定できない。Windows ではコマンドに指定した環境変数はコマンドのシェル・スクリプトで置換される）

ww…w は比較条件式の番号を、xx…x はパラメタ名を、yy…y はパラメタ番号を示します。

#### (O) 条件値を正しく指定して再度実行してください。

パラメタファイルに記述した日本語を含む文字条件値がエラーとなる場合は、16 進数の文字コード（「X'16 進数'」形式）で条件値を指定して再実行してください。

**KBLS1316-E**

---

invalid logical operator specification: ww...w xx...x yy...y (zz...z)

論理演算子の指定が正しくありません。ww...w xx...x yy...y (zz...z)

論理演算子の指定が間違っています。例えば、次のような場合が考えられます。

- 論理演算子が指定されていない
- 論理演算子以外を指定している（指定できる記号は、「AND」または「OR」の2種類）

ww...w は比較条件式の番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 論理演算子を正しく指定して再度実行してください。

**KBLS1317-E**

---

invalid position of parenthesis: ww...w xx...x yy...y (zz...z)

括弧の位置が正しくありません。ww...w xx...x yy...y (zz...z)

丸括弧の指定位置が間違っています。例えば、次のような場合が考えられます。

- 論理演算子（「AND」または「OR」）と丸括弧の間に、空白またはタブが指定されていない
- 比較演算子（「=」または「>」など）の前後に丸括弧の指定がある

ww...w は比較条件式の番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 括弧を指定できる位置に移動して再度実行してください。

**KBLS1318-E**

---

left parenthesis and right parenthesis is not pair: xx...x yy...y (zz...z)

左括弧と右括弧が対で指定されていません。xx...x yy...y (zz...z)

左括弧と右括弧が対になっていません。

xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 左括弧と右括弧が対になるように修正して再度実行してください。

**KBLS1319-E**

---

too many total compare-field length: ww...w xx...x yy...y (zz...z)

比較項目の合計長が制限を超えています。ww...w xx...x yy...y (zz...z)

比較項目の合計長が制限サイズを超えています。

ww...w はパラメタ名を、xx...x はパラメタ番号を、yy...y は制限サイズを示します。

(O) 比較項目の個数、項目長を正しく指定して再度実行してください。または、環境ファイル「Exsortconf.ini ファイル（64ビット版は「Exsortconf64.ini ファイル）」に、比較項目の最大個数の倍数を設定または変更して再度実行してください。比較項目の最

大個数の倍数については、「15.1.1(16)EXS\_SELNUMMAX」または「15.1.1(19)EXS\_CSVSELNUMMAX」を参照してください。

## KBLS1320-E

invalid compare-field data: ww...w xx...x yy...y (zz...z)

比較項目にエラーデータを検出しました。ww...w xx...x yy...y (zz...z)

10 進数の比較項目について、入力データ中に不正な形式のデータが見つかりました (CSV ファイルの場合は、入力セルデータに引用符、通貨記号、符号、桁修飾文字 (コンマ、小数点、空白など) が含まれていてもエラーとしません。引用符、通貨記号、符号、桁修飾文字を除いた内部データ形式に正規化し、キー比較します)。

ww...w はファイル名を、xx...x はレコード番号を、yy...y は比較条件式の番号を示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

**(O)** 比較項目の属性の指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。あらかじめ 10 進数および日付のデータ形式と一致しないデータ (空白、Low\_value (X'0000'), High\_value (X'FFFF') など) が入力データに含まれることがわかっている場合は、10 進キーのチェックオプション (-cmdopt KEYNOCHK) または日付キーのチェックオプション (-cmdopt DATENOCHK) の適用を検討してください。CSV ファイルの場合は、次の確認を行ってください。

- 環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」) に設定したレコードの改行文字「EXS\_RECSEPA」、セルのデリミタ「EXS\_CELLSEPA」、または -inpf, -catinp, -splitinp パラメタの recsepa または cellsepa オペランドの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。
- 入力セルデータに数字および引用符、通貨記号、符号、桁修飾文字以外の文字が含まれていないか確認してください。CSV 数値データ (固定小数点数) の形式チェックオプション (-cmdopt CSVFRMCHK) が適用されている場合は、入力セルデータの有効範囲を確認してください。数字および引用符、通貨記号、符号、桁修飾文字以外の文字が含まれることがわかっている場合は、10 進キーのチェックオプション (-cmdopt KEYNOCHK)、または日付キーのチェックオプション (-cmdopt DATENOCHK) の適用を検討してください。
- 日付時刻データに、数字、引用符、項目区切り文字、および桁修飾文字以外の文字が含まれる、または日付時刻データが指定可能な範囲外であることがわかっている場合は、日付時刻キーのチェックオプション (-cmdopt DATETIMENOCHK) の適用を検討してください。日付時刻データの形式については「付録 J 日付時刻データの形式 (SORT EE の機能)」を参照してください。

**KBLS1321-E**

---

nest level of parenthesis exceeds limited: vv...v ww...w xx...x yy...y (zz...z)

括弧のネストレベルが制限を超えています。vv...v ww...w xx...x yy...y (zz...z)

括弧のネストレベルが制限を超えています。

vv...v は比較条件式の番号を、ww...w はパラメタ名を、xx...x はパラメタ番号を、yy...y は制限値を示します。

(O) 括弧のネストレベルが制限を超えないように修正して、再度実行してください。

**KBLS1322-E**

---

invalid condition-expression specification: xx...x yy...y (zz...z)

選択条件式の指定が正しくありません。xx...x yy...y (zz...z)

選択条件式の指定が間違っています。例えば、次のような場合が考えられます。

- CSV ファイルで「include」または「omit」オペランドを指定している
- CSV ファイル以外で「csvinclude」または「csvomit」オペランドを指定している

xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 正しいオペランドを指定して再度実行してください。

**KBLS1323-E**

---

cell number of compare-field must be specified : ww...w xx...x yy...y (zz...z)

比較項目のセル番号が指定されていません。ww...w xx...x yy...y (zz...z)

比較項目のセル番号が指定されていません。

ww...w は選択条件式の番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 比較項目のセル番号を指定して再度実行してください。

**KBLS1324-E**

---

invalid cell number of compare-field specification : ww...w xx...x yy...y (zz...z)

比較項目のセル番号の指定が正しくありません。ww...w xx...x yy...y (zz...z)

比較項目のセル番号の指定が間違っています。例えば、次のような場合が考えられます。

- 1 ～ 5 桁の 10 進数で指定していない。
- 1 ～ 16,384 の範囲で指定していない。

ww...w は選択条件式の番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 比較項目のセル番号を指定して再度実行してください。

### **KBLS1325-E**

---

compare-field-precision must be specified : ww...w xx...x yy...y (zz...z)

比較項目の有効桁が指定されていません。ww...w xx...x yy...y (zz...z)

比較項目の有効桁が指定されていません。

ww...w は選択条件式の番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 比較項目の有効桁を指定して再度実行してください。

### **KBLS1326-E**

---

invalid compare-field-precision specification : ww...w xx...x yy...y (zz...z)

比較項目の有効桁の指定が正しくありません。ww...w xx...x yy...y (zz...z)

比較項目の有効桁の指定が間違っています。例えば、次のような場合が考えられます。

- 指定できる範囲以上の数値を指定している（比較項目の属性によって指定できる有効桁の範囲は異なる）
- 数値以外を指定している

(O) 比較項目の有効桁を修正して再度実行してください。

### **KBLS1327-E**

---

dateform of compare-field must be specified : ww...w xx...x yy...y (zz...z)

日付項目の書式番号が指定されていません。ww...w xx...x yy...y (zz...z)

項目属性「g」の比較項目の書式番号が指定されていません。

ww...w は選択条件式の番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 日付項目の書式番号を指定して再度実行してください。

### **KBLS1328-E**

---

invalid dateform of compare-field specification : ww...w xx...x yy...y (zz...z)

日付項目の書式番号の指定が正しくありません。ww...w xx...x yy...y (zz...z)

項目属性「g」の比較項目の書式番号の指定が間違っています。例えば、次のような場合が考えられます。

- 指定できる範囲以外の数値を指定している
- 数値以外を指定している

ww...w は選択条件式の番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 比較項目の書式番号を修正して再度実行してください。

## KBLS1400-E

---

too many sum-fields: xx...x (zz...z)

集約項目の個数が制限を超えています。xx...x (zz...z)

集約項目の個数が制限数を超えています。

xx...x は集約項目の制限個数を示します。

(O) 集約項目の個数を正しく指定して再度実行してください。または、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」」の集約項目の最大個数の倍数を、設定または変更して再度実行してください。集約項目の最大個数の倍数については、「15.1.1(15)EXS\_SUMNUMMAX」または「15.1.1(18)EXS\_CSVSUMNUMMAX」を参照してください。

## KBLS1401-E

---

sum-field attribute must be specified: xx...x (zz...z)

集約項目の属性が指定されていません。xx...x (zz...z)

集約項目の属性が指定されていません。

xx...x は集約項目の番号を示します。

(O) 集約項目の属性を指定して再度実行してください。

## KBLS1402-E

---

invalid sum-field attribute specification: xx...x (zz...z)

集約項目の属性の指定が正しくありません。xx...x (zz...z)

集約項目の属性の指定が間違っています。例えば、次のような場合が考えられます。

- 「n, m, p, z, x, s, t」以外の文字を指定した (CSV ファイルの場合は、「z」だけ指定できる)
- 大文字の英字で指定した

xx...x は集約項目の番号を示します。

(O) 集約項目の属性を修正して再度実行してください。

## KBLS1403-E

---

starting byte position of sum-field must be specified: xx...x (zz...z)

集約項目の開始バイト位置が指定されていません。xx...x (zz...z)

集約項目の開始バイト位置が指定されていません。

xx...x は集約項目の番号を示します。

(O) 集約項目の開始バイト位置を指定して再度実行してください。

## KBLS1404-E

---

invalid starting byte position of sum-field: xx...x (zz...z)

集約項目の開始バイト位置の指定が正しくありません。xx...x (zz...z)

集約項目の開始バイト位置の指定が間違っています。例えば、次のような場合が考えられます。

- 6桁以上の数値を指定した
- 数値以外を指定した
- 入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-16LE または UTF-16BE）」を指定し、かつ集約項目の開始バイト位置に 2 の倍数ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-32LE または UTF-32BE）」を指定し、かつ集約項目の開始バイト位置に 4 の倍数ではない数値を指定した

xx...x は集約項目の番号を示します。

(O) 集約項目の開始バイト位置を修正して再度実行してください。

入力ファイルがテキストファイルまたは CSV ファイルの場合は、入力データのエンコードの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。入力データのエンコードは、環境ファイル「Exsortconf.ini ファイル（64 ビット版は「Exsortconf64.ini ファイル）」の「EXS\_CODETYPE」または `-inpf`、`-catinp`、`-splitinp` パラメタの `codetype` オペランドで指定します。

## KBLS1405-E

---

sum-field byte length must be specified: xx...x (zz...z)

集約項目のバイト長が指定されていません。xx...x (zz...z)

集約項目のバイト長が指定されていません。

xx...x は集約項目の番号を示します。

(O) 集約項目のバイト長を指定して再度実行してください。

## KBLS1406-E

---

invalid byte length for sum-field: xx...x (zz...z)

集約項目のバイト長の指定が正しくありません。xx...x (zz...z)

集約項目のバイト長の指定が間違っています。例えば、次のような場合が考えられます。

- 指定できる範囲以上の数値を指定した（集約項目の属性によって指定できるバイト長の範囲は異なる）
- 数値以外を指定した

xx...x は集約項目の番号を示します。



- (O) 集約項目のバイト長を修正して再度実行してください。

### **KBLS1407-E**

---

sum of sum-field position and length is greater than record length: xx...x (zz...z)

集約項目の項目位置+項目長がレコード長を超えています。xx...x (zz...z)

集約項目の項目位置+項目長が入力ファイルのレコード長を超えています。可変長ファイルの場合、最小レコード長を超えています。または、入力ファイルに不当に短いデータがあります。

xx...x は集約項目の番号を示します（可変長ファイル、またはテキストファイルの場合は 0 が表示されます）。

(S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

- (O) キーの開始バイト位置、バイト長を正しく指定して再度実行してください。

### **KBLS1408-E**

---

sum of sum-field position and length include record length area: xx...x (zz...z)

集約項目の項目位置+項目長がレコード長領域を含んでいます。xx...x (zz...z)

可変長レコードの場合、集約項目の項目位置+項目長が入力レコードのレコード長領域を含んでいます。次の条件を両方満たす場合は該当します。

- 「-locatajst」パラメタの指定している
- 集約項目の項目位置+項目長が 1 ～ 4 バイト目を含んでいる

xx...x は集約項目の番号を示します。

- (O) レコード長領域を除いた集約項目の項目位置、項目長を指定して再度実行してください。

### **KBLS1409-E**

---

duplicate starting byte position of sum-field and key: xx...x (zz...z)

集約項目とキーが重なっています。xx...x (zz...z)

集約項目とキーの位置が重なっています。

xx...x は集約項目の番号を示します。

(S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

- (O) 集約項目の項目位置、項目長を正しく指定して再度実行してください。

### **KBLS1410-E**

---

starting byte position of sum-field and another sum-field are identical: xx...x (zz...z)

**集約項目が他の集約項目と重なっています。xx…x (zz…z)**

集約項目の項目位置がほかの集約項目と重なっています。  
xx…x は集約項目の番号を示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。
- (O) 集約項目の項目位置、項目長を正しく指定して再度実行してください。

**KBLS1411-E**

---

invalid sum-field data: ww…w xx…x yy…y (zz…z)

**集約項目にエラーデータを検出しました。ww…w xx…x yy…y (zz…z)**

10 進数のキーについて、入力データ中に不正な形式のデータが見つかりました (CSV ファイルの場合は、入力セルデータに引用符、通貨記号、符号、桁修飾文字 (コンマ、小数点、空白) が含まれていてもエラーとしません。編集文字を除いた内部データ形式に正規化し、キー比較します)。

ww…w はファイル名を、xx…x はレコード番号を、yy…y は集約項目の番号を示します。

(S) オープンしたファイルをすべてクローズし、処理を中断します。またデバッグ情報ファイルを生成します。

(O) 集約項目の属性の指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。あらかじめ 10 進数のデータ形式と一致しないデータ (空白, Low\_value (X'0000'), High\_value (X'FFFF') など) が入力データに含まれることがわかっている場合、10 進集約項目のチェックオプション (-cmdopt SUMNOCHK) の適用を検討してください。CSV ファイルの場合は、次の確認を行ってください。

- 環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」に設定したレコードの改行文字「EXS\_RECSEPA」、セルのデリミタ「EXS\_CELLSEPA」、または -inpf, -catinp, -splitinp パラメタの recsepa または cellsepa オペランドの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。
- 入力セルデータに数字および編集文字以外の文字が含まれていないか確認してください。数字および編集文字以外の文字が含まれることがわかっている場合は、10 進キーのチェックオプション (-cmdopt KEYNOCHK) の適用を検討してください。

**KBLS1412-E**

---

sum-field data overflowed: xx…x (zz…z)

**集約結果がオーバーフローしました。xx…x (zz…z)**

集約処理で、集約結果がオーバーフローしました。  
xx…x は、集約項目の番号を示します。

(S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情

報ファイルを生成します。

(O) 集約項目の指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。

### **KBLS1413-E**

---

too many total sum-field length : xx...x yy...y (zz...z)

集約項目の合計長が制限を超えています。xx...x yy...y (zz...z)

集約項目の合計長が制限サイズを超えています。

「-sum」パラメタの場合は、「-len」に指定したバイト長の合計となります。

「-csvsum」パラメタの場合は、「-p」に指定した有効桁の合計となります。

xx...x は集約項目の番号を、yy...y は制限サイズを示します。

(O) 集約項目の個数、バイト長を正しく指定して再度実行してください。または、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル」)」の集約項目の合計長の制限サイズの倍数を、設定または変更して再度実行してください。集約項目の合計長の制限サイズの倍数については、「15.1.1(15)EXS\_SUMNUMMAX」または「15.1.1(18)EXS\_CSVSUMNUMMAX」を参照してください。

### **KBLS1414-E**

---

cell number of sum-field must be specified : xx...x (zz...z)

集約項目のセル番号が指定されていません。xx...x (zz...z)

集約項目のセル番号が指定されていません。

xx...x は集約項目の番号を示します。

(O) 集約項目の集約対象セル番号を指定して再度実行してください。

### **KBLS1415-E**

---

invalid cell number of sum-field specification : xx...x (zz...z)

集約項目のセル番号の指定が正しくありません。xx...x (zz...z)

集約項目のセル番号の指定が間違っています。例えば、次のような場合が考えられます。

- 1 ~ 5 桁の 10 進数で指定していない
- 1 ~ 16,384 の範囲で指定していない

xx...x は集約項目の番号を示します。

(O) 集約項目の集約対象セル番号を修正して再度実行してください。

### **KBLS1416-E**

---

sum-field precision must be specified : xx...x (zz...z)

集約項目の有効桁が指定されていません。xx...x (zz...z)

集約項目の有効桁が指定されていません。

xx…x は集約項目の番号を示します。

(O) 集約項目の有効桁を指定して再度実行してください。

## **KBLS1417-E**

---

invalid sum-field precision specification : xx…x (zz…z)

集約項目の有効桁の指定が正しくありません。xx…x (zz…z)

集約項目の有効桁の指定が間違っています。例えば、次のような場合が考えられます。

- 指定できる範囲以上の数値を指定している
- 数値以外を指定している

xx…x は集約項目の番号を示します。

(O) 集約項目の有効桁を修正して再度実行してください。

## **KBLS1500-E**

---

invalid insertion start byte position: ww…w xx…x yy…y (zz…z)

挿入開始バイト位置の指定が正しくありません。ww…w xx…x yy…y (zz…z)

編集フィールドの挿入開始バイト位置の指定が正しくありません。例えば、次のような場合が考えられます。

- 6 桁以上の数値を指定した
- 数値以外を指定した
- 入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定し、かつ編集フィールドの挿入開始バイト位置に 2 の倍数ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定し、かつ編集フィールドの挿入開始バイト位置に 4 の倍数ではない数値を指定した

ww…w は編集フィールド番号を、xx…x はパラメタ名を、yy…y はパラメタ番号を示します。

(O) 挿入開始バイト位置を修正して再度実行してください。

入力ファイルがテキストファイルまたは CSV ファイルの場合は、入力データのエンコードの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。入力データのエンコードは、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」の「EXS\_CODETYPE」または -inpf, -catinp, -splitinp パラメタの codetype オペランドで指定します。

**KBLS1501-E**

insertion start byte position is in the edit field before it or insert data: ww...w xx...x yy...y (zz...z)

挿入開始バイト位置が、直前の編集フィールド又は挿入データに重なっています。ww...w xx...x yy...y (zz...z)

編集フィールドの挿入開始バイト位置が、直前の編集フィールドまたは挿入データに重なっています。例えば、次のような場合が考えられます。

- 挿入開始バイト位置に、直前の編集フィールドより前の位置を指定している
- 挿入開始バイト位置に、直前の編集フィールドと重なる位置を指定している

ww...w は編集フィールド番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

**(O)** 挿入開始バイト位置を直前の編集フィールドに重ならないように修正して再度実行してください。

**KBLS1502-E**

data position of edit field or insertion data format must be specified: ww...w xx...x yy...y (zz...z)

編集フィールドのデータ位置、または挿入データ形式が指定されていません。ww...w xx...x yy...y (zz...z)

編集フィールドのデータ位置、または挿入データが指定されていません。

ww...w は編集フィールド番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

**(O)** 編集フィールドのデータ位置、または挿入データを指定して再度実行してください。

**KBLS1503-E**

data position of edit field or insertion data format is invalid: ww...w xx...x yy...y (zz...z)

編集フィールドのデータ位置、または挿入データ形式の指定が正しくありません。ww...w xx...x yy...y (zz...z)

編集フィールドのデータ位置、または挿入データ形式の指定が正しくありません。例えば、次のような場合が考えられます。

- データ位置として 6 桁以上の数値を指定した
- データ位置として数値以外を指定した
- 入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定し、かつデータ位置に 2 の倍数ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定し、かつデータ位置に 4 の倍数ではない数値

を指定した

- 挿入データとして「X」, 「Z」, 「X'16進数」, または「C' 文字列」以外を指定した
- 挿入データの 16 進数に, 奇数桁の数値を指定した
- 入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定し, かつ挿入データの 16 進数に, 4 の倍数桁ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定し, かつ挿入データの 16 進数に, 8 の倍数桁ではない数値を指定した
- コマンド指定で, 編集フィールド全体をダブルクォーテーション (") で囲んでいない
- コマンドおよびパラメタファイルに記述した日本語 (マルチバイト文字) を含む文字列挿入データで, 入力データのエンコードに変換できない文字がある

ww...w は編集フィールド番号を, xx...x はパラメタ名を, yy...y パラメタ番号を示します。

**(O)** 編集フィールドのデータ位置, または挿入データを修正して再度実行してください。

入力ファイルがテキストファイルまたは CSV ファイルの場合は, 入力データのエンコードの指定が正しいか確認し, 間違っているときは正しく指定して再度実行してください。入力データのエンコードは, 環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル」)」の「EXS\_CODETYPE」または -inpfiler, -catinp, -splitinp パラメタの codeltype オペランドで指定します。パラメタファイルに記述した日本語を含む文字列挿入データがエラーとなる場合は, 16 進数の文字コード (「X'16 進数」形式) で挿入データを指定してください。

## **KBLS1504-E**

---

byte length of edit field must be specified: ww...w xx...x yy...y (zz...z)

編集フィールドのバイト長が指定されていません。ww...w xx...x yy...y (zz...z)

編集フィールドのバイト長が指定されていません。

ww...w は編集フィールド番号を, xx...x はパラメタ名を, yy...y はパラメタ番号を示します。

**(O)** 編集フィールドのバイト長を指定して再度実行してください。

## **KBLS1505-E**

---

invalid byte length of edit field: ww...w xx...x yy...y (zz...z)

編集フィールドのバイト長の指定が正しくありません。ww...w xx...x yy...y (zz...z)

編集フィールドのバイト長の指定が正しくありません。例えば, 次のような場合が考えられます。

- 6桁以上の数値を指定した
- 数値以外を指定した
- 0を指定した
- 入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-16LE または UTF-16BE）」を指定し、かつ編集フィールドのバイト長に2の倍数ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-32LE または UTF-32BE）」を指定し、かつ編集フィールドのバイト長に4の倍数ではない数値を指定した

ww...w は編集フィールド番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

**(O)** 編集フィールドのバイト長を修正して再度実行してください。

入力ファイルがテキストファイルまたは CSV ファイルの場合は、入力データのエンコードの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。入力データのエンコードは、環境ファイル「Exsortconf.ini ファイル（64 ビット版は「Exsortconf64.ini ファイル」）」の「EXS\_CODETYPE」または -infile、-catinp、-splitinp パラメタの codetype オペランドで指定します。

## KBLS1507-E

---

data position of edit field and byte length is greater than input record length: ww...w xx...x yy...y (zz...z)

編集フィールドのデータ位置+バイト長が入力レコード長を超えています。ww...w xx...x yy...y (zz...z)

編集フィールドのデータ位置+バイト長が入力レコード長を超えています。

ww...w は編集フィールド番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

**(O)** データ位置+バイト長が入力レコード長を超えないように修正して再度実行してください。

## KBLS1508-E

---

repeat number of insertion data must be specified: ww...w xx...x yy...y (zz...z)

挿入データの繰り返し回数が指定されていません。ww...w xx...x yy...y (zz...z)

挿入データの繰り返し回数が指定されていません。

ww...w は編集フィールド番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示し

ます。

(O) 挿入データの繰り返し回数を指定して再度実行してください。

### **KBLS1509-E**

---

invalid repeat number of insertion data: ww...w xx...x yy...y (zz...z)

挿入データの繰り返し回数の指定が正しくありません。ww...w xx...x yy...y (zz...z)

挿入データの繰り返し回数の指定が正しくありません。例えば、次のような場合が考えられます。

- 6桁以上の数値を指定した
- 数値以外を指定した
- 0を指定した

ww...w は編集フィールド番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 挿入データの繰り返し回数を修正して再度実行してください。

### **KBLS1510-E**

---

invalid boundary: ww...w xx...x yy...y (zz...z)

境界調整の指定が正しくありません。ww...w xx...x yy...y (zz...z)

編集フィールドの境界調整の指定が正しくありません。例えば、「H」、「F」、または「D」以外を指定した場合などが考えられます。

ww...w は編集フィールド番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 編集フィールドの境界調整を修正して再度実行してください。

### **KBLS1511-E**

---

record length after record editing is greater than maximum record length : vv...v ww...w xx...x yy...y (zz...z)

レコード編集後のレコード長が、最大レコード長を超えています。vv...v ww...w xx...x yy...y (zz...z)

レコード編集後のレコード長が、最大レコード長を超えています。例えば、次のような場合が考えられます。

- レコード編集後のレコード長が、最大レコード長を超えている
- レコード編集後のレコード長が、`-outfile`、`-catout`、または `-splitout` パラメタの「reclen」オペランドで指定したレコード長を超えている
- 挿入データのバイト数×繰り返し回数が、最大レコード長を超えている

vv...v はレコード番号を、ww...w は最大レコード長を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。



(O) レコード編集後のレコード長が、最大レコード長を超えないように修正して再度実行してください。

### **KBLS1512-E**

---

record length after record editing is not equal specified record length: vv...v ww...w xx...x yy...y (zz...z)

レコード編集後のレコード長が、指定レコード長と一致しません。vv...v ww...w xx...x yy...y (zz...z)

レコード編集後のレコード長が、`-outfile`、`-catout`、または `-splitout` パラメタの「reclen」オペランドで指定したレコード長と一致しません。

vv...v はレコード編集後のレコード長を、ww...w は reclen オペランドの指定値を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) レコード編集後のレコード長と指定レコード長が一致するように修正して再度実行してください。

### **KBLS1513-E**

---

record length after record editing is less than minimum record length: uu...u vv...v ww...w xx...x yy...y (zz...z)

レコード編集後のレコード長が、最小レコード長を下回っています。uu...u vv...v ww...w xx...x yy...y (zz...z)

レコード編集後のレコード長が、`-outfile`、`-catout`、または `-splitout` パラメタの「minlen」オペランドで指定した最小レコード長を下回っています。

uu...u はレコード番号を、vv...v はレコード編集後のレコード長を、ww...w は「minlen」オペランドの指定値を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) レコード編集後のレコード長が最小レコード長以上となるように修正して再度実行してください。

### **KBLS1514-E**

---

number of edit format exceeds limited: ww...w xx...x yy...y (zz...z)

フォーマットの個数が制限を超えています。ww...w xx...x yy...y (zz...z)

編集フィールドのフォーマットの個数が制限を超えています。

ww...w は制限数を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) フォーマットの個数を制限数以下に修正して再度実行してください。

### **KBLS1515-E**

---

insertion start byte position and byte length include record length area: ww...w xx...x yy...y (zz...z)

挿入開始バイト位置+バイト長がレコード長領域を含んでいます。ww...w xx...x yy...y (zz...z)

編集フィールドとして指定した挿入開始バイト位置+バイト長が、レコード長領域を含んでいます。

ww...w は編集フィールド番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 挿入開始バイト位置+バイト長がレコード長領域を含まないように修正して、再度実行してください。

### **KBLS1516-E**

---

data position of edit field and byte length include record length area: ww...w xx...x yy...y (zz...z)

編集フィールドのデータ位置+バイト長がレコード長領域を含んでいます。ww...w xx...x yy...y (zz...z)

編集フィールドとして指定したデータ位置+バイト長が、レコード長領域を含んでいます。

ww...w は編集フィールド番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) データ位置+バイト長がレコード長領域を含まないように修正して、再度実行してください。

### **KBLS1517-E**

---

include new line at insert data: ww...w xx...x yy...y (zz...z)

挿入データに改行コードを含んでいます。ww...w xx...x yy...y (zz...z)

編集フィールドとして指定した挿入データに改行文字があります。

ww...w は編集フィールド番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 挿入データに改行文字を含まないように修正して再度実行してください。

### **KBLS1518-E**

---

invalid insertion cell position specification : ww...w xx...x yy...y (zz...z)

セルの挿入位置の指定が正しくありません。ww...w xx...x yy...y (zz...z)

セルの挿入位置の指定が正しくありません。例えば、次のような場合が考えられます。

- 1～5桁の10進数で指定していない
- 1～16,384の範囲で指定していない

ww...w は編集フォーマット番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) 挿入位置を修正して再度実行してください。

**KBLS1519-E**

insertion cell position is before edit cell : ww...w xx...x yy...y (zz...z)

セルの挿入位置が、前の編集セルに重なっています。ww...w xx...x yy...y (zz...z)

セルの挿入位置が、前の編集セルに重なっています。例えば、次のような場合が考えられます。

- 前の編集セルの範囲より前の位置を指定している
- 前の編集セルの範囲と重なる位置を指定している

ww...w は編集フォーマット番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

**(O)** 挿入位置を前の編集セルに重ならないように修正して再度実行してください。

**KBLS1520-E**

edit cell position or insertion data format must be specified : ww...w xx...x yy...y (zz...z)

編集セルの位置、または挿入データが指定されていません。ww...w xx...x yy...y (zz...z)

編集セル位置、または挿入データが指定されていません。

ww...w は編集フォーマット番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

**(O)** 編集セル位置、または挿入データを指定して再度実行してください。

**KBLS1521-E**

edit cell position or insertion data format is invalid : ww...w xx...x yy...y (zz...z)

編集セルの位置、または挿入データの指定が正しくありません。ww...w xx...x yy...y (zz...z)

編集セルの位置、または挿入データの指定が正しくありません。例えば、次のような場合が考えられます。

- セル位置として 1～5 桁の 10 進数で指定していない
- セル位置として 1～16,384 の範囲で指定していない
- 挿入データとして「X」、「Z」、「NULL」、「C'文字列'」、「X'16 進数'」以外を指定した
- 挿入データの 16 進数に、奇数桁の数値を指定した
- 入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定し、かつ挿入データの 16 進数に、4 の倍数桁ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定し、かつ挿入データの 16 進数に、8 の倍数桁ではない数値を指定した
- コマンド指定で、編集フォーマット全体をダブルクォーテーション (") で囲んでいない

ww...w は編集フォーマット番号を、xx...x はパラメタ名を、yy...y はパラメタ番号を示

します。

(O) 編集セル位置, または挿入データを修正して再度実行してください。

### **KBLS1522-E**

---

invalid edit field specification:xx...x yy...y (zz...z)

編集フィールドの指定が正しくありません。xx...x yy...y (zz...z)

編集フィールドの指定が間違っています。例えば、次のような場合が考えられます。

- CSV ファイルで「reorg」オペランド, または「-reorg」パラメタを指定している。
- CSV ファイル以外で「csvreorg」オペランド, または「-csvreorg」パラメタを指定している。

xx...x はパラメタ名を, yy...y はパラメタ番号を示します。

(O) 正しいオペランド, またはパラメタを指定して再度実行してください。

### **KBLS1523-E**

---

edit cell number must be specified: ww...w xx...x yy...y (zz...z)

編集セルの個数が指定されていません。ww...w xx...x yy...y (zz...z)

編集セルの個数が指定されていません。

ww...w は編集フォーマット番号を, xx...x はパラメタ名を, yy...y はパラメタ番号を示します。

(O) 編集セルの個数を指定して再度実行してください。

### **KBLS1524-E**

---

edit cell number is invalid: ww...w xx...x yy...y (zz...z)

編集セルの個数の指定が正しくありません。ww...w xx...x yy...y (zz...z)

編集セルの個数の指定が正しくありません。例えば、次のような場合が考えられます。

- 6 桁以上の数値を指定した
- 数値以外を指定した
- 0 を指定した

ww...w は編集フォーマット番号を, xx...x はパラメタ名を, yy...y はパラメタ番号を示します。

(O) 編集セルの個数を修正して再度実行してください。

### **KBLS1600-E**

---

duplicate parameter file name and input file name: xx...x (zz...z)

パラメタファイル名と入力ファイル名が同じです。xx...x (zz...z)

exsort パラメタファイル名と入力ファイル名に同じファイル名が指定されています。xx

…x はファイル名を示します。

(O) `exsort` パラメタファイル名, または入力ファイル名を変更して再度実行してください。

### **KBLS1601-E**

---

duplicate parameter file name and output file name: xx…x (zz…z)

パラメタファイル名と出力ファイル名が同じです。xx…x (zz…z)

`exsort` パラメタファイル名と出力ファイル名に同じファイル名が指定されています。xx…x はファイル名を示します。

(O) `exsort` パラメタファイル名, または出力ファイル名を変更して再度実行してください。

### **KBLS1602-E**

---

two or more parameter file are specified: (zz…z)

パラメタファイルを複数指定することはできません。(zz…z)

「-p」パラメタに複数の `exsort` パラメタファイル名が指定されています。

(O) `exsort` パラメタファイル名を 1 個にして再度実行してください。

### **KBLS1603-E**

---

must specify parameter file: (zz…z)

パラメタファイルが指定されていません。(zz…z)

「-p」パラメタに `exsort` パラメタファイル名が指定されていません。

(O) `exsort` パラメタファイル名を指定して再度実行してください。

### **KBLS1604-E**

---

invalid parameter file: xx…x (zz…z)

パラメタファイル名が正しくありません。xx…x (zz…z)

`exsort` パラメタファイル名の指定が間違っています。例えば、ファイル名が `MAX_PATH` を超える文字数で指定した場合などが考えられます。

xx…x はファイル名を示します。

(O) `exsort` パラメタファイル名を正しく指定して再度実行してください。

### **KBLS1605-E**

---

I/O error on parameter file: ww…w xx…x yy…y (zz…z)

パラメタファイルに I/O エラーが発生しました。ww…w xx…x yy…y (zz…z)

`exsort` パラメタファイルに I/O エラーが発生しました。エラーの原因については、シス

テムエラーコードを参照してください。

ww...w はファイル名を, xx...x はシステムエラーコードを, yy...y は SORT EE 内部コードを示します。

(O) エラーの原因を取り除いて再度実行してください。

## **KBLS1606-E**

---

parameter file not found: xx...x (zz...z)

パラメタファイルが見つかりません。xx...x (zz...z)

exsort パラメタファイルが見つかりません。例えば, ファイル名の指定が間違っている場合などが考えられます。

xx...x はファイル名を示します。

(O) exsort パラメタファイルの指定を修正して再度実行してください。

## **KBLS1607-E**

---

too many size of parameter file: xx...x yy...y (zz...z)

パラメタファイルのサイズが制限を超えています。xx...x yy...y (zz...z)

exsort パラメタファイルのサイズが制限を超えています。

xx...x はファイル名を, yy...y はファイル制限サイズを示します。

(O) exsort パラメタファイルのサイズを制限サイズ以内に修正して, 再度実行してください。exsort パラメタファイルのサイズが制限を超える場合は, パラメタファイルサイズの制限解除オプション (環境変数「EXSORTPRMSIZEUNLIMIT (64 ビット版は「EXSORT64PRMSIZEUNLIMIT)」, または環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」) の「EXS\_PRMSIZEUNLIMIT」) の適用を検討してください。

## **KBLS1608-E**

---

too many size of parameter line: xx...x yy...y (zz...z)

パラメタ行のサイズが制限を超えています。xx...x yy...y (zz...z)

exsort パラメタファイルの 1 行のサイズが制限を超えています。

xx...x はファイル名を, yy...y は行制限サイズを示します。

(O) exsort パラメタファイルの 1 行サイズを制限サイズ以内に修正して, 再度実行してください。

## **KBLS1609-E**

---

size of parameter is 0 bytes: xx...x (zz...z)

パラメタファイルのサイズが 0 バイトです。xx...x (zz...z)

exsort パラメタファイルの中身に何も指定されていません。または, ファイル名の指定

が間違っている場合などが考えられます。  
 xx…x はファイル名を示します。

(O) 正しい `exsort` パラメタファイルを指定して再度実行してください。

### **KBLS1620-E**

---

input file organization must be specified: (zz…z)

入力ファイルのファイル編成が指定されていません。(zz…z)

「`-inpfile`」パラメタ, 「`-splitinp`」パラメタ, 「`-catinp`」パラメタ, および「`-cpyinp`」パラメタの「`fileorg`」オペランドが指定されていません。入力ファイルのファイル編成を必ず指定してください。

(O) 入力ファイル編成を指定して再度実行してください。

### **KBLS1621-E**

---

invalid input file organization specification: xx…x (zz…z)

入力ファイルのファイル編成の指定が正しくありません。xx…x (zz…z)

「`-inpfile`」パラメタ, 「`-splitinp`」パラメタ, 「`-catinp`」パラメタ, および「`-cpyinp`」パラメタの「`fileorg`」オペランドの指定が間違っています。例えば, 次のような場合が考えられます。

- ファイル編成以外の値を指定している
- ファイル編成を小文字の英字で指定している

xx…x はファイル編成指定値を示します。

(O) ファイル編成の指定値を修正して再実行してください。

### **KBLS1622-E**

---

invalid input file name specification: xx…x (zz…z)

入力ファイル名の指定が正しくありません。xx…x (zz…z)

入力ファイル名の指定が間違っています。例えば, 次のような場合が考えられます。

- ファイル名に `MAX_PATH` を超える文字数を指定している
- 環境変数名を「%%」だけで指定している（%～%で囲んだ変数名を指定する）
- 環境変数名「%～%」にファイル名が関連づけされていない
- `exsort` コマンドに「%～%」で囲んだ環境変数名を指定している（環境変数名「%～%」はパラメタファイルだけ指定できる。UNIXではコマンドで指定できない。

Windowsではコマンドに指定した環境変数はコマンドのシェル・スクリプトで置換される)

xx…x はファイル名, または環境変数名を示します。

(O) 入力ファイル名を正しく指定して再度実行してください。

### **KBLS1625-E**

---

duplicate input file name: xx...x (zz...z)

入力ファイル名が重複しています。xx...x (zz...z)

同じ入力ファイル名を複数指定しています。

xx...x はファイル名を示します。

(O) 入力ファイル名を正しく指定して再度実行してください。

### **KBLS1626-E**

---

too many number of input files: xx...x (zz...z)

入力ファイル名の指定が制限を超えています。xx...x (zz...z)

入力ファイルの個数が制限を超えています。

xx...x はファイル制限数を示します。

(O) 入力ファイル数を減らして再度実行してください。

### **KBLS1627-E**

---

directory name is specified in input file name: xx...x (zz...z)

入力ファイル名にディレクトリ名を指定しています。xx...x (zz...z)

入力ファイル名にディレクトリ名を指定しています。

xx...x はディレクトリ名を示します。

(O) 入力ファイル名を正しく指定して再度実行してください。

### **KBLS1628-E**

---

duplicate input file name and output file name: xx...x (zz...z)

入力ファイル名と出力ファイル名が同じです。xx...x (zz...z)

入力ファイル名と出力ファイル名に同じファイル名を指定しています。同じファイル名を指定できるのは、レコード処理機能が指定されていないソート機能だけです。

- ソート機能以外は、入力ファイル名と出力ファイル名が異ならなければならない

xx...x はファイル名を示します。

(O) 入力ファイル名、または出力ファイル名を変更して再度実行してください。

### **KBLS1629-E**

---

record length must be specified: (zz...z)

入力ファイルのレコード長が指定されていません。(zz...z)

固定長ファイルのレコード長 (「-inpfiler」パラメタ, 「-splitinp」パラメタ, および「-catinp」パラメタの「reclen」オペランド) が指定されていません。

(O) レコード長を指定して再度実行してください。



**KBLS1630-E**

---

invalid record length specification: (zz...z)

入力ファイルのレコード長の指定が正しくありません。(zz...z)

「-inpfile」パラメタ、「-splitinp」パラメタ、および「-catinp」パラメタの「reclen」オペランドの指定が間違っています。例えば、次のような場合が考えられます。

- 入力ファイルが固定長順編成ファイル、またはテキストファイルのとき、レコード長を 1 ～ 65,535 の範囲で指定していない
- 指定したレコード長が、入力ファイルのレコード長と異なっている

(O) レコード長を正しく指定して再度実行してください。

**KBLS1631-E**

---

invalid number that specified by skiprec operand: (zz...z)

skiprec オペランドで指定した件数が正しくありません。(zz...z)

「-inpfile」パラメタ、「-splitinp」パラメタ、および「-catinp」パラメタの「skiprec」オペランドの指定が間違っています。例えば、次のような場合が考えられます。

- 件数を 0 ～ 10 桁の 10 進数で指定していない
- 件数を 0 ～ 2,147,483,647 の範囲で指定していない
- 件数を指定していない

(O) 件数を修正、または指定して再度実行してください。

**KBLS1632-E**

---

invalid number that specified by stopaft operand: (zz...z)

stopaft オペランドで指定した件数が正しくありません。(zz...z)

「-inpfile」パラメタ、「-splitinp」パラメタ、および「-catinp」パラメタの「stopaft」オペランドの指定が間違っています。例えば、次のような場合が考えられます。

- 件数を 0 ～ 10 桁の 10 進数で指定していない
- 件数を 0 ～ 2,147,483,647 の範囲で指定していない
- 件数を指定していない

(O) 件数を修正、または指定して再度実行してください。

**KBLS1633-E**

---

I/O error on input file: ww...w xx...x yy...y (zz...z)

入力ファイルに I/O エラーが発生しました。ww...w xx...x yy...y (zz...z)

入力ファイルに I/O エラーが発生しました。エラーの原因については、システムエラーコードを参照してください。

ww...w はファイル名を、xx...x はエラーレコード番号を、yy...y はシステムエラーコードを示します。

(S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

(O) エラーの原因を取り除いて再度実行してください。

### **KBLS1634-E**

---

cannot open input file: already in use: xx…x (zz…z)

入力ファイルが他で使用中のためオープンできません。xx…x (zz…z)

入力ファイルがほかの処理で使用されているため、オープンできません。例えば、ほかの処理で、SORT プログラムが同じファイルを排他モードでオープンしている場合などが考えられます。

xx…x はファイル名を示します。

(S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

(O) ほかの処理が終わるのを待って、再度実行してください。

### **KBLS1635-E**

---

input file not found: xx…x (zz…z)

入力ファイルが見つかりません。xx…x (zz…z)

入力ファイルが見つかりません。例えば、ファイル名の指定が間違っている場合が考えられます。

xx…x はファイル名を示します。

(O) 入力ファイルを正しく指定して再度実行してください。

### **KBLS1636-E**

---

invalid organized file or corrupted file: xx…x (zz…z)

ファイル編成不正又はファイル破壊が発生しました。xx…x (zz…z)

入力ファイルのファイル編成が不正です。例えば、次のような場合が考えられます。

- 入力ファイル編成と、指定したファイル編成とが一致していない
- 入力ファイルのレコード長領域などが破壊されている
- 入力ファイルに可変長順編成の /dev/null (Windows の場合は nul) を指定している
- 入力ファイルがテキストファイルで、かつ「空行削除オプション (-cmdopt DELBLANKLINE)」の指定がない場合、データがないレコード (改行文字だけの空のレコード) を入力した
- 入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定し、かつファイルサイズが 2 の倍数ではないテキストファイルまたは CSV ファイルを入力ファイルに指定した
- 入力データのエンコードに「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」

または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定し、かつファイルサイズが 4 の倍数ではないテキストファイルまたは CSV ファイルを入力ファイルに指定した

xx...x はファイル名を示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

**(O)** 入力ファイルのファイル編成を正しく指定して再度実行してください。それでもエラーになる場合は、入力ファイルを作成し直して実行してください。

入力ファイルがテキストファイルまたは CSV ファイルの場合は、入力データのエンコードの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。入力データのエンコードは、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル」)」の「EXS\_CODETYPE」または `-inpfile`、`-catinp`、`-splitinp` パラメタの `codetype` オペランドで指定します。

空行を含むテキストファイルを入力したい場合、「空行削除オプション (`-cmdopt DELBLANKLINE`)」を指定して実行してください。

### **KBLS1637-E**

---

two or more input file are specified: (zz...z)

入力ファイルが複数指定されています。(zz...z)

次の場合、入力ファイルを 2 個以上指定することはできません。

- 集約機能だけを実行 (`-function` パラメタの `copy` 機能オプションと、`-sum` パラメタを組み合わせて指定した)

**(O)** 入力ファイルの個数を 1 個に修正して再度実行してください。

### **KBLS1639-E**

---

invalid number that specified by startrec operand: (zz...z)

startrec オペランドで指定した件数が正しくありません。(zz...z)

「`-inpfile`」パラメタ、「`-splitinp`」パラメタ、および「`-catinp`」パラメタの「startrec」オペランドの指定が間違っています。例えば、次のような場合が考えられます。

- 件数を 0 ~ 10 桁の 10 進数で指定していない
- 件数を 0 ~ 2,147,483,647 の範囲で指定していない
- 件数を指定していない

**(O)** 件数を修正、または指定して再度実行してください。

### **KBLS1650-E**

---

invalid output file organization specification: (zz...z)

出力ファイルのファイル編成の指定が正しくありません。(zz...z)

「-outfile」パラメタ、「-splitout」パラメタ、および「-catout」パラメタの「fileorg」オペランドの指定が間違っています。例えば、次のような場合が考えられます。

- ファイル編成以外の値を指定している
- ファイル編成を小文字の英字で指定している
- 入力ファイルと出力ファイルのレコード形式（固定長、可変長、テキスト、または CSV）が異なっている

(O) ファイル編成の指定値を修正して再実行してください。

## **KBLS1651-E**

---

invalid output file name specification: xx...x (zz...z)

出力ファイル名の指定が正しくありません。xx...x (zz...z)

出力ファイル名の指定が間違っています。例えば、次のような場合が考えられます。

- ファイル名を MAX\_PATH を超える文字数で指定している
- 環境変数名を「%%」だけで指定している（%～% で囲んだ変数名を指定する）
- 環境変数名「%～%」にファイル名が関連づけされていない
- exsort コマンドに「%～%」で囲んだ環境変数名を指定している（環境変数名「%～%」は、パラメタファイルだけを指定できる。UNIX ではコマンドで指定できない。Windows ではコマンドに指定した環境変数はコマンドのシェル・スクリプトで置換される）

xx...x はファイル名、または環境変数名を示します。

(O) 出力ファイル名を正しく指定して再度実行してください。

## **KBLS1652-E**

---

duplicate output file name: xx...x (zz...z)

出力ファイル名が重複しています。xx...x (zz...z)

同じ出力ファイル名を複数指定しています。

xx...x はファイル名を示します。

(O) 出力ファイル名を正しく指定して再実行してください。

## **KBLS1653-E**

---

too many number of output files: xx...x (zz...z)

出力ファイル名の指定が制限を超えています。xx...x (zz...z)

出力ファイルの個数が制限を超えています。

xx...x はファイル制限数を示します。

(O) 出力ファイル数を減らして再度実行してください。

**KBLS1654-E**

---

invalid record length of output file: xx...x (zz...z)

出力ファイルのレコード長の指定が正しくありません。xx...x (zz...z)

出力ファイルのレコード長の指定が間違っています。

xx...x はパラメタ番号を示します。

**(O)** レコード長を正しく指定して再度実行してください。

**KBLS1655-E**

---

I/O error on output file: xx...x yy...y (zz...z)

出力ファイルに I/O エラーが発生しました。xx...x yy...y (zz...z)

出力ファイルに I/O エラーが発生しました。エラーの原因については、システムエラーコードを参照してください。

xx...x はファイル名を、yy...y はシステムエラーコードを示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

**(O)** エラーの原因を取り除いて再度実行してください。

**KBLS1656-E**

---

cannot open output file: already in use: xx...x (zz...z)

出力ファイルが他で使用中のためオープンできません。xx...x (zz...z)

出力ファイルがほかの処理で使用されているため、オープンできません。例えば、ほかの処理で、SORT プログラムが同じファイルを排他モードでオープンしている場合などが考えられます。

xx...x はファイル名を示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

**(O)** ほかの処理が終わるのを待って再度実行します。または、出力ファイル名を変更して再度実行してください。

**KBLS1657-E**

---

output file already exists: xx...x (zz...z)

出力ファイルが既に存在するため処理を中止しました。xx...x (zz...z)

出力ファイル名に指定したファイルはすでにあります。

xx...x はファイル名を示します。

**(S)** 処理を中断します。

**(O)** 存在しないファイル名を指定して、再度実行してください。または、-cmdopt

OVWRT（既存ファイル上書きオプション）を指定してください。

### **KBLS1658-E**

---

insufficient memory to create output file: (zz...z)

出力ファイルの領域が確保できません。(zz...z)

ディスクの空き容量が少ないため、出力ファイルが確保できません。

(S) 処理を中断します。

(O) ディスクの領域を空けて、再度実行してください。

### **KBLS1659-E**

---

record length of output file must be specified: xx...x (zz...z)

出力ファイルのレコード長が指定されていません。xx...x (zz...z)

出力ファイルのレコード長が指定されていません。

xx...x はパラメタ番号を示します。

(O) 出力ファイルのレコード長を指定して再度実行してください。

### **KBLS1660-E**

---

record length of output file must be specified: xx...x (zz...z)

出力ファイルの最小レコード長が指定されていません。xx...x (zz...z)

出力ファイルの最小レコード長が指定されていません。

xx...x はパラメタ番号を示します。

(O) 出力ファイルの最小レコード長を指定して再度実行してください。

### **KBLS1661-E**

---

invalid minimum record length of output file: xx...x (zz...z)

出力ファイルの最小レコード長の指定が正しくありません。xx...x (zz...z)

出力ファイルの最小レコード長の指定が正しくありません。例えば、次のような場合が考えられます。

- 6桁以上の数値を指定した
- 数値以外を指定した
- 0を指定した

xx...x はパラメタ番号を示します。

(O) 出力ファイルの最小レコード長を正しく指定して再度実行してください。

### **KBLS1662-E**

---

minimum record length of output file is greater than maximum record length: xx...x (zz...z)

出力ファイルの最小レコード長の指定値が最大レコード長の指定値を超えています。xx...x (zz...z)

出力ファイルの最小レコード長の指定値が、出力ファイルの最大レコード長の指定値を超えています。

xx...x はパラメタ番号を示します。

**(O)** 出力ファイルの最小レコード長、または最大レコード長を正しく指定して、再度実行してください。

### **KBLS1663-E**

---

when record editing is not used, unable to specify record length and minimum record length of output file: xx...x (zz...z)

編集機能を使用しない場合、出力ファイルのレコード長と最小レコード長は指定できません。xx...x (zz...z)

編集機能を使用しない場合、出力ファイルのレコード長と最小レコード長は指定できません。

xx...x はパラメタ番号を示します。

**(O)** 編集機能を使用するか、またはレコード長と最小レコード長の指定を削除して、再度実行してください。

### **KBLS1664-E**

---

when input file organization is S, unable to specify minimum record length of output file: xx...x (zz...z)

入力ファイルが固定長順編成ファイルの場合、出力ファイルの最小レコード長は指定できません。xx...x (zz...z)

入力ファイルが固定長順編成ファイルの場合、出力ファイルの最小レコード長は指定できません。

xx...x はパラメタ番号を示します。

**(O)** 出力ファイルの最小レコード長の指定を削除して再度実行してください。

### **KBLS1665-E**

---

output file name must be specified: (zz...z)

出力ファイル名が指定されていません。(zz...z)

出力ファイル名が指定されていません。「-outfile」パラメタ、「-splitout」パラメタ、および「-catout」パラメタを複数指定した場合は、出力ファイル名を省略することはできません。

**(O)** 出力ファイル名を指定して再実行してください。

## **KBLS1670-E**

---

invalid directory specification: xx...x (zz...z)

一時ファイルのディレクトリの指定が正しくありません。xx...x (zz...z)

一時ファイルのディレクトリの指定が間違っています。例えば、次のような場合が考えられます。

- ディレクトリ名の指定が間違っている
- ディレクトリ名に (MAX\_PATH-1) を超える文字数を指定した
- 環境変数名「% ~ %」が、設定されていない

xx...x はディレクトリ名を示します。

**(O)** ディレクトリ名を正しく指定して再度実行してください。

## **KBLS1671-E**

---

file name is specified in directory of temporary file: xx...x (zz...z)

一時ファイルのディレクトリにファイル名を指定しています。xx...x (zz...z)

一時ファイルのディレクトリにファイル名を指定しています。

xx...x はファイル名を示します。

**(O)** ディレクトリ名を正しく指定して再度実行してください。

## **KBLS1672-E**

---

I/O error on temporary file: xx...x yy...y (zz...z)

一時ファイルに I/O エラーが発生しました。xx...x yy...y (zz...z)

一時ファイルに I/O エラーが発生しました。エラーの原因については、システムエラーコードを参照してください。

xx...x はディレクトリ名を、yy...y はシステムエラーコードを示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成する場合があります。

**(O)** エラーの原因を取り除いて再度実行してください。

## **KBLS1673-E**

---

insufficient memory to create temporary file: (zz...z)

一時ファイルの領域が確保できません。(zz...z)

ディスクの空き容量が少ないため、一時ファイルが確保できません。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。

**(O)** 一時ファイルを割り当てるディスクの領域を空ける、またはディレクトリを変更して再度実行してください。



**KBLS1674-E**

---

directory must be specified: (zz...z)

一時ファイルのディレクトリが指定されていません。(zz...z)

「-sorttmpdir」パラメタ、または「-splittmpdir」パラメタに一時ファイルのディレクトリを指定していません。

(O) 一時ファイルのディレクトリを指定して再度実行してください。

**KBLS1691-E**

---

I/O error on debugging information file: (zz...z)

デバッグ情報ファイルに I/O エラーが発生しました。(zz...z)

デバッグ情報ファイルに I/O エラーが発生しました。

(O) 保守員に連絡してください。

**KBLS1692-E**

---

I/O error on message file: ww...w xx...x yy...y (zz...z)

メッセージファイルに I/O エラーが発生しました。ww...w xx...x yy...y (zz...z)

メッセージ出力先ファイルに I/O エラーが発生しました。エラーの原因については、システムエラーコードを参照してください。

ww...w はファイル名を、xx...x はシステムエラーコードを、yy...y は SORT EE 内部コードを示します。

(S) オープンしたファイルをすべてクローズし、処理を中断します。

(O) エラーの原因を取り除いて再度実行してください。

**KBLS1693-E**

---

I/O error on message file: ww...w xx...x yy...y (zz...z)

メッセージファイルに I/O エラーが発生しました。ww...w xx...x yy...y (zz...z)

Windows の場合：

メッセージライブラリのローディングに失敗しました。

UNIX の場合：

メッセージカタログのオープンに失敗しました。

エラーの原因については、システムエラーコードを参照してください。

ww...w はパス名を、xx...x はシステムエラーコードを、yy...y は SORT EE 内部コードを示します。

(S) オープンしたファイルをすべてクローズし、処理を中断します。

(O) エラーの原因を取り除いて再度実行してください。

## **KBLS1694-E**

---

I/O error on library file: ww...w xx...x yy...y (zz...z)

ライブラリのローディングで I/O エラーが発生しました。ww...w xx...x yy...y (zz...z)

SORT EE ライブラリのローディングに失敗しました。エラーの原因については、システムエラーコードを参照してください。

ww...w はパス名を、xx...x はシステムエラーコードを、yy...y は SORT EE 内部コードを示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。

**(O)** エラーの原因を取り除いて再度実行してください。

## **KBLS1700-E**

---

invalid programming logic: xx...x (zz...z)

プログラム論理エラーが発生しました。xx...x (zz...z)

プログラム不良が発生しました。

xx...x は詳細コードを示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

**(O)** 保守員に連絡してください。

## **KBLS1710-E**

---

SORT program nothing: (zz...z)

SORT が組み込まれていません。(zz...z)

前提 PP の SORT が組み込まれていません。

**(O)** 前提 PP の SORT Version8 を組み込んで再度実行してください。

## **KBLS1711-E**

---

SORT version is old: (zz...z)

SORT のバージョンが古いため処理を中断しました。(zz...z)

前提 PP より古い SORT が組み込まれています。

**(O)** 前提 PP の SORT Version8 を組み込んで再度実行してください。

## **KBLS1720-E**

---

cannot obtain memory of this size: (zz...z)

メモリが確保できません。(zz...z)

ワークバッファ、または、I/O バッファを確保できません。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。
- (O) ワークバッファサイズ、または I/O サイズを修正して再度実行してください。
- ワークバッファサイズの指定  
「-worksize」パラメタ、  
環境変数 EXSORTWORKSIZE、  
環境変数 EXSORT64WORKSIZE、  
環境変数 EXSORT\_HI\_WORKSIZE、  
または、環境変数 EXSORT64\_HI\_WORKSIZE
  - I/O サイズの指定  
「-iosize」パラメタ、  
環境変数 EXSORTIOSIZE、  
環境変数 EXSORT64IOSIZE、  
環境変数 EXSORTINPIOSIZE、  
環境変数 EXSORT64INPIOSIZE、  
環境変数 EXSORTOUTIOSIZE、  
環境変数 EXSORT64OUTIOSIZE、  
環境変数 EXSORTSORTTMPIOSIZE、  
環境変数 EXSORT64SORTTMPIOSIZE、  
環境変数 EXSORTMERGETMPIOSIZE、  
環境変数 EXSORT64MERGETMPIOSIZE、  
環境変数 EXSORTSPLITTMPIOSIZE、  
または、環境変数 EXSORT64SPLITTMPIOSIZE

### **KBLS1721-E**

---

insufficient memory to execute: (zz...z)

メモリが不足しました。(zz...z)

「-worksize」パラメタ、環境変数「EXSORTWORKSIZE」、環境変数「EXSORT64WORKSIZE」、環境変数「EXSORT\_HI\_WORKSIZE」、または環境変数「EXSORT64\_HI\_WORKSIZE」で指定したワークバッファサイズが小さいため、SORT EE が実行できません。

(S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

(O) ワークバッファサイズを大きくして、再度実行してください。

### **KBLS1722-E**

---

Insufficient memory: xx...x yy...y (zz...z)

メモリ不足が発生しました。xx...x yy...y (zz...z)

exsort コマンドを実行するための領域が確保できません。

xx…x は確保しようとした領域のサイズを、yy…y はシステムエラーコードを示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。

**(O)** 再度実行してください。それでもエラーになる場合は保守員に連絡してください。

## **KBLS1730-E**

---

drive is not ready: (zz…z)

ドライブの準備ができていません。(zz…z)

ドライブの準備ができていません。例えば、ドライブにフロッピーディスクがセットされていない場合などが考えられます。

**(O)** エラーの原因を取り除いて再度実行してください。

## **KBLS1731-E**

---

media is read only: (zz…z)

メディアは書き込み禁止です。(zz…z)

メディアへの書き込みができません。例えば、フロッピーディスクが書き込み禁止になっている場合などが考えられます。

**(O)** エラーの原因を取り除いて再度実行してください。

## **KBLS1750-E**

---

cannot input because data is shorter than record length: (zz…z)

データがレコード長より短いため入力できません。(zz…z)

入力ファイルが標準入力ファイルのとき、入力データが指定したレコード長より短いため入力できません。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。

**(O)** 入力データの長さをレコード長の整数倍にして、再度実行してください。

## **KBLS1751-E**

---

cannot input because data is longer than maximum record length: ww…w xx…x yy…y (zz…z)

データが最大レコード長より長い場合入力できません。ww…w xx…x yy…y (zz…z)

入力ファイルがテキストファイルまたは CSV ファイルのとき、入力データに最大レコード長より長いデータがあるため入力できません。

ww…w はファイル名を、xx…x はレコード番号を、yy…y は最大レコード長を示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。

**(O)** 入力ファイルがテキストファイルの場合は、入力ファイルの最大レコード長を大きくして、再度実行してください。入力ファイルが CSV ファイルの場合は、環境ファイル

「Exsortconf.ini ファイル」に設定したレコードの改行文字「EXS\_RECSEPA」または、  
 -inpfile, -catinp, -splitinp パラメタの recsepa オペランドの指定が正しいかどうかを確認し、間違っているときは正しく指定して再度実行してください。

## KBLS1752-E

---

number of cells exceeds limited: ww...w xx...x yy...y (zz...z)

セル数が制限を超えています。ww...w xx...x yy...y (zz...z)

入力ファイルが CSV ファイルのとき、セル数が制限を超えている入力データがあるため入力できません。例えば、次のような場合が考えられます。

- 最大セル数 (16,384 個) を超えている

ww...w はファイル名、xx...x はレコード番号、yy...y は制限数を示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。

**(O)** 環境ファイル「Exsortconf.ini ファイル」に設定したレコードの改行文字「EXS\_RECSEPA」または、-inpfile, -catinp, -splitinp パラメタの recsepa オペランドの指定が正しいかどうかを確認し、間違っているときは正しく指定して再度実行してください。

## KBLS1753-E

---

data exceeded maximum record length : ww...w xx...x yy...y (zz...z)

データが最大レコード長を超えました。ww...w xx...x yy...y (zz...z)

入力ファイルがテキストファイルまたは CSV ファイルのとき、データが最大レコード長を超えたため処理を中断しました。エラー要因として、次のような場合が考えられます。

「テキストファイルの場合」

- -outfile, -catout または -splitout パラメタの recsepa オペランドで、1 バイトの改行文字から 2 バイトの改行文字に変更した。

「CSV ファイルの場合」

- レコード編集機能で編集した結果が最大レコード長を超えた。
- -outfile, -catout, または -splitout パラメタの quottype オペランドで、引用符なしのセルデータに引用符を付加した。
- -outfile, -catout, または -splitout パラメタの cellnum オペランドで、出力セル数を増やした。
- -outfile, -catout, または -splitout パラメタの recsepa オペランドで、1 バイトの改行文字から 2 バイトの改行文字に変更した。
- 集約機能で集約した結果が最大レコード長を超えた（集約した結果のセルの桁は、指定された有効桁に合わせる。また、小数点、符号、桁区切り（コンマ）などを付加した結果、元のセルの長さを超える場合がある）。

ww...w はパラメタ名を、xx...x はパラメタ番号を、yy...y は最大レコード長を示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 最大レコード長を超えないように指定して再度実行してください。

### **KBLS1760-E**

---

system call error: xx···x yy···y (zz···z)

システムコールエラーが発生しました。xx···x yy···y (zz···z)

システムコールエラーが発生しました。エラーの原因については、システムエラーコードを参照してください。

xx···x はシステムエラーコードを、yy···y は SORT EE 内部コードを示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) エラーの原因については、システムエラーコードを参照してください。

### **KBLS1770-E**

---

environment variable is not directory: xx···x yy···y (zz···z)

環境変数の設定値はディレクトリではありません。xx···x yy···y (zz···z)

環境変数、または環境ファイルで指定された値はディレクトリではありません。

xx···x は SORT EE の環境変数名、または SORT の環境変数名（Windows の場合は SORT の環境ファイル名）を、yy···y は SORT の環境変数のオプション、または SORT の環境ファイルのパラメタを示します（SORT EE の環境変数の場合、yy···y は表示されません）。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 環境変数の設定値を正しく指定して再度実行してください。

### **KBLS1771-E**

---

environment variable is not defined: xx···x yy···y (zz···z)

環境変数の設定値が見つかりません。xx···x yy···y (zz···z)

環境変数、または環境ファイルで指定されたディレクトリが見つかりません。

xx···x は SORT EE の環境変数名、または SORT の環境変数名（Windows の場合は SORT の環境ファイル名）を、yy···y は SORT の環境変数のオプション、または SORT の環境ファイルのパラメタを示します（SORT EE の環境変数の場合、yy···y は表示されません）。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 環境変数の設定値を正しく指定して再度実行してください。

### **KBLS1773-E**

---

environment variable invalid: ww···w xx···x yy···y (zz···z)

環境変数の設定値の内容が正しくありません。ww…w xx…x yy…y (zz…z)

環境変数、または環境ファイルで指定された設定値の内容が間違っています。

ww…w は SORT EE の環境変数名、または SORT の環境変数名 (Windows の場合は SORT の環境ファイル名) を、xx…x は SORT の環境変数のオプション、または SORT の環境ファイルのパラメタを、yy…y は SORT EE 内部コードを示します (SORT EE の環境変数の場合、xx…x は表示されません)。

(S) オープンしたファイルをすべてクローズし、処理を中断します。

(O) 環境変数の設定値を正しく指定して再度実行してください。

## **KBLS1900-E**

---

function not support: (zz…z)

サポートしていない機能が指定されています。(zz…z)

サポートしていない機能が指定されています。

(S) オープンしたファイルをすべてクローズし、処理を中断します。

(O) コマンド、またはパラメタファイルを正しく指定して再度実行してください。

## **KBLS2000-E**

---

invalid number that specified by fileno operand (zz…z)

fileno オペランドで指定した値が正しくありません。(zz…z)

「-splitfile」パラメタの「fileno」オペランドの指定が間違っています。

例えば、次のような場合が考えられます。

- 指定できる桁数 (5 桁) を超えている
- 10 進数で指定していない
- 値を指定していない

(O) 「fileno」オペランドの指定値を修正、または指定して再度実行してください。

## **KBLS2001-E**

---

invalid method option specification: xx…x (zz…z)

method オプションの指定が正しくありません。xx…x (zz…z)

「-splitfile」パラメタの「method」オペランドに分割基準オプション (recnt | filesize) 以外が指定されています。

xx…x はオプション指定値を示します。

(O) 「method」オペランドの指定値を修正して再度実行してください。

## **KBLS2010-E**

---

invalid number that specified by recno operand (zz…z)

**recno オペランドで指定した値が正しくありません。(zz…z)**

「-splitdata」パラメタ、または「-splitkey」パラメタの「recno」オペランドの指定が間違っています。

例えば、次のような場合が考えられます。

- レコード数を数値以外で指定している
- 指定できる桁数（10 桁）を超えている
- レコード数に 0 を指定している
- 値を指定していない

**(O)** 「recno」オペランドの指定値を修正、または指定して再度実行してください。

---

**KBLS2011-E**

**invalid number that specified by spno operand (zz…z)**

**spno オペランドで指定した値が正しくありません。(zz…z)**

「-splitdata」パラメタ、または「-splitkey」パラメタで指定した「spno」オペランドの指定が間違っています。

例えば、次のような場合が考えられます。

- サイズを数値以外で指定している
- 単位の指定を、G または M 以外で指定している
- 指定できる桁数（4 桁または 7 桁）を超えている
- サイズに 0 を指定している
- 値を指定していない

**(O)** 「spno」オペランドの指定値を修正、または指定して再度実行してください。

---

**KBLS2012-E**

**equality split information must be specified (zz…z)**

**均等分割情報が指定されていません。(zz…z)**

均等分割情報が指定されていません。

例えば、次のような場合が考えられます。

- 「-splitfile」パラメタだけ指定している
- 「fileno」オペランドを指定していない
- 「method」オペランドを指定していない

**(O)** 「-splitfile」パラメタで、「fileno」オペランドおよび「method」オペランドを指定して再度実行してください。

---

**KBLS2013-E**

**constant split information must be specified (zz…z)**

**定数分割情報が指定されていません。(zz…z)**

定数分割情報が指定されていません。



例えば、次のような場合が考えられます。

- 「-splitdata」パラメタだけ指定している

**(O)** 「-splitdata」パラメタで、「recno」オペランド、または「spno」オペランドを指定して再度実行してください。

## **KBLS2014-E**

---

key split information must be specified (zz...z)

キー分割情報が指定されていません。(zz...z)

キー分割情報が指定されていません。

例えば、次のような場合が考えられます。

- 集約機能を使用しないコピー機能を使用し、「-splitkey」パラメタだけ指定している

**(O)** 「-splitkey」パラメタで、「key」オペランド、または「csvkey」オペランドを指定して再度実行してください。

## **KBLS2015-E**

---

split information must be specified (zz...z)

分割情報が指定されていません。(zz...z)

分割情報が指定されていません。

例えば、次のような場合が考えられます。

- 集約機能を使用しないコピー機能を使用し、「-splitfile」パラメタ、「-splitdata」パラメタ、および「-splitkey」パラメタのすべての指定がない

**(O)** 「-splitfile」パラメタ、「-splitdata」パラメタ、または「-splitkey」パラメタを指定して再度実行してください。「-splitkey」パラメタを指定する場合、「key」オペランドまたは「csvkey」オペランドを指定する必要があります。

## **KBLS2020-I**

---

split file number reached the maximum: xx...x (zz...z)

分割ファイル数が分割できる最大値に達しました。xx...x (zz...z)

指定された分割基準によるファイルの分割では、1要素に対する分割ファイル数が出力ファイル名で付与する追番の上限に達したため、それ以降のファイル分割は行わず、残りレコードを最終分割ファイルへ出力しました。

xx...x は最終分割ファイル名を示します。

**(O)** 必要であれば、分割基準を修正して再度実行してください。

## **KBLS2021-E**

---

invalid splitout specification: yy...y (zz...z)

分割出力ファイルの指定が正しくありません。yy...y (zz...z)

「-splitout」パラメタで指定した出力ファイル格納ディレクトリ、または出力ファイル名の末尾の指定が正しくありません。

yy…y はディレクトリ名、またはファイル名の総称を示します。

例えば、次のような場合が考えられます。

- ディレクトリの指定が間違っている
- 分割後のファイル名が、PATH\_MAX または NAME\_MAX を超える文字数となった
- 指定した出力ファイル格納ディレクトリ名の末尾にスラッシュ (/) が指定されていない
- 環境変数名を「%%」だけで指定している (% ~ % で囲んだ変数名を指定する)
- 環境変数名「% ~ %」に出力ファイル名の末尾、または出力ファイル格納ディレクトリ名が関連づけされていない
- exsort コマンドに「% ~ %」で囲んだ環境変数名を指定している (環境変数名「% ~ %」はパラメタファイルだけ指定でき、UNIX では exsort コマンドで指定できない、Windows では exsort コマンドに指定した環境変数はコマンドのシェル・スクリプトで置換される)

(O) ディレクトリ名または出力ファイル名の末尾を正しく指定して再度実行してください。

## **KBLS2022-E**

---

split-input-file must be specified (zz…z)

分割入力ファイルが指定されていません。(zz…z)

「-splitinp」パラメタで入力ファイルが指定されていません。

(O) 「file」オペランドで入力ファイルを指定して再度実行してください。

## **KBLS2030-E**

---

improper order of split-keys, must be in ascending or descending order: xx…x yy…y (zz…z)

分割キーが昇順または降順になっていません。xx…x yy…y (zz…z)

強制ファイル分割オプション (-cmdopt FSPLIT) を指定しないで昇順で分割したとき、入力ファイルが昇順にソートされていません。または、強制ファイル分割オプション (-cmdopt FSPLIT) を指定しないで降順で分割したとき、入力ファイルが降順にソートされていません。

xx…x はファイル名を、yy…y はレコード番号を示します。

(S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを作成します。

(O) 分割キー属性の降順指定 (r) が正しいか確認し、間違っているときは正しく指定して再度実行してください。昇順になっていないファイルを昇順で分割したい場合、または降順になっていないファイルを降順で分割したい場合は、強制ファイル分割オプション (-cmdopt FSPLIT) を指定してください。またはファイル分割前に入力ファイルをソートしてください。

**KBLS2031-E**

---

invalid number of split-keys: xx...x (zz...z)

分割キーの個数が制限を超えています。xx...x (zz...z)

分割キーの個数が 1 以上、制限数以下の範囲にありません。

xx...x は、分割キーの制限個数を示します。

(O) 分割キーの個数を正しく指定して再度実行してください。

**KBLS2032-E**

---

split-key must be specified: (xx...x)

分割キーが指定されていません。(xx...x)

「-splitkey」パラメタの「key」オペランド、または「csvkey」オペランドが指定されていません。次の機能を使用する場合、「key」オペランドを必ず指定してください（入力ファイルが CSV ファイルの場合は、「csvkey」オペランドとなります）。

- 「-function」パラメタに copy を指定し、「-splitkey」パラメタを指定した

(O) 「key」または「csvkey」オペランドを指定して再度実行してください。

**KBLS2033-E**

---

split-key attribute must be specified: xx...x (zz...z)

分割キーの属性が指定されていません。xx...x (zz...z)

分割キーの属性が指定されていません。

xx...x は分割キー番号を示します。

(O) 分割キーの属性を指定して再度実行してください。

**KBLS2034-E**

---

invalid split-key attribute specification: xx...x (zz...z)

分割キーの属性の指定が正しくありません。xx...x (zz...z)

分割キーの属性の指定が間違っています。次のような場合が考えられます。

- 「r, j, c, n, m, a, b, p, z, x, s, t, f, g, q, d, w, u, v」以外の文字を指定した（CSV ファイルの場合は、「r, j, c, z, g, h, q, d, w, u, v」だけ指定できる）
- 大文字の英字で指定した
- 入力ファイル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に「w, u, v」を指定した

xx...x は分割キー番号を示します。

(O) 分割キーの属性を修正して再度実行してください。

## **KBLS2035-E**

---

starting byte position of split-key must be specified: xx···x (zz···z)

分割キーの開始バイト位置が指定されていません。xx···x (zz···z)

分割キーの開始バイト位置が指定されていません。

xx···x は分割キー番号を示します。

(O) 分割キーの開始バイト位置を指定して再度実行してください。

## **KBLS2036-E**

---

invalid starting byte position of split-key: xx···x (zz···z)

分割キーの開始バイト位置の指定が正しくありません。xx···x (zz···z)

分割キーの開始バイト位置の指定が間違っています。例えば、次のような場合が考えられます。

- 6 桁以上の数値を指定した
- 数値以外を指定した
- 入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定し、かつ分割キーの開始バイト位置に 2 の倍数ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定し、かつ分割キーの開始バイト位置に 4 の倍数ではない数値を指定した

xx···x は分割キー番号を示します。

(O) 分割キーの開始バイト位置を修正して再度実行してください。

入力ファイルがテキストファイルまたは CSV ファイルの場合は、入力データのエンコードの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。入力データのエンコードは、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」の「EXS\_CODETYPE」または `-inpf`, `-catinp`, `-splitinp` パラメタの `codetype` オペランドで指定します。

## **KBLS2037-E**

---

split-key length must be specified: xx···x (zz···z)

分割キーのバイト長が指定されていません。xx···x (zz···z)

分割キーのバイト長、文字数またはマスクビットが指定されていません。

xx···x は分割キー番号を示します。

(O) 分割キーのバイト長、文字数またはマスクビットを指定して再度実行してください。

**KBLS2038-E**

invalid byte length of split-key: xx...x (zz...z)

分割キーのバイト長の指定が正しくありません。xx...x (zz...z)

分割キーのバイト長、文字数またはマスクビットの指定が間違っています。例えば、次のような場合が考えられます。

- 指定できる範囲以上の数値を指定した（分割キーの属性によって指定できるバイト長、文字数、またはマスクビットの範囲は異なる）
- 数値以外を指定した
- 入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-16LE または UTF-16BE）」を指定し、かつ分割キーのバイト長に 2 の倍数ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-32LE または UTF-32BE）」を指定し、かつ分割キーのバイト長に 4 の倍数ではない数値を指定した

xx...x は分割キー番号を示します。

**(O)** 分割キーのバイト長、文字数、またはマスクビットを修正して再度実行してください。

入力ファイルがテキストファイルまたは CSV ファイルの場合は、入力データのエンコードの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。入力データのエンコードは、環境ファイル「Exsortconf.ini ファイル（64 ビット版は「Exsortconf64.ini ファイル）」の「EXS\_CODETYPE」または `-inpf`、`-catinp`、`-splitinp` パラメタの `codetype` オペランドで指定します。

**KBLS2039-E**

sum of split-key position and length is greater than record length: xx...x (zz...z)

分割キー位置+分割キー長がレコード長を超えています。xx...x (zz...z)

分割キー位置+分割キー長が入力ファイルのレコード長を超えているか、可変長ファイルの場合は、最小レコード長を超えています。可変長順編成ファイルで、`-cmdopt POSNOCHK` の指定がある場合は、最大レコード長を超えています。または、入力ファイルに不当に短いデータがあります。

xx...x は分割キー番号を示します（可変長ファイル、またはテキストファイルの場合、0 が表示されます）。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

**(O)** 分割キーの開始バイト位置、バイト長を正しく指定して再度実行してください。

## KBLS2040-E

---

sum of split-key position and length include record length area: xx...x (zz...z)

分割キー位置+分割キー長がレコード長領域を含んでいます。xx...x (zz...z)

分割キー位置+分割キー長が入力ファイルのレコード長領域を含んでいます。次の条件が成り立つ場合が該当します。

- 「-locatajst」パラメタを指定している
- 可変長ファイルで、分割キー位置+分割キー長が 1～4 バイト目を含んでいる

xx...x は分割キー番号を示します。

**(O)** レコード長領域を除いた分割キーの開始バイト位置、バイト長を指定して再度実行してください。

## KBLS2041-E

---

too many total split-key length: xx...x yy...y (zz...z)

分割キーの合計長が制限を超えています。xx...x yy...y (zz...z)

分割キーの合計長が制限サイズを超えています。

「-splitkey」パラメタの「key」オペランドの場合は、バイト長、文字数、またはマスクビットの合計となります。

「-splitkey」パラメタの「csvkey」オペランドの場合は、有効桁の合計となります。

xx...x は分割キー番号を、yy...y は制限サイズを示します。

**(O)** 分割キーの個数、バイト長、文字数、またはマスクビットを正しく指定して再度実行してください。

## KBLS2042-E

---

invalid split-key data: ww...w xx...x yy...y (zz...z)

分割キーにエラーデータを検出しました。ww...w xx...x yy...y (zz...z)

入力データ中の 10 進数、日付または日付時刻の分割キーとして指定した範囲の値に、不正な形式のデータが見つかりました（CSV ファイルの場合は、入力セルデータに引用符、通貨記号、符号、桁修飾文字（コンマ、小数点、空白など）が含まれていてもエラーとしません。引用符、通貨記号、符号、桁修飾文字を除いた内部データ形式に正規化し、分割キーと比較します）。

ww...w はファイル名を、xx...x はレコード番号を、yy...y は分割キー番号を示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

**(O)** 分割キーの属性の指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。あらかじめ 10 進数または日付のデータ形式と一致しないデータ（空白、Low\_value (X'0000')、High\_value (X'FFFF') など）が入力データに含まれることがわかっている場合は、10 進キーのチェックオプション（-cmdopt KEYNOCHK）ま

たは日付キーのチェックオプション (-cmdopt DATENOCHK) の適用を検討してください。

CSV ファイルの場合は、次の確認を行ってください。

- 環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」) に設定したレコードの改行文字「EXS\_RECSEPA」、セルのデリミタ「EXS\_CELLSEPA」、または -inpfile、-catinp、-splitinp パラメタの recsepa または cellsepa オペランドの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。
- 入力セルデータに数字および引用符、通貨記号、符号、桁修飾文字以外の文字が含まれていないか確認してください。数字および引用符、通貨記号、符号、桁修飾文字以外の文字が含まれることがわかっている場合は、10 進キーのチェックオプション (-cmdopt KEYNOCHK)、または日付キーのチェックオプション (-cmdopt DATENOCHK) の適用を検討してください。
- 日付時刻データに、数字、引用符、項目区切り文字、および桁修飾文字以外の文字が含まれる、または日付時刻データが指定可能な範囲外であることがわかっている場合は、日付時刻キーのチェックオプション (-cmdopt DATETIMENOCHK) の適用を検討してください。日付時刻データの形式については「付録 J 日付時刻データの形式 (SORT EE の機能)」を参照してください。

### **KBLS2043-E**

---

invalid split-key specification: xx...x (zz...z)

分割キーの指定が正しくありません。xx...x (zz...z)

分割キーの指定が間違っています。

例えば、次のような場合が考えられます。

- CSV ファイルで「-splitkey」パラメタの「key」オペランドを指定している
- CSV ファイル以外で「-splitkey」パラメタの「csvkey」オペランドを指定している

xx...x はオペランド名を示します。

**(O)** 正しいオペランドを指定して再度実行してください。

### **KBLS2044-E**

---

cell-number must be specified: xx...x (zz...z)

分割キーのセル番号が指定されていません。xx...x (zz...z)

分割キーのセル番号が指定されていません。

xx...x は分割キー番号を示します。

**(O)** 分割キーのセル番号を指定して再度実行してください。

### **KBLS2045-E**

---

invalid cell-number specification: xx...x (zz...z)

**分割キーのセル番号の指定が正しくありません。xx…x (zz…z)**

分割キーのセル番号の指定が間違っています。例えば、次のような場合が考えられます。

- 1～5桁の10進数で指定していない
- 1～16384の範囲で指定していない

xx…xは分割キー番号を示します。

**(O)** 分割キーのセル番号を修正して再度実行してください。

---

**KBLS2046-E**

split-key-precison must be specified: xx…x (zz…z)

**分割キーの有効桁が指定されていません。xx…x (zz…z)**

分割キーの有効桁が指定されていません。

xx…xは分割キー番号を示します。

**(O)** 分割キーの有効桁を指定して再度実行してください。

---

**KBLS2047-E**

invalid split-key-precison specification: xx…x (zz…z)

**分割キーの有効桁の指定が正しくありません。xx…x (zz…z)**

分割キーの有効桁の指定が間違っています。

例えば、次のような場合が考えられます。

- 指定できる範囲より上の数値を指定している（分割キーの属性によって指定できる有効桁の範囲は異なる）
- 数値以外を指定している

xx…xは分割キー番号を示します。

**(O)** 分割キーの有効桁を修正して再度実行してください。

---

**KBLS2048-E**

dateform must be specified: xx…x (zz…z)

**分割キーの書式番号が指定されていません。xx…x (zz…z)**

「-splitkey」パラメタの「csvkey」オペランドに指定した分割キー（分割キー属性「g」）の書式番号が指定されていません。

xx…xは分割キー番号を示します。

**(O)** 分割キーの書式番号を指定して再度実行してください。

---

**KBLS2049-E**

invalid dateform specification: xx…x (zz…z)

**分割キーの書式番号の指定が正しくありません。xx…x (zz…z)**



「-splitkey」パラメタの「csvkey」オペランドに指定した分割キー（分割キー属性「g」）の書式番号の指定が間違っています。例えば、次のような場合が考えられます。

- 指定できる範囲以上の数値を指定している
- 数値以外を指定している

xx...x は分割キー番号を示します。

(O) 分割キーの書式番号を修正して再度実行してください。

### **KBLS2050-E**

---

invalid number of no-split-data: xx...x (zz...z)

非分割データの個数が制限を超えています。xx...x (zz...z)

非分割データの個数が 1 以上、制限数以下の範囲にありません。

xx...x は、非分割データの制限個数を示します。

(O) 非分割データの個数を正しく指定して再度実行してください。

### **KBLS2051-E**

---

no-split-data attribute must be specified: xx...x (zz...z)

非分割データの属性が指定されていません。xx...x (zz...z)

非分割データの属性が指定されていません。

xx...x は非分割データ番号を示します。

(O) 非分割データの属性を指定して再度実行してください。

### **KBLS2052-E**

---

invalid no-split-data attribute specification: xx...x (zz...z)

非分割データの属性の指定が正しくありません。xx...x (zz...z)

非分割データの属性の指定が間違っています。例えば、次のような場合が考えられます。

- 「j, c, n, m, a, b, p, z, x, s, t, f, g, q, d, w, u, v」以外の文字を指定した（CSV ファイルの場合は、「j, c, z, g, h, q, d, w, u, v」だけ指定できる）
- 大文字の英字で指定した
- 入力ファイル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に「w, u, v」を指定した

xx...x は非分割データ番号を示します。

(O) 非分割データの属性を修正して再度実行してください。

### **KBLS2053-E**

---

starting byte position of no-split-data must be specified: xx...x (zz...z)

非分割データの開始バイト位置が指定されていません。xx...x (zz...z)

非分割データの開始バイト位置が指定されていません。

xx…x は非分割データ番号を示します。

(O) 非分割データの開始バイト位置を指定して再度実行してください。

## **KBLS2054-E**

---

invalid starting byte position of no-split-data: xx…x (zz…z)

非分割データの開始バイト位置の指定が正しくありません。xx…x (zz…z)

非分割データの開始バイト位置の指定が間違っています。

例えば、次のような場合が考えられます。

- 6桁以上の数値を指定した
- 数値以外を指定した
- 入力データのエンコードに「UCS-2LE」, 「UCS-2BE」, 「UTF-16LE」, 「UTF-16BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-16LE または UTF-16BE)」を指定し、かつ非分割データの開始バイト位置に2の倍数ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」, 「UCS-4BE」, 「UTF-32LE」, 「UTF-32BE」または「UNICODE (入力ファイルの Unicode シグニチャ (BOM) の形式が UTF-32LE または UTF-32BE)」を指定し、かつ非分割データの開始バイト位置に4の倍数ではない数値を指定した

xx…x は非分割データ番号を示します。

(O) 非分割データの開始バイト位置を修正して再度実行してください。

入力ファイルがテキストファイルまたは CSV ファイルの場合は、入力データのエンコードの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。入力データのエンコードは、環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」の「EXS\_CODETYPE」または `-inpf`, `-catinp`, `-splitinp` パラメタの `codetype` オペランドで指定します。

## **KBLS2055-E**

---

no-split-data length must be specified: xx…x (zz…z)

非分割データのバイト長が指定されていません。xx…x (zz…z)

非分割データのバイト長、文字数、またはマスクビットが指定されていません。

xx…x は非分割キー番号を示します。

(O) 非分割データのバイト長、文字数、またはマスクビットを指定して再度実行してください。

## **KBLS2056-E**

---

invalid byte length of no-split-data: xx…x (zz…z)

非分割データのバイト長の指定が正しくありません。xx…x (zz…z)

非分割データのバイト長、文字数、またはマスクビットの指定が間違っています。例えば、次のような場合が考えられます。

- 指定できる範囲以上の数を指定した（非分割データの属性によって指定できるバイト長、文字数、またはマスクビットの範囲は異なる）
- 数値以外を指定した
- 入力データのエンコードに「UCS-2LE」、「UCS-2BE」、「UTF-16LE」、「UTF-16BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-16LE または UTF-16BE）」を指定し、かつ非分割データのバイト長に 2 の倍数ではない数値を指定した
- 入力データのエンコードに「UCS-4LE」、「UCS-4BE」、「UTF-32LE」、「UTF-32BE」または「UNICODE（入力ファイルの Unicode シグニチャ（BOM）の形式が UTF-32LE または UTF-32BE）」を指定し、かつ非分割データのバイト長に 4 の倍数ではない数値を指定した

xx...x は非分割データ番号を示します。

**(O)** 非分割データのバイト長、文字数、またはマスクビットを修正して再度実行してください。

入力ファイルがテキストファイルまたは CSV ファイルの場合は、入力データのエンコードの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。入力データのエンコードは、環境ファイル「Exsortconf.ini ファイル（64 ビット版は「Exsortconf64.ini ファイル」）」の「EXS\_CODETYPE」または `-inpfiler`、`-catinpf`、`-splitinpf` パラメタの `codetype` オペランドで指定します。

## **KBLS2057-E**

sum of no-split-data position and length is greater than record length: xx...x (zz...z)

非分割データ位置+非分割データ長がレコード長を超えています。xx...x (zz...z)

非分割データ位置+非分割データ長が入力ファイルのレコード長を超えているか、可変長ファイルの場合は、最小レコード長を超えています。可変長順編成ファイルで、`-cmdopt POSNOCHK` の指定がある場合は、最大レコード長を超えています。または、入力ファイルに不当に短いデータがあります。

xx...x は非分割データ番号を示します（可変長ファイル、またはテキストファイルの場合、0 が表示されます）。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

**(O)** 非分割データの開始バイト位置、バイト長を正しく指定して再度実行してください。

## **KBLS2058-E**

sum of no-split-data position and length include record length area: xx...x (zz...z)

非分割データ位置+非分割データ長がレコード長領域を含んでいます。xx...x (zz...z)

非分割データ位置+非分割データ長が入力ファイルのレコード長領域を含んでいます。  
次の条件が成り立つ場合が該当します。

- 「-locatajst」パラメタを指定している
- 可変長ファイルで、非分割データ位置+非分割データ長が 1～4 バイト目を含んでいる

xx…x は非分割データ番号を示します。

**(O)** レコード長領域を除いた非分割データの開始バイト位置、バイト長を指定して再度実行してください。

## KBLS2059-E

---

too many total no-split-data length: xx…x yy…y (zz…z)

非分割データの合計長が制限を超えています。xx…x yy…y (zz…z)

非分割データの合計長が制限サイズを超えています。

「-splitdata」または「-splitkey」パラメタの「nosplitdata」オペランドの場合は、バイト長、文字数、またはマスクビットの合計となります。

「-splitdata」または「-splitkey」パラメタの「nosplitcsvdata」オペランドの場合は、有効桁の合計となります。

xx…x は非分割データ番号を、yy…y は制限サイズを示します。

**(O)** 非分割データの個数、バイト長、文字数、またはマスクビットを正しく指定して再度実行してください。

## KBLS2060-E

---

invalid data in no-split-data: ww…w xx…x yy…y (zz…z)

非分割データにエラーデータを検出しました。ww…w xx…x yy…y (zz…z)

入力データ中の 10 進数、日付または日付時刻の非分割データとして指定した範囲の値に、不正な形式のデータが見つかりました (CSV ファイルの場合は、入力セルデータに引用符、通貨記号、符号、桁修飾文字 (コンマ、小数点、空白など) が含まれていてもエラーとしません。引用符、通貨記号、符号、桁修飾文字を除いた内部データ形式に正規化し、非分割データと比較します)。

ww…w はファイル名を、xx…x はレコード番号を、yy…y は非分割データ番号を示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。

**(O)** 非分割データの属性の指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。あらかじめ 10 進数または日付のデータ形式と一致しないデータ (空白、Low\_value (X'0000'), High\_value (X'FFFF') など) が入力データに含まれることがわかっている場合は、10 進キーのチェックオプション (-cmdopt KEYNOCHK) または日付キーのチェックオプション (-cmdopt DATENOCHK) の適用を検討してください。

さい。

CSV ファイルの場合は、次の確認を行ってください。

- 環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」)に設定したレコードの改行文字「EXS\_RECSEPA」、セルのデリミタ「EXS\_CELLSEPA」、または `-inpf`、`-catinp`、`-splitinp` パラメタの `recsepa` または `cellsepa` オペランドの指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。
- 入力セルデータに数字および引用符、通貨記号、符号、桁修飾文字以外の文字が含まれていないか確認してください。数字および引用符、通貨記号、符号、桁修飾文字以外の文字が含まれることがわかっている場合は、10 進キーのチェックオプション (`-cmdopt KEYNOCHK`)、または日付キーのチェックオプション (`-cmdopt DATENOCHK`) の適用を検討してください。
- 日付時刻データに、数字、引用符、項目区切り文字、および桁修飾文字以外の文字が含まれる、または日付時刻データが指定可能な範囲外であることがわかっている場合は、日付時刻キーのチェックオプション (`-cmdopt DATETIMENOCHK`) の適用を検討してください。日付時刻データの形式については「付録 J 日付時刻データの形式 (SORT EE の機能)」を参照してください。

## **KBLS2061-E**

---

invalid no-split-data specification: xx...x (zz...z)

非分割データの指定が正しくありません。xx...x (zz...z)

非分割データの指定が間違っています。

例えば、次のような場合が考えられます。

- CSV ファイルで「`-splitdata`」または「`-splitkey`」パラメタの「`nosplitdata`」オペランドを指定している
- CSV ファイル以外で「`-splitdata`」または「`-splitkey`」パラメタの「`nosplitcsvdata`」オペランドを指定している

xx...x はオペランド名を示します。

**(O)** 正しいオペランドを指定して再度実行してください。

## **KBLS2062-E**

---

cell-number must be specified: xx...x (zz...z)

非分割データのセル番号が指定されていません。xx...x (zz...z)

非分割データのセル番号が指定されていません。

xx...x は非分割データ番号を示します。

**(O)** 非分割データのセル番号を指定して再度実行してください。

## **KBLS2063-E**

---

invalid cell-number specification: xx...x (zz...z)

非分割データのセル番号の指定が正しくありません。xx...x (zz...z)

非分割データのセル番号の指定が間違っています。

例えば、次のような場合が考えられます。

- 1～5桁の10進数で指定していない
- 1～16384の範囲で指定していない

xx...x は非分割データ番号を示します。

**(O)** 非分割データのセル番号を修正して再度実行してください。

## **KBLS2064-E**

---

no-split-data-precision must be specified: xx...x (zz...z)

非分割データの有効桁が指定されていません。xx...x (zz...z)

非分割データの有効桁が指定されていません。

xx...x は非分割データ番号を示します。

**(O)** 非分割データの有効桁を指定して再度実行してください。

## **KBLS2065-E**

---

invalid no-split-data-precision specification: xx...x (zz...z)

非分割データの有効桁の指定が正しくありません。xx...x (zz...z)

非分割データの有効桁の指定が間違っています。

例えば、次のような場合が考えられます。

- 指定できる範囲より上の数値を指定している（非分割データの属性によって指定できる有効桁の範囲は異なる）
- 数値以外を指定している

xx...x は非分割データ番号を示します。

**(O)** 非分割データの有効桁を修正して再度実行してください。

## **KBLS2066-E**

---

dateform must be specified: xx...x (zz...z)

非分割データの書式番号が指定されていません。xx...x (zz...z)

「-splitdata」または「-splitkey」パラメタの「nosplitcsvdata」オペランドに指定した非分割データ（非分割データ属性「g」）の書式番号が指定されていません。

xx...x は非分割データ番号を示します。

**(O)** 非分割データの書式番号を指定して再度実行してください。

**KBLS2067-E**

---

invalid dateform specification: xx...x (zz...z)

非分割データの書式番号の指定が正しくありません。xx...x (zz...z)

「-splitdata」または「-splitkey」パラメタの「nosplitcsvdata」オペランドに指定した非分割データ（非分割データ属性「g」）の書式番号の指定が間違っています。

例えば、次のような場合が考えられます。

- 指定できる範囲以上の数値を指定している
- 数値以外を指定している

xx...x は非分割データ番号を示します。

**(O)** 非分割データの書式番号を修正して再度実行してください。

**KBLS2100-I**

---

catinp file not found: yy...y (zz...z)

連結入力ファイルが見つかりません。yy...y (zz...z)

「-catinp」パラメタで指定した入力ファイルが見つかりません。

yy...y はファイル名を示します。

**(S)** 存在する入力ファイルで、処理を続行する。

**KBLS2101-E**

---

invalid catinp specification: yy...y (zz...z)

連結入力ファイルの指定が正しくありません。yy...y (zz...z)

「-catinp」パラメタで指定した入力ファイル格納ディレクトリ、または入力ファイル名の総称の指定が正しくありません。

yy...y はディレクトリ名、またはファイル名の総称を示します。

例えば、次のような場合が考えられます。

- ディレクトリの指定が間違っている
- 連結ファイル名に (PATH\_MAX-1) または NAME\_MAX を超える文字数を指定した
- 指定した入力ファイル格納ディレクトリ名の末尾にスラッシュ (/) が指定されていない
- 環境変数名を「%%」だけで指定している (% ~ % で囲んだ変数名を指定する)
- 環境変数名「% ~ %」に入力ファイル名の末尾、または入力ファイル格納ディレクトリ名が関連づけされていない
- `exsort` コマンドに「% ~ %」で囲んだ環境変数名を指定している (環境変数名「% ~ %」はパラメタファイルだけ指定でき、UNIX では `exsort` コマンドで指定できない、Windows では `exsort` コマンドに指定した環境変数はコマンドのシェル・スクリプトで置換される)

**(O)** ディレクトリ名、または入力ファイル名の総称を正しく指定して再度実行してくだ

さい。

### **KBLS2102-E**

---

catinp file must be specified (zz...z)

連結入力ファイルが指定されていません。(zz...z)

「-catinp」パラメタで入力ファイルが指定されていません。

(O) 「file」オペランド、「prefix」オペランド、または「suffix」オペランドで入力ファイルを指定して再度実行してください。

### **KBLS2105-E**

---

input sequence number of catinp file must be specified (zz...z)

連結入力ファイルの入力順対応コードの指定がありません。(zz...z)

「-catinp」パラメタの「nameseq」オペランドで入力順対応コードが指定されていません。

(O) 入力順対応コードを指定して再度実行してください。

### **KBLS2106-E**

---

invalid input sequence number of catinp file (zz...z)

連結入力ファイルの入力順対応コードの指定が正しくありません。(zz...z)

「-catinp」パラメタの「nameseq」オペランドの指定が間違っています。例えば、次のような場合が考えられます。

- 入力順対応コードの値が、0～3の範囲で指定していない
- 入力順対応コードの値が、1桁でない
- 使用 OS で、指定できない値を指定している

(O) 入力順対応コードを正しく指定して再度実行してください。

### **KBLS2200-E**

---

invalid line feed specification : xx...x yy...y (zz...z)

改行文字の指定が正しくありません。xx...x yy...y (zz...z)

-infile, -outfile, -catinp, -catout, -splitinp, または -splitout パラメタの recsepa オペランドに指定した改行文字の指定が間違っています。例えば、次のような場合が考えられます。

- -infile, -catinp, または -splitinp パラメタの場合、「CRLF」、「LF」、または「LFCRLF」以外を指定した
- -outfile, -catout, または -splitout パラメタの場合、「CRLF」または「LF」以外を指定した
- 入力ファイルのファイル編成に、固定長順編成ファイル (fileorg=S)、または可変長順



編成ファイル (fileorg=V) を指定した

xx…x はパラメタ名を, yy…y はパラメタ番号を示します。

(O) 改行文字の指定を修正して再度実行してください。

## **KBLS2201-E**

---

invalid cell separate character specification : xx…x yy…y (zz…z)

セル区切り文字の指定が正しくありません。xx…x yy…y (zz…z)

-inpfiler, -outfile, -catinp, -catout, -splitinp, または -splitout パラメタの cellsepa オペランドに指定したセル区切り文字の指定が間違っています。例えば, 次のような場合が考えられます。

- セル区切り文字に, 「TAB」以外の 2 文字以上の文字列を指定した
- 6 個以上のセル区切り文字を指定した
- セル区切り文字を指定していない
- 入力ファイルのファイル編成に, 固定長順編成ファイル (fileorg=S), 可変長順編成ファイル (fileorg=V), またはテキストファイル (fileorg=A) を指定した

xx…x はパラメタ名を, yy…y はパラメタ番号を示します。

(O) セル区切り文字の指定を修正して再度実行してください。

## **KBLS2202-E**

---

invalid cell type specification : xx…x (zz…z)

セル数調整の指定が正しくありません。xx…x (zz…z)

-outfile, -catout, または -splitout パラメタの cellnum オペランドの指定が間違っています。例えば, 次のような場合が考えられます。

- セル数および「ADJUST」の指定がない
- 数字または「ADJUST」以外の文字列を指定している
- セル数の指定で, 1 ~ 5 桁の 10 進数で指定していない
- セル数の指定で, 1 ~ 16,384 の範囲で指定していない
- 入力ファイルのファイル編成に, 固定長順編成ファイル (fileorg=S), 可変長順編成ファイル (fileorg=V), またはテキストファイル (fileorg=A) を指定した

xx…x はパラメタ番号を示します。

(O) 指定を修正して再度実行してください。

## **KBLS2204-E**

---

invalid quote character of cell specification : xx…x yy…y (zz…z)

セルの引用符文字の指定が正しくありません。xx…x yy…y (zz…z)

-inpfiler, -outfile, -catinp, -catout, -splitinp または -splitout パラメタの quottype オ

ペラントに指定したセルの引用符文字の指定が間違っています。例えば、次のような場合が考えられます。

「-inpfiler, -catinp または -splitinp パラメタの場合」

- 「SQQUOT」または「DQUOT」の指定がない
- 「SQQUOT」, 「DQUOT」以外の文字列を指定している
- クォーテーションタイプを指定していない
- 入力ファイルのファイル編成に、固定長順編成ファイル (fileorg=S), 可変長順編成ファイル (fileorg=V), またはテキストファイル (fileorg=A) を指定した

「-outfile, -catout または -splitout パラメタの場合」

- 1 桁の 10 進数で指定していない
- 0 ~ 5 の範囲で指定していない
- オプションコード番号を指定していない

xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

**(O)** セルの引用符文字の指定を修正して再度実行してください。

## **KBLS2205-E**

---

invalid currency specification : (zz...z)

通貨記号の指定が正しくありません。(zz...z)

-inpfiler, -catinp, または -splitinp パラメタの currency オペラントに指定した通貨記号の指定が間違っています。例えば、次のような場合が考えられます。

- 通貨記号として 2 文字以上を指定した
- 通貨記号を指定していない
- 入力ファイルのファイル編成に、固定長順編成ファイル (fileorg=S), 可変長順編成ファイル (fileorg=V), またはテキストファイル (fileorg=A) を指定した

**(O)** 通貨記号の指定を修正して再度実行してください。

## **KBLS2206-E**

---

invalid round-form specification : (zz...z)

丸め方式の指定が正しくありません。(zz...z)

-inpfiler, -catinp, または -splitinp パラメタの round オペラントに指定した丸め方式の指定が間違っています。例えば、次のような場合が考えられます。

- 「0」, 「1」, 「2」以外の文字を指定した
- 丸め方式を指定していない
- 入力ファイルのファイル編成に、固定長順編成ファイル (fileorg=S), 可変長順編成ファイル (fileorg=V), またはテキストファイル (fileorg=A) を指定した

**(O)** 丸め方式の指定を修正して再度実行してください。

**KBLS2207-E**

invalid positional option code of null character specification : (zz...z)

ナル文字列の配置オプションコードの指定が正しくありません。(zz...z)

-inpfiler, -catinpfiler, または -splitinpfiler パラメタの nullseq オペランドに指定したナル文字列の配置オプションコードの指定が間違っています。例えば、次のような場合が考えられます。

- 「0」, 「1」以外の文字を指定した
- 2桁以上の数値を指定した
- オプションコードを指定していない

(O) ナル文字列の配置オプションコードの指定を修正して再度実行してください。

**KBLS2208-E**

invalid padding character specification: xx...x yy...y (zz...z)

パディング文字の指定が正しくありません。xx...x yy...y (zz...z)

-outfile, -catoutfile, または -splitoutfile パラメタの padding オペランドに指定したパディング文字の指定が間違っています。例えば、次のような場合が考えられます。

- 「X」, 「Z」, 「C'パディング文字」, または「X'16進数」以外を指定した。
- 「X'16進数」の16進数に、奇数桁または3桁以上の数値を指定した。
- 「C'パディング文字」のパディング文字に、2バイト以上の文字を指定した。
- パディング文字に、改行文字またはマルチバイト文字（日本語）を指定した。
- コマンド指定で、タブ以外の文字をダブルクォーテーション「"」で囲んでいない。
- パディング文字を指定していない。

xx...x はパラメタ名を、yy...y はパラメタ番号を示します。

(O) パディング文字の指定を修正して再度実行してください。

**KBLS2209-E**

invalid codetype specification: (zz...z)

文字コードの指定が正しくありません。(zz...z)

-inpfiler, -catinpfiler, または -splitinpfiler パラメタの codetype オペランドに指定した文字コードの指定が間違っています。例えば、次のような場合が考えられます。

- 次の文字コード以外の文字コードを指定した。

ASCII

SJIS

EUC-JP

EUC-HJ

UTF-8

UCS-2LE

UCS-2BE

UTF-16LE

UTF-16BE

UCS-4LE

UCS-4BE

UTF-32LE

UTF-32BE

- 文字コードを指定していない
- 入力ファイルのファイル編成に、固定長順編成ファイル (`fileorg=S`) または可変長順編成ファイル (`fileorg=V`) を指定した

(O) 文字コードの指定、または入力ファイルのファイル編成の指定を修正して再度実行してください。

## 付録 K.3 SORT メッセージ

### (1) メッセージの種類

SORT のメッセージには、次の 3 種類があります。

- 正常終了時のメッセージ  
ソート、マージ、選択および集約が正常に終了した場合は表示されます。
- 異常終了時のメッセージ  
ソート、マージ、選択、集約および関数実行中にエラーが発生した場合は表示されません。
- システムエラー発生時のメッセージ  
システムコールによるエラーが発生した場合は表示されます。

### (2) メッセージの記述形式

このマニュアルでは、次に示す形式でメッセージを記載します。

#### メッセージ ID

---

##### メッセージテキスト (英語)

英語文によるメッセージは、UNIX の場合だけ表示されます。

##### メッセージテキスト (日本語)

日本語文によるメッセージの説明を示す。

- (S) システムの処理を示す。
- (O) 使用者、または運用者の処置を示す。

##### メッセージ ID の説明

メッセージ ID には、メッセージを識別するコードを「KBL\$nnn-x」という形式で示します。

**KBLS** : プログラム識別コードを示す。

**nnn** : メッセージ番号を示す。

**x** : メッセージの種類を示す。

- **I** : 正常に終了した処理の結果を知らせるメッセージ
- **E** : エラーの発生を知らせるメッセージ

### (3) 終了コードの種類

**SORT** の終了コードは次のとおりです。

メッセージのほかにソート、選択および集約の終了状態を表す 3 種類の終了コードがあります。終了コードの種類を次の表に示します。

表 K-3 終了コードの種類

終了状態		終了コード
正常終了		0
異常終了	KBLS300-E 以外のメッセージ出力時	1
	KBLS300-E のメッセージ出力時、または <b>SORT</b> プログラムのメッセージがないとき	システムの異常終了コードが設定される

## 付録 K.4 SORT のメッセージ一覧

### **KBLS001-I**

number of data items sorted : xx...x

ソート済みデータ件数 xx...x

ソート済みデータは、xx...x 件です。

### **KBLS002-E**

input file organization must be specified

入力ファイルのファイル編成が指定されていません。

Windows の場合 :

入力ファイルのファイル編成 (/gs, /gv, /gr, /gw, /gi, /gx, /ga または file\_org=) の指定がありません。

UNIX の場合 :

入力ファイルのファイル編成 (-S, -V, -R, -W, -I, -X, -A) の指定がありません。

**(O)** 入力ファイルのファイル編成を指定して再度実行してください。

## **KBLS003-E**

---

invalid input-file organization specification

入力ファイルのファイル編成の指定が正しくありません。

Windows の場合 :

入力ファイルのファイル編成 (/gs, /gv, /gr, /gw, /gi, /gx, /ga または file\_org=) の指定が間違っています。

UNIX の場合 :

入力ファイルのファイル編成 (-S, -V, -R, -W, -I, -X, -A) の指定が間違っています。

(O) 入力ファイルのファイル編成を正しく指定して再度実行してください。

## **KBLS004-E**

---

record length must be specified

レコード長が指定されていません。

固定長ファイルのレコード長 (Windows の場合 : /l または record\_leng=, UNIX の場合 : -L) が指定されていません。

(O) レコード長を指定して再度実行してください。

## **KBLS005-E**

---

invalid record-length specification

レコード長の指定が正しくありません。

レコード長 (Windows の場合 : /l または record\_leng=, UNIX の場合 : -L) の指定が間違っています。例えば、次のような場合が考えられます。

- 入力ファイルが固定長順編成ファイルまたはテキストファイルのとき、レコード長を 1 ~ 65,535 の範囲で指定していない
- 指定したレコード長が、入力ファイルのレコード長と異なっている

(O) レコード長を正しく指定して再度実行してください。

## **KBLS006-E**

---

invalid output-file name specification

出力ファイル名の指定が正しくありません。

出力ファイル名 (Windows の場合 : /o または output\_file=, UNIX の場合 : -o) の指定が間違っています。例えば、ファイル名を 256 文字以上で指定した場合などが考えられます。

(O) 出力ファイル名を正しく指定して再度実行してください。

**KBLS007-E**

---

key option must be specified

キーオプションが指定されていません。

キー属性 (Windows の場合 : /c, /n または sort\_key= など, UNIX の場合 : -c, -n など) の指定がありません。

(O) キー属性を指定して再度実行してください。

**KBLS008-E**

---

invalid key option specification

キーオプションの指定が正しくありません。

Windows の場合 :

キー属性 (/c, /n または sort\_key= など) の指定が間違っています。例えば, /r 以外の属性を複数指定する際に, 連続して指定した場合 (例 : /cn) などが考えられます。

UNIX の場合 :

キーオプションの属性 (-c, -n など) の指定が間違っています。例えば, -r 以外の属性を複数指定する際に, 連続して指定した場合 (例 : -cn) などが考えられます。

(O) キー属性を正しく指定して再度実行してください。

**KBLS009-E**

---

starting byte position of key must be specified

キーの開始バイト位置が指定されていません。

キーの開始バイト位置の指定がありません。

(O) キーの開始バイト位置を指定して再度実行してください。

**KBLS010-E**

---

key byte length must be specified

キーのバイト長が指定されていません。

キーのバイト長の指定がありません。

(O) キーのバイト長を指定して再度実行してください。

**KBLS011-E**

---

invalid input-file name specification

入力ファイル名の指定が正しくありません。

入力ファイル名の指定が間違っています。例えば, ファイル名を 256 文字以上で指定した場合などが考えられます。

- (O) 入力ファイル名を正しく指定して再度実行してください。

### **KBLS012-E (UNIX 用)**

---

duplicate input file name

入力ファイル名が重複しています。

入力ファイル名が重複しています。

- (O) それぞれ異なるファイル名を指定して再度実行してください。

### **KBLS013-E**

---

invalid parameter specification

不正なパラメタが指定されました。

パラメタとして指定できない文字が指定されています。

- (O) 正しいパラメタを指定して再度実行してください。

### **KBLS014-E**

---

invalid buffer size specification

バッファサイズの指定が正しくありません。

ワークバッファサイズ (メモリ量) (Windows の場合 : /y または work\_size=, UNIX の場合 : -y) の指定が間違っています。例えば, サイズを数字以外で指定した場合などが考えられます。

- (O) ワークバッファサイズ (メモリ量) を正しく指定して再度実行してください。

### **KBLS015-E**

---

invalid directory specification

ディレクトリの指定が正しくありません。

一時ファイルを割り当てるディレクトリ (Windows の場合 : /w または temp\_dir1=, temp\_dir3=, UNIX の場合 : -T または -T2) の指定が間違っています。例えば, /w のあとにディレクトリ名を指定していない場合などが考えられます。

- (O) ディレクトリを正しく指定して再度実行してください。

### **KBLS016-E**

---

directory name is specified in input file name

入力ファイル名にディレクトリ名を指定しています。

入力ファイル名にディレクトリ名を指定しています。

- (O) 入力ファイル名を正しく指定して再度実行してください。



**KBLS017-E**

---

invalid sort option specification

ソートオプションの指定が正しくありません。

Windows の場合 :

ソートオプション (`/d`, `/i` または `sort_opt=`) の指定が間違っています。例えば, `/dc` などと指定した場合 (`/c` というソートオプションはない) などが考えられます。

UNIX の場合 :

ソートオプション (`-d`, `-e`) の指定が間違っています。例えば, `-ec` などと指定した場合 (`-c` というソートオプションはない) などが考えられます。

**(O)** ソートオプションを正しく指定して再度実行してください。**KBLS018-E**

---

output file organization must be specified

入力が索引ファイルのとき, 出力ファイルのファイル編成は省略できません。

入力ファイルが固定長索引ファイルまたは可変長索引ファイルの場合, 出力ファイルのファイル編成の指定は省略できません。

**(O)** 出力ファイルのファイル編成を正しく指定して再度実行してください。**KBLS019-E**

---

invalid output-file organization specification

出力ファイルのファイル編成の指定が正しくありません。

Windows の場合 :

出力ファイルのファイル編成 (`:s`, `:v`, `:r`, `:w`, `:a`) の指定が間違っています。例えば, 次の場合が考えられます。

- 出力ファイルのファイル編成に「`:s`, `:v`, `:r`, `:w`, `:a`」以外を指定した
- 入力ファイルと出力ファイルのレコードの形式 (固定長, 可変長またはテキスト) が異なっている

UNIX の場合 :

出力ファイルのファイル編成 (`:S`, `:V`, `:R`, `:W`, `:A`) の指定が間違っています。例えば, 次の場合が考えられます。

- 出力ファイルのファイル編成に「`:S`, `:V`, `:R`, `:W`, `:A`」以外を指定した
- 入力ファイルと出力ファイルのレコードの形式 (固定長, 可変長またはテキスト) が異なっている

**(O)** 出力ファイルのファイル編成を正しく指定して再度実行してください。**KBLS020-E**

---

input file name must be specified

入力が索引ファイルのとき、入力ファイル名は省略できません。

入力ファイルが固定長索引ファイルまたは可変長索引ファイルの場合、入力ファイル名の指定は省略できません。

(O) 入力ファイル名を指定して再度実行してください。

### **KBLS021-E (Windows 用)**

---

入力ファイル名の指定が 256 個を超えています。

入力ファイル名の指定が 256 個を超えています。

(O) ファイル名の個数を正しく指定して再度実行してください。

### **KBLS030-E (Windows 用)**

---

パラメタファイル名の指定が正しくありません。

パラメタファイル名の指定が間違っています。例えば、拡張子「.wsp」が間違っている場合などが考えられます。

(O) パラメタファイル名を正しく指定して再度実行してください。

### **KBLS031-E (Windows 用)**

---

パラメタファイルに I-O エラーが発生しました。xx...x ee...e

パラメタファイルに I/O エラーが発生しました。xx...x はエラーファイル名、ee...e はシステムエラーコードを示します。

(S) オープンしたファイルをすべてクローズし、処理を中断します。

(O) エラー原因を取り除いて再度実行してください。

### **KBLS032-E (Windows 用)**

---

パラメタファイルが他で使用中のためオープンできません。

パラメタファイルをほかの処理で使用中のため、オープンできません。例えば、COBOL プログラムが使用している場合などが考えられます。

(S) オープンしたファイルをすべてクローズし、処理を中断します。

(O) ほかの処理が終わるのを待って、再度実行してください。

### **KBLS033-E (Windows 用)**

---

パラメタファイルが見つかりません。

パラメタファイルが見つかりません。例えば、パラメタファイル名などの指定が間違っている場合が考えられます。

(O) パラメタファイルの指定を正しくして再度実行してください。

**KBLS034-E (Windows 用)**

---

パラメタファイルが指定されていません。

パラメタファイルの指定がありません。

- (O) パラメタファイルを指定して再度実行してください。

**KBLS035-E (Windows 用)**

---

実行オプションが指定されていません。

実行オプション (select, sort, sum, または exec\_opt=) の指定がありません。

- (O) 実行オプションを指定して再度実行してください。

**KBLS036-E (Windows 用)**

---

実行オプションの指定が正しくありません。

実行オプションの指定が間違っています。

- (O) 実行オプションを正しく指定して、再度実行してください。

**KBLS037-E (Windows 用)**

---

予約語 [WNSORTEX] が指定されていません。

拡張 SORT コマンドで指定されたパラメタファイルの先頭に, [WNSORTEX] の指定がありません。

- (O) パラメタファイルの先頭に [WNSORTEX] を指定して再度実行してください。

**KBLS051-I**

---

number of data items merged : xx...x

マージ済みデータ件数 xx...x

マージ済みデータ件数は, xx...x 件です。

**KBLS052-E**

---

invalid merge option specification

マージオプションの指定が正しくありません。

Windows の場合 :

マージオプション (/i, /k など) の指定が間違っています。例えば, /ic と指定した場合 (/c というマージオプションはない) などが考えられます。

UNIX の場合 :

マージオプション (-e, -k) の指定が間違っています。例えば, -ec などと指定した場合 (-c というマージオプションはない) などが考えられます。

- (O) マージオプションを正しく指定して再度実行してください。

### **KBLS053-E**

---

duplicate input file name and output file name

入力ファイル名と出力ファイル名が同じです。

入力ファイル名と出力ファイル名に同じファイル名を指定しました（選択機能、マージ機能、集約機能では入力ファイル名と出力ファイル名は異ならなければなりません）。

(O) 入力ファイル名または出力ファイル名を変更して再度実行してください。

### **KBLS061-I**

---

number of data items selected : xx...x

選択済みデータ件数 xx...x

選択済みデータは、xx...x 件です。

### **KBLS062-E**

---

invalid select option specification

選択オプションの指定が正しくありません。

Windows の場合 :

選択オプション (/u, /i) の指定が間違っています。例えば、/ud などと指定した場合 (/d という選択オプションはない) などが考えられます。

UNIX の場合 :

選択オプション (-e, -i) の指定が間違っています。例えば、-ed などと指定した場合 (-d という選択オプションはない) などが考えられます。

(O) 選択オプションを正しく指定して再度実行してください。

### **KBLS063-E**

---

selection-condition-expression must be specified

選択条件式が指定されていません。

選択条件式 (Windows の場合 : /sl または select\_exp=, UNIX の場合 : -sl) の指定がありません。

(O) 選択条件式を指定して再度実行してください。

### **KBLS064-E**

---

compare-field option must be specified

比較項目の属性が指定されていません。

比較項目の属性 (Windows の場合 : /c, /n など, UNIX の場合 : -c, -n など) の指定がありません。

- (O) 比較項目を指定して再度実行してください。

### **KBLS065-E**

---

invalid compare-field option specification

比較項目の属性の指定が正しくありません。

Windows の場合：

比較項目の属性 (*/c*, */n* など) の指定が間違っています。例えば、*/b* などと指定した場合 (*/b* という属性はない) などが考えられます。

UNIX の場合：

比較項目の属性 (*-c*, *-n* など) の指定が間違っています。例えば、*-b* などと指定した場合 (*-b* という属性はない) などが考えられます。

- (O) 比較項目を正しく指定して再度実行してください。

### **KBLS066-E**

---

starting byte position of compare-field must be specified

比較項目の開始バイト位置が指定されていません。

比較項目の開始バイト位置の指定がありません。

- (O) 比較項目の開始バイト位置を指定して再度実行してください。

### **KBLS067-E**

---

invalid starting byte position of compare-field

比較項目の開始バイト位置の指定が正しくありません。

比較項目の開始バイト位置が間違っています。例えば、数字以外の値を指定した場合などが考えられます。

- (O) 比較項目の開始バイト位置を正しく指定して再度実行してください。

### **KBLS068-E**

---

compare-field byte length must be specified

比較項目のバイト長が指定されていません。

比較項目のバイト長の指定がありません。

- (O) 比較項目のバイト長を指定して再度実行してください。

### **KBLS069-E**

---

invalid byte length for compare-field

比較項目のバイト長の指定が正しくありません。

比較項目のバイト長の指定が間違っています。例えば、数字以外の値を指定した場合などが考えられます。

- (O) 比較項目のバイト長を正しく指定して再度実行してください。

### **KBLS070-E**

---

invalid comparison expression specification

比較条件式の指定が正しくありません。

比較条件式の指定が間違っています。

- (O) 比較条件式を正しく指定して再度実行してください。

### **KBLS071-E**

---

qualification must be specified

条件値が指定されていません。

条件値の指定がありません。

- (O) 条件値を指定して再度実行してください。

### **KBLS072-E**

---

invalid qualification

条件値の指定が正しくありません。

条件値の指定が間違っています。例えば、Windows の場合 : 「c」, 「n」, 「x」 または 「d」, UNIX の場合 : 「C」 または 「N」 が指定されていない場合などが考えられます。

- (O) 条件値を正しく指定して再度実行してください。

### **KBLS073-E**

---

logical operator must be specified

論理演算子が指定されていません。

論理演算子の指定がありません。

- (O) 論理演算子を指定して再度実行してください。

### **KBLS074-E**

---

invalid logical operator specification

論理演算子の指定が正しくありません。

論理演算子の指定が間違っています。

- (O) 論理演算子を正しく指定して再度実行してください。

### **KBLS075-E**

---

relative operator must be specified

比較演算子が指定されていません。

比較演算子の指定がありません。

- (O) 比較演算子を指定して再度実行してください。

### **KBLS076-E**

---

invalid relative operator specification

比較演算子の指定が正しくありません。

比較演算子の指定が間違っています。

- (O) 比較演算子を正しく指定して再度実行してください。

### **KBLS077-E (Windows 用)**

---

比較条件式が 128 個を超えて指定されています。

比較条件式が 128 個を超えています。

- (O) 比較条件式の個数を 128 以下にして再度実行してください。

### **KBLS077-E (UNIX 用)**

---

number of comparison expression exceeds 32

比較条件式が 32 個を超えて指定されています。

比較条件式が 32 個を超えています。

- (O) 比較条件式を 32 個以内で指定して再度実行してください。

### **KBLS091-I**

---

number of data items summation: xx...x

集約済みデータ件数 xx...x

集約済みデータは、xx...x 件です。

### **KBLS092-E**

---

invalid sum option specification

集約オプションの指定が正しくありません。

Windows の場合：

集約オプション (/b, /i) の指定が間違っています。例えば、/bf などと指定した場合 (/f という集約オプションはありません) などが考えられます。

UNIX の場合：

集約オプション (-e, -i) の指定が間違っています。例えば、-eb などと指定した場合 (-b という集約オプションはない) などが考えられます。

- (O) 集約オプションを正しく指定して再度実行してください。

### **KBLS093-E**

---

sum-field must be specified

集約項目が指定されていません。

集約項目の指定がありません。

(O) 集約項目を指定して再度実行してください。

### **KBLS094-E**

---

sum-field option must be specified

集約項目の属性が指定されていません。

集約項目の属性の指定がありません。

(O) 集約項目の属性を指定して再度実行してください。

### **KBLS095-E**

---

invalid sum-field option specification

集約項目の属性の指定が正しくありません。

集約項目の属性 (n, m など) の指定が間違っています。例えば、b などと指定した場合 (b という属性はありません) などが考えられます。

(O) 集約オプションを正しく指定して再度実行してください。

### **KBLS096-E**

---

starting byte position of sum-field must be specified

集約項目の開始バイト位置が指定されていません。

集約項目の開始バイト位置の指定がありません。

(O) 集約項目の開始バイト位置を指定して再度実行してください。

### **KBLS097-E**

---

invalid starting byte position of sum-field

集約項目の開始バイト位置の指定が正しくありません。

集約項目の開始バイト位置が間違っています。例えば、数字以外の値を指定した場合などが考えられます。

(O) 比較項目の開始バイト位置を正しく指定して再度実行してください。

### **KBLS098-E**

---

sum-field byte length must be specified

集約項目のバイト長が指定されていません。

集約項目のバイト長の指定がありません。



- (O) 集約項目のバイト長を指定して再度実行してください。

### **KBLS099-E**

---

invalid byte length for sum-field

集約項目のバイト長の指定が正しくありません。

集約項目のバイト長の指定が間違っています。例えば、バイト長を 1～19 の範囲で指定していない場合などが考えられます。

- (O) バイト長を正しく指定して再度実行してください。

### **KBLS190-E (Windows 用)**

---

ISAM が見つかりません。

ISAM ライブラリ (ISAMNT.DLL, または ISAMNT64.DLL) がインストールされていない、またはパス名が見つかりません。

- (O) ISAM ライブラリをインストールするか、パス名を正しく指定して再度実行してください。

### **KBLS201-E**

---

invalid key data for sort or merge

xx...x yy...y zz

ソート又はマージのキーにエラーデータを検出しました。 xx...x yy...y zz

数字項目のキーデータ形式が間違っています。

xx...x はエラーファイル名を、yy...y はエラーレコード番号を、zz はエラーキー番号を示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) キー属性の指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。または、キーを正しい形式に修正して再度実行してください。

### **KBLS202-E**

---

cannot obtain memory of this size

メモリが確保できません。

指定したメモリサイズが確保できません。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) メモリサイズを修正して再度実行してください。

### **KBLS203-E**

---

invalid number of keys

キーの個数が正しくありません。

キーの個数が 1 以上 64 以下の範囲にありません。

(S) オープンしたファイルをすべてクローズし、処理を中断します。

(O) キーの個数を正しく指定して再度実行してください。

## **KBLS204-E**

---

invalid key item attribute specification

キーの項目属性の指定が正しくありません。

キーの項目属性の指定が間違っています。

(S) オープンしたファイルをすべてクローズし、処理を中断します。

(O) キーの項目属性を正しく指定して再度実行してください。

## **KBLS205-E**

---

invalid starting byte position of key

キーの開始バイト位置が正しくありません。

Windows の場合：

キーの開始バイト位置が間違っています。例えば、入力ファイルの最小レコード長を超えて指定した場合などが考えられます。

(S) オープンしたファイルをすべてクローズし、処理を中断します。

(O) キーの開始バイト位置を正しく指定して再度実行してください。

UNIX の場合：

キーの開始バイトに 7 桁以上を指定しています。

(S) オープンしたファイルをすべてクローズし、処理を中断します。デバッグ情報を生成します。

(O) キーの開始バイト位置を正しく指定して再度実行してください。

## **KBLS206-E**

---

invalid byte length for key

キーのバイト長が正しくありません。

Windows の場合：

キーのバイト長が間違っています。例えば、レコード長より大きい値を指定した場合などが考えられます。

(S) オープンしたファイルをすべてクローズし、処理を中断します。

(O) キーのバイト長を正しく指定して再度実行してください。

UNIX の場合：

キーのバイト長に 7 桁以上を指定しています。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。デバッグ情報を生成します。
- (O) キーのバイト長を正しく指定して再度実行してください。

### **KBLS207-E**

---

sum of key position and length is greater than record length

キー位置+キー長がレコード長を超えています。

キー位置+キー長が入力ファイルの最小レコード長を超えています。  
または、入力ファイルに不当に短いデータがあります。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) キーの開始バイト位置、バイト長を正しく指定して再度実行してください。または、入力ファイルのデータを正しく修正して再度実行してください。

### **KBLS208-E**

---

total key length exceeds 4096 bytes

キーの合計長が 4096 バイトを超えています。

キーの合計長が 4,096 バイトを超えています。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) キーの個数、バイト長を正しく指定して再度実行してください。

### **KBLS209-E (Windows 用)**

---

入力ファイルに I-O エラーが発生しました。 xx...x yy...y zz...z

入力ファイルに I/O エラーが発生しました。

xx...x はエラーファイル名を、yy...y はエラーレコード番号を、zz...z はシステムエラーコードを示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) エラーの原因を取り除いて再度実行してください。

### **KBLS209-E (UNIX 用)**

---

I/O error on input file xx...x yy...y

入力ファイルに I-O エラーが発生しました。

xx...x yy...y

入力ファイルに I/O エラーが発生しました。

xx...x はエラーファイル名を、yy...y はエラーレコード番号を示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。
- (O) エラー要因を除去して再度実行してください。

### **KBLS210-E (Windows 用)**

---

一時ファイルに I/O エラーが発生しました。 ee…e

一時ファイルに I/O エラーが発生しました。  
ee…e はシステムエラーコードを示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) エラーの原因を取り除いて再度実行してください。

### **KBLS210-E (UNIX 用)**

---

I/O error on temporary file

一時ファイルに I/O エラーが発生しました。

一時ファイルに I/O エラーが発生しました。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。
- (O) エラー要因を除去して再度実行してください。

### **KBLS211-E (Windows 用)**

---

出力ファイルに I/O エラーが発生しました。 xx…x zz…z

出力ファイルに I/O エラーが発生しました。

xx…x はエラーファイル名を、zz…z はシステムエラーコードを示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) エラーの原因を取り除いて再度実行してください。

### **KBLS211-E (UNIX 用)**

---

I/O error on output file

xx…x

出力ファイルに I/O エラーが発生しました。 xx…x

出力ファイルに I/O エラーが発生しました。

xx…x はエラーファイル名を示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。
- (O) エラー要因を除去して再度実行してください。

**KBLS212-E**

---

cannot input because data is shorter than record length

データがレコード長より短いため入力できません。

入力ファイルが標準入力ファイルのとき、入力データが指定したレコード長より短いです。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 入力データの長さをレコード長の整数倍にして再度実行してください。

**KBLS213-E**

---

invalid organized file or corrupted file

xx...x

ファイル編成不正又はファイル破壊が発生しました。 xx...x

入力ファイルの編成と指定したファイル編成（Windows の場合：/gs, /gv, /gr, /gw, /gi, /gx, /ga または file\_org=, UNIX の場合：-S, -V, -R, -W, -I, -X, -A）とが一致していません。または、入力ファイルのレコード長領域などが破壊されています。

xx...x はエラーファイル名を示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 入力ファイルのファイル編成を正しく指定して再度実行してください。それでもエラーになる場合は、入力ファイルを作成し直して実行してください。

**KBLS214-E**

---

I/O error on debugging information file

デバッグ情報ファイルに I-O エラーが発生しました。

デバッグ情報ファイルに I/O エラーが発生しました。

- (O) 保守員に連絡してください。

**KBLS216-E**

---

file name is specified in directory of temporary file

一時ファイルのディレクトリにファイル名を指定しています。

ソート用一時ファイルを割り当てるディレクトリ（Windows の場合：/w, UNIX の場合：-T または -T2）の指定に、ファイル名を指定しました。

- (S) 処理を中断します。
- (O) ディレクトリ名を正しく指定して再度実行してください。

**KBLS217-E**

---

cannot open input file as it is already in use

入力ファイルが他で使用中のためオープンできません。

入力ファイルがほかの処理で使用されているため、オープンできません。例えば、ほかの処理で、SORT プログラムが同じファイルをオープンしている場合などが考えられます。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) ほかの処理が終わるのを待って、再度実行してください。

### **KBLS218-E**

---

cannot open output file as it is already in use

出力ファイルが他で使用中のためオープンできません。

出力ファイルがほかの処理で使用されているため、オープンできません。例えば、ほかの処理で、SORT プログラムが同じファイルをオープンしている場合などが考えられます。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) ほかの処理が終わるのを待って、再度実行してください。

### **KBLS219-E**

---

output file already exists

出力ファイルが既に存在するため処理を中止しました。

ソートオプションで既存ファイルの書き換え防止機能（Windows の場合 : /i, UNIX の場合 : -i）が指定されているとき、出力ファイル名に、すでにあるファイル名を指定しました。

- (S) 処理を中断します。
- (O) 出力ファイル名に、すでにあるファイル名と異なるファイル名を指定して、再度実行してください。

### **KBLS220-E (Windows 用)**

---

入力ファイルが見つかりません。 xx…x

存在しないファイル名を指定しています。xx…x は、入力したファイル名を示します。

- (O) 存在するファイル名を指定し、再度実行してください。

### **KBLS221-E (Windows 用)**

---

出力ファイルの領域が確保できません。

ディスクの空き容量が少ないため、出力ファイルが確保できません。

- (O) ディスクの領域を空けて再度実行してください。

**KBLS222-E (Windows 用)**

---

索引ファイルに I/O エラーが発生しました。 xx…x zz…z

索引ファイルに I/O エラーが発生しました。xx…x はエラーファイル名、zz…z はシステムエラーコードを示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) エラーの原因を取り除いて再度実行してください。

**KBLS225-E (Windows 用)**

---

ドライブの準備ができていません。

ドライブにフロッピーディスクがセットされていません。

- (O) フロッピーディスクをセットして再度実行してください。

**KBLS226-E (Windows 用)**

---

ドライブの準備ができていません。

ドライブにフロッピーディスクがセットされていません。

- (O) フロッピーディスクをセットして再度実行してください。

**KBLS227-E (Windows 用)**

---

ドライブの準備ができていません。

ドライブにフロッピーディスクがセットされていません。

- (O) フロッピーディスクをセットして再度実行してください。

**KBLS228-E (Windows 用)**

---

メディアは書き込み禁止です。

フロッピーディスクが書き込み禁止になっている場合などが考えられます。

- (O) エラーの原因を取り除いて再度実行してください。

**KBLS229-E (Windows 用)**

---

メディアは書き込み禁止です。

フロッピーディスクが書き込み禁止になっている場合などが考えられます。

- (O) エラーの原因を取り除いて再度実行してください。

**KBLS230-E (Windows 用)**

---

一時ファイルの領域が確保できません。

一時ファイルの領域が確保できません。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。

- (O) 一時ファイルを割り当てるディレクトリを変更して再度実行してください。

### **KBLS231-E (Windows 用)**

---

入力ファイル名と出力ファイル名が同じです。

入力ファイル名と出力ファイル名に、同じ名前を指定しています。

- (O) 異なるファイル名を指定して、再度実行してください。

### **KBLS232-E**

---

cannot input because data is greater then max-record length

データが最大レコード長より長い場合入力できません。

Windows の場合：

入力ファイルがテキストファイルのとき、入力データに最大レコード長より長いデータがあります。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 入力ファイルの最大レコード長を大きくして、再度実行してください。

UNIX の場合：

指定した最大レコード長より長いデータを入力しようとしてしました。

- (S) 処理を中断します。
- (O) 最大レコード長を正しく指定して再度実行してください。

### **KBLS290-E**

---

insufficient memory to execute

メモリが不足しました。

Windows の場合：

SORT オプションで指定した使用メモリ量が小さいため、SORT が実行できません。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) wnsort コマンドの /y オペランド、wnsortex コマンドで指定するパラメタファイルの work\_size オペランド、または環境ファイルの work\_size パラメタで指定したワークバッファサイズを大きくして再度実行してください。

UNIX の場合：

SORT オプションで指定した使用メモリ量では不足しているため、処理を継続できません。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。
- (O) rsort コマンドの -y オペランド、または環境変数 SORTOPT の -y オプションで指定したワークバッファサイズを大きくして再度実行してください。



**KBLS300-E**

---

invalid programming logic, error code = xxxx

プログラム論理エラーが発生しました。(xxxx)

詳細コード xxxx のプログラム不良が発生しました。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 詳細コードを保守員に連絡してください。

**KBLS401-E**

---

improper order of merge keys, must be in ascending or descending order

xx...x yy...y

マージのキーが昇順または降順になっていません。

xx...x yy...y

次の場合が考えられます。

- マージオプションの強制マージ (Windows の場合 : /k, UNIX の場合 : -k) を指定しないで昇順でマージしたとき、入力ファイルが昇順でソートされていない
- マージオプションの強制マージ (Windows の場合 : /k, UNIX の場合 : -k) を指定しないで降順でマージ (Windows の場合 : /r, UNIX の場合 : -r を指定) したとき、入力ファイルが降順でソートされていない

xx...x はエラーファイル名を、yy...y はエラーレコード番号を示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) キー属性 (Windows の場合 : /r, UNIX の場合 : -r) の指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。  
昇順になっていないファイルを昇順でマージしたい場合、または降順になっていないファイルを降順でマージしたい場合は、マージオプションで強制マージ (Windows の場合 : /k, UNIX の場合 : -k) を指定するか、ソート機能を使ってください。

**KBLS402-E**

---

mismatch in position between merge key and indexed key

マージのキー位置が索引ファイルのキー位置と一致していません。

マージ機能で入力ファイルが索引ファイルのとき、マージの第 1 キーが索引ファイルの主キーまたは副キーになっていません。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。デバッグ情報ファイルを生成します。
- (O) キー位置およびキー長が正しいか確認し、間違っている場合は正しく指定して再度実行してください。

索引ファイルの主キーまたは副キーでない項目をキーとして索引ファイルをマージ

したい場合は、ソート機能を使ってください。

### **KBLS501-E**

---

invalid compare-field data for select

xx...x yy...y zz

比較項目にエラーデータを検出しました。 xx...x yy...y zz

比較項目のデータ形式が間違っています。

xx...x はエラーファイル名を、yy...y はエラーコード番号を、zz はエラーのある比較項目番号を示します。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。

**(O)** 比較項目の属性の指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。または、データを正しい形式に修正して再度実行してください。

### **KBLS503-E**

---

比較項目の属性の指定が正しくありません。

比較項目の属性の指定が間違っています。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。

**(O)** 比較項目の属性を正しく指定して再度実行してください。

### **KBLS504-E**

---

比較項目のバイト長が正しくありません。

比較項目のバイト長が間違っています。例えば、指定できる値の範囲を超えて指定した場合などが考えられます。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。

**(O)** 比較項目のバイト長を正しく指定して再度実行してください。

### **KBLS505-E**

---

sum of compare-field position and length is greater than record length

比較項目の項目位置+項目長がレコード長を超えています。

比較項目の項目位置+項目長が入力ファイルの最小レコード長を超えています。

**(S)** オープンしたファイルをすべてクローズし、処理を中断します。

**(O)** 比較項目の項目位置、項目長を正しく指定して再度実行してください。

### **KBLS506-E**

---

total compare-field length exceeds 4096 bytes

比較項目の合計長が 4096 バイトを超えています。

比較項目の合計長が 4,096 バイトを超えています。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 比較項目の個数、項目長を正しく指定して再度実行してください。

### **KBLS601-E**

---

invalid sum-field data for sum

xx...x yy...y zz

集約項目にエラーデータを検出しました。 xx...x yy...y zz...z

集約項目のデータ形式が間違っています。

xx...x はエラーファイル名、yy...y はエラーコード番号、zz...z はエラーのある集約項目番号を示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 集約項目の属性の指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。または、集約項目を正しい形式に修正して再度実行してください。

### **KBLS602-E**

---

invalid key data for sum

xx...x yy...y zz

集約キーにエラーデータを検出しました。 xx...x yy...y zz

数字項目のキーデータ形式が間違っています。xx...x はエラーファイル名を、yy...y はエラーコード番号を、zz はエラーキー番号を示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) キー属性の指定が正しいか確認し、間違っているときは正しく指定して再度実行してください。または、キーを正しい形式に修正して再度実行してください。

### **KBLS603-E (Windows 用)**

---

集約項目の属性の指定が正しくありません。

集約項目の属性の指定が間違っています。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 集約項目の属性を正しく指定して再度実行してください。

### **KBLS604-E (Windows 用)**

---

集約項目のバイト長が正しくありません。

集約項目のバイト長が間違っています。例えば、指定できる値の範囲を超えて指定した場合などが考えられます。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。

- (O) 集約項目のバイト長を正しく指定して再度実行してください。

### **KBLS605-E (Windows 用)**

---

集約項目の項目位置+項目長がレコード長を超えています。

集約項目の項目位置+項目長が入力ファイルの最小レコード長を超えています。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 集約項目の項目位置、項目長を正しく指定して再度実行してください。

### **KBLS605-E (UNIX 用)**

---

sum of sum-field position and length is greater than record length

集約項目の項目位置がレコード長を超えています。

集約項目の項目位置が入力ファイルの最小レコード長を超えています。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。また、デバッグ情報ファイルを生成します。
- (O) 集約項目の項目位置、項目長を正しく指定して再度実行してください。

### **KBLS606-E**

---

duplicate starting byte position of sum-field and key

集約項目と集約キーが重なっています。

集約項目と集約キーの位置が重なっています。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 集約項目の項目位置、項目長を正しく指定して再度実行してください。

### **KBLS607-E**

---

too many sum-fields

集約項目の個数が制限を超えています。

集約項目が 64 個を超えています。

- (O) 集約項目の個数を減らして再度実行してください。

### **KBLS608-E**

---

duplicate starting byte position of sum-field and another sum-field are identical

集約項目が他の集約項目と重なっています。

項目位置、項目長で指定した集約項目の位置が、ほかの集約項目の指定位置と重なっています。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 集約項目の項目位置、項目長を正しく指定して再度実行してください。

**KBLS701-E (Windows 用)**

---

中間ファイルに I/O エラーが発生しました。 ee...e

中間ファイルに I/O エラーが発生しました。 ee...e はシステムエラーコードを示します。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) エラーの原因を取り除いて再度実行してください。

**KBLS702-E (Windows 用)**

---

ドライブの準備ができていません。

ドライブにフロッピーディスクがセットされていません。

- (O) フロッピーディスクをセットして再度実行してください。

**KBLS703-E (Windows 用)**

---

メディアは書き込み禁止です。

フロッピーディスクが書き込み禁止になっている場合などが考えられます。

- (O) エラーの原因を取り除いて再度実行してください。

**KBLS704-E (Windows 用)**

---

中間ファイルの領域が確保できません。

中間ファイルの領域が確保できません。

- (S) オープンしたファイルをすべてクローズし、処理を中断します。
- (O) 中間ファイルのサイズを修正して再度実行してください。

**KBLS705-E (Windows 用)**

---

入力ファイルが複数指定されています。

拡張 SORT コマンドの input\_file オペランドに、入力ファイルが複数指定されました。実行オプションで sum を単独で指定した場合、指定できる入力ファイル数は 1 個です。

- (O) 入力ファイル名を正しく指定して再度実行してください。

**KBLS901-E (Windows 用)**

---

関数実行時の指定が正しくありません。 xxx

次の関数の実行時の指定が間違っています。 xxx は、詳細エラーコードを示します。

- WNS\_Sort
- WNS\_Select
- WNS\_Sum
- WNS\_SortEX

- (S) オープンしたファイルをすべてクローズし、処理を中断します。

- (O) 各ハンドルの設定内容，設定漏れを確認して再度実行してください。

### **KBLS950-I**

---

insufficient buffer to execute

ソート用ワークバッファサイズが不足しています。

- (O) 適正なワークバッファサイズを指定してください。

### **KBLS951-E**

---

too many size of string file

ストリング情報ファイルの制限を越えました。

- (S) オープンしたファイルをすべてクローズし，処理を中断します。
- (O) 適正なワークバッファサイズを指定してください。

### **KBLS952-E**

---

I/O error on string file xxx

ストリング情報ファイルに I-O エラーが発生しました。xxx

ストリング情報ファイルに I/O エラーが発生しました。xxx は，システムエラーコードを示します。

- (S) オープンしたファイルをすべてクローズし，処理を中断します。
- (O) 適正なワークバッファサイズを指定してください。

## 付録 L 各バージョンの変更内容

各バージョンの変更内容を示します。

変更内容 (3020-3-N73-60) SORT Version8 - Extended Edition(64) 08-40, SORT Version8 - Extended Edition 08-40

### 追加・変更内容

SORT EE の簡易コピー機能について説明を追加した。

各機能を使用する場合に指定するオプション名または参照先を記載した。

エンコードに ASCII, UTF-16LE, UTF-16BE, UTF-32LE, UTF-32BE を追加した。

SORT EE の機能として、次を追加した。

- シフト JIS コード比較オプション
- ゼロデータ比較オプション
- 空行削除オプション
- ファイルの上書きオプション
- Unicode シグニチャ (BOM) 関連オプション
- マルチバイト文字データの比較
- 内部 10 進数符号オプション
- 外部 10 進数データ形式オプション
- 符号なし 2 進数のマスクビット指定
- 簡易コピー機能

SORT EE の空行削除オプションについての説明を追加した。

外部 10 進数データ形式オプションについての説明を追加した。

SORT EE で指定できるデータの種類の次に追加した。

- シングルバイト文字
- マルチバイト文字
- 符号なし 2 進数

次のパラメタの説明を追加した。

- `-cpyinp`
- `-cpyout`

• SORT EE のバージョンごとの説明を変更，または追加した。

- 前版で「4.5 パラメタおよびオペランドの詳細」に記載していた内容を「`exsort` コマンドのパラメタおよびオペランドの詳細」として 5 章に移動した。

テキストファイルおよび CSV ファイルでは，マルチバイト文字に対応した文字属性コード「`u`」および「`v`」を指定できるようにした。

## 追加・変更内容

次のパラメタの説明を変更した。

- -p パラメタ
- -function パラメタ
- -cmdopt パラメタ
- -key パラメタ
- -sum パラメタ
- -infile パラメタ
- -outfile パラメタ
- -sorttmpdir パラメタ
- -mergetmpdir パラメタ
- -csvkey パラメタ
- -splitdata パラメタ
- -splitinp パラメタ
- -splitout パラメタ
- -catinp パラメタ
- -catout パラメタ

- -cmdopt パラメタに次のオプションを追加した。
- シフト JIS コード比較オプション (SJIS オプション)
- ゼロデータ比較オプション (DIFFZERO オプション)
- 空行削除オプション (DELBLANKLINE オプション)

次のソート機能の使用例を追加した。

- ソート+符号なし 2 進数のマスクビット指定キー
- ソート+ CSV ファイル+ UTF-16LE
- 簡易コピー

altseq 変換ファイルの作成例を追加した。

SORT EE の環境変数に次を追加した。

- EXSORTPACKSIGNTYPE (EXSORT64PACKSIGNTYPE)
- EXSORTUNPACKDATATYPE (EXSORT64UNPACKDATATYPE)
- EXSORTDELBLANKLINE (EXSORT64DELBLANKLINE)

環境ファイルに指定できる項目に次を追加した。

- EXS\_CODETYPE
- EXS\_PACK\_SIGNTYPE
- EXS\_UNPACK\_DATATYPE

インコアソートの場合のワークバッファサイズ見積もり式の変数値を変更, または追加した。

- SORT から SORT EE に移行する場合のコマンド, およびワークバッファサイズの移行内容を記載した。

次のメッセージを変更した。

KBLS1001-I, KBLS1003-I, KBLS1004-I, KBLS1005-I, KBLS1006-I, KBLS1007-I, KBLS1008-I, KBLS1009-I, KBLS1100-E, KBLS1111-E, KBLS1203-E, KBLS1205-E, KBLS1206-E, KBLS1207-E, KBLS1210-E, KBLS1211-E, KBLS1303-E, KBLS1306-E, KBLS1307-E, KBLS1308-E, KBLS1314-E, KBLS1320-E, KBLS1404-E, KBLS1500-E, KBLS1503-E, KBLS1505-E, KBLS1521-E, KBLS1620-E, KBLS1621-E, KBLS1631-E, KBLS1632-E, KBLS1636-E, KBLS1639-E, KBLS2034-E, KBLS2036-E, KBLS2037-E, KBLS2038-E, KBLS2041-E, KBLS2042-E, KBLS2052-E, KBLS2054-E, KBLS2055-E, KBLS2056-E, KBLS2059-E, KBLS2060-E, KBLS2209-E



## 追加・変更内容

- 次のメッセージを追加した。
- KBLS1014-I

用語解説に次を追加した。

- IVS
- UTF-16BE
- UTF-16LE
- UTF-32BE
- UTF-32LE
- VS
- 簡易コピー機能
- 基底文字
- レコードヘッダ

変更内容 (3020-3-N73-50) SORT Version8 - Extended Edition(64) 08-30, SORT Version8 - Extended Edition 08-30

## 追加・変更内容

EBCDIK コードキー比較機能に関する説明を変更した。

Windows 版 SORT EE で、キャッシュファイルの同期出力機能をサポートした。

キー・項目の上限値拡張機能 (SORT EE の機能) に関する説明を変更した。

「UCS-2」および「UCS-4」に関する説明を追加した。また、この機能に関連する説明を変更した。

キー属性と形式に関する説明を変更した。

日付時刻データに関する説明を追加した。また、この機能に関連する説明を変更した。

指定できる値の範囲 (パラメタ) に関する説明を変更した。

exsort パラメタファイルに記述できる 1 行の長さを、10KB 以内から 50KB 以内に変更した。

次のパラメタの説明を変更した。

- -cmdopt パラメタ
- -key パラメタ
- -sum パラメタ
- -infile パラメタ
- -outfile パラメタ
- -worksize パラメタ
- -csvkey パラメタ
- -splitdata パラメタ
- -splitinp パラメタ
- -splitout パラメタ
- -catinp パラメタ
- -catout パラメタ

追加・変更内容

次の環境変数の説明を変更した。

- EXSORTMGTOPT (DATETIME\_INVALID\_NOCHK オプション追加)
- EXSORTBOMLNGIN
- EXSORTBOMLNGOUT
- EXSORT64MGTOPT (DATETIME\_INVALID\_NOCHK オプション追加)
- EXSORT64BOMLNGIN
- EXSORT64BOMLNGOUT
- SORTOPT

次の環境変数の説明を追加した。

- EXSORTBOMGETPUT
- EXSORTGOPTION
- EXSORT64BOMGETPUT
- EXSORT64GOPTION

環境ファイルに指定できる項目に EXS\_REPOLOG\_FILESIZE の説明を追加した。

環境ファイルに指定する次の項目の説明を変更した。

- EXS\_RECSEPA
- EXS\_CELLSEPA
- EXS\_CURRENCY
- EXS\_OUTRECSEPA
- EXS\_PADDING
- EXS\_OUTCELLSEPA
- EXS\_KEYNUMMAX
- EXS\_SELNUMMAX
- EXS\_CSVKEYNUMMAX
- EXS\_CSVSELNUMMAX

一時ファイルの作成条件とファイル容量の算出方法の説明を変更した。

ワークバッファサイズと処理時間に関する説明を変更した。

ワークバッファサイズの自動見積もり機能に関する説明を変更した。

**SORT** の機能拡張オプションの注意事項と見直しポイントに関する説明を変更した。

ログファイルに関する説明を修正した。

CSV ファイルのデータの扱い (**SORT EE** の機能) に関する次の説明を変更した。

- 文字列データの端数処理
- 日付データの端数処理

次のメッセージを変更した。

KBLS1007-I, KBLS1203-E, KBLS1205-E, KBLS1207-E, KBLS1210-E, KBLS1211-E,  
KBLS1303-E, KBLS1306-E, KBLS1308-E, KBLS1314-E, KBLS1320-E, KBLS1404-E,  
KBLS1500-E, KBLS1503-E, KBLS1505-E, KBLS1517-E, KBLS1636-E, KBLS2034-E,  
KBLS2036-E, KBLS2038-E, KBLS2042-E, KBLS2052-E, KBLS2054-E, KBLS2056-E,  
KBLS2060-E, KBLS2208-E, KBLS2209-E

## 追加・変更内容

用語解説に次を追加した。

- EUC-JP
- Microsoft コードページ
- Shift\_JIS
- UCS-2LE
- UCS-2BE
- UCS-4LE
- UCS-4BE
- Unicode
- Unicode シグニチャ (BOM)
- UTF-8
- UTF-16
- UTF-32
- エンディアン
- 結合文字
- コードページ 932 (CP932)
- コードページ 1200 (CP1200)
- コードページ 1201 (CP1201)
- コードページ 12000 (CP12000)
- コードページ 12001 (CP12001)
- コードページ 51932 (CP51932)
- コードページ 65001 (CP65001)
- サロゲートペア
- 正規化
- 世界協定時
- マルチバイト文字
- ワイド文字

用語解説の「環境ファイル」の説明を変更した。

次のとおり表現を変更した。

- マルチバイトコード → マルチバイト文字
- 改行コード → 改行文字
- シフト JIS → Shift\_JIS

また、改行文字の 16 進数表記を削除した (SORT EE の場合)。

変更内容 (3020-3-N73-40) SORT Version8 - Extended Edition(64) 08-20, SORT Version8 - Extended Edition 08-20

## 追加・変更内容

ファイル編成変更機能に関する説明を追加した。また、この機能に関連する説明を変更した。

AIX, HP-UX, Linux(x86) 版 SORT EE で次の機能をサポートした。

- ファイル分割機能
- ファイル連結機能
- 強制ファイル分割機能
- ファイル逐次 OPEN/CLOSE 機能
- メッセージ抑止機能
- 0 件ファイル出力抑止機能

CSV ファイルの選択, 集約, レコード編集機能に関する説明を追加した。また、この機能に関連する説明を変更した。

追加・変更内容

キー位置チェック緩和機能の変更に伴い、関連する説明を変更した。

キー・項目の上限値拡張機能に関する説明を追加した。また、この機能に関連する説明を変更した。

メッセージ出力オプションに関する説明を追加した。

集約結果オーバーフロー検知オプションに関する説明を追加した。

改行文字の自動判定機能に関する説明を追加，変更した。

属性コード **j** および **c** のマニュアル上の表現を次のとおり変更した。

- **j**: 英数字→文字 (JIS8 単位コード)
- **c**: 文字 (英数字, 漢字) →文字

左分離符号付き数字 (文字) 形式サポートに伴い、関連説明を追加した。

機能の実行順序の説明を変更した。

次のパラメタの説明を追加した。

- `-csvsum` パラメタ
- `-csvreorg` パラメタ

次のパラメタの説明を変更した。

- `-p` パラメタ
- `-cmdopt` パラメタ
- `-key` パラメタ
- `-sum` パラメタ
- `-infile` パラメタ
- `-outfile` パラメタ
- `-worksize` パラメタ
- `-csvkey` パラメタ
- `-iosize` パラメタ
- `-splitfile` パラメタ
- `-splitdata` パラメタ
- `-splitkey` パラメタ
- `-splitinp` パラメタ
- `-splitout` パラメタ
- `-splittmpdir` パラメタ
- `-catinp` パラメタ
- `-catout` パラメタ

次の環境変数の説明を追加した。

- `EXSORTOVFERROPT`
- `EXSORT64OVFERROPT`

## 追加・変更内容

次の環境変数の説明を変更した。

- EXSORTWORKSIZE
- EXSORTMSGOPT
- EXSORT\_HI\_WORKSIZE
- EXSORTSUPPRESSSPLITMSG
- EXSORTSUPPRESSCATMSG
- EXSORTNOCREATEZEROFIELD
- EXSORTCATINPNOERR
- EXSORTFILESPLITNUMB
- EXSORTSPLITNUMBONE
- EXSORTSPLITTMPDIR
- EXSORTSPLITTMPIOSIZE
- EXSORT64WORKSIZE
- EXSORT64MSGOPT
- EXSORT64MSGFILE
- EXSORT64\_HI\_WORKSIZE
- EXSORT64SUPPRESSSPLITMSG
- EXSORT64SUPPRESSCATMSG
- EXSORT64NOCREATEZEROFIELD
- EXSORT64CATINPNOERR
- EXSORT64FILESPLITNUMB
- EXSORT64SPLITNUMBONE
- EXSORT64SPLITTMPDIR
- EXSORT64SPLITTMPIOSIZE

次の環境ファイルに指定する項目の説明を追加した。

- EXS\_LOGPATH
- EXS\_ALTSEQPATH
- EXS\_COMPLETE\_MSGOPT
- EXS\_NULLSEQ
- EXS\_OUTRECSEPA
- EXS\_PADDING
- EXS\_OUTCELLSEPA
- EXS\_EXSORTDMP\_OPT
- EXS\_KEYNUMMAX
- EXS\_SUMNUMMAX
- EXS\_SELNUMMAX
- EXS\_CSVKEYNUMMAX
- EXS\_CSVSUMNUMMAX
- EXS\_CSVSELNUMMAX

次の環境ファイルに指定する項目の説明を変更した。

- EXS\_RECSEPA
- EXS\_CELLSEPA
- EXS\_QUOTTYPE
- EXS\_CURRENCY
- EXS\_ROUND

SORT EE のワークバッファサイズ自動見積もり機能の説明を変更した。

追加・変更内容

次のメッセージを追加した。

KBLS1012-I, KBLS1013-I, KBLS1108-E, KBLS1322-E, KBLS1323-E, KBLS1324-E,  
KBLS1325-E, KBLS1326-E, KBLS1327-E, KBLS1328-E, KBLS1412-E, KBLS1413-E,  
KBLS1414-E, KBLS1415-E, KBLS1416-E, KBLS1417-E, KBLS1518-E, KBLS1519-E,  
KBLS1520-E, KBLS1521-E, KBLS1522-E, KBLS1523-E, KBLS1524-E, KBLS1753-E,  
KBLS2200-E, KBLS2201-E, KBLS2202-E, KBLS2204-E, KBLS2205-E, KBLS2206-E,  
KBLS2207-E, KBLS2208-E, KBLS2209-E

次のメッセージを変更した。

KBLS1007-I, KBLS1010-I, KBLS1011-I, KBLS1100-E, KBLS1200-E, KBLS1201-E,  
KBLS1203-E, KBLS1210-E, KBLS1211-E, KBLS1213-E, KBLS1300-E, KBLS1301-E,  
KBLS1303-E, KBLS1314-E, KBLS1319-E, KBLS1320-E, KBLS1400-E, KBLS1402-E,  
KBLS1411-E, KBLS1511-E, KBLS1512-E, KBLS1513-E, KBLS1751-E, KBLS1752-E,  
KBLS2034-E, KBLS2042-E, KBLS2052-E, KBLS2060-E

変更内容 (3020-3-N73-30) SORT Version8 - Extended Edition(64) 08-11, SORT  
Version8 - Extended Edition 08-11

追加・変更内容

ファイル分割機能の説明を追加した。

ファイル連結機能の説明を追加した。

数値の **big endian** オプション機能に **Linux** が対応した。

強制ファイル分割機能の説明を追加した。

ファイル逐次 **OPEN/CLOSE** 機能の説明を追加した。

メッセージ抑止機能の説明を追加した。

0 件ファイル出力抑止機能の説明を追加した。

パラメタ指定時に必要な値として指定できる最小値および最大値の表に次の項目を追加した。

- 入力ファイルの数
  - inpfiler 指定
  - splitinp 指定
  - catinp 指定
- 出力ファイルの数
  - outfile 指定
  - splitout 指定
  - catout 指定

ソート機能でワークバッファサイズに指定するメモリ量に指定できる最大値の説明を追加した。

exsort64 コマンドの説明を追加した。

## 追加・変更内容

次のパラメタの説明を追加した。

- `-splitfile` パラメタ
- `-splitdata` パラメタ
- `-splitkey` パラメタ
- `-splitinp` パラメタ
- `-splitout` パラメタ
- `-splittmpdir` パラメタ
- `-catinp` パラメタ
- `-catout` パラメタ

次のパラメタの説明を変更した。

- `-cmdopt` パラメタ
- `-key` パラメタ
- `-infile` パラメタ
- `-outfile` パラメタ
- `-sorttmpdir` パラメタ
- `-mergetmpdir` パラメタ
- `-dateopt` パラメタ
- `-iosize` パラメタ

使用例で使用するサンプルファイルの格納ディレクトリの説明に 64 ビット版の格納ディレクトリを追加した。

次のソート機能の使用例を追加した。

- ソート+均等分割
- ソート+定数分割
- ソート+キー分割
- ソート+ファイル連結

オプションのパラメタに `-B` を追加した。

ソート用一時ファイルを割り当てるディレクトリを指定についての説明を変更した。

次の環境変数の説明を追加した。

- `EXSORTSUPPRESSSPLITMSG`
- `EXSORTSUPPRESSCATMSG`
- `EXSORTNOCREATEZEROFIELD`
- `EXSORTCATINPNOERR`
- `EXSORTFILESPLITNUMB`
- `EXSORTSPLITNUMBONE`
- `EXSORTSPLITTMPDIR`
- `EXSORTSPLITTMPIOSIZE`
- `EXSORTCATNAMESEQ`
- `EXSORT64SUPPRESSSPLITMSG`
- `EXSORT64SUPPRESSCATMSG`
- `EXSORT64NOCREATEZEROFIELD`
- `EXSORT64CATINPNOERR`
- `EXSORT64FILESPLITNUMB`
- `EXSORT64SPLITNUMBONE`
- `EXSORT64SPLITTMPDIR`
- `EXSORT64SPLITTMPIOSIZE`
- `EXSORT64CATNAMESEQ`

ワークサイズ設定の優先順位の説明を変更した。

追加・変更内容

SORTOPT のオプションに `-SORT_EE` を追加した。

Exsortconf.ini ファイルに設定できる `EXS_CELLSEPA` 項目の注意事項を変更した。

SORT の環境ファイル (Windows の場合) の形式の説明を変更した。

SORT EE および SORT が作成する一時ファイルの種類と目的の説明を追加した。

SORT EE の一時ファイルの作成条件に 32 ビット版と 64 ビット版の説明を追加した。

SORT EE の一時ファイル容量の算出方法に 64 ビット版の説明を追加した。

SORT EE のファイル分割用一時ファイルの作成条件を追加した。

SORT EE のファイル分割用一時ファイル容量の算出方法を追加した。

SORT の機能拡張オプションの環境設定の使用する SORT EE 機能を有効にする説明を変更した。

SORT EE のデバッグ情報ファイルを出力するメッセージ ID の表を追加した。

SORT のデバッグ情報ファイルを出力するメッセージ ID の表のデバッグ情報ファイル (sortidmp) の出力可否を変更した。

次のメッセージを追加した。

KBLS1010-I, KBLS1011-I, KBLS1106-E, KBLS1107-E, KBLS2000-E, KBLS2001-E,  
KBLS2010-E, KBLS2011-E, KBLS2012-E, KBLS2013-E, KBLS2014-E, KBLS2015-E,  
KBLS2020-I, KBLS2021-E, KBLS2022-E, KBLS2030-E, KBLS2031-E, KBLS2032-E,  
KBLS2033-E, KBLS2034-E, KBLS2035-E, KBLS2036-E, KBLS2037-E, KBLS2038-E,  
KBLS2039-E, KBLS2040-E, KBLS2041-E, KBLS2042-E, KBLS2043-E, KBLS2044-E,  
KBLS2045-E, KBLS2046-E, KBLS2047-E, KBLS2048-E, KBLS2049-E, KBLS2050-E,  
KBLS2051-E, KBLS2052-E, KBLS2053-E, KBLS2054-E, KBLS2055-E, KBLS2056-E,  
KBLS2057-E, KBLS2058-E, KBLS2059-E, KBLS2060-E, KBLS2061-E, KBLS2062-E,  
KBLS2063-E, KBLS2064-E, KBLS2065-E, KBLS2066-E, KBLS2067-E, KBLS2100-I,  
KBLS2101-E, KBLS2102-E, KBLS2105-E, KBLS2106-E, KBLS705-E

次のメッセージを変更した。

KBLS1007-I, KBLS1008-I, KBLS1009-I, KBLS1101-E, KBLS1152-E, KBLS1620-E,  
KBLS1621-E, KBLS1629-E, KBLS1630-E, KBLS1650-E, KBLS1665-E, KBLS1674-E,  
KBLS1720-E, KBLS037-E, KBLS053-E, KBLS402-E, KBLS503-E, KBLS504-E

変更内容 (3020-3-N73-20) SORT Version8 08-10, SORT Version8 - Extended Edition 08-10

追加・変更内容

ExSORT を SORT EE という名称に変更しました。

SORT EE と SORT の処理の流れの概要を追加しました。

SORT のファイル排他機能の記述を追加しました。

SORT EE のファイル排他機能の記述を追加しました。

SORT EE の I/O サイズ変更機能の記述を追加しました。

SORT EE のワークバッファサイズ自動見積もり機能の記述を追加しました。

SORT EE の 2GB を超えるワークバッファサイズの指定の記述を追加しました。



## 追加・変更内容

SORT の機能拡張オプションの記述を追加しました。

パラメタの記述を追加しました。

選択オプションの記述を追加しました。

SORT EE の環境変数の記述を追加しました。

デバッグ情報ファイルのレポートログ出力オプションの記述を追加しました。

内容を全面的に変更しました。

SORT の機能拡張オプションの使い方の記述を追加しました。

SORT EE のデバッグ情報ファイルの記述を変更しました。

次のメッセージを追加しました。

KBLS1060-I, KBLS1061-I, KBLS1133-I, KBLS1134-E, KBLS1135-E, KBLS1160-E,  
KBLS1161-E および KBLS1162-E

また、次のメッセージの説明を追加・修正しました。

KBLS1007-I, KBLS1130-E, KBLS1131-E, KBLS1132-E, KBLS1693-E, KBLS1720-E,  
KBLS1721-E

変更内容 (3020-3-N73-10) SORT Version8 08-00, SORT Version8 - Extended Edition  
08-00

## 追加・変更内容

レコード編集機能を編集機能という呼び方に変更しました。

文字コードの比較順序変更機能を文字順序変更機能という呼び方に変更しました。

UNIX 版 SORT の記述追加に伴い、ExSORT の機能、SORT の機能、Windows の場合、UNIX の場合  
など、タイトルに追加しました。

ExSORT と SORT の機能の概要を追加しました。

UNIX 版 SORT の記述を追加しました。

CSV ファイルの説明を変更し、ファイルおよびデータ形式の参照先を追加しました。

今版 (3020-3-N73-10) では、目次の構成を全面的に変更しました。前版  
(3020-3-N73) との対応は次のようになっています。

旧 (3020-3-N73)	新 (3020-3-N73-10)
第 1 編 機能編	
1. ExSORT	1. ExSORT と SORT
1.1 ExSORT の概要	1.1 ExSORT と SORT の概要
1.1.1 複数機能を 1 コマンドに集約	1.1 ExSORT と SORT の概要
1.1.2 選択機能の拡張	削除
1.1.3 複数ファイル出力機能	2.14 複数ファイル出力機能 (ExSORT の機能)
1.1.4 レコード編集機能	1.7 編集機能とは (ExSORT の機能)

付録 L 各バージョンの変更内容

旧 (3020-3-N73)	新 (3020-3-N73-10)
1.1.5 文字コードの比較順序変更機能	2.16 文字順序変更機能 (ExSORT の機能)
1.1.6 CSV ファイルの入出力	付録 C CSV ファイル形式と CSV データ形式 (ExSORT の機能)
1.1.7 前提条件	削除
1.2 入出力構成と入出力データ形式	削除
1.2.1 入出力構成	削除
1.2.2 入出力データ形式	削除
1.3 ExSORT の機能詳細	1.1 ExSORT と SORT の概要
1.3.1 基本機能	1.1 ExSORT と SORT の概要
1.3.2 付加機能	1.1 ExSORT と SORT の概要
1.3.3 機能の実行順序と実行形態	4.1 機能の実行順序と実行形態
1.3.4 CSV ファイル形式	付録 C.1 CSV ファイル形式
1.3.5 CSV データ形式	付録 C.2 CSV データ形式
1.4 ExSORT の使い方	4. ExSORT の使い方
1.4.1 exsort コマンドの形式	4.2.1(1) exsort コマンド形式 [ソート機能] 4.2.2(1) exsort コマンド形式 [マージ機能] 4.2.3(1) exsort コマンド形式 [コピー機能]
1.4.2 コマンドの表記規則	4.3 コマンドの表記規則
1.4.3 exsort パラメタファイルの表記規則	4.4 exsort パラメタファイルの表記規則
1.4.4 パラメタおよびオペランドの詳細	2.1 重複レコードの入力順序保存機能 2.2 EBCDIK コードキー比較機能 2.3 強制マージ機能 2.4 無条件レコード抽出機能 2.5 10 進数オーバーフロー判定機能 2.6 数値の big endian 操作機能 (Windows の場合) 2.9 内部 10 進数符号保障機能 (ExSORT の機能) 2.10 10 進キーのキーチェック緩和機能 (ExSORT の機能) 2.11 10 進集約項目のチェック緩和機能 (ExSORT の機能) 2.12 キー位置のチェック緩和機能 (ExSORT の機能) 2.13 日付データのチェック緩和機能 (ExSORT の機能) 2.15 入力レコードの読み飛ばし・打ち切り機能 (ExSORT の機能) 4.5 パラメタおよびオペランドの詳細
1.5 使用例	5. ExSORT の使用例
1.5.1 ソート機能の使用例	5.1 ソート機能の使用例

旧 (3020-3-N73)	新 (3020-3-N73-10)
1.5.2 マージ機能の使用例	5.2 マージ機能の使用例
1.5.3 コピー機能の使用例	5.3 コピー機能の使用例
1.5.4 altseq 変換ファイルの使用例	削除
2. SORT	1. ExSORT と SORT
2.1 SORT の概要	1.1 ExSORT と SORT の概要
2.2 入出力機能と処理の流れ	1.1 ExSORT と SORT の概要
2.3 SORT の使い方	1.2 ソート機能とは
2.3.1 ソート機能の概要	2.17 既存ファイルの書き換え防止機能 (SORT の機能)
2.3.2 ソート機能の使い方	6. ソートコマンドの使い方 6.1 ソートコマンドの使い方 (Windows 版の場合) 6.1.1 コマンドの使い方 6.1.2 オペランドの意味 6.1.3 コマンド指定時の注意事項 6.1.4 使用例
2.3.3 マージ機能	1.3 マージ機能とは 2.3 強制マージ機能
2.3.4 マージ機能の使い方	7. マージコマンドの使い方 7.1 マージコマンドの使い方 (Windows の場合) 7.1.1 コマンドの使い方 7.1.2 オペランドの意味 7.1.3 コマンド指定時の注意事項 7.1.4 使用例
2.3.5 選択機能	1.5 選択機能とは
2.3.6 選択機能の使い方	8. 選択コマンドの使い方 8.1 選択コマンドの使い方 (Windows の場合) 8.1.1 コマンドの使い方 8.1.2 オペランドの意味 8.1.3 コマンド指定時の注意事項 8.1.4 使用例
2.3.7 集約機能	1.6 集約機能とは

旧 (3020-3-N73)	新 (3020-3-N73-10)
2.3.8 集約機能の使い方	9. 集約コマンドの使い方 9.1 集約コマンドの使い方 (Windows の場合) 9.1.1 コマンドの使い方 9.1.2 オペランドの意味 9.1.3 コマンド指定時の注意事項 9.1.4 使用例 10. 拡張 SORT コマンドの使い方 (Windows の場合) 10.1 コマンドの使い方 10.2 オペランドの意味 10.3 コマンド指定時の注意事項 10.4 使用例
第 2 編 運用編	
3. ExSORT のメモリの見積もり	削除
3.1 ExSORT の実行メモリサイズ	
4. ExSORT の入出力障害時のファイルの扱い	17. 障害時の対処方法
4.1 ExSORT の入出力障害時のファイルの扱い	17.1.1 ExSORT の入出力障害時のファイルの扱い
4.1.1 デバッグ情報ファイル	17.1.2 デバッグ情報ファイル
5. ExSORT の障害情報の出力	17.1 ExSORT の障害情報
5.1 ExSORT の障害情報の出力	
5.1.1 デバッグ情報ファイル	17.1.2 デバッグ情報ファイル
5.1.2 ログファイル	17.1.3 ログファイル
5.1.3 完了メッセージオプション	17.1.4 完了メッセージオプション
6. ExSORT のキーまたは項目の不正による障害の対処方法	17.1.5 キーまたは項目の不正による障害の対処方法
6.1 キーまたは項目の不正による障害の対処方法	
6.1.1 メッセージを確認する	
6.1.2 sortrdmp の内容を確認する	
7. 環境変数の設定 (ExSORT の場合)	13. 環境変数の設定
7.1 環境変数	13.1 ExSORT の環境変数の設定
7.2 環境変数の設定	
8. altseq 変換ファイル	付録 E altseq 変換ファイル (ExSORT の機能)
8.1 altseq 変換ファイル	
9. 設定ファイル (Exsortconf.ini)	14.1 ExSORT の環境ファイルの設定
9.1 設定ファイル (Exsortconf.ini)	
9.2 CSV ファイル操作項目	14.1.1 CSV ファイル操作項目
10. SORT の入出障害時のファイルの扱い	17. 障害時の対処方法
10.1 入出力障害時のファイルの扱い	17.2 SORT の障害情報
11. SORT の障害の対処方法	17. 障害時の対処方法

旧 (3020-3-N73)	新 (3020-3-N73-10)
11.1 キーまたは項目の不正による障害の場合	17.2.3 キーまたは項目の不正による障害の場合
11.2 C 言語で作成したプログラムの不正による障害の場合	17.2.4 C 言語で作成したプログラムの不正による障害の場合
12. SORT の環境ファイル	14. 環境ファイルの設定
12.1 SORT の環境ファイルの形式	14.2 SORT の環境ファイルの設定 (Windows の場合)
12.2 SORT の環境ファイルのパラメタの意味	14.2.1 SORT の環境ファイルのパラメタの意味
12.3 SORT の環境ファイルの有効範囲	14.2.2 SORT の環境ファイルの有効範囲
12.4 SORT の環境ファイル指定時の注意事項	14.2.3 SORT の環境ファイル指定時の注意事項
13. 一時ファイルの作成条件とファイル容量の算出方法	15. ソート機能の一時ファイルの作成条件とファイル容量の算出方法
13.1 ExSORT の一時ファイルの作成条件 13.1.1 ソート用一時ファイルおよびマージ用一時ファイルを作成する場合のワークバッファサイズ 13.1.2 ソート用一時ファイルだけを作成する場合のワークバッファサイズ 13.1.3 一時ファイルを作成しない場合のワークバッファサイズ	15.1 ExSORT の一時ファイルの作成条件
13.2 ExSORT の一時ファイルのファイル容量の算出方法	15.2 ExSORT の一時ファイル容量の算出方法
13.3 SORT の一時ファイルの作成条件	15.3 SORT の一時ファイルの作成条件
13.4 SORT の一時ファイル容量の算出方法	15.4 SORT の一時ファイル容量の算出方法
14. ExSORT および SORT の効果的な使い方	16. ソート機能の効果的な使い方
14.1 ソート処理時間を短くするには	16.1 ソート機能の実行時間を短くするためには
14.2 テキストファイルをソートするには	16.2 テキストファイルをソートするには
15. ExSORT および SORT で指定できるデータの種類と形式	3.2 キー属性と形式
15.1 指定できるデータの種類と形式	
16. 指定できる値の範囲	3.3 指定できる値の範囲
16.1 指定できる値の範囲 (ExSORT の場合)	
16.2 指定できる値の範囲 (SORT の場合)	
17. CSV ファイルのデータ入力時の端数処理	付録 B CSV ファイルのデータとキーの扱い (ExSORT の機能)
17.1 数値データ	付録 B.1 数値データ
17.2 文字列データ	付録 B.2 文字列データ
17.3 日付データ	付録 B.3 日付データ

付録 L 各バージョンの変更内容

旧 (3020-3-N73)	新 (3020-3-N73-10)
18. CSV ファイル内のナル文字列の扱い	付録 D CSV ファイル内のナル文字列の扱い (ExSORT の機能)
18.1 キー	
19. C 言語からの呼び出し	12. C 言語プログラムでの使い方 (Windows の場合)
19.1 関数の機能と形式	12.1 関数の機能と形式
19.2 コーディング例	12.2 コーディング例
19.3 コンパイル方法	12.3 コンパイル方法
付録 A JIS8 単位コード体系と EBCDIK コード体系の変換表	付録 A JIS8 単位コード体系と EBCDIK コード体系の変換表
付録 B メッセージ	付録 F メッセージ
付録 B.1 メッセージの記述形式	付録 F.1 ExSORT メッセージ 付録 F.3 SORT メッセージ
付録 B.2 メッセージ一覧	付録 F.2 ExSORT のメッセージ一覧 付録 F.4 SORT のメッセージ一覧
付録 B.3 詳細エラーコード	12.4 詳細エラーコード
付録 C 用語解説	付録 G 用語解説

## 付録 M 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 M.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。

- 索引順編成ファイル管理 ISAM (3020-3-D88)  
Windows 版の索引順編成ファイル管理 ISAM について説明しています。
- 索引順編成ファイル管理 ISAM (3000-3-046)  
UNIX 版の索引順編成ファイル管理 ISAM について説明しています。

### 付録 M.2 このマニュアルでの表記

#### (1) 製品名

このマニュアルでは、製品名称を次に示す略称で表記します。

正式名称	このマニュアルでの表記
AIX 5L V5.2	AIX
AIX 5L V5.3	
AIX V6.1	
AIX V7.1	
COBOL2002 Net Client Runtime	COBOL2002
COBOL2002 Net Client Suite	
COBOL2002 Net Developer	
COBOL2002 Net Developer(64)	
COBOL2002 Net Server Runtime	
COBOL2002 Net Server Runtime(64)	
COBOL2002 Net Server Suite	
COBOL2002 Net Server Suite(64)	

正式名称	このマニュアルでの表記	
COBOL85 Version 6	COBOL85	
COBOL85 Version 7		
COBOL85 Run Time System Version 6		
COBOL85 Run Time System Version 7		
COBOL85 Server Version 6		
COBOL85 Server Version 7		
COBOL85 Server Run Time System Version 6		
COBOL85 Server Run Time System Version 7		
HP-UX 11i V2(IPF)	HP-UX	
HP-UX 11i V3(IPF)		
ISAM Version7	ISAM	
ISAM/D Version7		
Red Hat Enterprise Linux <sup>(R)</sup> 5 Advanced Platform(x86)	Linux(x86)	Red Hat Linux または Linux
Red Hat Enterprise Linux <sup>(R)</sup> 5(x86)		
Red Hat Enterprise Linux <sup>(R)</sup> Server 6 (32-bit x86)		
Red Hat Enterprise Linux <sup>(R)</sup> 5 Advanced Platform(AMD/Intel 64)	Linux(x64)	
Red Hat Enterprise Linux <sup>(R)</sup> 5(AMD/Intel 64)		
Red Hat Enterprise Linux <sup>(R)</sup> Server 6 (64-bit x86_64)		
Red Hat Enterprise Linux <sup>(R)</sup> Server 7 (64-bit x86_64)		
SORT Version8	SORT	
SORT Version8(64)		
SORT Version8 - Extended Edition	SORT EE(32)	SORT EE
SORT Version8 - Extended Edition(64)	SORT EE(64)	

AIX, HP-UX, Red Hat Linux または Linux を総称する場合は、UNIX と表記します。  
COBOL2002 および COBOL85 を総称する場合は、COBOL と表記します。

## (2) 製品の違いによる表記について

SORT Version8 - Extended Edition と SORT Version8 - Extended Edition(64) では、コマンド名、インストールパス名、サンプルファイル名、環境変数名、環境ファイル名、デバッグ情報ファイル名、およびログファイル名が異なります。次の対応表に基づき、SORT Version8 - Extended Edition(64) をご使用になる場合は、次のように読み替えてください。



種別	SORT Version8 - Extended Edition	SORT Version8 - Extended Edition(64)
コマンド名	exsort	exsort64
インストールパス名	/opt/HIEXSORT	/opt/HIEXSORT64
サンプルファイル名	exsort_sample.txt	exsort64_sample.txt
	exsort_sample1_sort.txt	exsort64_sample1_sort.txt
	exsort_sample2_merge.txt	exsort64_sample2_merge.txt
	exsort_sample3_select.txt	exsort64_sample3_select.txt
	exsort_sample4_sum.txt	exsort64_sample4_sum.txt
	exsort_sample5_sortsel.txt	exsort64_sample5_sortsel.txt
	exsort_sample6_mergesum.txt	exsort64_sample6_mergesum.txt
	exsort_sample7_sortselsum.txt	exsort64_sample7_sortselsum.txt
	exsort_sample8_reorg.txt	exsort64_sample8_reorg.txt
	exsort_sample9_reorg.txt	exsort64_sample9_reorg.txt
	exsort_sample10_outfile.txt	exsort64_sample10_outfile.txt
	exsort_sample11_skiprec.txt	exsort64_sample11_skiprec.txt
	exsort_sample12_altseq.txt	exsort64_sample12_altseq.txt
	exsort_sample13_csv.txt	exsort64_sample13_csv.txt
	exsort_sample14_sortsplitfile.txt	exsort64_sample14_sortsplitfile.txt
	exsort_sample15_sortsplitdata.txt	exsort64_sample15_sortsplitdata.txt
	exsort_sample16_sortsplitkey.txt	exsort64_sample16_sortsplitkey.txt
exsort_sample17_sortcat.txt	exsort64_sample17_sortcat.txt	
exsort_altseq_sample.txt	exsort64_altseq_sample.txt	
環境変数名	EXSORTDATEOPT	EXSORT64DATEOPT
	EXSORTSORTTMPDIR	EXSORT64SORTTMPDIR
	EXSORTMERGETMPDIR	EXSORT64MERGETMPDIR
	EXSORTWORKSIZE	EXSORT64WORKSIZE
	EXSORTMSGOPT	EXSORT64MSGOPT
	EXSORTMSGFILE	EXSORT64MSGFILE
	EXSORTMGTOPT	EXSORT64MGTOPT
	EXSORTDUPOPT	EXSORT64DUPOPT
環境ファイル名	Exsortconf.ini	Exsortconf64.ini
デバッグ情報ファイル名	exsortdmp	exsortdmp64
	exsortrdmp	exsortrdmp64
	EXSORTIODMP	EXSORTIODMP64

種別	SORT Version8 - Extended Edition	SORT Version8 - Extended Edition(64)
ログファイル名	ExSORT.log	ExSORT64.log
	ExSORT1.log	ExSORT641.log

### 付録 M.3 英略語

このマニュアルでは、次に示す英略語を使用します。

英略語	説明
CSV	Comma Separated Values
EBCDIK	Extended Binary-Coded-Decimal Interchange Kana Code
IPF	Itanium <sup>(R)</sup> Processor Family
IVS	Ideographic Variation Sequence
JIS	Japanese Industrial Standards
OS	Operating System
PP	Program Product
RDW	Record Descriptor Word

### 付録 M.4 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 N 用語解説

---

### (特殊文字)

---

¥

シングルクォーテーション (') などの文字をエスケープする場合に指定します。UNIX 環境によっては、バックスラッシュ (\) と表示されます。

---

### (英字)

---

#### EUC-JP

UNIX 系 OS の国際化対応のために開発された文字コード体系です。日本語対応ワークステーションで、内部コードとして広く使用されています。

#### IVS

IVS とは、同じ意味の漢字文字を複数の字形で表現できる仕組みで、ISO/IEC 10646 で規定されている Unicode で実装されています。漢字を表すコードの直後に VS (Variation Selector, 異体字セレクタ) と呼ばれるコードを付加することで、その漢字の「異体字」を表現する方法です。また、基底文字 + VS を並べた文字符号列を IVS (Ideographic Variation Sequence) と呼びます。

#### Microsoft コードページ

Windows で実装されている各国語別にまとめられた符号化文字集合を示す文字コード体系です。一般的な日本語環境の Windows システムで使用するコードページは、コードページ 932 (CP932) です。

#### Shift\_JIS

1 バイト文字 (JIS X0201) と 2 バイト文字 (JIS X0208) を、エスケープシーケンスで切り替えることなく、JIS X0208 を移動 (シフト) させることで、混在して扱えるようにした文字コード体系です。

#### SORT

SORT は、ファイルに保管されたデータを、それぞれの利用目的に合った形で入出力するソート・マージ機能を提供するソフトウェアです。

SORT には、ソート機能、マージ機能、選択機能および集約機能があります。

#### SORT EE

SORT EE (SORT Version8 - Extended Edition) は、SORT より豊富なソート・マージ機能を提供するソフトウェアです。SORT が提供する機能に、編集機能や複数ファイル出力機能が追加されました。また、可変長データの CSV ファイルを入出力ファイルとして扱えます。

SORT EE には、基本機能 (ソート機能、マージ機能、コピー機能) とレコード処理機能 (選択機能、集約機能、編集機能、ファイル分割機能、ファイル連結機能) があります。SORT EE は、1 コマンドで基本機能とレコード処理機能を組み合わせて使用できます。

## UCS-2LE

little endian 形式の UCS-2 文字コードのことです。

## UCS-2BE

big endian 形式の UCS-2 文字コードのことです。

## UCS-4LE

little endian 形式の UCS-4 文字コードのことです。

## UCS-4BE

big endian 形式の UCS-4 文字コードのことです。

## Unicode

ISO/IEC で標準化された符号化文字集合の規格で、UCS-2 と UCS-4 の規格があります。

- UCS-2 (Universal Multiple-Octet Coded Character Set-2) は、2 オクテット (16bit) で文字を表現します。
- UCS-4 (Universal Multiple-Octet Coded Character Set-4) は、4 オクテット (32bit) で文字を表現します。

## Unicode シグニチャ (BOM)

テキストファイルの先頭に付けられる、Unicode のエンコード方式を示す数バイトのデータのことです。BOM (byte order mark) ともいいます。

## UTF-8

UTF-8 (Unicode Transformation Format 8) とは、Unicode を 8 ビット符号単位で表現するエンコード (符号化) 方式です。

## UTF-16

UTF-16 (Unicode Transformation Format 16) とは、Unicode を 16 ビット符号単位で表現するエンコード (符号化) 方式です。

## UTF-16BE

UTF-16 big endian のエンコード方式のことです。

## UTF-16LE

UTF-16 little endian のエンコード方式のことです。

## UTF-32

UTF-32 (Unicode Transformation Format 32) とは、Unicode を 32 ビット符号単位で表現するエンコード (符号化) 方式です。

## UTF-32BE

UTF-32 big endian のエンコード方式のことです。

## UTF-32LE

UTF-32 little endian のエンコード方式のことです。

## VS

Variation Selector (異体字セレクタ) と呼ばれるコードで、Unicode での IVS (異体字) を表現します。

VS の範囲としてサロゲートエリア内の「U+E0100 ~ U+E01EF」が割り当てられており、UTF-16 ではサロゲートペアで VS が表現されることになります。

## (ア行)

---

### 円記号 (¥)

シングルクォーテーション (') などの文字をエスケープする場合に指定します。UNIX 環境によっては、バックスラッシュ (\) と表示されます。

### エンディアン

バイトデータを格納する際のバイトオーダー (格納順序) のことです。上位バイトを低位番地の数値として扱うバイトオーダーを **big endian** といい、上位バイトを高位番地の数値として扱うバイトオーダーを **little endian** といいます。

例えば、英小文字の「a (U+0061)」は、UTF-16 big endian では「0x0061」、UTF-16 little endian では「0x6100」と表現されます。

## (カ行)

---

### 簡易コピー機能

簡易コピー機能とは、入力ファイルを出力ファイルに複写する機能です。レコード処理機能 (選択、集約、編集、ファイル分割、ファイル連結) と組み合わせることはできませんが、コマンド指定が簡単で使いやすく、また、レコード長とレコード件数の制限がありません。

### 環境ファイル

SORT の環境ファイル「wmsort.wse」(Windows の場合だけ) とは、各コマンドの実行時に省略したパラメタの初期値を設定するファイルです。

SORT EE の環境ファイル「Exsortconf.ini ファイル (64 ビット版は「Exsortconf64.ini ファイル)」とは、SORT EE の実行環境を初期化するときの値を設定するファイルです。

### 環境変数

SORT EE の環境変数は、exsort コマンドのパラメタ省略時に設定するデフォルト値を定義します。

SORT EE の環境変数の指定がなく、SORT の環境ファイルの指定がある場合は、SORT の環境ファイルの指定を有効とします。

### 基底文字

Unicode での IVS (異体字) の表現で、漢字を表すコード部分を「基底文字 (Base Character)」と呼びます。

### 結合文字

Unicode には、1 つの文字に別の文字を結合することで、修飾された 1 つの文字として表現する方法があります。結合 (修飾) する側の文字を「結合文字 (Combining Character)」といいます。ま

た、結合（修飾）される側の文字を「基底文字（Base Character）」といい、基底文字と結合文字を組み合わせた結果を「合成済み文字（Precomposed Character）」といいます。

また、合成済み文字と同じ字形の基底文字がある場合、その 2 つの文字は等価です。結合文字と合成済み文字の例を次に示します。

**【例】**

基底文字「は (U+306F)」 + 結合文字「゛ (U+3099)」

= 合成済み文字「ば (U+306F U+3099)」

= 基底文字「ば (U+3070)」

SORT EE では、結合文字については対応していません。合成済み文字と同じ字形の基底文字は等価と扱われませんので、注意してください。

### コードページ 932 (CP932)

Windows システムでの Shift\_JIS コード対応の文字コードを示します。

### コードページ 1200 (CP1200)

Windows システムでの UTF-16 エンコード対応の文字コードを示します。

### コードページ 1201 (CP1201)

Windows システムでの UTF-16 (Big Endian) エンコード対応の文字コードを示します。

### コードページ 12000 (CP12000)

Windows システムでの UTF-32 エンコード対応の文字コードを示します。

### コードページ 12001 (CP12001)

Windows システムでの UTF-32 (Big Endian) エンコード対応の文字コードを示します。

### コードページ 51932 (CP51932)

Windows システムでの EUC-JP コード対応の文字コードを示します。

### コードページ 65001 (CP65001)

Windows システムでの UTF-8 エンコード対応の文字コードを示します。

### コピー機能

コピー機能とは、入力ファイルを出力ファイルに複写する機能です。

レコード処理機能（選択、集約、編集、ファイル分割、ファイル連結）と組み合わせると、レコード処理機能だけを使用できます。

## (サ行)

---

### サロゲートペア

Unicode は、BMP と呼ばれる基本多言語面（Basic Multilingual Plane）と、拡張面の 2 種類で構成されます。

Unicode (UCS-2) の BMP では、65,535 文字までしか表現できないため、Unicode2.0 より拡張面を使用するサロゲートペアの方式が採用されました。

BMP の未定義領域 1,024 × 2 個 (0xD800 ~ 0xDBFF, 0xDC00 ~ 0xDFFF) を使用し、ペア (4

バイト) で 1 つの字を表現します。

JIS 規格の第三水準、第四水準の一部の漢字は、サロゲートペアで表現されます。

### 集約機能

集約機能とは、データ中の文字または数字をキーとして、指定した複数のレコードを一つのレコードに併合し、指定した項目の値を合計する機能です。

### 正規化

SORT での正規化とは、キー・項目に対応する入力データを、大小比較できる形式に変形することです。

### 世界協定時

UTC (Coordinated Universal Time) とは、世界共通の標準時刻のことで、グリニッジ標準時 (GMT) と同義です。

### 選択機能

選択機能とは、条件値とデータを比較し、条件に一致したデータを持つレコードをファイルから選択する機能です。

### ソート機能

ソート機能とは、ファイル内のレコードを、データ中の文字または数字をキーとして、昇順または降順に並べ替える機能です。

## (ハ行)

---

### 比較項目

比較項目とは、選択機能で、条件値と比較されるデータを格納している項目です。

### 編集機能

編集機能とは、レコード内の任意の部分を選択したり、任意のデータを埋め込んだりすることでレコードを再編成する機能です。

## (マ行)

---

### マージ機能

マージ機能とは、複数のファイルのデータを、データ中の文字または数字をキーとして、一つのファイルに併合する機能です。

### マルチバイト文字

1 文字の長さを可変長の多バイトで扱う文字のことです。

## (ラ行)

---

### レコードヘッダ

可変長順編成ファイルで、各レコードの先頭にあるレコード長を格納している領域のことです。日立メインフレームでは RDW (Record Descriptor Word, レコード記述語) といいます。

## (ワ行)

---

### ワイド文字

1文字の長さを固定長の多バイトで扱う文字のことです。

SORT EE では、1文字を UCS-2 と UTF-16 を 2 バイト、UCS-4 と UTF-32 を 4 バイトの固定長として扱います。



# 索引

## 記号

- + キー位置・キー長 [SORT の集約コマンド (UNIX の場合)] 585
- + キー位置・キー長 [シェル・スクリプトコマンド] 616
- + 項目位置・項目長 [SORT の集約コマンド (UNIX の場合)] 586
- + 項目位置・項目長 [SORT の選択コマンド (UNIX の場合)] 560
- altseq パラメタ 373
- a 集約項目情報 [SORT のシェル・スクリプトコマンド (UNIX の場合)] 616
- a 集約項目情報 [SORT の集約コマンド (UNIX の場合)] 585
- catinp パラメタ 441
- catout パラメタ 451
- cmdopt パラメタ 174
- cpyinp パラメタ 455
- cpyout パラメタ 457
- csvkey パラメタ 375
- csvreorg パラメタ 395
- csvsum パラメタ 389
- dateopt パラメタ 372
- E オプション 698
- function パラメタ 171
- g オプション 697
- inpfile パラメタ 258
- iosize パラメタ 396
- key パラメタ 241
- locatajst パラメタ 172
- L レコード長 [SORT の集約コマンド (UNIX の場合)] 583
- L レコード長 [SORT の選択機能 (UNIX の場合)] 558
- L レコード長 [シェル・スクリプトコマンド] 615
- mergetmpdir パラメタ 366
- outfile パラメタ 338
- o 出力ファイル名 [SORT の集約コマンド (UNIX の場合)] 584
- o 出力ファイル名 [SORT の選択コマンド (UNIX の場合)] 559
- o 出力ファイル名 [シェル・スクリプトコマンド] 615
- p パラメタ 166
- reorg パラメタ 337
- sl" 選択条件式" 559
- sorttmpdir パラメタ 364
- splitdata パラメタ 404
- splitfile パラメタ 400
- splitinp パラメタ 426
- splitkey パラメタ 421
- splitout パラメタ 431
- splittmpdir パラメタ 440
- sum パラメタ 252
- T2 オプション 697
- T2 マージ用一時ファイルのディレクトリ名 2 [シェル・スクリプトコマンド] 616
- T オプション 697
- T ソート用一時ファイルのディレクトリ名 [シェル・スクリプトコマンド] 615
- worksize パラメタ 368
- w 中間ファイルのディレクトリ [シェル・スクリプトコマンド] 616
- y オプション 696
- y ワークバッファサイズ [シェル・スクリプトコマンド] 615
- ¥¥ [用語解説] 1013
- {csvinclude | csvomit} オペランド [-catinp パラメタ] 450
- {csvinclude | csvomit} オペランド [-catout パラメタ] 454
- {csvinclude | csvomit} オペランド [-inpfile パラメタ] 315
- {csvinclude | csvomit} オペランド [-outfile パラメタ] 363
- {csvinclude | csvomit} オペランド [-splitinp パラメタ] 430
- {csvinclude | csvomit} オペランド [-splitout パラメタ] 439

- {include | omit} オペランド [-catinp パラメタ] 449
- {include | omit} オペランド [-catout パラメタ] 454
- {include | omit} オペランド [-inpfle パラメタ] 299
- {include | omit} オペランド [-outfile パラメタ] 363
- {include | omit} オペランド [-splitinp パラメタ] 430
- {include | omit} オペランド [-splitout パラメタ] 439
- {suffix | prefix} オペランド [-catinp パラメタ] 443
- {suffix | prefix} オペランド [-splitout パラメタ] 432

## 数字

---

- 10 進キーのキーチェック緩和機能 (SORT EE の機能) 36
- 10 進キーのチェックオプション (KEYNOCHK オプション) 185
- 10 進集約項目のチェックオプション (SUMNOCHK オプション) 192
- 10 進集約項目のチェック緩和機能 (SORT EE の機能) 38
- 10 進数オーバーフロー判定オプション (OVFLW オプション) 184
- 10 進数オーバーフロー判定機能 29
- 32 ビット版 SORT EE の環境変数 654
- 64 ビット版 SORT EE の環境変数 674

## A

---

- altseq 変換ファイル (SORT EE の機能) 833
- altseq 変換ファイルの作成例 488
- altseq 変換ファイルの指定方法 833
- altseq 変換ファイルの指定例 834
- altseq 変換ファイルの詳細 833

## B

---

- bom\_lgin=Unicode シグニチャの読み飛ばし機能オプション 722
- bom\_lngout=Unicode シグニチャの付加機能オプション [環境ファイル] 722

## C

---

- c' 文字条件値' [拡張 SORT コマンド] 603
- c' 文字条件値' [選択コマンド] 543
- cellnum オペランド [-catout パラメタ] 453
- cellnum オペランド [-outfile パラメタ] 357
- cellnum オペランド [-splitout パラメタ] 438
- cellsepa オペランド [-catinp パラメタ] 448
- cellsepa オペランド [-catout パラメタ] 452
- cellsepa オペランド [-inpfle パラメタ] 268
- cellsepa オペランド [-outfile パラメタ] 354
- cellsepa オペランド [-splitinp パラメタ] 428
- cellsepa オペランド [-splitout パラメタ] 437
- codetype オペランド [-catinp パラメタ] 449
- codetype オペランド [-inpfle パラメタ] 274
- codetype オペランド [-splitinp パラメタ] 429
- csvkey オペランド [-splitkey パラメタ] 422
- csvreorg オペランド [-catinp パラメタ] 449
- csvreorg オペランド [-catout パラメタ] 454
- csvreorg オペランド [-inpfle パラメタ] 290
- csvreorg オペランド [-outfile パラメタ] 362
- csvreorg オペランド [-splitinp パラメタ] 429
- csvreorg オペランド [-splitout パラメタ] 438

- CSV 数値データ (固定小数点数) の形式  
 チェックオプション (CSVFRMCHK オプション) 235
- CSV 数値データ (固定小数点数) の除外文字の位置規則 [CSV ファイルのデータの扱い] 813
- CSV データ形式 816
- CSV ファイル形式 815
- CSV ファイル形式と CSV データ形式 (SORT EE の機能) 815
- CSV ファイル内のナル文字列の扱い (SORT EE の機能) 820
- CSV ファイルのデータの扱い (SORT EE の機能) 808
- currency オペランド [-catinp パラメタ] 448
- currency オペランド [-inpfile パラメタ] 270
- currency オペランド [-splitinp パラメタ] 428
- C 言語 622
- C 言語で作成したプログラムの不正による障害の場合 (Windows の場合) 773
- C 言語プログラムでの使い方 (Windows の場合) 621
- D**
- 
- d' 日付条件値' [拡張 SORT コマンド] 603
- d' 日付条件値' [選択コマンド] 543
- DATE\_INVALID\_NOCHK オプション [32 ビット版 SORT EE] 659
- DATE\_INVALID\_NOCHK オプション [64 ビット版 SORT EE] 679
- date\_opt= 日付項目の基準年度 720
- DATETIME\_INVALID\_NOCHK オプション [32 ビット版 SORT EE] 660
- DATETIME\_INVALID\_NOCHK オプション [64 ビット版 SORT EE] 679
- E**
- 
- EBCDIK コードキー比較機能 26
- EBCDIK コード比較オプション (EBCJ, EBCE オプション) 180
- EUC-JP [用語解説] 1013
- EXS\_ALTSEQPATH 705
- EXS\_CELLSEPA 706
- EXS\_CODETYPE 715
- EXS\_COMPLETE\_MSGOPT 705
- EXS\_CSVKEYNUMMAX 712
- EXS\_CSVSELNUMMAX 714
- EXS\_CSVSUMNUMMAX 713
- EXS\_CURRENCY 707
- EXS\_DUMPLOG\_FILESIZE 718
- EXS\_EXSORTDMP\_OPT 710
- EXS\_KEYNUMMAX 710
- EXS\_LOGPATH 705
- EXS\_NULLSEQ 708
- EXS\_OUTCELLSEPA 709
- EXS\_OUTFILEENUMUNLIMIT 717
- EXS\_OUTRECSEPA 708
- EXS\_PACK\_SIGNTYPE 716
- EXS\_PADDING 708
- EXS\_PRMSIZEUNLIMIT 717
- EXS\_QUOTTYPE 706
- EXS\_RECSEPA 705
- EXS\_REPOLOG\_FILESIZE 714
- EXS\_ROUND 707
- EXS\_SELNUMMAX 712
- EXS\_SUMNUMMAX 711
- EXS\_UNPACK\_DATATYPE 716
- EXSORT\_HI\_WORKSIZE 661
- EXSORT64\_HI\_WORKSIZE 681
- EXSORT64BOMGETPUT 682
- EXSORT64BOMLNGIN 681
- EXSORT64BOMLNGOUT 682
- EXSORT64CATINPNOERR 685
- EXSORT64CATNAMESEQ 686
- EXSORT64DATEOPT 674
- EXSORT64DELBLANKLINE 691
- EXSORT64DUPOPT 679
- EXSORT64FILESPLITNUMB 686
- EXSORT64GOPTION 687
- EXSORT64INPIOSIZE 680
- EXSORT64IOSIZE 680

- EXSORT64LOGSUPPRESS 693  
 EXSORT64MERGETMPDIR 674  
 EXSORT64MERGETMPIO SIZE 681  
 EXSORT64MGTOPT 678  
 EXSORT64MSGFILE 677  
 EXSORT64MSGOPT 675  
 EXSORT64NOCREATEZEROFILE 685  
 EXSORT64OUTFILENODUPCHK 692  
 EXSORT64OUTFILENUMUNLIMIT 692  
 EXSORT64OUTIOSIZE 680  
 EXSORT64OVFERROPT 687  
 EXSORT64PACKSIGNTYPE 687  
 EXSORT64PRMSIZEUNLIMIT 692  
 EXSORT64SORTLIBOPTION 693  
 EXSORT64SORTTMPDIR 674  
 EXSORT64SORTTMPIO SIZE 680  
 EXSORT64SPLITNUMBONE 686  
 EXSORT64SPLITTMPDIR 686  
 EXSORT64SPLITTMPIO SIZE 686  
 EXSORT64STRINGDIR 684  
 EXSORT64SUPPRESSCATMSG 685  
 EXSORT64SUPPRESSSPLITMSG 685  
 EXSORT64UNPACKDATATYPE 688  
 EXSORT64WORKSIZE 674  
 EXSORTBOMGETPUT 663  
 EXSORTBOMLNGIN 662  
 EXSORTBOMLNGOUT 662  
 EXSORTCATINPNOERR 666  
 EXSORTCATNAMESEQ 667  
 Exsortconf.ini ファイルの指定 833  
 EXSORTDATEOPT 654  
 EXSORTDELBLANKLINE 672  
 EXSORTDUPOPT 660  
 EXSORTFILESPLITNUMB 666  
 EXSORTGOPTION 667  
 EXSORTINPIO SIZE 660  
 EXSORTIOSIZE 660  
 EXSORTLOGSUPPRESS 673  
 EXSORTMERGETMPDIR 655  
 EXSORTMERGETMPIO SIZE 661  
 EXSORTMGTOPT 658  
 EXSORTMSGFILE 658  
 EXSORTMSGOPT 656  
 EXSORTNOCREATEZEROFILE 666  
 EXSORTOUTFILENODUPCHK 673  
 EXSORTOUTFILENUMUNLIMIT 672  
 EXSORTOUTIOSIZE 661  
 EXSORTOVFERROPT 667  
 EXSORTPACKSIGNTYPE 668  
 EXSORTPRMSIZEUNLIMIT 673  
 EXSORTSORTTMPDIR 655  
 EXSORTSORTTMPIO SIZE 661  
 EXSORTSPLITNUMBONE 666  
 EXSORTSPLITTMPDIR 667  
 EXSORTSPLITTMPIO SIZE 667  
 EXSORTSTRINGDIR 665  
 EXSORTSUPPRESSCATMSG 666  
 EXSORTSUPPRESSSPLITMSG 665  
 EXSORTUNPACKDATATYPE 669  
 EXSORTWORKSIZE 655  
 exsort コマンド形式 [コピー機能] 126  
 exsort コマンド形式 [ソート機能] 118  
 exsort コマンド形式 [ファイル分割機能]  
     130  
 exsort コマンド形式 [ファイル連結機能]  
     134  
 exsort コマンド形式 [マージ機能] 122  
 exsort コマンドの形式 117  
 exsort コマンドのパラメタおよびオペランド  
     の詳細 157  
 exsort パラメタファイル形式 [コピー機能]  
     128  
 exsort パラメタファイル形式 [ソート機能]  
     120  
 exsort パラメタファイル形式 [ファイル分割  
     機能] 132  
 exsort パラメタファイル形式 [ファイル連結  
     機能] 136  
 exsort パラメタファイル形式 [マージ機能]  
     124  
 exsort パラメタファイルの表記規則 146
- ## F
- 
- fileno オペランド [-splitfile パラメタ] 400  
 fileorg オペランド 455  
 fileorg オペランド [-catinp パラメタ] 443

fileorg オペランド [-catout パラメタ] 451  
 fileorg オペランド [-inpfile パラメタ] 258  
 fileorg オペランド [-outfile パラメタ] 339  
 fileorg オペランド [-splitinp パラメタ] 426  
 fileorg オペランド [-splitout パラメタ] 432  
 file オペランド [-catinp パラメタ] 443  
 file オペランド [-catout パラメタ] 451  
 file オペランド [-inpfile パラメタ] 259  
 file オペランド [-outfile パラメタ] 342  
 file オペランド [-splitinp パラメタ] 427

---

## I

ISAM [SORT の選択コマンド] 558  
 ISAM [SORT のソートコマンド] 505  
 ISAM [SORT のマージコマンド] 529  
 IVS [用語解説] 1013  
 IVS 対応 864

---

## J

JIS8 単位コード体系と EBCDIK コード体系  
 の変換表 806

---

## K

KEY\_INVALID\_NOCHK オプション [32  
 ビット版 SORT EE] 658  
 KEY\_INVALID\_NOCHK オプション [64  
 ビット版 SORT EE] 678  
 key オペランド [-splitkey パラメタ] 421

---

## L

LOCK\_INVALID オプション [32 ビット版  
 SORT EE] 660  
 LOCK\_INVALID オプション [64 ビット版  
 SORT EE] 679

---

## M

method オペランド [-splitfile パラメタ]  
 400  
 Microsoft コードページ [用語解説] 1013  
 minlen オペランド [-catout パラメタ] 452  
 minlen オペランド [-outfile パラメタ] 347

minlen オペランド [-splitout パラメタ]  
 437  
 msg\_file= メッセージの出力先ファイル名  
 [環境ファイル] 720  
 msg\_opt= メッセージ出力オプション [環境  
 ファイル] 720

---

## N

n' 数値条件値' [拡張 SORT コマンド] 603  
 n' 数値条件値' [選択コマンド] 543  
 nameseq オペランド [-catinp パラメタ]  
 445  
 nosplitcsvdata オペランド [-splitdata パラ  
 メタ] 413  
 nosplitdata オペランド [-splitdata パラメ  
 タ] 405  
 nullseq オペランド [-catinp パラメタ] 449  
 nullseq オペランド [-inpfile パラメタ] 273  
 nullseq オペランド [-splitinp パラメタ]  
 429

---

## P

padding オペランド [-catout パラメタ] 453  
 padding オペランド [-outfile パラメタ] 360  
 padding オペランド [-splitout パラメタ]  
 438

---

## Q

quottype オペランド [-catinp パラメタ]  
 448  
 quottype オペランド [-catout パラメタ]  
 453  
 quottype オペランド [-inpfile パラメタ]  
 269  
 quottype オペランド [-outfile パラメタ]  
 358  
 quottype オペランド [-splitinp パラメタ]  
 428  
 quottype オペランド [-splitout パラメタ]  
 438

## R

- 
- reclen オペランド [-catinp パラメタ] 447
  - reclen オペランド [-catout パラメタ] 452
  - reclen オペランド [-inpfile パラメタ] 259
  - reclen オペランド [-outfile パラメタ] 342
  - reclen オペランド [-splitinp パラメタ] 427
  - reclen オペランド [-splitout パラメタ] 437
  - recno オペランド [-splitdata パラメタ] 404
  - recno オペランド [-splitkey パラメタ] 423
  - recsepa オペランド [-catinp パラメタ] 448
  - recsepa オペランド [-catout パラメタ] 452
  - recsepa オペランド [-inpfile パラメタ] 266
  - recsepa オペランド [-outfile パラメタ] 349
  - recsepa オペランド [-splitinp パラメタ] 428
  - recsepa オペランド [-splitout パラメタ] 437
  - reorg オペランド [-catinp パラメタ] 449
  - reorg オペランド [-catout パラメタ] 453
  - reorg オペランド [-inpfile パラメタ] 275
  - reorg オペランド [-outfile パラメタ] 362
  - reorg オペランド [-splitinp パラメタ] 429
  - reorg オペランド [-splitout パラメタ] 438
  - rmerge 527
  - rmerge64 527
  - round オペランド [-catinp パラメタ] 448
  - round オペランド [-inpfile パラメタ] 271
  - round オペランド [-splitinp パラメタ] 428
  - rselect 551
  - rselect64 551
  - rsort 502
  - rsort64 502
  - rsortsum 614
  - rsortsum (rsortsum64) コマンドの形式 614
  - rsortsum64 614
  - rsortsum コマンドの指定内容 620
  - rsum 581
  - rsum64 581
- 
- S
  - skiprec オペランド [-catinp パラメタ] 447
  - skiprec オペランド [-inpfile パラメタ] 262
  - skiprec オペランド [-splitinp パラメタ] 427
  - SORT\_BOMLNGIN 698
  - SORT\_BOMLNGOUT 698
  - SORT\_DMPDIR 699
  - SORT\_STRINGDIR 699
  - SORT [用語解説] 1013
  - SORT EE [用語解説] 1013
  - SORT EE と SORT 1
  - SORT EE と SORT の概要 3
  - SORT EE の一時ファイルの作成条件 728
  - SORT EE の一時ファイル容量の算出方法 730
  - SORT EE の環境ファイルの設定 702
  - SORT EE の環境変数の設定 648
  - SORT EE の障害情報 758
  - SORT EE の使用例 459
  - SORT EE の使い方 109
  - SORT EE の入出力障害時のファイルの扱い 758
  - SORT EE のパラメタの最小値および最大値 99
  - SORT EE のファイル排他機能 32
  - SORT EE のワークバッファサイズ自動見張り機能 742
  - SORT EE メッセージ出力オプション 880
  - SORT EE メッセージに出力されるデバッグ情報 878
  - SORT EE メッセージの記述形式 878
  - SORT EE メッセージの種類 878
  - sortidmp 773
  - SORTOPT 696
  - sortrdmp 772
  - sortrdmp64 772
  - sortrdmp の内容を確認する [キーまたは項目の不正による障害の対処方法] 767
  - SORT から SORT EE への移行ガイド 775
  - SORT の一時ファイルの作成条件 733
  - SORT の一時ファイル容量の算出方法 735
  - SORT の環境ファイル指定時の注意事項 723
- 
- Shift\_JIS [用語解説] 1013

SORT の環境ファイルの設定 (Windows の場合) 719  
 SORT の環境ファイルのパラメタの意味 719  
 SORT の環境ファイルの有効範囲 722  
 SORT の環境変数の設定 (UNIX の場合) 696  
 SORT の機能拡張オプションの環境設定 750  
 SORT の機能拡張オプションの注意事項と見直しポイント 754  
 SORT の障害情報 769  
 SORT の入出力障害時のファイルの扱い 769  
 SORT のパラメタの最小値および最大値 98  
 SORT のファイル排他機能 31  
 SORT のメッセージ一覧 967  
 SORT メッセージ 966  
 SORT メッセージの記述形式 966  
 SORT メッセージの種類 966  
 spno オペランド [-splitdata パラメタ] 405  
 spno オペランド [-splitkey パラメタ] 423  
 startrec オペランド [-catinp パラメタ] 447  
 startrec オペランド [-inpfile パラメタ] 264  
 startrec オペランド [-splitinp パラメタ] 427  
 stopaft オペランド [-catinp パラメタ] 447  
 stopaft オペランド [-inpfile パラメタ] 263  
 stopaft オペランド [-splitinp パラメタ] 427  
 string\_dir= スtring情報格納一時ファイルディレクトリ名 721  
 SUM\_INVALID\_NOCHK オプション [32 ビット版 SORT EE] 659  
 SUM\_INVALID\_NOCHK オプション [64 ビット版 SORT EE] 678

## T

---

temp\_dir1= ソート用一時ファイルディレクトリ [環境ファイル] 720  
 temp\_dir2= 中間ファイルディレクトリ [環境ファイル] 721  
 temp\_dir3= マージ用一時ファイルディレクトリ [環境ファイル] 721  
 TMPDIR 616

## U

---

UCS-2BE [用語解説] 1014  
 UCS-2LE [用語解説] 1014  
 UCS-4BE [用語解説] 1014  
 UCS-4LE [用語解説] 1014  
 Unicode [用語解説] 1014  
 Unicode シグニチャ (BOM) [用語解説] 1014  
 Unicode シグニチャ (BOM) 付きパラメタファイルの扱い 154  
 Unicode 操作文字コード一覧 (SORT EE) 835  
 UTF-16 [用語解説] 1014  
 UTF-16BE [用語解説] 1014  
 UTF-16LE [用語解説] 1014  
 UTF-32 [用語解説] 1014  
 UTF-32BE [用語解説] 1014  
 UTF-32LE [用語解説] 1014  
 UTF-8 [用語解説] 1014

## V

---

VKEY\_POS\_NOCHK オプション [32 ビット版 SORT EE] 659  
 VKEY\_POS\_NOCHK オプション [64 ビット版 SORT EE] 678  
 VS [用語解説] 1015

## W

---

wnmerge 518  
 wnmerge64 518  
 WNS\_Create 624  
 WNS\_Date\_Set 638  
 WNS\_File\_Create 624  
 WNS\_File\_Init 625  
 WNS\_File\_Release 626  
 WNS\_File\_Set 626  
 WNS\_GetError 627  
 WNS\_GetMsg 627  
 WNS\_Key\_Create 628  
 WNS\_Key\_Init 628  
 WNS\_Key\_Release 629  
 WNS\_Key\_Set 629

WNS\_Release 630  
 WNS\_Sel\_Create 631  
 WNS\_Sel\_Init 632  
 WNS\_Sel\_Release 632  
 WNS\_Sel\_Set 633  
 WNS\_Select 631  
 WNS\_Sort 634  
 WNS\_SortEX 635  
 WNS\_Sum 635  
 WNS\_Sum\_Create 636  
 WNS\_Sum\_Init 637  
 WNS\_Sum\_Release 637  
 WNS\_Sum\_Set 637  
 WNS\_Work\_Set 639  
 wnselect 538  
 wnselect64 538  
 wnsort 490  
 wnsort64 490  
 wnsortev 719  
 wnsortex 594  
 wnsortex64 594  
 wnsum 572  
 wnsum64 572  
 work\_size= ワークバッファサイズ [環境  
 ファイル] 721

## X

---

x'16 進数条件値' [拡張 SORT コマンド]  
 603  
 x'16 進数条件値' [選択コマンド] 543

## あ

---

アウトコアソート 741

## い

---

一時ファイルの作成条件とファイル容量の算  
 出方法 725  
 一時ファイルを作成しない場合のワークバッ  
 ファサイズ [SORT] 733  
 一時ファイルを作成しない場合のワークバッ  
 ファサイズ [SORT EE] 728  
 インコアソート 738

引用符のエスケープ [CSV ファイルのデー  
 タの扱い] 812

## え

---

円記号 (¥¥) [用語解説] 1015  
 エンディアン [用語解説] 1015  
 エンディアンオプション機能 30

## お

---

オペランド 138  
 オペランド形式 [コマンドの表記規則] 138  
 オペランドの意味 [SORT の集約機能  
 (UNIX の場合)] 581  
 オペランドの意味 [SORT の集約機能  
 (Windows の場合)] 572  
 オペランドの意味 [SORT の選択機能  
 (UNIX の場合)] 551  
 オペランドの意味 [SORT の選択機能  
 (Windows の場合)] 538  
 オペランドの意味 [SORT のソート機能  
 (UNIX の場合)] 502  
 オペランドの意味 [SORT のソート機能  
 (Windows の場合)] 490  
 オペランドの意味 [SORT のマージ機能  
 (UNIX の場合)] 527  
 オペランドの意味 [SORT のマージ機能  
 (Windows の場合)] 518  
 オペランドの意味 [拡張 SORT コマンド]  
 595  
 オペランドの意味 [シェル・スクリプトコマ  
 ンド] 615

## か

---

拡張 SORT コマンドの形式 594  
 拡張 SORT コマンドの使い方 (Windows の  
 場合) 593  
 拡張 SORT コマンドのパラメタの意味 597  
 簡易コピー [ソート機能の使用例] 486  
 簡易コピー機能 [用語解説] 1015  
 環境ファイル [用語解説] 1015  
 環境ファイルの設定 701



環境変数〔**exsort** パラメタファイルの表記規則〕 148  
 環境変数〔用語解説〕 1015  
 環境変数の設定 647  
 関数一覧 622  
 関数の機能と形式 622  
 関数の組み合わせ 623  
 完了メッセージオプション〔**SORT EE** の障害情報の出力〕 766

## き

キー〔集約機能〕 9  
 キー〔ソート機能〕 5  
 キー〔マージ機能〕 6  
 キー位置のチェックオプション (POSNOCHK オプション) 195  
 キー位置のチェック緩和機能 (**SORT EE** の機能) 39  
 キー属性〔**SORT** の集約機能〕 584  
 キー属性〔シェル・スクリプトコマンド〕 616  
 キー属性と形式 82  
 キー属性のパラメタ (集約機能) 584  
 キー属性のパラメタ (マージ機能) 530  
 キーの指定範囲 585  
 キーの指定範囲〔**SORT** のソート機能 (UNIX の場合)〕 509  
 キーの指定範囲〔**SORT** のマージ機能 (UNIX の場合)〕 531  
 キーまたは項目の不正による障害の対処方法 767  
 キーまたは項目の不正による障害の場合〔**SORT**〕 771  
 既存ファイルの書き換え防止機能 (**SORT** の機能) 45  
 基底文字〔用語解説〕 1015  
 機能オプション〔シェル・スクリプトコマンド〕 615  
 機能の実行形態〔**SORT EE**〕 115  
 機能の実行順序〔**SORT EE**〕 110  
 機能の実行順序と実行形態〔**SORT EE**〕 110  
 キャッシュファイルの同期出力機能 34

強制マージオプション (**FMRG** オプション) 183  
 強制マージ機能 27

## く

空行削除オプション (**DELBLANKLINE** オプション) 232

## け

結合文字〔用語解説〕 1015

## こ

コーディング規則〔**exsort** パラメタファイルの表記規則〕 146  
 コーディング例〔C 言語〕 640  
 コードページ 1200 (**CP1200**)〔用語解説〕 1016  
 コードページ 12000 (**CP12000**)〔用語解説〕 1016  
 コードページ 12001 (**CP12001**)〔用語解説〕 1016  
 コードページ 1201 (**CP1201**)〔用語解説〕 1016  
 コードページ 51932 (**CP51932**)〔用語解説〕 1016  
 コードページ 65001 (**CP65001**)〔用語解説〕 1016  
 コードページ 932 (**CP932**)〔用語解説〕 1016  
 コピー+選択+CSV ファイル〔コピー機能の使用例〕 484  
 コピー+ファイル編成変更〔ソート機能の使用例〕 485  
 コピー+レコード長変更〔ソート機能の使用例〕 486  
 コピー機能〔**exsort** コマンドの形式〕 126  
 コピー機能〔用語解説〕 1016  
 コピー機能とは (**SORT EE** の機能) 7  
 コピー機能の処理の流れ 19  
 コマンド実行時のコードページ (Windows の場合) 144

- コマンド実行時のロケール (UNIX の場合)  
142
- コマンド指定時の注意事項 [SORT の集約機能 (UNIX の場合)] 586
- コマンド指定時の注意事項 [SORT の集約機能 (Windows の場合)] 578
- コマンド指定時の注意事項 [SORT の選択機能 (UNIX の場合)] 563
- コマンド指定時の注意事項 [SORT の選択機能 (Windows の場合)] 545
- コマンド指定時の注意事項 [SORT のソート機能 (UNIX の場合)] 509
- コマンド指定時の注意事項 [SORT のソート機能 (Windows の場合)] 496
- コマンド指定時の注意事項 [SORT のマージ機能 (UNIX の場合)] 532
- コマンド指定時の注意事項 [SORT のマージ機能 (Windows の場合)] 522
- コマンド指定時の注意事項 [拡張 SORT コマンド] 609
- コマンド指定時の注意事項 [シェル・スクリプトコマンド] 618
- コマンドの形式 [SORT の集約コマンド (UNIX の場合)] 581
- コマンドの形式 [SORT の選択コマンド (UNIX の場合)] 551
- コマンドの形式 [シェル・スクリプトコマンド] 614
- コマンドの使い方 [SORT の集約機能 (UNIX の場合)] 581
- コマンドの使い方 [SORT の集約機能 (Windows の場合)] 572
- コマンドの使い方 [SORT の選択機能 (UNIX の場合)] 551
- コマンドの使い方 [SORT の選択機能 (Windows の場合)] 538
- コマンドの使い方 [SORT のソート機能 (UNIX の場合)] 502
- コマンドの使い方 [SORT のソート機能 (Windows の場合)] 490
- コマンドの使い方 [SORT のマージ機能 (UNIX の場合)] 527
- コマンドの使い方 [SORT のマージ機能 (Windows の場合)] 518
- コマンドの使い方 [拡張 SORT コマンド] 594
- コマンドの使い方 [シェル・スクリプトコマンド] 614
- コマンドの表記規則 138
- コマンド予約語の表記 [コマンドの表記規則] 139
- コンパイル方法 [C 言語] 643

---

 さ
 

---

- 左分離符号付き数字のデータ形式 (SORT EE の機能) 827
- サロゲートペア [用語解説] 1016

---

 し
 

---

- シェル・スクリプトコマンドの使い方 (UNIX の場合) 613
- 指定できる値の範囲 98
- 指定できる条件値の長さ (選択コマンド) 543
- 指定できる条件値の長さ [拡張 SORT コマンド] 603
- シフト JIS コード比較オプション (SJIS オプション) 216
- 集約 [コピー機能の使用例] 481
- 集約オプション 581
- 集約オプションのパラメタ 581
- 集約キー [CSV ファイル内のナル文字列の扱い] 822
- 集約キーの指定範囲 [SORT の集約機能 (Windows の場合)] 576
- 集約機能 [用語解説] 1017
- 集約機能とは 9
- 集約結果の範囲の違い 574
- 集約項目 9
- 集約項目 [CSV ファイル内のナル文字列の扱い] 825
- 集約項目情報の指定 588
- 集約項目の指定範囲 [SORT の集約機能 (Windows の場合)] 577

集約項目の属性のパラメタ 585  
 集約コマンドの形式 581  
 集約コマンドの指定内容 [集約項目を複数指定して集約する例 (UNIX の場合)] 591  
 集約コマンドの指定内容 [文字項目をキーとして集約する例 (UNIX の場合)] 589  
 集約コマンドの使い方 (UNIX の場合) 581  
 集約コマンドの使い方 (Windows の場合) 572  
 終了コードの種類 (SORT の場合) 967  
 出力ファイルのファイル編成 [SORT の集約コマンド (UNIX の場合)] 583  
 出力ファイルのファイル編成 [SORT の選択コマンド (UNIX の場合)] 558  
 出力ファイルのファイル編成 [シェル・スクリプトコマンド] 615  
 出力ファイルのファイル編成のパラメタ (集約機能) 583  
 出力ファイルのファイル編成のパラメタ (選択機能) 558  
 出力ファイルのファイル編成のパラメタ (マージ機能) 529  
 出力ファイル名 457  
 障害時の対処方法 757  
 条件値 [選択機能] 8  
 条件値の扱い 561  
 詳細エラーコード 644  
 詳細エラーコードの内容 644  
 使用例 [SORT の集約機能 (UNIX の場合)] 588  
 使用例 [SORT の集約機能 (Windows の場合)] 579  
 使用例 [SORT の選択機能 (UNIX の場合)] 565  
 使用例 [SORT の選択機能 (Windows の場合)] 547  
 使用例 [SORT のソートコマンド (UNIX の場合)] 510  
 使用例 [SORT のソートコマンド (Windows の場合)] 498  
 使用例 [SORT のマージコマンド (UNIX の場合)] 534

使用例 [SORT のマージコマンド (Windows の場合)] 523  
 使用例 [拡張 SORT コマンド] 610  
 使用例 [シェル・スクリプトコマンド] 619  
 処理の流れの概要 16

## す

---

数値 (固定小数点数) データの端数処理 [CSV ファイルのデータ入力時の端数処理] 808  
 スクリプト・ファイル 618  
 スペース (空白) の扱い [SORT の集約コマンド (UNIX の場合)] 587  
 スペース (空白) の扱い [SORT の選択コマンド (UNIX の場合)] 563

## せ

---

正規化 [用語解説] 1017  
 性能チューニング方法 737  
 世界協定時 [用語解説] 1017  
 ゼロデータ比較オプション (DIFFZERO オプション) 222  
 選択+ソート+集約 [ソート機能の使用例] 462  
 選択+ソート [ソート機能の使用例] 461  
 選択 [コピー機能の使用例] 481  
 選択オプション 551  
 選択オプションのパラメタ 552  
 選択機能 [用語解説] 1017  
 選択機能とは 8  
 選択コマンドの形式 551  
 選択コマンドの指定内容 [1 種類の条件式でデータを選択する例 (UNIX の場合)] 565  
 選択コマンドの指定内容 [2 種類の条件式でデータを選択する例 (UNIX の場合)] 567  
 選択コマンドの指定内容 [ソート処理と組み合わせで選択する例 (UNIX の場合)] 568  
 選択コマンドの使い方 (UNIX の場合) 551  
 選択コマンドの使い方 (Windows の場合) 538  
 選択条件式の書き方 [exsort パラメタファイルの表記規則] 148

選択条件式の書き方〔コマンドの表記規則〕  
140

選択条件式のブレイクオプション (SELBRK  
オプション) 238

## そ

ソート + altseq〔ソート機能の使用例〕 463

ソート + CSV ファイル + UCS-2LE〔ソート  
機能の使用例〕 474

ソート + CSV ファイル + UTF-16LE〔ソート  
機能の使用例〕 476

ソート + csv ファイル〔ソート機能の使用  
例〕 464

ソート + キー分割〔ソート機能の使用例〕  
467

ソート + 均等分割〔ソート機能の使用例〕  
465

ソート + 集約 + CSV ファイル〔ソート機能  
の使用例〕 469

ソート + 出力レコード選択 + 出力レコード編  
集 + ファイル分割 (キー分割) + CSV  
ファイル〔ソート機能の使用例〕 470

ソート + 定数分割〔ソート機能の使用例〕  
466

ソート + ファイル連結〔ソート機能の使用  
例〕 468

ソート + 符号なし 2 進数のマスクビット指定  
キー〔ソート機能の使用例〕 475

ソート〔ソート機能の使用例〕 461

ソートキー・マージキー〔CSV ファイル内  
のナル文字列の扱い〕 820

ソートキーの指定範囲〔SORT のソート機能  
(Windows の場合)〕 496

ソート機能〔exsort コマンドの形式〕 118

ソート機能〔用語解説〕 1017

ソート機能以外の機能の処理時間を短くする  
ためには 746

ソート機能とは 5

ソート機能の処理時間を短くするためには  
738

ソート機能の処理の流れ 16

ソートコマンドの使い方 (UNIX の場合)  
502

ソートコマンドの使い方 (Windows の場合)  
490

ソート用一時ファイルおよびマージ用一時  
ファイルを作成する場合のワークバッファ  
サイズ〔SORT〕 733

ソート用一時ファイルおよびマージ用一時  
ファイルを作成する場合のワークバッファ  
サイズ〔SORT EE〕 728

ソート用一時ファイルだけを作成する場合の  
ワークバッファサイズ〔SORT〕 733

ソート用一時ファイルだけを作成する場合の  
ワークバッファサイズ〔SORT EE〕 728

## ち

中間ファイルのエラー〔シェル・スクリプト  
コマンド〕 618

中間ファイルの削除〔シェル・スクリプトコ  
マンド〕 618

中間ファイルの容量〔シェル・スクリプトコ  
マンド〕 618

重複レコードの入力順序保存機能 24

重複レコードの入力順序保存機能 (DUP |  
NODUP オプション) 178

## て

デバッグ情報ファイル〔SORT EE の障害情  
報の出力〕 758

デバッグ情報ファイル〔SORT の障害情報の  
出力〕 769

## と

同一パラメタの複数指定〔SORT の集約機能  
(UNIX の場合)〕 587

同一パラメタの複数指定〔SORT の選択コマ  
ンド (UNIX の場合)〕 563

## な

内部 10 進数符号保障オプション (ABS オプ  
ション) 185

内部 10 進数符号保障機能 35

## に

---

- 入出力ファイル共用・排他オプション  
(NOLOCK | EXC オプション) 209
- 入出力ファイル名の指定 [環境変数の設定]  
694
- 入力ファイルのエンディアンオプション  
(BIG | INPBIG | INPLTL オプション)  
181
- 入力ファイルのファイル編成 [SORT の集約  
コマンド (UNIX の場合)] 583
- 入力ファイルのファイル編成 [SORT の選択  
コマンド (UNIX の場合)] 557
- 入力ファイルのファイル編成 [シェル・スクリプト  
コマンド] 615
- 入力ファイルのファイル編成のパラメタ (集約  
機能) 583
- 入力ファイルのファイル編成のパラメタ (選  
択機能) 557
- 入力ファイルのファイル編成のパラメタ  
(マージ機能) 529
- 入力ファイル名 456
- 入力ファイル名 [SORT の集約コマンド  
(UNIX の場合)] 586
- 入力ファイル名 [SORT の選択コマンド  
(UNIX の場合)] 562
- 入力ファイル名 [シェル・スクリプトコマン  
ド] 616
- 入力レコード件数とファイルサイズ 107
- 入力レコードの読み飛ばし・打ち切り機能  
(SORT EE の機能) 43

## は

---

- パラメタ 138
- パラメタ [指定できる値の範囲] 98
- パラメタおよびオペランドの組み合わせ 159
- パラメタおよびオペランドの指定順序 [コマ  
ンドの表記規則] 138
- パラメタ形式 [コマンドの表記規則] 138
- パラメタの指定の組み合わせ [拡張 SORT  
コマンド] 595
- パラメタの入力順序 [SORT の集約コマンド  
(UNIX の場合)] 586

- パラメタの入力順序 [SORT の選択コマンド  
(UNIX の場合)] 563
- パラメタファイル作成時の注意事項 [拡張  
SORT コマンド] 609
- パラメタファイルの形式 [拡張 SORT コマ  
ンド] 595
- パラメタファイルを記述する文字コード 152

## ひ

---

- 比較演算子 560
- 比較演算子のパラメタ 560
- 比較項目 [CSV ファイル内のナル文字列の  
扱い] 823
- 比較項目 [選択機能] 8
- 比較項目 [用語解説] 1017
- 比較項目の指定範囲 560
- 比較項目の指定範囲 [コマンド] 542
- 比較項目の属性 559
- 比較項目の属性のパラメタ 559
- 日付時刻データのチェックオプション  
(DATETIMENOCK オプション) 207
- 日付データのチェックオプション  
(DATENOCK オプション) 202
- 日付データのチェック緩和機能 (SORT EE  
の機能) 41
- 日付データの端数処理 [CSV ファイルの  
データ入力時の端数処理] 811

## ふ

---

- ファイル形式 [exsort パラメタファイルの表  
記規則] 146
- ファイルコピー [コピー機能の使用例] 482
- ファイルの上書きオプション (OVWRT |  
INPOVWRT オプション) 179
- ファイル排他機能 (UNIX の場合) 31
- ファイル名およびディレクトリ名 [exsort パ  
ラメタファイルの表記規則] 151
- ファイル名およびディレクトリ名の指定形式  
[コマンドの表記規則] 139
- 複数指定できるパラメタ [SORT の集約コマ  
ンド (UNIX の場合)] 587

複数指定できるパラメタ〔SORTの選択コマンド (UNIXの場合)〕 563

複数のオプションの指定例〔UNIXの場合〕 696

複数ファイル出力機能 (SORT EE の機能) 42

---

## へ

編集機能〔用語解説〕 1017

編集機能とは (SORT EE の機能) 10

便利な機能 21

---

## ま

マージ+集約〔マージ機能の使用例〕 478

マージ+出力選択+出力レコード編集〔マージ機能の使用例〕 479

マージ〔マージ機能の使用例〕 478

マージオプションのパラメタ 527

マージキーの指定範囲〔SORTのマージ機能 (Windowsの場合)〕 521

マージ機能〔exsort コマンドの形式〕 122

マージ機能〔用語解説〕 1017

マージ機能とは 6

マージ機能の処理の流れ 19

マージコマンドの指定内容〔2種類の文字項目で索引ファイルをマージする例〕 536

マージコマンドの指定内容〔文字項目で相対ファイルをマージする例 (UNIXの場合)〕 534

マージコマンドの使い方 (UNIXの場合) 527

マージコマンドの使い方 (Windowsの場合) 518

マルチバイト文字〔用語解説〕 1017

---

## む

無条件レコード抽出機能 28

---

## め

メッセージ 878

メッセージを確認する〔SORTの障害の対処方法〕 772

メッセージを確認する〔キーまたは項目の不正による障害の対処方法〕 767

メモリ量〔指定できる値の範囲〕 106

メモリ量の最小値および最大値 107

---

## も

文字項目をキーとしてソートし集約する例〔シェル・スクリプトコマンド〕 619

文字順序変更機能 (SORT EE の機能) 44

文字列データの端数処理〔CSVファイルのデータ入力時の端数処理〕 810

---

## り

利用できるファイルの種類とデータ形式 79

リンケージ 643

---

## れ

レコードの読み飛ばしと打ち切り〔コピー機能の使用例〕 484

レコードヘッダ〔用語解説〕 1018

レコード編集〔コピー機能の使用例〕 483

レコード編集処理で基になるレコード

(SORT EE の機能) 830

レコード編集の挿入データの書き方〔exsort パラメタファイルの表記規則〕 148

レコード編集の挿入データの書き方〔コマンドの表記規則〕 142

レコードを選択したあとにソートする例〔拡張 SORT コマンド〕 610

レコードをソートしたあとに集約する例〔拡張 SORT コマンド〕 611

---

## ろ

ログファイル〔SORT EE の障害情報の出力〕 762

論理演算子 562

---

## わ

ワークサイズ設定の優先順位 693

ワークバッファサイズと処理時間 738  
ワイド文字〔用語解説〕 1018