

# SAS<sup>®</sup> 9.3ファイルの 移動とアクセス

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2011. *SAS® 9.3 ファイルの移動とアクセス*. Cary, NC: SAS Institute Inc.

### **SAS® 9.3 ファイルの移動とアクセス**

Copyright © 2011, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

**For a hardcopy book:** No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

**For a Web download or e-book:** Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

**U.S. Government Restricted Rights Notice:** Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

Electronic book 1, 2011 July

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at [support.sas.com/publishing](http://support.sas.com/publishing) or call 1-800-727-3228.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

---

# 目次

本書について	vii
SAS 9.3 ファイルの移動とアクセスの新機能	xi
推奨資料	xiii
1部 はじめに	1
<b>1章・異なる動作環境間での SAS ファイルの移動とアクセス</b>	<b>3</b>
異なる動作環境間での SAS ファイルの移動	3
異なる動作環境での SAS ファイルへのアクセス	3
SAS ファイルの移動とアクセスに使用する方法	4
方法の機能概要	5
各国語サポートを使用した、異なるコンピュータ間での SAS ファイルの移動	6
例で使用するデータセット	7
例で使用する命名規則	7
SAS 製品のユーザー補助機能	8
2部 SAS ファイルの移動とアクセスに使用する方法	9
<b>2章・CEDA(クロス環境データアクセス)</b>	<b>11</b>
CEDA の概要	11
CEDA の利点	12
CEDA の制限事項	12
SAS ファイルの形式の作成、変更	13
異なるコンピュータ間での SAS ファイルの転送	16
SAS ファイル形式の特定	16
非ネイティブファイルの読み込みと書き込み	17
<b>3章・CPORT プロシジャと CIMPORT プロシジャ</b>	<b>19</b>
CPORT プロシジャと CIMPORT プロシジャを使用した SAS ファイルの移動の概要	19
CPORT プロシジャと CIMPORT プロシジャを使用した SAS ファイルの移動の制限事項	20
CPORT プロシジャと CIMPORT プロシジャを使用した SAS ファイルの移動の弱点	20
移送元コンピュータでの移送ファイルの作成	20
移送先コンピュータへの移送ファイルの転送	22
移送先コンピュータでの移送ファイルの復元	22
<b>4章・DATA ステップまたは PROC COPY での XPORT エンジンの使用</b>	<b>27</b>
XPORT エンジンの概要	27
XPORT エンジンの利点	28
XPORT エンジンの制限事項	28
SAS データセットを SAS 6 形式に戻す	28
移送元コンピュータでの移送ファイルの作成	30
ネットワークを経由した移送ファイルの転送	31

移送先コンピュータでの移送ファイルの復元	31
<b>5 章 • DATA ステップまたは PROC COPY での XML エンジンの使用</b>	<b>33</b>
XML エンジンの概要	33
XML エンジンの利点	34
XML エンジンの制限事項	34
移送元コンピュータでの XML ドキュメントの作成	34
ネットワークを経由した XML ドキュメントの転送	35
移送先コンピュータで XML ドキュメントをデータセットに復元する	36
3 部 移送ファイルと非ネイティブファイルの転送	37
<b>6 章 • ファイルの転送</b>	<b>39</b>
ファイル転送の概要	39
移送ファイルの属性	40
非ネイティブファイルや移送ファイルへの FILENAME ステートメントまたは FTP の使用	41
4 部 動作環境固有の情報	45
<b>7 章 • OpenVMS 動作環境</b>	<b>47</b>
OpenVMS システムファイルの属性の表示	47
OpenVMS のファイル属性	48
OpenVMS でファイルの作成に使用された SAS バージョンを特定する	48
OpenVMS へのテープデバイスのマウント	49
OpenVMS のエラーメッセージ	49
<b>8 章 • z/OS 動作環境</b>	<b>53</b>
z/OS ファイル属性の表示	53
z/OS でファイルの作成に使用された SAS バージョンを特定する	53
z/OS ファイルと UNIX System Services Directory	54
ファイル移送に使用する z/OS バッチステートメント	54
z/OS 移送先コンピュータでの転送の問題	54
z/OS 動作環境での移送ファイルの読み込み	55
<b>9 章 • Windows 動作環境</b>	<b>57</b>
Windows のファイル属性	57
Windows でファイルの作成に使用された SAS バージョンを特定する	57
エラーメッセージ: 暗号化したデータが無効です	58
<b>10 章 • UNIX 動作環境</b>	<b>59</b>
UNIX のファイル属性	59
UNIX でファイルの作成に使用された SAS バージョンを特定する	59
例: テープでの移送ファイルの作成	61
例: UNIX 移送元コンピュータでの移送ファイルのコピー(ディスクからテープ)	61
例: 移送先コンピュータでの移送ファイルのコピー(テープからディスク)	61
<b>11 章 • SAS ファイル拡張子とファイルヘッダー</b>	<b>63</b>
ファイル名拡張子: SAS ファイルの作成に使用された SAS エンジンと動作環境を特定する	63

PROC CONTENTS: SAS ファイルの作成に使用された Base SAS エンジン特定する .....	64
ファイルヘッダー: 移送ファイルの作成に使用された方法を調べる .....	65

## 5 部 **トラブルシューティング** 67

<b>12 章・問題の回避と修正</b> .....	<b>69</b>
トラブルシューティング: 移送ファイルの転送と復元 .....	70
移送ファイルに関するエラーメッセージと警告メッセージ .....	73
転送形式と移送ファイルの属性の検証 .....	79
移送ファイルの再ブロック化 .....	80

## 6 部 **サンプルとログ** 83

<b>13 章・SAS ファイルの移動の例</b> .....	<b>85</b>
異なるコンピュータ間での SAS ファイルの移動例の概要 .....	85
例: OpenVMS から UNIX へのファイル移送 .....	86
例: z/OS から Windows へのファイル移送 .....	93
例: z/OS JCL Batch から UNIX へのファイル移送 .....	98
移送ファイルの検証方法 .....	106
<b>用語集</b> .....	<b>109</b>
<b>キーワード</b> .....	<b>115</b>



# 本書について

---

## SAS 言語の構文規則

### SAS 言語の構文規則の概要

SAS では、SAS 言語要素の構文ドキュメントに共通の規則を使用しています。これらの規則により、SAS 構文の構成要素を簡単に識別できます。規則は、次の項目に分類されます。

- 構文の構成要素
- スタイル規則
- 特殊文字
- SAS ライブラリと外部ファイルの参照

### 構文のコンポーネント

言語要素の多くでは、その構文の構成要素はキーワードと引数から構成されます。キーワードのみ必要な言語要素もあります。また、キーワードに等号(=)が続く言語要素もあります。

#### キーワード

プログラムの作成時に使用する SAS 言語要素名です。キーワードはリテラルであり、通常、構文の先頭の単語です。CALL ルーチンでは、最初の 2 つの単語がキーワードです。

次の SAS 構文の例では、構文の最初の単語がキーワードです。

```
CHAR (string, position)
CALL RANBIN (seed, n, p, x);
ALTER (alter-password)
BEST w.
REMOVE <data-set-name>
```

次の例では、CALL ルーチンの最初の 2 つの単語がキーワードです。

```
CALL RANBIN(seed, n, p, x)
```

引数なしで 1 つのキーワードから構成される SAS ステートメント構文もあります。

```
DO;
... SAS code ...
END;
```

2つのキーワード値のいずれか1つの指定が必要なシステムオプションもあります。

#### DUPLEX | NODUPLEX

##### 引数

数値定数、文字定数、変数、式のいずれかです。引数は、キーワードに続くか、キーワードの後ろの等号に続きます。SASでは、引数を使用して、言語要素を処理します。引数が必須の場合もオプションの場合もあります。構文では、オプションの引数にはかぎカッコが付けられます。

次の例では、*string* と *position* がキーワード CHAR に続きます。これらの引数は、CHAR 関数の必須引数です。

#### CHAR (*string*, *position*)

引数ごとに値が指定されます。次の例の SAS コードでは、引数 *string* の値として 'summer'、引数 *position* の値として 4 が指定されています。x=char ('summer', 4);

次の例では、*string* と *substring* は必須引数ですが、*modifiers* と *startpos* はオプションの引数です。

#### FIND(*string*, *substring* <*modifiers*> <*startpos*>)

注: 通常、SASドキュメントのサンプルコードは、小文字の固定幅フォントを使用して表記されます。コードの作成には、大文字も、小文字も、大文字と小文字の両方も使用できます。

## スタイル規則

SAS 構文の説明に使用されるスタイル規則には、大文字太字、大文字、斜体の規則も含まれます。

### 大文字太字

関数名やステートメント名などの SAS キーワードを示します。次の例では、キーワード ERROR の表記には大文字太字が使用されています。

```
ERROR<message>;
```

### 大文字

リテラルの引数を示します。

次の CMPMODEL=システムオプションの例では、BOTH、CATALOG、XML がリテラルです。

```
CMPMODEL = BOTH | CATALOG | XML
```

### 斜体

ユーザー指定の引数または値を示します。斜体表記の項目は、ユーザー指定値であり、次のいずれかを表します。

- 非リテラルの引数。次の LINK ステートメントの例では、引数 *label* はユーザー指定値であるため、斜体で表記されています。

```
LINK label;
```

- 引数に割り当てられる非リテラル値。

次の FORMAT ステートメントの例では、引数 DEFAULT に変数の *default-format* が割り当てられます。

```
FORMAT = variable-1 <, ..., variable-n format ><DEFAULT = default-format>;
```



斜体表記の項目は、選択可能な引数リストの総称でもあります(*attribute-list* など)。複数の斜体表記の項目が使用される場合、項目は *item-1*, ..., *item-n* という形式で表記されます。

## 特殊文字

SAS 言語要素の構文には、次の特殊文字も使用されます。

=

等号は、一部の言語要素(システムオプションなど)のリテラル値を示します。

次の MAPS システムオプションの例では、等号は MAPS の値を設定します。

**MAPS** = *location-of-maps*

<>

かぎかっこはオプションの引数を示します。かぎかっこ付きでない引数は必須引数です。

次の CAT 関数の例では、少なくとも項目が 1 つ必要です。

**CAT** (*item-1* <, ..., *item-n*>)

|

縦棒は、値グループから 1 つの値を選択できることを示します。縦棒で区切られている値は、相互排他です。

次の CMPMODEL=システムオプションの例では、属性を 1 つのみ選択できます。

**CMPMODEL** = BOTH | CATALOG | XML

...

省略記号は、省略記号に続く引数や引数グループの繰り返しを示します。省略記号とその後の引数にかぎかっこが付けられている場合、その引数はオプションです。

次の CAT 関数の例では、省略記号はオプションの項目を複数指定できることを示しています。

**CAT** (*item-1* <, ..., *item-n*>)

'value' or "value"

単一引用符や二重引用符付きの引数は、その値も単一引用符または二重引用符を付ける必要があることを示します。

次の FOOTNOTE ステートメントの例では、引数 *text* には引用符が付けられています。

**FOOTNOTE** <*n*> <*ods-format-options* 'text' | "text">;

;

セミコロンは、ステートメントまたは CALL ルーチンの終わりを示します。

次の例では、それぞれのステートメントはセミコロンで終了しています。data namegame; length color name \$8; color = 'black'; name = 'jack'; game = trim(color) || name; run;

## SAS ライブラリと外部ファイルへの参照

多くの SAS ステートメントなどの言語要素では、SAS ライブラリと外部ファイルを参照します。論理名(ライブラリ参照名またはファイル参照名)から参照を作成するのか、引用符付きの物理ファイル名を使用するかを選択できます。論理名を使用する場合、通

## x 本書について

常、関連付けに SAS ステートメント(LIBNAME または FILENAME)を使用するのか、動作環境のコントロール言語を使用するのかを選択します。複数の方法を使用して、SAS ライブラリと外部ファイルを参照できます。動作環境によっては使用できない方法があります。

SAS ドキュメントでは、外部ファイルを使用する例には斜体のフレーズ *file-specification* を使用します。また、SAS ライブラリを使用する例には斜体フレーズ *SAS-library* を使用します。*SAS-library* は引用符付きであることに注意してください。

```
infile file-specification obs = 100;  
libname libref 'SAS-library';
```

# SAS 9.3 ファイルの移動とアクセス の新機能

---

## 概要

今回の更新で、CEDA と、CPORT プロシジャおよび CIMPORT プロシジャに関する情報がドキュメントに追加されました。

---

## ドキュメントの拡張

クロス環境データアクセス(CEDA)機能に関する拡張は次のとおりです。

- z/OS 上の UNIX ファイルシステムライブラリですべての CEDA データ表現をサポートします。ただし、SAS にバインドされたライブラリが z/OS でサポートするのは、MVS\_32 の CEDA データ表現を持つ SAS データセットのみです。確認されている制限には、“CEDA の制限事項” (12 ページ)があります。

CPORT プロシジャおよび CIMPORT プロシジャに関する拡張は次のとおりです。

- 埋め込み空白を含む SAS 名リテラルを CPORT プロシジャおよび CIMPORT プロシジャで使用できるようになりました。詳細については、“SAS 名前リテラル” (SAS 言語リファレンス: 解説編 3 章)を参照してください。
- VALIDVARNAME=ANY または VALIDMEMNAME=EXTEND を指定すると、最大長 32 バイトのデータセット名またはメンバ名を CIMPORT プロシジャおよび CPORT プロシジャで使用できるようになりました。名前とメンバ名には、大文字と小文字を混在させて使用できます。詳細については、“VALIDMEMNAME=システムオプション” (SAS システムオプション: リファレンス)および “VALIDVARNAME=システムオプション” (SAS システムオプション: リファレンス)を参照してください。
- CPORT の SELECT ステートメントおよび EXCLUDE ステートメントで、ACCESS エンジンの大文字と小文字を区別する名前をサポートするようになりました。
- CIMPORT の SELECT ステートメントおよび EXCLUDE ステートメントで、CPORT ファイルの大文字と小文字を区別する名前をサポートするようになりました。



# 推奨資料

---

このタイトルに関連した推奨される参考資料のリストを次に示します。

- *SAS/CONNECT User's Guide*
- *SAS/SHARE User's Guide*
- *SAS ステートメント: リファレンス*
- *SAS システムオプション: リファレンス*
- *Base SAS プロシジャガイド*
- *SAS 言語リファレンス: 解説編*
- *Communications Access Methods for SAS/CONNECT and SAS/SHARE*
- *SAS XML LIBNAME Engine: ユーザーガイド*
- ご使用の動作環境の SAS ドキュメント (UNIX 版 SAS、または Windows 版 SAS)

SAS の刊行物の総一覧については、[support.sas.com/bookstore](http://support.sas.com/bookstore) にてご確認ください。  
必要な書籍についてのご質問は、下記までお寄せください。

SAS Publishing Sales  
SAS Campus Drive  
Cary, NC 27513-2414  
電話: 1-800-727-3228  
ファクシミリ: 1-919-677-8166  
電子メール: [sasbook@sas.com](mailto:sasbook@sas.com)  
Web アドレス: [support.sas.com/bookstore](http://support.sas.com/bookstore)



## 1 部

---

# はじめに

1 章	異なる動作環境間での SAS ファイルの移動とアクセス.....	3
-----	----------------------------------	---





## 1 章

## 異なる動作環境間での SAS ファイルの移動とアクセス

---

異なる動作環境間での SAS ファイルの移動 .....	3
異なる動作環境での SAS ファイルへのアクセス .....	3
SAS ファイルの移動とアクセスに使用する方法 .....	4
方法の機能概要 .....	5
各国語サポートを使用した、異なるコンピュータ間での SAS ファイルの移動 .....	6
例で使用するデータセット .....	7
例で使用する命名規則 .....	7
SAS 製品のユーザー補助機能 .....	8

---

## 異なる動作環境間での SAS ファイルの移動

動作環境間での SAS ファイルの移動は一般的なタスクです。動作環境間で SAS ファイルを移動する理由には、次のようなものがあります。

- 異なるコンピュータの新しい動作環境に SAS ファイルを移動(たとえば、HP-UX ファイルを RedHat Linux 動作環境に移動)する。
- ファイルとその処理を処理速度の高い動作環境に移動してから、要求側の動作環境にファイルに戻す。
- 継続的にデータを処理するため、物理的に別個の動作環境で使用可能な SAS ファイルの静的なコピーを作成する。NFS でマウントされたファイルシステムでは受信側の動作環境で SAS ファイルが使用できないため、受信側の動作環境でできるようにファイルを複製します。

これらすべてのシナリオでの移動操作は動作環境のアーキテクチャ間および SAS リリース間の差異を認識するため、元のファイルを受信側の動作環境で使用できます。

---

## 異なる動作環境での SAS ファイルへのアクセス

場合によっては、ファイルのコピーを独自に所有して維持するよりも単にアクセスする方が望ましいことがあります。または、他の場所で作成され、ローカルにマウントされたテープからデータを読み込んだり、ネットワークにリモートでマウントされたデータの読み込み、書き込みまたは更新を実行したりする必要がある場合があります。

注: アクセスという用語を製品 SAS/ACCESS と混同しないようにしてください。異なる動作環境での SAS ファイルの移動とアクセスにおいて、アクセスとは SAS ファイルを開いて処理することを意味します。SAS/ACCESS では、ユーザーがサードパーティ DBMS ファイルを使用できます。

リモート SAS ファイルにアクセスするには、次の方法を使用できます。

- CEDA(クロス環境データアクセス)を使用して、SAS 8 以降の SAS ファイルにアクセスできます。
- クライアントで SAS/SHARE を使用して、SAS/SHARE サーバーが実行されている動作環境に存在するリモート SAS ファイルにアクセスできます。SAS/SHARE は、複数のユーザー間のリモートデータへの透過的同時アクセスを容易にします。異なるリリース間の SAS データへのアクセスには制限が適用されます。

また、SAS/SHARE では、SAS/ACCESS でサポートされているエンジンを使用して特定のサードパーティ DBMS ファイルにアクセスできます。

- SAS/SHARE または CEDA を使用せずに、ネットワークサービスを使用してリモートファイル(SAS ファイルとサードパーティ DBMS ファイルの両方)にアクセスできます。通常、クライアントとサーバーは互換性のあるアーキテクチャを共有する必要があり、同じリリースの SAS ソフトウェアを実行している必要があります。動作環境、ネットワークソフトウェアおよびセキュリティソフトウェアによって、特定のリモートファイルに対するユーザーの権限が制御される可能性があります。詳細については、ご使用の動作環境の SAS ドキュメント(UNIX 版 SAS、または Windows 版 SAS)、および使用するネットワークソフトウェアとセキュリティソフトウェア用のサードパーティのドキュメントを参照してください。

## SAS ファイルの移動とアクセスに使用する方法

SAS ファイルの移動またはアクセスには、次の方法を使用できます。

### CEDA(クロス環境データアクセス)

SAS のこの機能は、あらゆるディレクトリベース動作環境(Solaris、Windows、HP-UX、OpenVMS など)で作成された SAS ファイルを、別のディレクトリベースの環境で実行された SAS セッションで処理できるようにします。詳細については、[“CEDA\(クロス環境データアクセス\)” \(11 ページ\)](#)を参照してください。

### CPORT プロシジャと CIMPORT プロシジャ

移送元環境では、PROC CPORT を使用して、データセットまたはカタログを移送形式で書き込みます。移送先環境では、PROC CIMPORT を使用して、移送ファイルを移送先環境のネイティブ形式に変換できます。詳細については、[“CPORT プロシジャと CIMPORT プロシジャ” \(19 ページ\)](#)を参照してください。

### DATA ステップまたは PROC COPY での XPORT エンジンの使用

移送元環境では、LIBNAME ステートメントを XPORT エンジン(DATA ステップまたは PROC COPY)で使用して、SAS データセットから移送ファイルを作成できます。移送先環境では、同じ方法を使用して、移送ファイルを移送先環境のネイティブ形式に変換できます。[“DATA ステップまたは PROC COPY での XPORT エンジンの使用” \(27 ページ\)](#)を参照してください。

注: XPORT エンジンでは、SAS 8 以降の機能(ログファイルや変数名など)はサポートされていません。

### DATA ステップまたは PROC COPY での XML エンジンの使用

移送元環境では、LIBNAME ステートメントを XML エンジン(DATA ステップまたは PROC COPY)で使用して、SAS データセットから XML ドキュメントを作成できます。移送先環境では、同じ方法を使用して、XML ドキュメントを移送先環境のネイ

タイプ形式に変換できます。詳細については、“[DATA ステップまたは PROC COPY での XML エンジンの使用](#)” (33 ページ)を参照してください。

SAS/CONNECT のデータ転送サービス(DTS)

この機能は、データセットとカタログを移送元環境から移送先環境に転送できます。DTS は、必要に応じて動作環境の表現間と SAS バージョン間でデータを変換します。転送を実行するには、SIGNON ステートメントを使用して 2 つの SAS セッションに接続し、PROC UPLOAD または PROC DOWNLOAD を使用してデータを移動します。

SAS/SHARE および SAS/CONNECT の REMOTE エンジンとリモートライブラリサービス

これらの機能では、REMOTE エンジンと LIBNAME ステートメントを使用してリモートデータに透過的にアクセスできます。

## 方法の機能概要

SAS ファイルの移動またはアクセスに使用できる各方法の機能概要を次に示します。

表 1.1 SAS ファイルの移動またはアクセスに使用できる方法の機能概要

機能	使用できる方法					
					SAS/CONNECT DTS	SAS/CONNECT RLS および SAS/SHARE RLS
サポートされている SAS メンバの種類	データファイル、PROC SQL ビュー*、SAS/ACCESS ビュー(Oracle および Sybase)、MDDB*	ライブラリ、データセット、カタログ、カタログエントリ	ライブラリ、データセット	データファイル	ライブラリ、データセット、カタログ、カタログエントリ、PROC SQL ビュー、MDDB、外部サードパーティデータベース***	ライブラリ、データセット、Catalog**、カタログエントリ**、PROC SQL ビュー、MDDB、DATA ステップビュー、SAS/ACCESS ビュー、外部サードパーティデータベース***
動的変換またはファイル形式の作成	動的	移送†	移送†	XML	動的	動的
サポートされている SAS バージョン	SAS 8 以降	SAS 6 以降	SAS 6 以降†	SAS 8.2 以降	SAS 6 以降	SAS 6 以降

機能	使用できる方法					
					SAS/CONNECT DTS	SAS/CONNECT RLS および SAS/SHARE RLS
新しい SAS リリースから旧 SAS リリース への変換	いいえ	いいえ	はい	いいえ	はい	はい
ディレクトリベースのファイル構造を使用する動作環境のみに制限	はい	いいえ	いいえ	いいえ	いいえ	いいえ
必要な SAS 製品ライセンス	Base SAS	Base SAS	Base SAS	Base SAS	SAS/CONNECT	SAS/CONNECT または SAS/SHARE

\* データセット(ファイル)への読み込み、書き込みおよび更新アクセスが可能です。PROC SQL ビューと MDDDB は読み込み専用です。

\*\* SAS 9 では、互換性のない動作環境でのカタログエントリまたはカタログへのクロス動作環境アクセスはサポートされていません。アーキテクチャグループの詳細については、*SAS/CONNECT User's Guide* または *SAS/SHARE User's Guide* を参照してください。

\*\*\* SAS/CONNECT は、外部テキストファイルとバイナリファイルをサポートしています。SAS/CONNECT と SAS/SHARE は、リモート SQL パススルー機能を使用したサードパーティの外部データベースをサポートしていますが、これらのデータベースへのアクセスには SAS/ACCESS ライセンスが必要です。

† XPORT エンジンでは、SAS 8 で導入された機能(ログファイルや変数名など)はサポートされていません。SAS 8 以降の SAS ファイルを旧リリースに戻すために XPORT エンジンを使用した場合、SAS 8 以降に固有の機能は SAS ファイルから削除されます。また、XPORT エンジンで生成された移送形式と PROC CPORT で生成された移送形式は互換性がありません。

リレーショナルデータベースの詳細については、*SAS/ACCESS for Relational Databases: Reference* を参照してください。非リレーショナルデータベースの詳細については、*SAS/ACCESS(R) Interface to ADABAS: Reference*、*SAS/ACCESS(R) 9.3 Interface to IMS: Reference*、*SAS/ACCESS DATA Step Interface to CA-IDMS: Reference* または *SAS/ACCESS Interface to SYSTEM 2000: Reference* を参照してください。

## 各国語サポートを使用した、異なるコンピュータ間での SAS ファイルの移動

2つのコンピュータと動作環境間で移送ファイルを正常に移動するには、移送元と移送先の SAS セッションのエンコーディングに互換性がなければなりません。たとえば、スペイン語(メキシコ)ロケールに関連付けられた Wlatin1 エンコーディングを使用する移送元 SAS セッションは、イタリア語(イタリア)ロケールに関連付けられた Wlatin1 エンコーディングを使用する移送先 SAS セッションと互換性があります。両方のセッションは Wlatin1 エンコーディングを使用しています。

ただし、各国語サポート(NLS)がなく、互換性のない移送元と移送先の SAS セッション間では、移送ファイルを移動できません。たとえば、チェコ語(チェコスロバキア)ロケールに関連付けられた Wlatin2 エンコーディングを使用する移送元 SAS セッションは、ドイツ語(ドイツ)ロケールに関連付けられた open\_ed-1141 z/OS エンコーディングを使用する移送先 SAS セッションとは互換性がありません。Wlatin2 エンコーディングと open\_ed-1141 エンコーディングは互換性がありません。

適切な方法(たとえば、XPORT エンジンまたは PROC CPORT および PROC CIMPORT)を使用してデータを移動する前に、移送ファイルを作成した移送元 SAS セッションのロケールに移送先 SAS セッションのロケールを再設定する必要があります。ロケールまたはエンコーディングの指定方法は、移送元コンピュータと移送先コンピュータで実行されている SAS のバージョンによって異なります。

異なるロケールまたはエンコーディング間で SAS ファイルを移動する場合は、LOCALE=オプションと ENCODING=オプションを使用します。この情報については、*SAS 各国語サポート(NLS): リファレンスガイド*を参照してください。PROC CIMPORT を使用して、ロケールとエンコーディングが異なる移送元コンピュータと移送先コンピュータ間で移送ファイルを移動する方法の詳細については、*Base SAS プロシジャガイド*を参照してください。

---

## 例で使用するデータセット

テストを行う場合、ライブラリ内にいくつかの単純なデータセットを作成できます。データセット GRADES を作成するサンプル SAS プログラムを次に示します。

```
data grades;
input student $ test1 test2 final;
datalines;
Fred 66 80 70
Wilma 97 91 98
;
proc print data=grades;
run;
```

次のよう出力されます。

```
The SAS System 10:59 Friday, April 25, 2008

Obs student test1 test2 final
1 Fred 66 80 70
2 Wilma 97 91 98
```

---

## 例で使用する命名規則

このドキュメントの例では、次の命名規則が使用されます。

### WORK

データセット GRADES を含むライブラリを示す、デフォルトのライブラリ参照名です。

### XPORTOUT

XPORT エンジンで移送ファイルが作成される場所を示す、ライブラリ参照名です。

### XPORTIN

移送ファイルを転送した移送先動作環境の場所を示す、ライブラリ参照名です。

### XMLOUT

XML エンジンで XML ファイルが作成される場所を示す、ライブラリ参照名です。

### XMLIN

XML ファイルを転送した移送先動作環境の場所を示す、ライブラリ参照名です。

CPORTOUT

PROC CPORT で移送ファイルが作成される場所を示す、ファイル参照名です。

IMPORTIN

移送ファイルを転送した移送先動作環境の場所を示す、ファイル参照名です。

SOURCE

移送形式または XML 形式に変換される移送元ファイルの場所を示す、ライブラリ参照名です。

LIST

カテゴリエントリの種類です。

GRADES

データセット名です。

TARGET

復元された SAS ファイルが作成される場所を示す、ライブラリ参照名です。

TESTCAT

カタログ名です。

TESTNPGM

カタログエントリ名です。

---

## SAS 製品のユーザー補助機能

このドキュメントで言及された製品のユーザー補助機能の詳細については、その製品のドキュメントを参照してください。SAS 製品のユーザー補助機能についてのご質問やご意見は、[accessibility@sas.com](mailto:accessibility@sas.com) まで電子メールでお寄せください。

## 2 部

---

# SAS ファイルの移動とアクセスに使用する方法

2 章	
CEDA(クロス環境データアクセス) .....	11
3 章	
CPORT プロシジャと CIMPORT プロシジャ .....	19
4 章	
DATA ステップまたは PROC COPY での XPORT エンジンの使用 .....	27
5 章	
DATA ステップまたは PROC COPY での XML エンジンの使用 .....	33





## 2 章

## CEDA(クロス環境データアクセス)

CEDA の概要	11
CEDA の利点	12
CEDA の制限事項	12
SAS ファイルの形式の作成、変更	13
SAS ファイルを外部形式で作成する	13
例: DATA ステップでの OUTREP=を使用した外部形式の作成	13
SAS ファイルを外部形式に変更する	14
例: LIBNAME ステートメントの OUTREP=オプションと PROC COPY の NOCLONE オプションを使用してファイ ルの形式を変更する	14
例: 変更したファイルの形式を移送元コンピュータの SAS ログで確認する	15
異なるコンピュータ間での SAS ファイルの転送	16
SAS ファイル形式の特定	16
例: CEDA が使用されていることの報告	16
例: PROC CONTENTS を使用したファイルの形式の特定	16
非ネイティブファイルアクセスの制限	17
非ネイティブファイルの読み込みと書き込み	17

## CEDA の概要

CEDA は、ネットワーク経由のファイルアクセスに使用する単純な方法です。CEDA を使用すると、アクセス対象 SAS ファイルのファイル形式に関係なく、SAS 8 以降を実行するあらゆるディレクトリベース動作環境から、ネットワークにマウントされた SAS ファイルを読み込むことができます。たとえば、CEDA を使用すると、ネットワークにマウントされた UNIX ファイル形式の SAS ファイルを読み込むことができます。

注: SAS 8.2 より前のバージョンでは、CEDA は SAS/CONNECT とパッケージ化されていたましたが、別途ライセンスが必要でした。現在、CEDA は Base SAS の一部として含まれています。

CEDA は透過的に実行されます。ファイルの形式が不明でも、サポートされている SAS ファイルにアクセスできます。CEDA では、アクセスする側のコンピュータの形式を検出し、"ネイティブ"形式から"外部"(アクセスする側の)コンピュータの表現に自動的に変換されます。

CEDA は、一元管理された SAS ファイルからのデータ読み込み、データ処理、レポート作成が複数のアプリケーションで行われるような、異種環境がネットワーク接続され

ている企業で最も役に立ちます。たとえば、SAS データセットを UNIX コンピュータに配置し、UNIX コンピュータには外部形式となるデータ表現を使用するコンピュータからアクセスすることができます。たとえば、UNIX コンピュータと Windows コンピュータではデータの表現が異なります。CEDA を使用しなければ、SAS ファイルにアクセスしたときに動的に変換することはできません。代わりに、アクセスする側のコンピュータ用に移送ファイルまたはファイルを外部形式で生成する必要があります。

---

## CEDA の利点

CEDA には次のような利点があります。

- CEDA は透過的に実行されます。ユーザーは、ファイルのネイティブ形式が不明でも、データセットを読み込むことができます。
- ネイティブ形式から移送形式へ、移送形式からネイティブ形式へと、複数の変換を行う代わりに、CEDA ではネイティブ形式と外部形式間の変換を一度行うだけで済みます。
- 中間の移送ファイルは作成されません。
- CEDA を使用すると、ファイルにアクセスするために明示的なステップを実行する必要がなくなります。
- CEDA には、SAS/SHARE で必要とされる専用サーバーや、SAS/CONNECT で必要とされる明示的なサインオンは必要ありません。

---

## CEDA の制限事項

CEDA は、ネットワークファイルにアクセスするために、すべての状況で推奨される方法というわけではありません。CEDA には、次のような制限があります。

- CEDA 機能は、SAS 9 または SAS 8 のデータセット、PROC SQL ビュー、Oracle および Sybase 用の SAS/ACCESS ビュー、および MDDDB 用に実装されています。SAS 9 または SAS 8 の格納プログラムまたはカタログ、および SAS 6 以前のファイルはサポートされません。CEDA による SAS ファイルへのアクセスの種類は、使用されるエンジンと要求されたファイルアクセスの種類(読み込み、書き込み、更新)によって異なります。ファイルアクセスの制限の詳細については、“CEDA を使用した SAS ファイルの処理”(SAS 言語リファレンス: 解説編 32 章)を参照してください。
- CEDA では SAS ファイルの更新処理はサポートされません。
- CEDA ではインデックスを使用したサブセットはサポートされません。
- CEDA では監査証跡の読み込みができますが、更新はできません。
- 一貫性制約の処理はサポートされません。
- z/OS では、SAS 連結ライブラリは CEDA データ表現 MVS\_32 を使用する SAS データセットのみをサポートします。ただし、z/OS の UNIX ファイルシステムライブラリでは、すべての CEDA データ表現がサポートされます。
- CEDA でファイルが変換されるたびに、ネットワークリソースが消費されます。
- エンコーディングに互換性がない場合、トランスコーディングで文字データが失われる可能性があります。エンコーディングおよびトランスコーディングの詳細については、*SAS 各国語サポート(NLS): リファレンスガイド*を参照してください。

- 破損したデータセットの修復に使用する方法である更新処理が CEDA でサポートされていないため、外部形式のファイルが破損している場合、ファイルの修復はできません。ファイルを修復するには、移送元環境にファイルを移動する必要があります。破損したデータセットの修復の詳細については、*Base SAS プロシジャガイド* の DATASETS プロシジャの REPAIR ステートメントを参照してください。
- 動作環境に応じて、数値変数の最小長は 2 バイトまたは 3 バイトになります。3 バイトの最小長をサポートする動作環境(Windows や UNIX)では、CEDA は(たとえば、z/OS で)2 バイトの長さで作成された数値変数を処理できません。この制限が発生した場合は、CEDA の代わりに XPORT エンジンか、CPORT および CIMPORT プロシジャを使用してください。
- 動作環境間でデータを移動するときに、数値変数の精度が劣化する可能性があります。数値変数の長さが短く定義されている場合は、変数の長さを増やしてみてください。フルサイズの数値変数では、CEDA を使用するとき精度の劣化が発生する可能性が、短い数値変数よりも低くなります。詳細については、*SAS 言語リファレンス: 解説編*の数値精度に関するトピックを参照してください。

処理速度に関する問題がある場合は、ファイルアクセスのパターンを分析し、データセットが正しいコンピュータ上にあるかどうかを確認してください。たとえば、SAS データセットが UNIX データ形式で表現されていて、ほとんどの読み込み操作が Windows コンピュータから行われている場合、Windows コンピュータにデータセットを移動し、データセットの UNIX ファイル形式を Windows 形式に変更することを検討します。ネットワークにマウントされた Windows 形式のファイルに Windows からアクセスする場合、CEDA は必要ありません。ファイルの形式を変更すると、処理速度が向上し、書き込みおよび更新アクセスが許可されます。ただし、アクセス対象 SAS ファイルの Windows ネイティブ形式と、アクセスする側の Windows 以外のコンピュータ(UNIX、z/OS、OpenVMS など)の間の変換には CEDA が使用されます。

CEDA でサポートされるデータの種類の詳細については、“クロス環境データアクセス(CEDA)を用いたデータ処理”(SAS 言語リファレンス: 解説編 32 章)を参照してください。

## SAS ファイルの形式の作成、変更

### SAS ファイルを外部形式で作成する

デフォルトでは、DATA ステップまたは LIBNAME ステートメントを使用して新規に作成する SAS ファイルまたは SAS ライブラリは、移送元コンピュータのネイティブ形式で作成されます。たとえば、Windows では、新しいデータセットは Windows ネイティブ形式で作成されます。ただし、デフォルト設定より優先される外部形式でデータセットを作成することができます。たとえば Windows で、新しいデータセットを UNIX データ表現など外部形式で作成できます。

### 例: DATA ステップでの OUTREP=を使用した外部形式の作成

サポートされているメンバの種類的外部形式で SAS ファイルを作成するには、DATA ステップで OUTREP=オプションを使用します。OUTREP=オプションは指定したデータセットの外部形式に適用されます。

この例では、ネイティブ形式が Windows 表現であると想定します。これは、外部形式の HP\_UX\_64 で上書きされます。

```
data chem.grades (outrep=HP_UX_64);
input student $ test1 test2 final;
```

```
datalines;
Fred 66 80 70
Wilma 97 91 98
run;
```

この例では、データセット GARDES が外部形式の HP\_UX\_64 で作成されます。

OUTREP=オプションでサポートされる値については、“LIBNAME ステートメント” (SAS ステートメント: リファレンス) または *SAS データセットオプション: リファレンスの OUTREP=オプション* を参照してください。

## SAS ファイルを外部形式に変更する

移送元コンピュータまたは移送先コンピュータで、LIBNAME ステートメントおよび OUTREP=オプションを COPY プロシジャおよび NOCLONE オプションと一緒に使用して、ファイルのネイティブ形式を外部形式に変更することもできます。

移送元コンピュータで、ファイルのネイティブ形式を外部形式に変更し、このファイルを移送先コンピュータに転送できます。

または、移送元コンピュータで、ネイティブ形式のファイルを移送先コンピュータに転送できます。次に、移送先コンピュータで、ファイルのネイティブ形式を、移送先コンピュータで使用される外部形式に変更できます。

### 例: LIBNAME ステートメントの OUTREP=オプションと PROC COPY の NOCLONE オプションを使用してファイルの形式を変更する

SAS ファイルのネイティブ形式をサポートされているメンバの種類的外部形式に変更するプロセスを次に示します。

注: MDDDB ファイルのファイル形式は変更できません。CEDA では、MDDDB ファイルの読み込み専用アクセスがサポートされます。

1. LIBNAME ステートメントと OUTREP=オプションを使用して、外部形式で作成されるファイルを指します。
2. LIBNAME ステートメントを使用して、ネイティブ形式のファイルを指します。
3. COPY プロシジャを使用して、ネイティブ形式のファイルを外部形式のファイルにコピーします。さらに、NOCLONE オプションも使用します。このオプションでは、ネイティブ形式のファイルの代わりに、外部形式のファイルのデータ表現が選択されます。

次に、例を示します。

```
libname target 'path-to-target-library' outrep=HP_UX_64;
libname source 'path-to-source-library';
proc copy in=source out=target noclone memtype=data;
run;
```

この例では、Windows ネイティブ形式のデータセットのライブラリが、HP\_UX\_64 外部形式のデータセットのライブラリにコピーされます。

OUTREP=オプションのサポートされる値については、“LIBNAME ステートメント” (SAS ステートメント: リファレンス) または *SAS データセットオプション: リファレンスの OUTREP=オプション* を参照してください。NOCLONE オプションの詳細については、*Base SAS プロシジャガイド* の COPY プロシジャを参照してください。

**例: 変更したファイルの形式を移送元コンピュータの SAS ログで確認する**

SAS ファイルが HP\_UX\_64 外部形式に変更されたことを確認するには、Windows ネイティブ動作環境を実行する移送元コンピュータで SAS ログを表示します。

**アウトプット 2.1 SAS ログで指定されたデータ表現**

```

The SAS System 10:15 Friday, December 19, 2003 1

The CONTENTS Procedure

Data Set Name WORK.GRADES Observations
1
Member Type DATA Variables
4
Engine V9 Indexes
0
Created 11:03 Friday, December 19, 2003 Observation Length
32
Last Modified 11:03 Friday, December 19, 2003 Deleted Observations
0
Protection Compressed
NO
Data Set Type Sorted
NO
Label
Data Representation HP_UX_64, RS_6000_AIX_64, SOLARIS_64, HP_IA_64
1
Encoding latin1 Western (ISO)

Engine/Host Dependent Information

Data Set Page Size 4096
Number of Data Set Pages 1
First Data Page 1
Max Obs per Page 126
Obs in First Data Page 1
Number of Data Set Repairs 0
File Name C:\TEMP\SAS Temporary
Files\TD228\grades.sas7bdat
Release Created 9.0000M0
Host Created WIN_NT 2

Alphabetic List of Variables and Attributes

# Variable Type Len
4 final Num 8
1 student Char 8
2 test1 Num 8
3 test2 Num 8

```

- 1 データセットは HP\_UX\_64 形式で表現されています。これは、Windows ネイティブ環境には外部形式になります。
- 2 ネイティブ形式は WIN\_NT です。

---

## 異なるコンピュータ間での SAS ファイルの転送

移送先コンピュータから SAS ファイルにアクセスできるようにするには、次のいずれかの方法を使用します。

- NFS(Network File Services)を使用して、動作環境でアクセスできるようにネットワーク上のファイルをマウントします。NFS および動作環境向けドキュメントを参照してください。
- FTP(File Transfer Protocol)サービスで特定の移送先動作環境にバイナリ形式でファイルをコピーする。FTP の例については、“[例: FTP を使用した外部ファイルと移送ファイルの転送](#)” (42 ページ) を参照してください。

**注意:**

外部ファイルは BINARY 形式で転送する必要があります。

---

## SAS ファイル形式の特定

### 例: CEDA が使用されていることの報告

SAS 9 以降では、CEDA が使用されると、ログにメッセージが書き込まれます。次に、例を示します。

```
NOTE: Data file HEALTH.GRADES.DATA is in a format that is native to
another host, or the file encoding does not match the session encoding.
Cross Environment Data Access will be used, which might require
additional CPU resources and might reduce performance.
```

**注:** 外部ファイルを読み込むたびに、追加リソースが消費されます。

### 例: PROC CONTENTS を使用したファイルの形式の特定

CONTENTS プロシジャ(または PROC DATASETS の CONTENTS ステートメント)を使用して、指定した SAS ファイルの形式を調べることができます。

このコードの例は次のとおりです。

```
proc contents data=grades;
run;
```

この出力の一部を次に示します。

```
Data Representation HP_UX_64, RS_6000_AIX_64, SOLARIS_64, HP_IA64
```

前述の例では、ファイルが UNIX 形式で表現されていることが出力に示されています。

移送先コンピュータでファイル形式と同じ形式が使用されている場合は、ファイルの読み込み、書き込み、更新ができます。

**注:** 追加リソースは消費されません。

移送先コンピュータでファイル形式と異なる形式(この例では UNIX)が使用されている場合は、読み書きはできますが、ファイルの更新はできません。

注: 外部ファイルを読み込むたびに、追加リソースが消費されます。

### 非ネイティブファイルアクセスの制限

外部ファイルは更新できません。ただし、次の操作を実行できます。

- ファイルを読み込みます。

注: 外部ファイルを読み込むたびに、追加リソースが消費されます。

- ファイルの外部形式(たとえば、UNIX)をネイティブ(アクセスする側の)コンピュータ(たとえば、Windows)の形式に変更します。外部形式からネイティブ形式に変更すると、ファイルへの完全アクセス(読み込み、書き込み、更新)が許可され、中間変換を行う必要はありません。

注: ファイルの形式を変更した後は、ファイルにアクセスするときに追加リソースは消費されません。

アクセスする側のコンピュータの外部形式を使用する SAS ファイルを更新しようとすると、エラーメッセージが表示されます。

注: CEDA に許可されるアクセスの種類は、使用されるエンジンと要求されたファイルアクセスの種類(読み込み、書き込み、更新)によって異なります。ファイルアクセスの制限の詳細については、“CEDA を使用した SAS ファイルの処理”(SAS 言語リファレンス: 解説編 32 章)を参照してください。

通常のエラーメッセージは次のとおりです。

```
ERROR: File TEST.CMVS cannot be updated because its
encoding does not match the session encoding or the file is in a
format native to another host, such as SOLARIS, HP_UX, RS_6000_AIX,
MIPS_ABI.
```

---

## 非ネイティブファイルの読み込みと書き込み

外部ファイルがネットワーク経由で移送先コンピュータに転送されると、移送先コンピュータで SAS 8 以降が実行されている場合は、移送先コンピュータで SAS ファイルの読み書きができます。移送先コンピュータでは、ファイルを更新する場合はアクセスできませんが、読み込みまたは書き込みを行う場合は外部ファイルに透過的にアクセスできます。

ファイルの読み書きができますが、更新はできません。

注: 外部ファイルを読み込む、または書き込むたびに、追加リソースが消費されます。





## 3 章

## CPORT プロシジャと CIMPORT プロシジャ

---

CPORT プロシジャと CIMPORT プロシジャを使用した SAS ファイルの移動の概要 .....	19
CPORT プロシジャと CIMPORT プロシジャを使用した SAS ファイルの移動の制限事項 .....	20
CPORT プロシジャと CIMPORT プロシジャを使用した SAS ファイルの移動の弱点 .....	20
移送元コンピュータでの移送ファイルの作成 .....	20
PROC CPORT を使用して移送ファイルを作成する .....	20
移送先コンピュータへの移送ファイルの転送 .....	22
移送先コンピュータでの移送ファイルの復元 .....	22
移送ファイルの内容の確認 .....	22
PROC CIMPORT を使用して移送ファイルを復元する .....	23

---

## CPORT プロシジャと CIMPORT プロシジャを使用した SAS ファイルの移動の概要

PROC CPORT は、移送形式のファイルを作成します。このファイルは、環境に依存しない文字エンコーディングと数値表現の標準を使用します。PROC CPORT によって作成された移送ファイルは異なる動作環境間で転送可能で、PROC CIMPORT を使用して読み込むことができます。

移送元コンピュータで移送ファイルを作成し、移送先コンピュータでそのファイルを読み込むプロセスを次に示します。

1. PROC CPORT を使用して、移送元コンピュータで移送ファイルが作成されます。
2. ファイルが移送元コンピュータから移送先コンピュータへ転送されます。
3. PROC CIMPORT を使用して、移送先コンピュータで移送ファイルが読み込まれます。

注: PROC CPORT を使用して作成された移送ファイルは、XPORT エンジンを使用して作成された移送ファイルとは互換性がありません。

---

## CPORT プロシジャと CIMPORT プロシジャを使用した SAS ファイルの移動の制限事項

CPORT プロシジャと CIMPORT プロシジャは、DATA と CATALOG の両方の種類のメンバの移動に適しています。PROC COPY は、種類が DATA のみのメンバの移動に使用されます。

---

## CPORT プロシジャと CIMPORT プロシジャを使用した SAS ファイルの移動の弱点

PROC CPORT と PROC CIMPORT を使用した SAS ファイルの移動には、次の弱点があります。

- PROC CPORT と PROC CIMPORT では、種類がビューまたは MDDDB の移送はサポートされていません。
- PROC CPORT と PROC CIMPORT では、新しいバージョンから古いバージョン(たとえば、SAS 9 から SAS 6)へのファイル移送(旧バージョンへの変換)は許可されていません。PROC CPORT と PROC CIMPORT は、古いバージョンから新しいバージョン(たとえば、SAS 6 から SAS 9)にファイルを移動します。また、これらのプロシジャは同じバージョン間(たとえば、SAS 9 動作環境から別の SAS 9 動作環境)でもファイルを移動します。

ただし、SAS 6 のリリース間(たとえば、SAS 6.12 から SAS 6.08)ではファイルを移動できます。これらのプロシジャの構文の詳細については、“CPORT プロシジャ”(Base SAS プロシジャガイド)および“CIMPORT プロシジャ”(Base SAS プロシジャガイド)を参照してください。

- PROC CPORT と PROC CIMPORT では、極端に小さい数値や極端に大きい数値の精度が失われる可能性があります。詳細については、“Loss of Numeric Precision and Magnitude”(SAS/CONNECT User's Guide 1 章)を参照してください。

---

## 移送元コンピュータでの移送ファイルの作成

1 つ以上の SAS データセットまたは 1 つ以上の SAS カタログを含む移送ファイルを作成するには、PROC CPORT を使用します。

### PROC CPORT を使用して移送ファイルを作成する

#### 例: PROC CPORT を使用したデータセットの移送ファイルの作成

この例では、CPORT プロシジャを使用して 1 つのデータセットの移送ファイルを作成します。

```
libname source 'SAS-data-library';
filename cportout 'transport-file';
proc cport data=source.grades file=cportout;
run;
```

前述の例では、ライブラリ参照名 SOURCE は、移送元コンピュータにあるデータセットの元の場所を指しています。ファイル参照名 CPORTOUT は、移送ファイルが作成される新しい場所を指しています。PROC CPORT ステートメントは、コピー元として、DATA=オプションで指定されたファイルを FILE=オプションで指定された新しい移送ファイルにコピーします。DATA=オプションでは、移送される1つのデータセットのみを指定します。

複数のカタログとデータセットが含まれる可能性のあるライブラリの内容全体を含めるには、PROC CPORT で DATA=オプションの代わりに LIBRARY=オプションを指定します。

ライブラリ内のすべてのデータセットを移送するように指定した PROC CPORT の例を次に示します。

```
proc cport library=source file=cportout memtype=data;
```

### **例: PROC CPORT を使用した複数のカタログの移送ファイルの作成**

この例では、CPORT プロシジャを使用してライブラリ内の複数のカタログの移送ファイルを作成します。

```
libname source 'SAS-data-library';
filename cportout 'transport-file';
proc cport library=source file=cportout memtype=catalog;
run;
```

前述の例では、ライブラリ参照名 SOURCE は、移送元コンピュータにあるカタログを含むライブラリを指しています。ファイル参照名 CPORTOUT は、移送ファイルが作成される新しい場所を指しています。PROC CPORT ステートメントは、指定したライブラリから MEMTYPE=オプションで指定された種類のすべてのメンバを FILE=オプションで指定された新しい移送ファイルにコピーします。

PROC CPORT で EXCLUDE ステートメントを使用して、不要なカタログエントリを明示的に除外できます。または、PROC CPORT で SELECT ステートメントを使用して、必要なカタログエントリのみを指定することもできます。

### **例: PROC CPORT を使用したカタログ全体の移送ファイルの作成**

この例では、CPORT プロシジャを使用してカタログ全体の移送ファイルを作成します。

```
libname source 'SAS-data-library';
filename cportout 'transport-file';
proc cport catalog=source.testcat file=cportout;
run;
```

前述の例では、ライブラリ参照名 SOURCE は、移送元コンピュータにあるカタログの元の場所を指しています。ファイル参照名 CPORTOUT は、移送ファイルが作成される新しい場所を指しています。PROC CPORT ステートメントは、コピー元として、CATALOG=オプションで指定されたファイルを FILE=オプションで指定された新しい移送ファイルにコピーします。SOURCE ではライブラリ参照名を指定し、TESTCAT ではカタログ名を指定します。PROC CPORT で SELECT ステートメントまたは EXCLUDE ステートメントを省略すると、カタログ全体がコピーされます。

### **例: PROC CPORT を使用した特定の種類のカタログエントリの移送ファイルの作成**

この例では、CPORT プロシジャを使用して特定の種類のカタログエントリの移送ファイルを作成します。

```
libname source 'SAS-data-library';
filename cportout 'transport-file';
proc cport catalog=source.testcat file=cportout et=list;
```

```
run;
```

前述の例では、ライブラリ参照名 SOURCE は、移送元コンピュータにあるカタログの元の場所を指しています。ファイル参照名 CPORTOUT は、移送ファイルが作成される新しい場所を指しています。PROC CPORT ステートメントは、コピー元として、CATALOG=オプションで指定されたファイルを FILE=オプションで指定された新しい移送ファイルにコピーします。PROC CPORT の ET=オプションでは、種類が LIST のすべてのカタログエントリを新しいライブラリに書き込むように指定しています。代わりに EET=オプションを使用することで、全種類のエントリを除外することもできます。

### 例: PROC CPORT を使用したカタログエントリの移送ファイルの作成

この例では、CPORT プロシジャを使用して 1 つ以上のカタログエントリの移送ファイルを作成します。

```
libname source 'SAS-data-library';
filename cportout 'transport-file';
proc cport catalog=source.mycat file=cportout;
select testnpgm.list;
run;
```

前述の例では、ライブラリ参照名 SOURCE は、移送元コンピュータにあるカタログの元の場所を指しています。ファイル参照名 CPORTOUT は、移送ファイルが作成される新しい場所を指しています。PROC CPORT ステートメントは、コピー元として、CATALOG=オプションで指定されたファイルを FILE=オプションで指定された新しい移送ファイルにコピーします。

この例では、SELECT TESTNPGM.LIST で明示的に 1 つのカタログエントリを指定しています。ただし、1 つ以上のカタログエントリを名前でも指定することもできます。

PROC CPORT で EXCLUDE ステートメントを使用して、不要なカタログエントリを明示的に除外できます。または、PROC CPORT で SELECT ステートメントを使用して、必要なカタログエントリのみを指定することもできます。

## 移送先コンピュータへの移送ファイルの転送

次のいずれかの方法を使用して移送ファイルにアクセスできます。

- NFS(Network File Services)を使用して、動作環境でアクセスできるようにネットワーク上のファイルをマウントします。NFS および動作環境向けドキュメントを参照してください。
- FTP(File Transfer Protocol)サービスを使用して、バイナリ形式のファイルを特定の移送先コンピュータにコピーします。FTP の例については、“[FTP ユーティリティ](#)” (41 ページ) を参照してください。

## 移送先コンピュータでの移送ファイルの復元

### 移送ファイルの内容の確認

移送先コンピュータでの移送ファイルの復元者が移送元コンピュータでの移送ファイルの作成者と異なる場合、ファイルの復元処理を実行する前に移送ファイルに関する情報を取得します。移送先コンピュータで移送ファイルをネイティブ形式に復元する場合に役立つ可能性のある、情報の種類の例を次に示します。

表 3.1 移送ファイルの説明

移送元動作環境の種類と使用された SAS リリース	移送ファイルの作成に使用された方法	移送ファイル名	データセット	カタログ	カタログエントリ
z/OS SAS 9	PROC CPORT	TPORT.D AT	TEST.CIT Y  TEST.CLA SS	TEST.FORM ATS	REGF MT  SALE FMT  SIZEF MT

移送ファイルの作成に使用された方法は、テキストエディタを使用して調べられます。動作環境の読み込みコマンドまたは表示コマンドを使用して、移送ファイルを読み込むこともできます。XPORT エンジンと PROC CPORT では、ヘッダーの異なる移送ファイルが作成されます。詳細については、“[ファイルヘッダー: 移送ファイルの作成に使用された方法を調べる](#)” (65 ページ) を参照してください。

また、PROC CATALOG、PROC CONTENTS および PROC DATASETS の SAS プロシジャを使用して、移送ファイルの内容を表示することもできます。これらのプロシジャの詳細については、*Base SAS* プロシジャガイドを参照してください。

## PROC CIMPORT を使用して移送ファイルを復元する

### 例: PROC CIMPORT を使用した移送ファイルからのデータセットのインポート

この例では、CIMPORT プロシジャを使用して移送ファイルから複数のデータセットをインポートします。

```
filename importin 'transport-file';
libname target 'SAS-data-library';
proc cimport infile=importin library=target memtype=data;
run;
```

前述の例では、ファイル参照名 IMPORTIN は、移送ファイルが移送先コンピュータに転送された場所を指しています。ライブラリ参照名 TARGET は、移送ファイルがコピーされる新しい場所を指しています。PROC CIMPORT ステートメントは、コピー元として、INFILE=オプションで指定されたファイルを LIBRARY=オプションで指定された場所にコピーします。PROC CIMPORT ステートメントは、暗黙的に移送ファイルを移送先コンピュータのネイティブ形式に変換します。

LIBRARY=オプションではデータセットとカタログの両方をライブラリにコピーできるため、MEMTYPE=DATA を指定して操作をライブラリのデータセットに制限する必要があります。MEMTYPE=オプションを省略すると、ファイル参照名 IMPORTIN で参照されたファイル内のデータセットとカタログの両方がライブラリ参照名 TARGET で参照された場所にコピーされます。

PROC CIMPORT で出力先メンバをサブセット化するには、SELECT ステートメント、EXCLUDE ステートメントまたは MEMTYPE=オプションのいずれかを使用します。サブセット化の例を次に示します。

```
filename importin 'transport-file';
libname target 'SAS-data-library';
proc cimport infile=importin library=target memtype=data;
select grades;
```

```
run;
```

前述の例では、ライブラリ参照名 TARGET と MEMTYPE=オプションは、移送ファイルがコピーされる新しい場所を指しています。ファイル参照名 IMPORTIN は、移送ファイルが移送先コンピュータに転送された場所を指しています。PROC CIMPORT ステートメントは、コピー元として、INFILE=オプションで指定されたファイルを LIBRARY=オプションで指定された場所にコピーします。PROC CIMPORT ステートメントは、暗黙的に移送ファイルを移送先コンピュータのネイティブ形式に変換します。

SELECT ステートメントでは、ライブラリ TARGET のデータセット GRADES のみを選択しています。

### PROC CPORT と PROC CIMPORT に互換性のある出力先メンバの種類を使用する

移送ファイルからカタログをインポートするには、次の表に示すように、PROC CPORT と PROC CIMPORT で互換性のある出力先メンバの種類を使用していることを確認してください。

移送元コンピュータでサポートされるステートメント	移送先コンピュータでサポートされるステートメント
CPORT LIBNAME=	CIMPORT LIBNAME=または DATA=
CPORT DATA=	CIMPORT LIBNAME=または DATA=
CPORT CATALOG=	CIMPORT LIBNAME=または CATALOG=

出力先メンバに互換性がない場合、エラーまたは警告メッセージが表示されます。一般的なエラーを解決するために実行できる回復操作については、“[問題の回避と修正](#)” (70 ページ) を参照してください。PROC CPORT と PROC CIMPORT の構文の詳細については、*Base SAS プロシジャガイド* を参照してください。

### 例: PROC CIMPORT を使用した移送ファイルからの複数のカタログのインポート

この例では、CIMPORT プロシジャを使用して移送ファイルから複数のカタログをインポートします。複数のカタログをインポートするには、PROC CIMPORT で LIBRARY=オプションと MEMTYPE=CATALOG を指定します。

```
filename importin 'transport-file';
libname target 'SAS-data-library';
proc cimport infile=importin library=target memtype=catalog;
run;
```

前述の例では、ファイル参照名 IMPORTIN は、移送ファイルが移送先コンピュータに転送された場所を指しています。ライブラリ参照名 TARGET は、移送ファイルがコピーされる新しい場所を指しています。PROC CIMPORT ステートメントは、コピー元として、INFILE=オプションで指定されたファイルを LIBRARY=オプションで指定された場所にコピーします。出力先はライブラリであるため、ライブラリ参照名のみが指定されています。MEMTYPE=オプションでインポートをカタログのみに制限しています。PROC CIMPORT は、暗黙的に移送ファイルを移送先コンピュータのネイティブ形式に変換します。

**例: PROC CIMPORT を使用した移送ファイルからの 1 つのカタログのインポート**

この例では、CIMPORT プロシジャを使用して移送ファイルから 1 つのカタログをインポートします。1 つのカタログをインポートするには、PROC CIMPORT で CATALOG= オプションを指定します。

```
filename importin 'transport-file';
libname target 'SAS-data-library';
proc cimport infile=importin catalog=target.testcat;
run;
```

**例: PROC CIMPORT を使用した移送ファイルからの 1 種類のカテゴリのエントリのインポート**

この例では、CIMPORT プロシジャを使用して移送ファイルから 1 種類のカテゴリのエントリをインポートします。1 種類のカテゴリのエントリをインポートするには、PROC CIMPORT で ET=オプションと CATALOG=オプションを指定します。

```
filename importin 'transport-file';
libname target 'SAS-data-library';
proc cimport infile=importin catalog=target.testcat et=list;
run;
```

**例: PROC CIMPORT を使用した移送ファイルからの選択したカテゴリのエントリのインポート**

この例では、CIMPORT プロシジャを使用して移送ファイルから選択したカテゴリのエントリをインポートします。SELECT ステートメントを使用して、必要なカテゴリのエントリの名前を指定します。この例では、SELECT TESTNPGM.LIST ONE.SCL で明示的に選択したカテゴリのエントリを指定しています。また、PROC CIMPORT で CATALOG=オプションを指定する必要があります。

```
filename importin 'transport-file';
libname target 'SAS-data-library';
proc cimport infile=importin catalog=target.testcat;
select testnpgm.list one.scl;
run;
```

または、PROC CIMPORT で EXCLUDE ステートメントを使用して、不要なカテゴリのエントリを明示的に除外できます。





## 4 章

# DATA ステップまたは PROC COPY での XPORT エンジンの使用

XPORT エンジンの概要 .....	27
XPORT エンジンの利点 .....	28
XPORT エンジンの制限事項 .....	28
SAS データセットを SAS 6 形式に戻す .....	28
移送元コンピュータでの移送ファイルの作成 .....	30
例: DATA ステップを使用した単一データセットの移送ファイルの作成 .....	30
例: PROC COPY を使用した 1 つ以上のデータセットの移送ファイルの作成 .....	30
ネットワークを経由した移送ファイルの転送 .....	31
移送先コンピュータでの移送ファイルの復元 .....	31
移送ファイルの内容の特定 .....	31
例: DATA ステップを使用した移送ファイルからの単一データセットの復元 .....	32
例: PROC COPY を使用した移送ファイルからのデータセットの復元 .....	32

## XPORT エンジンの概要

XPORT エンジンは移送形式でファイルを作成します。移送形式は環境に依存しない標準を文字エンコーディングと数値表現に使用します。XPORT エンジンで作成した移送ファイルを、動作環境にまたがって移送して読み込むには、DATA ステップまたは PROC COPY で XPORT エンジンを使用します。

移送元コンピュータで移送ファイルを作成し、移送先コンピュータでそのファイルを読み込むプロセスを次に示します。

1. 移送元コンピュータで、DATA ステップまたは PROC COPY で XPORT エンジンを使用して移送ファイルを作成します。
2. ファイルが移送元コンピュータから移送先コンピュータへ転送されます。
3. 移送先コンピュータで、DATA ステップまたは PROC COPY で XPORT エンジンを使用して移送ファイルを読み込みます。

注: PROC COPY を使用して作成された移送ファイルは、XPORT エンジンを使用して作成された移送ファイルとは互換性がありません。

---

## XPORT エンジンの利点

DATA ステップまたは COPY プロシジャで XPORT エンジンを使用することには次の利点があります。

- 移送ファイルの移動先で使用されている SAS リリースの新旧に関係なく、動作環境間でファイルを移動できます。

注: データセットを戻す(新しいリリースから以前のリリースに移動)と、新しいリリースに固有の機能が削除(無効化)されます。たとえば、SAS 9 から SAS 6 に移動すると、SAS 9 の長い変数名は 8 バイトに切り捨てられます。ファイルを戻す方法の詳細については、“[SAS データセットを SAS 6 形式に戻す](#)”(28 ページ)を参照してください。

- 移送先動作環境の SAS リリースが不明な場合は、XPORT エンジンを使用して移送ファイルを送信できます。
- いったん移送ファイルを作成すれば、同じファイルを異なる SAS リリースの複数移送先動作環境に送信できます。

DATA ステップで XPORT エンジンを使用する主な目的は、1 つ以上のデータセットを動的に作成して順序を指定し、移送形式に変換することです。これに対し、PROC COPY ではライブラリにすでに存在する複数のデータセットを変換できます。

---

## XPORT エンジンの制限事項

XPORT エンジンには次の制限があります。

- XPORT エンジンは種類が DATA のメンバのみをサポートしています。種類が CATALOG または VIEW のメンバはサポートされていません。
- XPORT エンジンは、SAS 6 と互換性のある機能セットをサポートします。長い変数名などの SAS 9 機能はサポートされていません。移送処理中に発生した制限は、警告メッセージまたはエラーメッセージに表示されます。一般的なエラーメッセージと回復操作の詳細については、“[ファイル library.member.DATA のメンバ名は XPORT エンジンでは長すぎます](#)”(75 ページ)を参照してください。
- PROC COPY で XPORT エンジンを使用する場合、種類がビューまたは MDDB の移送は一切サポートされていません。

---

## SAS データセットを SAS 6 形式に戻す

データセットを SAS 6 に戻す方法に利用できるのは、SAS/CONNECT および PROC COPY で XPORT エンジンを使用して UPLOAD プロシジャおよび DOWNLOAD プロシジャを行う方法のみです。

注: SAS/CONNECT の利用には、個別のライセンスが必要です。

SAS 9 と SAS 8 では長い変数名、長い変数ラベルおよび長いデータセットがサポートされたため、SAS 9 と SAS 8 のデータセットが SAS 6 のデータセットと非互換になる

場合があります。SAS 6 に戻すには、これらの長い名前は切り捨てを行って SAS 6 でサポートされる長さにする必要があります。切り捨てのルールは次のとおりです。

変換する SAS 9 および SAS 8 のデータセットオブジェクト名	SAS 6 の文字数
データセットラベル	40
変数ラベル	40
変数名	8

SAS 9 および SAS 8 のファイルを SAS 6 に移送して戻すには、ファイルを移送する SAS セッションで、ポータブルシステムオプション VALIDVARNAME の値を V6 に設定します。SAS システムオプションおよびマクロ変数の形で指定した例を次に示します。

```
options VALIDVARNAME=V6
%let VALIDVARNAME=V6;
```

VALIDVARNAME システムオプションの設定方法の詳細については、(SAS システムオプション: リファレンス)を参照してください。

8 文字の長さの変数名を生成する切り捨てアルゴリズムは、名前の競合も解決しません。

- 最初に出現した 8 文字より長い名前は 8 文字に切り捨てられます。たとえば、PROPERTYTAXRATE から PROPERTY への切り捨ては最初の切り捨てで行われます。
- 次に出現した 8 文字より長い名前は、まず 8 文字に切り捨てられます。その名前が既存の変数名と競合するときは、さらに 7 文字に切り捨てられて、接尾辞として 2 が追加されます。たとえば、PROPERTYTAXRATE は切り捨ての結果、PROPERT2 となります。
- 接尾辞は、切り捨てられた名前が既存名と競合するたびに 1 ずつ増加します。接尾辞が 9 に到達したときは、次に競合する変数名は長さが 6 文字に切り捨てられて、接尾辞として 10 が追加されます。たとえば、PROPERTYTAXRATE は切り捨ての結果、PROPER10 となります。

VALIDVARNAME オプションによって長い変数名の切り捨てに関する問題は解決されますが、次の SAS 9 または SAS 8 機能を SAS 6 に戻す方法はありません。

- 8 文字を超えるデータセット名
- 一貫性制約
- データセットの世代
- 監査証跡

これらの機能を使用しているデータセットに戻すには、SAS 9 または SAS 8 のセッションで、該当する SAS 9 または SAS 8 の機能を使用せずにデータセットを再作成します。

注: SAS/CONNECT は、SAS 9 または SAS 8 の一部のカタログエントリについて、SAS 6 へのアップロードまたは SAS 6 からのダウンロードをサポートします。詳細については、“UPLOAD プロシジャ” (SAS/CONNECT User's Guide) および “DOWNLOAD プロシジャ” (SAS/CONNECT User's Guide) を参照してください。

---

## 移送元コンピュータでの移送ファイルの作成

### 例: DATA ステップを使用した単一データセットの移送ファイルの作成

この例では、DATA ステップを使用して単一データセットの移送ファイルを作成します。

```
libname source 'SAS-data-library';
libname xportout xport 'transport-file';
data xportout.grades;
set source.grades;
run;
```

前述の例では、ライブラリ参照名 SOURCE は、移送元動作環境に存在するデータセットの元の場所を指しています。ライブラリ参照名 XPORTOUT は、移送ファイルが作成される新しい場所を指しています。この LIBNAME ステートメントの XPORT エンジンは、データセットが移送形式で作成されることを指定しています。SET ステートメントでデータセット GRADES が読み込まれて移送形式で再作成された後、DATAE ステートメントで指定した場所にコピーされます。

### 例: PROC COPY を使用した 1 つ以上のデータセットの移送ファイルの作成

この例では、COPY プロシジャを使用して複数データセットの移送ファイルを作成します。

```
libname source 'SAS-data-library';
libname xportout xport 'transport-file';
proc copy in=source out=xportout memtype=data;
run;
```

前述の例では、ライブラリ参照名 SOURCE は、移送元動作環境に存在するライブラリの元の場所を指しています。ライブラリ参照名 XPORTOUT は、移送ファイルがコピーされる新しい場所を指しています。この LIBNAME ステートメントの XPORT エンジンは、ライブラリが移送形式で作成されることを指定しています。PROC COPY ステートメントは、IN=オプションで指定されたライブラリのすべてのデータセットを、OUT=オプションで指定された新しいライブラリにコピーします。MEMTYPE=DATA オプションは、コピーされるファイルを種類が DATA のファイルに制限します(カタログおよびビューを除外します)。

#### 注意:

**MEMTYPE=DATA オプションは省略しないでください。**省略すると、カタログおよびビューを含むライブラリから内容全体を移送ファイルへコピーしようとします。XPORT エンジンでは、種類が CATALOG または VIEW のメンバはサポートされていません。エラーメッセージおよび警告メッセージが SAS ログに書き込まれます。

この例では、PROC COPY を使用して単一データセットの移送ファイルを作成します。

```
libname source 'SAS-data-library';
libname xportout xport 'transport-file';
proc copy in=source out=xportout memtype=data;
select grades;
run;
```

前述の例では、ライブラリ参照名 SOURCE は、移送元動作環境に存在するデータセットの元の場所を指しています。ライブラリ参照名 XPORTOUT は、移送ファイルがコ

コピーされる新しい場所を指しています。この LIBNAME ステートメントの XPORT エンジン、データセットが移送形式で作成されることを指定しています。PROC COPY ステートメントは、IN=オプションで指定されたすべてのデータセットを、OUT=オプションで指定された新しいライブラリにコピーします。MEMTYPE=DATA オプションは、コピーされるファイルを種類が DATA のファイルに制限します(カタログおよびビューを除外します)。SELECT ステートメントは、データセット GRADES のみが新しいライブラリにコピーされることを指定しています。ただし、ここに複数のデータセットを指定することもできます。SELECT ステートメントを省略すると、すべてのデータセットが移送ファイルにコピーされます。

注: 対象とするデータセットを SELECT ステートメントで指定する代わりに、対象から明示的に除外するデータセットを EXCLUDE ステートメントで指定することもできます。

## ネットワークを経由した移送ファイルの転送

次のいずれかの方法を使用して移送ファイルにアクセスできます。

- NFS(Network File Services)を使用して、動作環境でアクセスできるようにネットワーク上のファイルをマウントします。NFS および動作環境向けドキュメントを参照してください。
- FTP(File Transfer Protocol)サービスを使用して、バイナリ形式のファイルを特定の移送先コンピュータにコピーします。FTP の例については、“[ファイルの転送](#)” (39 ページ) を参照してください。

## 移送先コンピュータでの移送ファイルの復元

### 移送ファイルの内容の特定

移送元動作環境での移送ファイルの作成者が移送先動作環境での移送ファイルを復元者と異なる場合、復元処理を実行する前に、移送ファイルに関する情報を入手してください。移送ファイルを移送先動作環境でネイティブ形式に復元するのに役立つ情報の種類は、たとえば次のとおりです。

表 4.1 移送ファイルの説明

移送元動作環境の種類と使用された SAS リリース	移送ファイルの作成に使用された方法	移送ファイル名	データセット
z/OS	XPORT エンジン	TPORT.DAT	TEST.CITY
SAS 9			TEST.CLASS

移送ファイルの作成に使用された方法は、ファイルのヘッダーを調べればわかります。XPORT エンジンと PROC CPORT では、ヘッダーの異なる移送ファイルが作成されます。詳細については、“[ファイルヘッダー: 移送ファイルの作成に使用された方法を調べる](#)” (65 ページ) を参照してください。

また、PROC CONTENTS および PROC DATASETS を使用して、移送ファイルの内容を表示することもできます。これらの処理についての詳細は、*Base SAS プロシジャガイド*を参照してください。

### 例: DATA ステップを使用した移送ファイルからの単一データセットの復元

この例では、DATA ステップを使用して、移送ファイルからデータセットを復元します。

```
libname xportin xport
'transport-file';
libname target 'SAS-data-library';
data target.grades;
set xportin.grades;
run;
```

前述の例では、ライブラリ参照名 XPORTIN は、移送先動作環境に転送した、エクスポートされたデータセットの場所を指しています。XPORT エンジンは、データセットが移送形式で読み込まれること指定しています。ライブラリ参照名 TARGET は、変換されたファイルがコピーされる新しい場所を指しています。SET ステートメントで移送形式のデータセット XPORTIN.GRADES が読み込まれて変換された後、DATA ステートメントで指定した場所にコピーされます。DATA ステップで XPORT エンジンを使って移送元動作環境で作成したのは単一データセットの移送ファイルであったため、移送先動作環境で復元できるのは単一のデータセットです。

### 例: PROC COPY を使用した移送ファイルからのデータセットの復元

この例では、COPY プロシジャを使用して、移送ファイルから 1 つ以上のデータセットを復元します。

```
libname xportin xport
'transport-file';
libname target 'SAS-data-library';
proc copy in=xportin out=target;
select grades;
run;
```

前述の例では、ライブラリ参照名 XPORTIN は、移送先動作環境に転送した移送ファイルの場所を指しています。この LIBNAME ステートメントの XPORT エンジン、この場所の移送ファイルが移送形式で読み込まれることを指定しています。ライブラリ参照名 TARGET は、移送ファイルがネイティブ形式でコピーされる新しい場所を指しています。PROC COPY ステートメントは、IN=オプションで指定されたライブラリの選択されたデータセット GRADES を、OUT=オプションで指定された新しいライブラリにコピーします。

SELECT ステートメントを使用し、1 つ以上の特定データセットを指定して新しいライブラリにコピーします。移送ファイルのすべてのデータセットを指定してコピーするには、PROC COPY の SELECT ステートメントを省略します。

注: 対象とするデータセットを SELECT ステートメントで指定する代わりに、対象から明示的に除外するデータセットを PROC COPY 内の EXCLUDE ステートメントで指定することもできます。

## 5 章

## DATA ステップまたは PROC COPY での XML エンジンの使用

---

XML エンジンの概要 .....	33
XML エンジンの利点 .....	34
XML エンジンの制限事項 .....	34
移送元コンピュータでの XML ドキュメントの作成 .....	34
例: DATA ステップを使用したデータセットからの XML ドキュメントの作成 .....	34
例: PROC COPY を使用したデータセットからの XML ドキュメントの作成 .....	35
ネットワークを経由した XML ドキュメントの転送 .....	35
移送先コンピュータで XML ドキュメントをデータセットに復元する .....	36
例: DATA ステップを使用を使用した XML ドキュメントからのデータセットの復元 .....	36
例: PROC COPY を使用した XML ドキュメントからのデータセットの復元 .....	36

---

XML エンジンの概要

SAS データセットから XML ドキュメントへのエクスポートと、XML ドキュメントから SAS データセットへの復元には XML エンジンを使用します。XML ドキュメントを動作環境にまたがって移送して読み込むには、DATA ステップまたは PROC COPY で XML エンジンを使用します。

移送元コンピュータで XML ドキュメントを作成し、移送先コンピュータでそのファイルを読み込むプロセスを次に示します。

1. 移送元コンピュータで、DATA ステップまたは PROC COPY で XML エンジンを使用して XML ドキュメントを作成します。
2. ファイルが移送元コンピュータから移送先コンピュータへ転送されます。
3. 移送先コンピュータで、DATA ステップまたは PROC COPY で XML エンジンを使用して XML ドキュメントを読み込みます。

XML エンジンの詳細については、*SAS XML LIBNAME Engine: ユーザーガイド*を参照してください。

---

## XML エンジンの利点

DATA ステップまたは PROC COPY で XML エンジンを使用することには次の利点があります。

- XML データがテキストとして保存されます。SAS ファイルとは異なり、XML ドキュメントはテキストエディタで読み込んだり、更新したりできます。
- SAS アプリケーション以外のアプリケーションに XML ドキュメントをインポートできます。たとえば、XML ドキュメントを Oracle アプリケーションに入力したり、Web に配信したりできます。SAS データセットとして復元して処理を続行することもできます。他のプログラムとのデータ互換性を重視する場合には、XML エンジンの使用をお勧めします。
- XML エンジンには、SAS 8 以降の機能をサポートしています。XPORT エンジンとは異なり、XML エンジンには長い名前などの SAS 8 機能をサポートします。

---

## XML エンジンの制限事項

XML エンジンには次の制限があります。

- XML エンジンには SAS データファイルのみをサポートしています。ビューやその他の SAS ファイルの種類はサポートされません。
- XML エンジンには、SAS 6 以前のリリースではサポートされません。SAS 6 との間で移動する場合には、XPORT エンジンを使用する必要があります。
- XML エンジンには、他の方法よりも処理に長い時間がかかります。処理時間が問題となる場合は、XML の使用をお勧めしません。
- XML ドキュメントはサイズが大きい場合があります。ディスク容量やネットワーク帯域が問題となる場合は、XML の使用をお勧めしません。
- XML エンジンには、文字変換で転送方式に依存します。XML ドキュメントをバイナリファイルとして転送すると、移送先コンピュータで読め込めない場合があります。

---

## 移送元コンピュータでの XML ドキュメントの作成

### 例: DATA ステップを使用したデータセットからの XML ドキュメントの作成

この例では、DATA ステップで XML エンジンを使用して、データセットから XML ドキュメントを作成します。

```
libname source 'SAS-data-library';  
libname xmlout xml 'XML-document';  
data xmlout.grades;  
set source.grades;  
run;
```

前記の例では、ライブラリ参照名 SOURCE は、移送元コンピュータにあるライブラリの場所を指しています。ライブラリ参照名 XMLOUT は、XML ドキュメントを作成する場



所を指しています。この LIBNAME ステートメントの XML エンジン、ファイルが XML マークアップで作成されることを指定しています。SET ステートメントでデータセット GRADES が読み込まれ、LIBNAME ステートメントで指定した場所に XML マークアップが生成されます。

作成される XML ドキュメントの内容は次のとおりです。

**アウトプット 5.1** データセット GRADES から生成された XML 出力

```
<?xml version="1.0" encoding="windows-1252" ?>
<TABLE>
<GRADES>
<student> Fred </student>
<test1> 66 </test1>
<test2> 80 </test2>
<final> 90 </final>
</GRADES>
<GRADES>
<student> Wilma </student>
<test1> 97 </test1>
<test2> 91 </test2>
<final> 98 </final>
</GRADES>
</TABLE>
```

### 例: PROC COPY を使用したデータセットからの XML ドキュメントの作成

この例では、COPY プロシジャを使用して、データセットから XML ドキュメントを作成します。

```
libname source 'SAS-data-library';
libname xmlout xml 'XML-document';
proc copy in=source out=xmlout;
select grades;
run;
```

前記の例では、ライブラリ参照名 SOURCE は、移送元コンピュータにあるライブラリの場所を指しています。ライブラリ参照名 XMLOUT は、XML ドキュメントを作成する場所を指しています。この LIBNAME ステートメントの XML エンジン、ファイルが XML マークアップで作成されることを指定しています。PROC COPY ステートメントは、IN=オプションで指定されたライブラリのデータを、OUT=オプションで指定されたライブラリにコピーします。SELECT ステートメントは、入力ライブラリからコピーするデータセットを指定しています。

注: SELECT ステートメントで指定するデータセットが複数の場合は、XML エンジンで入力ライブラリの全メンバが処理されて、オブザベーションは連結されます。

---

## ネットワークを経由した XML ドキュメントの転送

XML ドキュメントを利用できるようにするには、次のいずれかの方法を使用します。

- NFS(Network File Services)を使用して、動作環境でアクセスできるようにネットワーク上のファイルをマウントします。NFS および動作環境向けドキュメントを参照してください。

- FTP(File Transfer Protocol)を使って特定の移送先コンピュータにファイルをコピーする。FTP の詳細については、“[ファイルの転送](#)” (39 ページ) を参照してください。

作成された XML ドキュメントを転送するときにデフォルトのエンコーディングを使用した場合、ファイルの転送に ASCII(テキスト)モードを使ってください。エンコーディングの値を明示的に指定した場合は、ファイルの転送にバイナリモードを使ってください。

## 移送先コンピュータで XML ドキュメントをデータセットに復元する

### 例: DATA ステップを使用を使用した XML ドキュメントからのデータセットの復元

この例では、DATA ステップを使用して、XML ドキュメントからデータセットを復元します。

```
libname xmlin xml 'XML-document';
libname target 'SAS-data-library';
data target.grades;
set xmlin.grades;
run;
```

前記の例では、ライブラリ参照名 XMLIN は、XML ドキュメントがある場所を指しています。XML エンジンは、SAS データセットを読み込むことを指定しています。ライブラリ参照名 TARGET は、変換した SAS データセットのコピー先を指しています。SET ステートメントで XML 形式のデータセット XMLIN.GRADES が読み込まれて変換された後、DATA ステートメントで指定した場所にコピーされます。

### 例: PROC COPY を使用した XML ドキュメントからのデータセットの復元

この例では、COPY プロシジャを使用して、XML ドキュメントからデータセットを復元します。

```
libname xmlin xml 'XML-document';
libname target 'SAS-data-library';
proc copy in=xmlin out=target;
run;
```

前記の例では、ライブラリ参照名 XMLIN は、XML ドキュメントがある場所を指しています。XML エンジンは、XML ドキュメントを XML 形式で読み込むことを指定しています。ライブラリ参照名 TARGET は、XML ドキュメントの内容のコピー先を指しています。PROC COPY ステートメントは、IN=オプションで指定されたライブラリの内容を、OUT=オプションで指定されたライブラリにコピーします。

## 3 部

---

# 移送ファイルと非ネイティブファイル の転送

6 章	
ファイルの転送	39



## 6 章

## ファイルの転送

---

ファイル転送の概要 .....	39
移送ファイルの属性 .....	40
非ネイティブファイルや移送ファイルへの FILENAME ステートメントまたは FTP の使用 .....	41
例: ファイル転送での FILENAME ステートメントの使用 .....	41
FTP ユーティリティ .....	41
例: FTP を使用した外部ファイルと移送ファイルの転送 .....	42
例: 磁気メディアを使用した外部ファイルと移送ファイルの転送 .....	43

---

## ファイル転送の概要

転送できるファイルの種類は次のとおりです。

## 外部ファイル

移送先コンピュータには外部形式になるファイル。たとえば、Windows ファイル形式は、UNIX 動作環境には外部形式となります。

## 移送ファイル

移送形式に形式が変更されたファイル。このファイルを後で読み込み、移送先コンピュータのネイティブ形式に変換できます。

転送とは、ネットワークを経由して、動作環境間で外部ファイルまたは移送ファイルを伝送するプロセスです。この操作を実行するために使用できるさまざまなサードパーティ製品があります。この例では、FTP(File Transfer Protocol)を使用して転送操作について説明します。

転送操作を行うには、次のいずれかのアクションを実行します。

## ファイルのプッシュ

移送元コンピュータで FTP の `put` コマンドを使用して、移送元コンピュータから移送先コンピュータにファイルをコピーします。

## ファイルのプル

移送先コンピュータで FTP の `get` コマンドを使用して、移送元コンピュータから移送先コンピュータにファイルをコピーします。

移送元コンピュータから移送先コンピュータにファイルをプッシュできるかどうかは、移送先コンピュータに書き込みができるアクセス権限があるかどうかによって決まります。詳細については、ネットワークのドキュメントを参照してください。

## 移送ファイルの属性

ファイル属性は、移送先コンピュータに転送される移送ファイルのデータの編成と形式を表します。移送ファイルには、次の属性値が必要です。

論理レコード長(LRECL)	80
ブロックサイズ(BLKSIZE)	8000 バイト
レコード形式(RECFM)	固定ブロック

注: 場合によっては、8000 バイト未満のブロックサイズ値を使用した方が、ストレージデバイスの効率が上がる場合があります。ブロックサイズ値は、論理レコード長値の倍数にする必要があります。

### 注意:

ただし、z/OS に限り、ブロックサイズに 80 または 80 の倍数(たとえば、160、240、320)を指定する必要があります。

必須ではありませんが、ファイル属性はその他のすべての移送元コンピュータに設定できます。ファイル属性は、移送ファイルが作成される移送元コンピュータと使用される転送方式に従って宣言されます。

さらに、ファイル属性を必要とする動作環境のファイルに、通信ソフトウェアプロトコルを使用してファイル属性を指定する必要があります。たとえば、UNIX 動作環境から z/OS 動作環境に移送ファイルを転送する場合、通信ソフトウェアを使用してファイル属性を指定する必要があります。

ファイル属性が必要な動作環境にファイル属性を設定する他に、動作環境のデフォルトのファイル属性設定が通信ソフトウェアによって変更されないようにする必要があります。

または、移送ファイルを移送元コンピュータからテープに、次にテープから移送先コンピュータのディスクに転送するには、転送にかかわる動作環境の入力デバイスと出力デバイスを定義する、動作環境に固有のコマンドを使用する必要があります。

移送ファイルを作成した後に、ネットワークを経由するか、ディスクやテープなどのマウント可能な磁気メディアを使用して、この移送ファイルを移送先コンピュータに転送する必要があります。

ファイル属性を正しく設定しないと、移送ファイルが破損したり、無効になったりする可能性があります。

これらの動作環境のファイル属性の設定またはテープコマンドの使用の詳細については、次の該当するトピックを参照してください。

- “OpenVMS 動作環境” (47 ページ)
- “z/OS 動作環境” (53 ページ)
- “UNIX 動作環境” (59 ページ)
- “Windows 動作環境” (57 ページ)

## 非ネイティブファイルや移送ファイルへの FILENAME ステートメントまたは FTP の使用

### 例: ファイル転送での FILENAME ステートメントの使用

#### 注意:

FILENAME ステートメントは、外部ファイルではなく、移送ファイルにのみ使用します。

FTP アクセス方式で FILENAME ステートメントを使用して、ファイル属性を指定し、ネットワーク経由で移送ファイルを移送先コンピュータに転送する例を次に示します。

```
filename tranfile ftp 'tport.dat' lrecl=80 blocksize=8000
recfm=f cd='mydir' host='myhost.mycompany.com'
user='myuser' pass='mypass'
rcmd='site umask 022 recfm=s';
```

FILENAME ステートメントでは、ファイル参照名 TRANFILE を指定します。これは、ネットワーク経由で転送する外部ファイル TPORT.DART を指定します。FTP オプションで、レコード長、ブロックサイズ、レコード形式というレコード属性の値を指定します。また、FTP オプションでは、移送先コンピュータ上のファイル転送先、移送先コンピュータへのアクセスを許可するユーザー ID とパスワードを指定します。最後に、移送先コンピュータのファイルモード作成マスクとバイナリ転送が指定されます。FILENAME ステートメントの FTP アクセス方式の詳細については、(SAS ステートメント: リファレンス) および動作環境向けの付随ドキュメントを参照してください。

注: FTP アクセス方式の他に、FILENAME ステートメントでは、SOCKET、URL、または SMTP アクセス方式も使用できます。ファイルの転送先は、FTP ではハードディスク、SOCKET では TCP/IP ポート、URL では Web、SMTP では電子メールになります。これらのアクセス方式の詳細については、“FILENAME ステートメント、SOCKET アクセス方式” (SAS ステートメント: リファレンス)、“FILENAME ステートメント、URL アクセス方式” (SAS ステートメント: リファレンス) または “FILENAME ステートメント、EMAIL (SMTP) アクセス方式” (SAS ステートメント: リファレンス) を参照してください。

### FTP ユーティリティ

FTP は、ファイル転送プロトコルのユーザーインターフェイスです。FTP では、移送元コンピュータと移送先コンピュータ間のネットワーク接続を経由してファイルがコピーされます。FTP は起動元のコンピュータから実行されます。これは、移送元コンピュータと移送先コンピュータのいずれかになります。

ネットワーク経由で移送先コンピュータにファイルを転送するには、バイナリ(またはイメージ)形式の転送を指定する必要があります。この形式を使用すると、SAS を実行するなどの動作環境でも一貫したファイル構造が保証されます。バイナリ形式を宣言するには FTP の BINARY コマンドを使用する必要があります。一般的な FTP コマンドの構文については、“例: FTP を使用した外部ファイルと移送ファイルの転送” (42 ページ) を参照してください。

ASCII 形式でファイルを転送すると、移送先コンピュータ上の移送ファイルに文字が追加されます。通常、これらの文字は、ラインフィード、キャリッジリターン、レコードの終端記号や、ファイル特性を定義するために一部の動作環境で使用されるその他の文字です。

SAS を実行する移送先コンピュータでは、これらの文字を含まない特定の構造形式になっている移送ファイルが要求されます。これらの文字が挿入されるとファイルが破損し、移送先コンピュータでファイルを正常に復元することができなくなります。通常は、ファイルの破損を警告するエラーメッセージが表示されます。ファイルの破損および修復操作の詳細については、“問題の回避と修正” (70 ページ) を参照してください。

注: SAS 6.11 から SAS 9 では、FTP アクセス方式で FILENAME ステートメントを使用できます。このステートメントでは、ファイル転送に使用するファイル属性を指定します。SAS 6.11 より前のリリースでは、FTP アクセス方式で FILENAME ステートメントはサポートされません。

### 例: FTP を使用した外部ファイルと移送ファイルの転送

外部ファイルは、移送ファイルと同じ方法で転送します。この 2 つのファイルで唯一違う点は、ファイル名です。SAS では、移送ファイル名に、適切なメンバの種類の前が追加されます(たとえば、データセットの場合は.DAT)。CEDA 機能で作成されたファイルには、適切な SAS 9 または SAS 8 のファイル名拡張子が追加されます(たとえば、データセットの場合は.SAS9BDAT)。

これらの例では、TRANFILE で、ネットワークを経由して転送される移送ファイルの名前を指定します。TARGET で、移送先コンピュータ上の外部形式ファイルまたは移送ファイルの転送先を指定します。

この例では、移送先コンピュータに外部ファイルまたは移送ファイルを配置するために移送元コンピュータで使用される FTP コマンドを示します。

```
/* putting transport file on the target computer */
> open target-computer
> binary
> put tranfile target-computer-filename
> close
> quit
```

この例では、移送元コンピュータから外部ファイルまたは移送ファイルを取得するために移送先コンピュータで使用される FTP コマンドを示します。

```
/* At the target computer, getting transport file from */
/* the source computer */
> open source-computer
> binary
> get tranfile source-computer-filename
> close
> quit
```

UNIX システムにアクセスできる場合、詳細については、ftp(1)マニュアルページを参照してください。

注: FTP の put コマンドでファイルをサーバー上の場所にコピーするには、サーバー上の移送先場所に対する書き込み権限が必要です。サーバーの場所にファイルを配置するローカルユーザーの権限は不明確なため、リモートユーザーは代わりに FTP の get コマンドを使用してクライアントからファイルを取得することをお勧めします。ローカルユーザーは、リモートユーザーがアクセスするファイルに対する読み書き権限を付与する必要があります。

このコードは、移送先コンピュータのユーザー JOE が、OpenVMS 移送元コンピュータから 2 つの移送ファイルを取得する例を示しています。



```

hp> ftp myhost.mycompany.com 1
Connected to myhost.mycompany.com
220 myhost.mycompany.com MultiNet FTP Server Process V4.0(15)
at Mon 13-Jan-03 12:59PM-EDT
Name (myhost.mycompany.com): joe
331 User name (joe) ok. Password, please.
Password:
230 User JOE logged into DISK01:[JOE] at Mon 13-Jan-03
12:59PM-EDT, job 27a34cef.
Remote system type is VMS.
ftp> cd [.xpttest] 2
250 Connected to DISK01:[JOE.XPTTEST].
ftp> binary 80 3
200 Type I ok.
ftp> get xptds.dat xptds.dat 4
200 Port 14.83 at Host 10.26.2.45 accepted.
150 IMAGE retrieve of DISK01:[JOE.XPTTEST]XPTDS.DAT;1 started.
226 Transfer completed. 1360 (8) bytes transferred. 5
1360 bytes received in 0.02 seconds (87.59 Kbytes/s)
ftp> get xptlib.dat xptlib.dat 6
200 Port 14.84 at Host 10.26.2.45 accepted.
150 IMAGE retrieve of DISK01:[JOE.XPTTEST]XPTLIB.DAT;1 started.
226 Transfer completed. 3120 (8) bytes transferred. 7
3120 bytes received in 0.04 seconds (85.81 Kbytes/s)
ftp> quit 8

```

- 1 HP-UX 動作環境から FTP を起動し、OpenVMS 動作環境の MYHOST.MYCOMPANY.COM に接続します。
- 2 HP-UX 移送元コンピュータと OpenVMS 移送先コンピュータ間の接続が確立された後、FTP プロンプトでユーザーの JOE は移送ファイル XPTTEST が含まれている移送先コンピュータのディレクトリに変更します。
- 3 移送ファイル属性の BINARY 80 は、OpenVMS 移送ファイルがレコードサイズ 80 バイトの BINARY 形式で移送元コンピュータに転送されることを示します。
- 4 FTP コマンドで、移送先コンピュータから XPTDS.DAT という移送ファイルが取得され、現在のディレクトリに、同じ XPTDS.DAT という名前の新しいファイルにコピーされます。
- 5 メッセージは、転送が正常に行われ、移送ファイルの長さが 1360 バイトであることを示しています。
- 6 FTP コマンドで、移送先コンピュータから XPTLIB.DAT という別の移送ファイルが取得され、現在のディレクトリに、同じ XPTLIB.DAT という名前の新しいファイルにコピーされます。
- 7 メッセージは、転送が正常に行われ、移送ファイルの長さが 3120 バイトであることを示しています。
- 8 ユーザーは FTP セッションを終了します。

### 例: 磁気メディアを使用した外部ファイルと移送ファイルの転送

テープを使用して移送ファイルを転送する場合は、必ずラベル付けされていないテープを使用してください。標準のラベル付けされたテープを使用することもできますが、通常は移送先コンピュータでファイルを読み込むときに追加の作業が必要となります。

また、移送ファイルが1個のテープの容量を超える場合は、復元プロセス中に問題が発生する可能性があります。マルチボリュームのテープを使用する代わりに、元のライブラリを2つ以上のライブラリに分割し、ライブラリごとに別個のテープを作成してください。元のライブラリは、移送先コンピュータで再構築できます。

次の例に示すように、移送元コンピュータで LIBNAME ステートメントを使用して、移送ファイルを磁気メディアに割り当てます。

UNIX

```
libname tranfile xport '/dev/tape';
```

Windows

```
libname tran xport 'a:\test';
```

ファイルパスの指定は、動作環境により異なります。

移送ファイルを物理テープへ移動するために使用される方式も、動作環境によって異なります。

UNIX の例を次に示します。

```
dd if=tranfile of=/dev/tape1 bs=8000;
```

移送元コンピュータで、UNIX の dd コマンドを使用して、指定した出力デバイスへ指定した入力ファイルをコピーします。ブロックサイズは 8000 です。

移送先コンピュータで、テープからディスクへ移送ファイルをコピーする必要があります。

UNIX の例を次に示します。

```
dd if=/dev/tape1 of=tranfile bs=8000;
```

移送先コンピュータで、LIBNAME ステートメントを使用して、移送ファイルをネイティブ形式に変換し、変換後のファイルを特定のファイルの場所に割り当てます。

UNIX の例を次に示します。

```
libname tranfile xport '/dev/tape1';
```

## 4 部

---

# 動作環境固有の情報

7 章		
	<b>OpenVMS 動作環境</b> .....	47
8 章		
	<b>z/OS 動作環境</b> .....	53
9 章		
	<b>Windows 動作環境</b> .....	57
10 章		
	<b>UNIX 動作環境</b> .....	59
11 章		
	<b>SAS ファイル拡張子とファイルヘッダー</b> .....	63



## 7 章

## OpenVMS 動作環境

---

OpenVMS システムファイルの属性の表示 .....	47
OpenVMS のファイル属性 .....	48
OpenVMS でファイルの作成に使用された SAS バージョンを特定する .....	48
OpenVMS へのテープデバイスのマウント .....	49
OpenVMS のエラーメッセージ .....	49
移送ファイルが損傷しています .....	49
ファイルに使用できないメンバまたはライブラリ .....	51
レコードが切り捨てられました .....	51
データ取得時の内部エラー .....	51

---

## OpenVMS システムファイルの属性の表示

OpenVMS 動作環境で作成されたファイルの属性を表示するには、次のコマンドを指定します。

```
DIR/FULL transport-file
```

通常は次のよう出力されます。

```
Directory DISK01: [JOE.XPTTEST]

XPTLIB.DAT;1 File ID: 31223,952,0)
Size: 7/8 Owner: [DISK01,JOE]
Created: 25-APR-2008 16:47:31.34
Revised: 25-APR-2008 16:47:31.69 (1)
Expires: <No backup recorded>
Effective: <None specified>
Recording: <None specified>
File organization: Sequential
Shelved state: Online
File attributes: Allocation: 8, Extend: 0,
Global buffer count: 0 Version limit: 2
Record format: Fixed length 512 byte records 1
Record attributes: None 2
RMS attributes: None
Journaling enabled: None
File protection: System:RWED, Owner:RWED,
Group:RE, World:
```

```

Access Cntrl List: None

Total of 1 file, 7/8 blocks.
$ dir/size xptlib.dat

Directory DISK01: [JOE.XPTTEST]

XPTLIB.DAT;1 7

Total of 1 file, 7 blocks.

```

- 1 OpenVMS RECORD FORMAT 属性は、固定レコードの種類と 512 バイトのレコード長を示しています。
- 2 RECORD ATTRIBUTES フィールドには、値 NONE を指定できます。

**注意:**

このフィールドに値 **CARRIAGE RETURN CARRIAGE CONTROL** が指定されている場合は、ファイルが破損します。破損を防ぐには、移送ファイルを転送する前に RECORD ATTRIBUTES フィールドからこの値を削除してください。破損したファイルを転送しようとした後に、この状態について警告するエラーメッセージが表示されます。

---

## OpenVMS のファイル属性

FILENAME ステートメントで、FTP または FTP アクセス方式オプションのいずれか適切な方式を使用して、移送ファイルの属性を指定できます。FILENAME ステートメントの構文の詳細については、*SAS(R) 9.2 Companion for OpenVMS on HP Integrity Servers* を参照してください。ファイル属性の指定方法の例については、“例: ファイル転送での FILENAME ステートメントの使用” (41 ページ) を参照してください。

---

## OpenVMS でファイルの作成に使用された SAS バージョンを特定する

次の表に、OpenVMS システムで作成された、サポートされているファイルの種類をメンバおよび SAS バージョン別に示します。

表 7.1 メンバおよび SAS バージョン別 OpenVMS ファイル名拡張子

メンバの種類	SAS 6 のファイル名拡張子	SAS 8 以降のファイル名拡張子
SAS	.SAS	.SAS
PROGRAM(DATA ステップ)	.SASEB\$PROGRAM	.sas7bpgm
DATA	.SASEB\$DATA	.sas7bdat
INDEX	.SASEB\$INDEX	.sas7bndx
CATALOG	.SASEB\$CATALOG	.sas7beat

メンバの種類	SAS 6 のファイル名拡張子	SAS 8 以降のファイル名拡張子
MDDDB	.SASEB\$MDDDB	.sas7bmdb
PROC SQL ビュー	.SASEB\$VIEW	.sas7bview

CONTENTS プロシジャを使用して、データに関する情報を表示することもできます。作成に使用されたメンバとエンジンを特定する、一般的な PROC CONTENTS 出力の一部を次に示します。

```
The SAS System
The CONTENTS Procedure
Data Set Name: TEST.RECORDS
Member Type: DATA
Engine: V9
```

この出力では、データセット TEST.RECORDS は種類が DATA のメンバであり、V9 エンジンで作成されたことが示されています。

---

## OpenVMS へのテープデバイスのマウント

移送ファイルを移送元コンピュータでディスクからテープに移動し、移送先コンピュータでテープからディスクに移動するには、SAS セッションを開始する前に次の DCL コマンドを指定してテープデバイスを割り当てます。

注: INITIALIZE コマンドは新しいテープがある場合にのみ使用します。INITIALIZE コマンドは、テープ内にあるすべてのファイルを破棄します。

```
$ DEFINE TRANFILE tape-name
$ ALLOCATE TRANFILE
$ INITIALIZE TRANFILE DUMMY
$ MOUNT/FOREIGN/BLOCKSIZE=8000 TRANFILE
```

注: DCL コマンドの TRANFILE は、移送ファイルの場所を示すライブラリ参照名と同一です。

---

## OpenVMS のエラーメッセージ

### 移送ファイルが損傷しています

このエラーメッセージが表示された場合の一般的な回復操作については、“[適切なでない移送ファイル](#)” (73 ページ) を参照してください。

移送ファイルが破損している可能性があります。

1. 移送ファイルに破損している文字が含まれているかどうかを調べます。

```
$DIR/FULL transport-file
```

この出力で、移送ファイルに破損している文字が含まれていることを確認します。

```
Record attributes: Carriage return Carriage control
```

2. この後の操作は、環境に適用される次の条件によって異なります。

- a. 動作環境に NFTCOPY(Network File Transfer Copy)コマンドがあり、移送ファイルを DOS 移送先コンピュータに移動する場合、キャリッジリターン(CC)属性を移送ファイルから削除し、その移送ファイルを再度移送先コンピュータに移動します。

```
NFTCOPY/IMAGE/FIXED/CC=NONE NODE"userid password"
::disk:[dir] tranfile target
```

次に、例を示します。

```
NFTCOPY/IMAGE/FIXED/CC=NONE CHEX "brown bird":
dua0[brown]tranfile c:\blue\target
```

- b. 移送元コンピュータがメンテナンスレベル TS405 以降で SAS 6.08 を実行している場合、LIBNAME または FILENAMENONE ステートメントのいずれかの適切な方法で CC=オプションに NONE 値を設定します。

注: SAS リリースとメンテナンスレベルを確認するには、SAS ログの上部を参照してください。

次に例を示します。

```
libname grades 'file-path';
libname tranfile xport 'file-path' cc=none;
proc copy in=grades out=tranfile;
run;
```

- c. メンテナンスレベル TS405 で SAS 6.08 より前の SAS リリースを実行している場合、移送ファイルを後処理してキャリッジリターンを削除する必要があります。

REMCC.FDL という名前の新しいファイルを作成し、CARRIAGE\_CONTROL NONE を含む次のエントリを挿入します。

```
RECORD
BLOCK_SPAN YES
CARRIAGE_CONTROL NONE
FORMAT FIXED
SIZE 80
```

次の DCL コマンドを指定して、NEWTRAN.SEQ という名前の新しいファイルを作成します。

```
$ CONVERT/FDL=REMCC.FDL TRAN.SEQ NEWTRAN.SEQ
$ DELETE TRAN.SEQ
```

新しい移送ファイルのファイル属性にキャリッジリターンが含まれていないことを確認します。

```
$ DIR/FULL NEWTRAN.SEQ
```

3. 移送元コンピュータで、移送ファイルを再度移送先コンピュータに転送します。

移送ファイルに正しい属性が設定されていてもインポートできない場合、再ブロック化プログラムを使用して再試行してください。詳細については、「[移送ファイルの再ブロック化](#)」(80 ページ)を参照してください。



## ファイルに使用できないメンバまたはライブラリ

移送ファイルが破損している可能性があります。回復操作については、“[移送ファイルが損傷しています](#)” (49 ページ) を参照してください。

## レコードが切り捨てられました

このエラーメッセージが表示された場合の一般的な回復操作については、“[レコードが切り捨てられました](#)” (78 ページ) を参照してください。

通常、このメッセージは DOS、Macintosh または UNIX などの動作環境で、移送ファイルが仮想ディスクまたは共有ディスクに移動されたときに表示されます。多くの場合、仮想ディスクまたは共有ディスクのディレクトリのレコード形式は FIXED ではなく STREAM となっています。

回復するには、次の操作を実行します。

1. DIR/FULL コマンドを使用して移送ファイルの属性を確認します。
2. レコード属性を正しく設定するには、FIXREC.FDL という名前の新しいファイルを作成して次のエントリを挿入します。

```
RECORD
BLOCK_SPAN YES
CARRIAGE_CONTROL NONE
FORMAT FIXED
SIZE 80
```

3. 次の DCL コマンドを指定して、NEWTRAN.FDL という名前の新しいファイルを作成します。

```
$ EXCHANGE/NETWORK/TRANSFER_MOD=BLOCK/FDL=TRAN.FDL
TRAN.SEQ NEWTRAN.SEQ
```

4. 新しい移送ファイルの属性にキャリッジリターンが含まれていないことを確認します。

```
$ DIR/FULL NEWTRAN.SEQ
```

5. 移送元コンピュータで、移送ファイルを再度移送先コンピュータに転送します。

## データ取得時の内部エラー

移送ファイルが破損している可能性があります。回復操作については、“[レコードが切り捨てられました](#)” (51 ページ) を参照してください。



## 8 章

## z/OS 動作環境

---

z/OS ファイル属性の表示 .....	53
z/OS でファイルの作成に使用された SAS バージョンを特定する .....	53
z/OS ファイルと UNIX System Services Directory .....	54
ファイル移送に使用する z/OS バッチステートメント .....	54
z/OS 移送先コンピュータでの転送の問題 .....	54
レコード長 .....	54
例: FTP と移送先の z/OS コンピュータ .....	54
Windows Attachmate と移送先の z/OS コンピュータ .....	55
z/OS 動作環境での移送ファイルの読み込み .....	55
z/OS は ASCII 移送ファイルを読み込みません .....	55
例: 移送ファイルの一部の EBCDIC への変換 .....	55
例: 16 進形式での移送ファイルの一部の読み込み .....	56

---

**z/OS ファイル属性の表示**

z/OS 動作環境で作成されたファイルの属性を一覧表示するには、TSO で次のコマンドを発行します。

```
listd 'file-name'
```

このコマンドの出力例を次に示します。

```
The transport file should have the following attributes:
RECFM: FB
LRECL: 80
BLKSIZE: 8000
DSORG: PS
```

---

**z/OS でファイルの作成に使用された SAS バージョンを特定する**

データに関する情報を表示するには、CONTENTS プロシジャを使用します。

作成に使用されたメンバとエンジンを特定する、一般的な PROC CONTENTS 出力の一部を次に示します。

```
The SAS System
The CONTENTS Procedure
Data Set Name: TEST.CONTENTS
Member Type: DATA
Engine: V9
```

この出力は、データセット TEST.CONTENTS のメンバの種類が DATA で、V9 エンジンを使用して作成されたことを示しています。

## z/OS ファイルと UNIX System Services Directory

SAS 8 では、z/OS 動作環境のファイル編成の連結ライブラリ方式の代わりにして、UNIX System Services ディレクトリが導入されました。CEDA の機能は、UNIX System Services ディレクトリを使用する z/OS 動作環境でファイルを作成するために使用できます。CEDA の詳細については、“CEDA(クロス環境データアクセス)” (11 ページ) を参照してください。

## ファイル移送に使用する z/OS バッチステートメント

SAS バッチジョブを使用して、移送ファイルを作成できます。この例については、“例: z/OS JCL Batch から UNIX へのファイル移送” (98 ページ) を参照してください。JCL ステートメントの詳細については、z/OS 版 SAS を参照してください。

## z/OS 移送先コンピュータでの転送の問題

### レコード長

一部の例では、移送先の z/OS コンピュータに転送する移送ファイルのファイル形式が正しくても、レコード長が正しくない場合があります。この問題が発生した場合の回復操作については、“移送ファイルが破損していないことを確認する” (70 ページ) を参照してください。

### 例: FTP と移送先の z/OS コンピュータ

移送元コンピュータから移送ファイルを取得する移送先の z/OS コンピュータの FTP の例を次に示します。

```
> ftp
> open source-host
> binary
> locsite recfm=fb blksize=8000 lrecl=80
> get xportout target
> close
> quit
```

移送先の z/OS コンピュータに移送ファイルを配置する移送元コンピュータの FTP の例を次に示します。

```
> ftp
> open target-host
> binary 80
> quote site recfm=fb blksize=8000 lrecl=80
> put xportout target
> close
> quit
```

注: Windows や UNIX などのディレクトリベースの動作環境に移送ファイルを転送する場合は、ファイル属性の宣言に FTP の QUOTE SITE コマンドまたは LOCSITE コマンドは使用しないでください。

### Windows Attachmate と移送先の z/OS コンピュータ

Extra for Windows を使用する場合は、変換に NONE を指定し、ファイル転送ダイアログボックスに次の情報が表示されることを確認します。

```
send a:grades xportout lrecl(80) blksize(8000)
recfm(f) space(10,10)
```

詳細については、動作環境のドキュメントを参照してください。

---

## z/OS 動作環境での移送ファイルの読み込み

### z/OS は ASCII 移送ファイルを読み込めません

移送形式には、ASCII エンコーディングが使用されます。これは z/OS 動作環境には外部形式となります。この非互換性により、z/OS 動作環境のテキストエディタでは移送ファイルを正しく読み込むことができません。

### 例: 移送ファイルの一部の EBCDIC への変換

次の SAS コードを使用すると、z/OS 動作環境で移送ファイルの最初の数行を読み込むことができます。

注: このプログラムでは、ファイルは EBCDIC には変換されません。ファイルの最初の 5 つのレコードのみが解釈され、SAS ログに書き込まれます。移送ファイルは同じ状態のままです。

#### 例のコード 8.1 移送ファイルのヘッダーを解釈するコード

```
//PEEK JOB (,X101), 'SMITH,B.', TIME=(,3)
/*JOBPARM FETCH
//STEP1 EXEC SAS
//transport-file DD
DSN=USERID.XPT6.FILE, DISP=SHR
//SYSIN DD *
data _null_;
infile tranfile obs=5;
input theline $ascii80.;
put theline;
```

```
run;
/*
```

ログ出力は、移送ファイルの作成に XPORT エンジンまたは PROC CPORT が使用されたことを示しています。

この SAS コードは、XPORT エンジンで作成される移送ファイルの最初の 40 文字を示しています。

```
HEADER RECORD*****LIBRARY HEADER RECORD!!!!!!00
```

この SAS コードは、PROC CPORT で作成される移送ファイルの最初の 40 文字を示しています。

```
**COMPRESSED** **COMPRESSED** **COMPRESSED** **COM
```

注: CPORT プロシジャで NOCOMPRESS オプションを指定すると、圧縮が抑制され、移送ファイルに前述のテキストは表示されません。

データセットに使用される移送形式の技術的な詳細については、テクニカルサポート記事 TS-140 The Record Layout of a SAS Transport Data Set を参照してください。

### 例: 16 進形式での移送ファイルの一部の読み込み

ISPF を使用して、16 進形式の移送ファイルを参照できます。または、次の SAS コードを使用して、16 進形式の移送ファイルの最初の 20 個の 80 バイトレコードを表示できます。

```
data _null_;
infile 'transport-file';
input;
list;
put '-----';
if _n_ > 20 then stop;
run;
```

この SAS コードは、XPORT エンジンで作成される移送ファイルの最初の ASCII 文字 40 個の 16 進表現を示しています。

```
484541444552205245434F52442A2A2A2A2A2A
4C5920484541444552205245434F524421212121
```

この SAS コードは、PROC CPORT で作成される移送ファイルの最初の ASCII 文字 40 個の 16 進表現を示しています。

```
2A2A434F4D505245535345442A2A202A2A434F4D
50442A2A202A2A434F4D505245535345442A2A20
```

## 9 章

## Windows 動作環境

---

Windows のファイル属性 .....	57
Windows でファイルの作成に使用された SAS バージョンを特定する .....	57
エラーメッセージ: 暗号化したデータが無効です .....	58

---

## Windows のファイル属性

ファイル属性を適用するには、用途に応じて FTP を使用するか、FILENAME ステートメントで FTP アクセスメソッドオプションを使用します。FILENAME ステートメントの構文の詳細については、*SAS ステートメント: リファレンス*を参照してください。属性を使用する FILENAME ステートメントの例については、“[例: ファイル転送での FILENAME ステートメントの使用](#)” (41 ページ) を参照してください。

## Windows でファイルの作成に使用された SAS バージョンを特定する

次の表は、Windows 動作環境で作成されるファイルについて、サポート対象のファイルの種類をメンバ別およびバージョン別に示しています。

表 9.1 メンバ別、SAS バージョン別の Windows ファイル名拡張子

メンバの種類	SAS 6 のファイル名拡張子	SAS 8 以降のファイル名拡張子
SAS	.sas	.sas
PROGRAM(DATA ステップ)	.ss2	.sas7bpgm
DATA	.sd2	.sas7bdat
INDEX	.si2	.sas7bndx
CATALOG	.sc2	.sas7bcats
MDDDB	.sm2	.sas7bmdb

メンバの種類	SAS 6 のファイル名拡張子	SAS 8 以降のファイル名拡張子
PROC SQL ビュー	.sv2	.sas7bview

SAS 9 と SAS 8 のファイル名拡張子は同一です。

CONTENTS プロシジャを使用して、データに関する情報を表示することもできます。

作成に使用されたメンバとエンジンを特定する、一般的な PROC CONTENTS 出力の一部を次に示します。

```
The SAS System
The CONTENTS Procedure
Data Set Name: TEST.CONTENTENTS
Member Type: DATA
Engine: V9
```

この出力は、データセット TEST.CONTENTENTS は種類が DATA のメンバで、V9 エンジンによって作成されたことを示しています。

---

## エラーメッセージ: 暗号化したデータが無効です

通常、このメッセージが表示されるのは、PROC CPORT および PROC CIMPORT を使ってファイル名拡張子を変更されたファイルを移動するときです。たとえば、ディレクトリ内の少なくとも 1 つのファイル名の拡張子が、そのファイルを作成するのに使用した SAS バージョンに一致しない拡張子に置き換えられた場合です。DOS の `rename` コマンドや Windows のファイルマネージャでファイル名拡張子を変更された可能性が考えられます。SAS バージョン別の有効な Windows ファイル名拡張子のリストについては、“SAS ファイル拡張子とファイルヘッダー” (63 ページ) を参照してください。

疑わしいファイル名拡張子を確認するには、次のコマンド構文を使用します。

```
type filename.extension
```

`more` コマンドを使ってパイプ出力もできます。

次に、例を示します。

```
type xportout.sd2 | more
```

SAS 9 データセット `xportout` のファイル名拡張子 `.sas7bdat` が誤って `.sd2` に変更された疑いがあるとします。

注: SAS 9 と SAS 8 のファイル名拡張子は同一です。

次のように出力されます。

```
SAS 9.00 WIN 6.09
```

右側の列から、SAS 6.09 に適するファイル名拡張子が誤って SAS 9 ファイルに適用されていることがわかります。この問題を修正するには、DOS の `rename` コマンドか Windows のファイルマネージャを使用し、ファイル名の拡張子に `.sas7bdat` に再適用する必要があります。



## 10 章

## UNIX 動作環境

---

UNIX のファイル属性 .....	59
UNIX でファイルの作成に使用された SAS バージョンを特定する .....	59
例: テープでの移送ファイルの作成 .....	61
例: UNIX 移送元コンピュータでの移送ファイルのコピー(ディスクからテープ) .....	61
例: 移送先コンピュータでの移送ファイルのコピー(テープからディスク) .....	61

---

## UNIX のファイル属性

FILENAME ステートメントで、FTP または FTP アクセス方式オプションのいずれか適切な方式を使用して、移送ファイルの属性を指定できます。FILENAME ステートメントの構文の詳細については、*UNIX 版 SAS* を参照してください。属性を使用する FILENAME ステートメントの例については、“[非ネイティブファイルや移送ファイルへの FILENAME ステートメントまたは FTP の使用](#)” (41 ページ) を参照してください。FTP の例については、“[例: ファイル転送での FILENAME ステートメントの使用](#)” (41 ページ) を参照してください。

---

**UNIX でファイルの作成に使用された SAS バージョンを特定する**

次の表は、UNIX 動作環境で作成されるファイルについて、サポート対象のファイルの種類をメンバ別およびバージョン別に示しています。

表 10.1 メンバ別、SAS バージョン別の UNIX ファイル名拡張子

メンバの種類	SAS 6 のファイル名拡張子	SAS 8 以降のファイル名拡張子
SAS	.sas	.sas
PROGRAM(DATA ステップ)	.sspnn	.sas7bpgm
DATA	.ssdnn	.sas7bdat

メンバの種類	SAS 6 のファイル名拡張子	SAS 8 以降のファイル名拡張子
INDEX	.snxnn	.sas7bndx
CATALOG	.sctnn	.sas7bcac
MDDDB	.ssmnn	.sas7bmdb
PROC SQL ビュー	.snvnn	.sas7bview

この表で、nn は UNIX コンピュータのアーキテクチャを区別するための拡張子です。拡張子と UNIX 動作環境グループの対応は次のとおりです。

表 10.2 UNIX 動作環境のファイル名拡張子

SAS ファイル名拡張子 nn	UNIX 動作環境グループ	SAS リリース別サポート			
		6.09	6.10	6.11	6.12
01	HP-UX	はい	使用不可	はい	はい
	Sun	はい	使用不可	はい	はい
	Solaris	はい	使用不可	はい	はい
	AIX	はい	使用不可	はい	はい
	MIPS ABI	使用不可	はい	はい	使用不可
02	ULTRIX	はい	使用不可	使用不可	使用不可
	INTEL-ABI	はい	使用不可	はい	はい
04	COMPAQ Digital UNIX	使用不可	はい	はい	はい

SAS 9 と SAS 8 のファイル名拡張子は同一です。

HP-UX、Sun、Solaris、AIX、MIPS の動作環境間ではデータセットを相互に交換可能なため、これらの動作環境間の移動に移送ファイルを作成する必要はありません。カタログも、AIX、HP-UX、Sun、Solaris、MIPS の動作環境間では相互に交換可能です。

CONTENTS プロシジャを使用して、データに関する情報を表示することもできます。

作成に使用されたメンバとエンジンを特定する、一般的な PROC CONTENTS 出力の一部を次に示します。

```
The SAS System
The CONTENTS Procedure
Data Set Name: TEST.RECORDS
Member Type: DATA
Engine: V9
```

この出力は、データセット TEST.RECORDS は種類が DATA のメンバで、V9 エンジンによって作成されたことを示しています。

---

## 例: テープでの移送ファイルの作成

テープで移送ファイルを作成するには、移送元コンピュータで LIBNAME ステートメントまたは FILENAME ステートメントを用途に応じて使用し、ファイルパスにテープデバイスを指定します。次に、例を示します。

```
libname tranfile xport '/dev/tape1';  
filename tranfile '/dev/tape1';
```

---

## 例: UNIX 移送元コンピュータでの移送ファイルのコピー (ディスクからテープ)

移送元コンピュータで移送ファイルをディスクからテープにコピーするには、UNIX の dd コマンドを発行します。次に、例を示します。

```
dd if=tranfile of=/dev/tape1 bs=8000
```

**dd**

指定した入力ファイルを指定した出力デバイスにコピーします。

**if=tranfile**

入力ファイル(移送ファイル)を指定します。

**of=/dev/tape1**

出力ファイル(テープデバイス)を指定します。

**bs=8000**

入力ファイルと出力ファイルのブロックサイズを 8000 に指定します。

詳細については、UNIX の dd(1)マニュアルページを参照してください。

---

## 例: 移送先コンピュータでの移送ファイルのコピー(テープ からディスク)

移送先コンピュータで移送ファイルをテープからディスクにコピーするには、UNIX の dd コマンドを発行します。次に、例を示します。

```
dd if=/dev/tape1 of=tranfile bs=8000
```

**dd**

指定した入力ファイルを指定した出力デバイスにコピーします。

**if=/dev/tape1**

入力ファイル(テープデバイス)を指定します。

**of=tranfile**

出力ファイルを指定します。

**bs=8000**

入力ファイルと出力ファイルのブロックサイズを 8000 に指定します。

詳細については、UNIX の dd(1)マニュアルページを参照してください。



## 11 章

## SAS ファイル拡張子とファイルヘッダー

ファイル名拡張子: SAS ファイルの作成に使用された SAS エンジンと動作環境を特定する .....	63
PROC CONTENTS: SAS ファイルの作成に使用された Base SAS エンジン进行特定する .....	64
ファイルヘッダー: 移送ファイルの作成に使用された方法を調べる .....	65

## ファイル名拡張子: SAS ファイルの作成に使用された SAS エンジンと動作環境を特定する

SAS ファイルの作成に使用された SAS エンジンと動作環境を SAS ファイル名拡張子から推測することができます。SAS 6 以降の動作環境では、SAS エンジンと作成される SAS メンバを反映するファイル名拡張子が使用されます。

- OpenVMS
- z/OS(SAS 8 以降の UNIX System Services ディレクトリ)
- UNIX
- Windows

次の表に、メンバの SAS ファイル名拡張子を動作環境および SAS バージョン別に示します。

表 11.1 動作環境の種類および SAS バージョン別の SAS ファイル名拡張子

メンバ	SAS ファイル名拡張子			
	SAS 6			SAS 8 以降
	UNIX	OpenVMS	Windows	UNIX、OpenVMS、z/OS*、Windows
.SAS	.sas	.SAS	.sas	.sas
PROGRAM (DATA ステップ)	.sspnn	.SASEB\$PROGRAM	.ss2	sas7bpgm
DATA	.ssdnn	.SASEB\$DATA	.sd2	.sas7bdat

メンバ	SAS ファイル名拡張子			
	SAS 6			SAS 8 以降
	UNIX	OpenVMS	Windows	UNIX、OpenVMS、z/OS*、Windows
INDEX	.snxnn	.SASEB\$INDEX	.si2	.sas7bndx
CATALOG	.sctnn	.SASEB\$CATALOG	.sc2	.sas7bcat
MDDDB	.ssmnn	.SASEB\$MDDDB	.sm2	.sas7bmdb
PROC SQL ビュー	.snvnn	SASEB\$VIEW	.sv2	.sas7bview
ITEMSTOR	n/a	n/a	n/a	.sas7bitm

UNIX コンピュータアーキテクチャを区別するために拡張子 *nn* が使用されます。UNIX 動作環境の *nn* の値の詳細については、表 10.2 (60 ページ)を参照してください。

\*SAS 8 以降の z/OS UNIX System Services ディレクトリを指します。

---

## PROC CONTENTS: SAS ファイルの作成に使用された Base SAS エンジン特定する

SAS 6 以降を使用するすべての動作環境で CONTENTS プロシジャを使用し、SAS ファイルの作成に使用された Base SAS エンジン特定できます。

注: z/OS 動作環境ではファイル名拡張子が使用されないため、SAS ファイルの作成に使用された Base SAS エンジン特定するには、PROC CONTENTS を使用する必要があります。

z/OS 環境のデータセットに PROC CONTENTS を使用する例を次に示します。

```
proc contents data=test.records;
run;
```

次は、出力の一部です。

```
The SAS System
The CONTENTS Procedure
Data Set Name: TEST.RECORDS
Member Type: DATA
Engine: V9
```

この出力は、データセット RECORDS のメンバの種類が DATA で、V9 エンジンを使用して作成されたことを示しています。

PROC CONTENTS を使用して、データセットの動作環境の形式が、アクセスする動作環境に対して外部形式なのかネイティブ形式なのかを調べることもできます。詳細については、“SAS ファイル形式の特定” (16 ページ) を参照してください。

---

## ファイルヘッダー: 移送ファイルの作成に使用された方法を調べる

移送ファイルがどの方法(PROC COPY または PROC CPORT および PROC CIMPORT で XPORT エンジンを使用)で作成されたのかを調べる方法は、動作環境によって異なります。

- 文字データを ASCII 形式で保存する動作環境では、テキストエディタを使用するか、動作環境の読み込みまたは表示コマンドを使用して、ファイルを読み込みます。

XPORT エンジンでは、次の ASCII テキストを最初の 40 文字に含むファイルが作成されます。

```
HEADER RECORD*****LIBRARY HEADER RECORD!!!!!!!!00
```

PROC CPORT では、次の ASCII テキストを最初の 40 文字に含むファイルが作成されます。

```
**COMPRESSED** **COMPRESSED** **COMPRESSED** **COM
```

注: PROC CPORT で NOCOMPRESS オプションを指定すると、圧縮が抑制され、移送ファイルに前述のテキストは表示されません。

データセットに使用される移送形式の技術的な詳細については、テクニカルサポート記事 TS-140 The Record Layout of a SAS Transport Data Set を参照してください。

- 移送形式には ASCII エンコーディングが使用されるため、z/OS などの非 ASCII 動作環境の場合はテキストエディタで移送ファイルを読み込むことができません。詳細については、“[z/OS 動作環境での移送ファイルの読み込み](#)” (55 ページ) を参照してください。





## 5 部

---

# トラブルシューティング

12 章	
問題の回避と修正	69



## 12 章

## 問題の回避と修正

<b>トラブルシューティング: 移送ファイルの転送と復元</b> .....	<b>70</b>
トラブルシューティングチェックリスト .....	70
移送ファイル(バイナリ形式)の転送 .....	70
移送ファイルが破損していないことを確認する .....	70
通信ソフトウェアによりファイル属性が変更されていないことを確認する .....	71
移送先コンピュータで通信ソフトウェアを起動する .....	71
移送元コンピュータと移送先コンピュータで互換性のある移送方法を使用する .....	71
移送ファイルの一貫性の検証 .....	72
ラベルなしテープの使用 .....	72
大きな移送ファイルをテープ用に小さなファイルに分割する .....	72
<b>移送ファイルに関するエラーメッセージと警告メッセージ</b> .....	<b>73</b>
適切でない移送ファイル .....	73
XPORT エンジンでは、カタログファイルを開く機能はサポートされていません .....	74
CATALOG=の代わりに、DATA=か LIBRARY=を指定してください .....	74
filename は SAS ファイルではありません .....	74
エントリの種類 catalog-entry-type は CPORT ではサポートされていません .....	75
エントリの種類 catalog-entry-type は以前のリリースとの互換性がありません .....	75
ファイル library.member.DATA のメンバ名は XPORT エンジンでは長すぎます .....	75
ファイル library.member.DATA のメンバ名は V6 エンジンでは長すぎます .....	75
ファイル libref.ALL は損傷しています。I/O 処理は完了していません .....	75
移送ファイルが損傷しています .....	76
データ取得時の内部エラー .....	76
無効なデータ長 .....	76
ファイル filename★で使用できないメンバまたはライブラリ .....	77
入力ファイルにはもっと多くのライブラリメンバが存在します。PROC ステートメントに LIBRARY=libref パラメータを指定して、すべてのメンバを変換してください .....	77
SQL プロシジャでは、V9 のビューを V6 ライブラリに保存しません .....	77
指定した機能をサポートしていません .....	78
レコードが切り捨てられました .....	78
libref.member-name は、異なるオペレーティングシステム向けに作成されているため更新できません .....	78
XPORT エンジンでは、UTILITY FILE OPEN 機能はサポートされていません .....	78
値 y コードは有効な SAS 名ではありません。エラーのためデータセットをスキップします .....	78
変数名 variable は Version-6-data-set では不正です .....	79
<b>転送形式と移送ファイルの属性の検証</b> .....	<b>79</b>
<b>移送ファイルの再ブロック化</b> .....	<b>80</b>

---

## トラブルシューティング: 移送ファイルの転送と復元

### トラブルシューティングチェックリスト

移送ファイルを移送先コンピュータに転送するときの潜在的な問題を回避するには、次の条件を満たしている必要があります。

- ネットワークを介して転送する場合、移送ファイルがバイナリ形式で転送されることを確認します。詳細については、“[移送ファイル\(バイナリ形式\)の転送](#)”(70 ページ)を参照してください。
- 移送ファイルが破損していないことを確認します。詳細については、“[移送ファイルが破損していないことを確認する](#)”(70 ページ)を参照してください。
- 通信ソフトウェアによりファイルの属性が変更されないことを確認します。詳細については、“[通信ソフトウェアによりファイル属性が変更されていないことを確認する](#)”(71 ページ)を参照してください。
- 移送先コンピュータで通信ソフトウェアを起動し、移送ファイルを移送元コンピュータから取得することを検討します。詳細については、“[移送先コンピュータで通信ソフトウェアを起動する](#)”(71 ページ)を参照してください。
- 移送元コンピュータで移送ファイルを作成して移送先コンピュータでその移送ファイルを復元する場合、異なる方法を併用しないでください。詳細については、“[移送元コンピュータと移送先コンピュータで互換性のある移送方法を使用する](#)”(71 ページ)を参照してください。
- 移送ファイルを移送先コンピュータに転送する前に、作成元の移送元コンピュータでその移送ファイルを復元し、ファイルの一貫性を検証します。詳細については、“[移送ファイルの一貫性の検証](#)”(72 ページ)を参照してください。
- テープを使用して転送する場合、ラベル付けされていないテープを使用します。詳細については、“[ラベルなしテープの使用](#)”(72 ページ)を参照してください。
- テープを使用して大きい移送ファイルを転送する場合、ライブラリを複数のライブラリに分割して各ライブラリをテープに移送します。詳細については、“[大きな移送ファイルをテープ用に小さなファイルに分割する](#)”(72 ページ)を参照してください。

### 移送ファイル(バイナリ形式)の転送

通信ソフトウェアを使用して移送ファイルを転送する場合、バイナリ(またはイメージ)形式でファイルが転送されることを確認します。ファイルの内容は変更せずに、順次バイト単位で転送する必要があります。

FTP を使用して移送ファイルを移送先コンピュータに移動する場合、ファイルを転送する前に BINARY 80 を指定する必要があります。

PATHWORKS を使用する場合、PCSA\_MANAGER を使用して file\_server サービスを設定するときに SEQUENTIAL\_FIXED 属性を使用します。デフォルトの属性は STREAM で、これは移送ファイルの移動には適していません。

### 移送ファイルが破損していないことを確認する

移送先コンピュータへの転送時に、移送ファイルのレコードの終了を表すキャリッジリターンが通信ソフトウェアによって挿入されないことを確認します。キャリッジリターン

やラインフィードが挿入されていると移送ファイルが破損し、移送先コンピュータでそのファイルを復元できなくなります。この状態の特定方法の詳細については、“[ファイル libref.ALL は損傷しています。I/O 処理は完了していません](#)” (75 ページ) の回復操作を参照してください。

### 通信ソフトウェアによりファイル属性が変更されていないことを確認する

通信ソフトウェアによりファイル属性が変更されないことを確認します。必須の属性と値を次に示します。

論理レコード長(LRECL)	80 または 80 の倍数の整数(たとえば、160、240、320)
ブロックサイズ(BLKSIZE)	8000 ブロック
レコード形式(RECFM)	固定ブロック

これらの属性の制御の詳細については、通信ソフトウェアのドキュメントを参照してください。

移送先コンピュータで、破損していない(キャリッジリターンまたはラインフィードが挿入されていない)移送ファイルがあり、そのレコードブロックサイズが不正なためブロック化された移送ファイルを正常に取得できない場合、再ブロック化プログラムを実行してブロックを正しいサイズに修正できます。詳細については、“[移送ファイルの再ブロック化](#)” (80 ページ) を参照してください。

### 移送先コンピュータで通信ソフトウェアを起動する

移送ファイルを移送先コンピュータに転送するには、通信ソフトウェアを移送元コンピュータではなく移送先コンピュータで起動した方が成功する可能性が高くなります。多くの場合、移送先コンピュータでは書き込み権限がないためファイルを配置できません。UNIX から z/OS に移送ファイルを転送する場合、z/OS コンピュータで通信ソフトウェアを起動することをお勧めします。通常は UNIX コンピュータでの読み込み権限があるため、その移送ファイルを取得して z/OS コンピュータに書き込むことができます。

### 移送元コンピュータと移送先コンピュータで互換性のある移送方法を使用する

移送元コンピュータで移送ファイルを作成して移送先コンピュータでその移送ファイルを復元する場合、異なる方法を併用しないでください。使用する方法は同一であるか、一組のペアである必要があります。たとえば、移送元コンピュータと移送先コンピュータの両方で、XPORT エンジンと PROC COPY を使用して移送ファイルを作成および復元します。または、移送元コンピュータで PROC CPORT を使用して移送ファイルを作成し、移送先コンピュータで PROC CIMPORT を使用してその移送ファイルをインポートすることもできます。移送元コンピュータで XPORT エンジンと PROC COPY を使用して移送ファイルを作成してから、移送先コンピュータで PROC CIMPORT を使用してその移送ファイルを復元しないでください。

移送ファイルの作成に使用された方法を特定するには、テキストエディタまたは動作環境の読み込みコマンドか表示コマンドを使用して、文字データを ASCII として表現するコンピュータで SAS 9 のファイルを読み込みます。

注: EBCDIC として文字データを表現する動作環境で移送ファイルを表示する方法の詳細については、“[z/OS 動作環境での移送ファイルの読み込み](#)” (55 ページ) を参照してください。

XPORT エンジン、最初の行に次の ASCII テキストが含まれるファイルを作成します。

```
HEADER RECORD*****LIBRARY HEADER RECORD!!!!!!00
```

PROC CPORT は、最初の行に次のテキストが含まれるファイルを作成します。

```
**COMPRESSED** **COMPRESSED** **COMPRESSED**
```

注: PROC CPORT で NOCOMPRESS オプションを指定すると、圧縮が抑制され、移送ファイルに前述のテキストは表示されません。

### 移送ファイルの一貫性の検証

移送ファイルを移送先コンピュータに転送する前にその一貫性を検証するには、適切な方法を使用して移送元コンピュータでファイルをネイティブ形式に戻して読み込んでみてください。

PROC COPY の例を次に示します。

```
/* This PROC COPY creates the transport file TRAN. */
libname tran xport 'transport-file';
libname grades 'SAS-data-library';
proc copy in=grades out=tran memtype=data;
run;
/* This PROC COPY reads back transport file TRAN. */
libname grades 'SAS-data-library';
libname tran xport 'transport-file';
proc copy in=tran out=test;
run;
```

PROC CPORT と PROC CIMPORT の例を次に示します。

```
/* This PROC CPORT creates the transport file. */
libname grades 'SAS-data-library';
filename tran 'transport-file';
proc cport library=grades file=tran;
run;
/* This PROC CIMPORT reads back the transport file. */
filename tran 'transport-file';
libname grades 'SAS-data-library';
proc cimport library=grades infile=tran;
run;
```

両方の例で、ログのエラーメッセージを確認します。

### ラベルなしテープの使用

テープを使用して移送ファイルを転送する場合、ラベル付けされていないテープを使用します。テープのラベルの処理はコンピュータによって異なるため、移送先コンピュータで標準ラベル付きのテープからファイルを読み込むことは複雑になる可能性があります。

### 大きな移送ファイルをテープ用に小さなファイルに分割する

テープを使用して移送ファイルを転送するときその移送ファイルが 1 つのテープの容量を超える場合、元のライブラリを複数のライブラリに分割し、各ライブラリに対して

個別のラベル付けされていないテープを作成する必要があります。移送先コンピュータで元のライブラリを復元できます。

## 移送ファイルに関するエラーメッセージと警告メッセージ

### 適切でない移送ファイル

このメッセージは次のいずれかの条件下で表示されます。

- PROC CIMPORT を使用して、SAS 9 で作成された移送ファイルを SAS 6 が実行されているコンピュータに移動しようとした場合。移送元コンピュータの SAS 9 のセッションから移送先コンピュータの SAS 6 のセッションに移送ファイルを移動することはできません。
- ファイルが BINARY 以外の形式で移送された場合、または移送先コンピュータへの転送中に移送ファイルの属性が変更された場合。回復操作については、“[転送形式と移送ファイルの属性の検証](#)” (79 ページ) を参照してください。
- サイトでデフォルト以外の変換テーブルを使用している場合。カスタマイズされた変換テーブルは、TRANTAB=システムオプションで設定されます。このオプションの詳細については、*SAS システムオプション: リファレンス* を参照してください。TRANTAB=システムオプションの値を確認するには、次のステートメントをサブミットします。

```
proc options option=trantab;
run;
```

サイトでデフォルト以外の変換テーブルを使用していることが判明した場合、次のオプションを指定してオプションをデフォルト値に復元する必要があります。

```
options trantab=( );
```

次に、移送ファイルを再度作成して移送先コンピュータに転送し、移送先コンピュータでそのファイルをインポートします。

- 移送元コンピュータが SAS 6.12 を実行していて、ファイルをインポートする移送先コンピュータが SAS 6.08、6.09E または 6.10 を実行している場合。データセットの並べ替え機能(SORTEDBY=データセットオプションを使用して指定)は SAS 6.12 の CPORT プロシジャに含まれていますが、SAS 6.08 の CIMPORT プロシジャには含まれていません。

この問題を解決するには、次の操作のいずれかを実行します。

- SAS 6.12 の CPORT プロシジャで SORTINFO=オプションを使用して、並べ替え機能を無効にします。次に、例を示します。

```
proc cport data=grades.junior
file='xgrades.junior'
sortinfo=no;
```

- SAS 6.12 の CPORT プロシジャで V608 または V609 エンジンオプションを使用して、SAS 6.12 の並べ替え機能を無効にします。次に、例を示します。

```
proc cport data=grades.junior
file='xgrades.junior' v609;
```

SORTEDBY=データセットオプションの情報は、SAS 6.12 の PROC CPORT に含まれています。

### **XPORT エンジンでは、カタログファイルを開く機能はサポートされていません**

このメッセージは、PROC COPY と XPORT エンジンを使用してカタログまたはカタログエントリの移送ファイルを作成しようとした場合に表示されます。PROC CPORT を使用してカタログまたはカタログエントリの移送ファイルを作成し、移送先コンピュータで PROC CIMPORT を使用してそのファイルをインポートする必要があります。

### **CATALOG=の代わりに、DATA=か LIBRARY=を指定してください**

このメッセージは、移送元コンピュータで LIBRARY=出力先メンバを含む PROC CPORT が使用され、移送先コンピュータで PROC CIMPORT に CATALOG=出力先メンバが含まれている場合に表示されます。移送先コンピュータでは DATA=または LIBRARY=のいずれかのメンバの種類を使用する必要があります。次に、例を示します。

```
proc cport file=in libname=out;
proc cimport infile=in catalog=new;
```

PROC CPORT の LIBNAME=オプションでは種類が LIBRARY の出力先メンバを指定しているため、PROC CIMPORT では LIBNAME=または DATA=オプションを指定する必要があります。

インポートされたライブラリからカタログエントリの種類のみを選択するには、PROC CIMPORT で ET=オプションを指定します。カタログエントリの種類を除外するには、EET=オプションを使用します。次に、例を示します。

```
proc cimport infile=in library=new et=program memtype=catalog;
proc cimport infile=in library=new eet=program memtype=catalog;
```

最初の例では、種類が PROGRAM のカタログエントリのみがインポートされます。2番目の例では、種類が PROGRAM のカタログエントリのみが除外されます。MEMTYPE=CATALOG でインポートをカタログのみに制限しています。

### **filename は SAS ファイルではありません**

通常、このメッセージは、移送先コンピュータで CIMPORT プロシジャを使用してデータセットをインポートするときに表示されます。2つの可能性が考えられます。

- PROC CIMPORT を使用してインポートしようとしている移送ファイルが、XPORT エンジンと COPY プロシジャまたは DATA ステップを使用して作成されている可能性があります。ファイルの先頭を参照して移送ファイルが作成された方法を確認してください。XPORT エンジンで移送ファイルが作成された場合、ファイルの先頭には次の ASCII テキストが含まれています。

```
HEADER RECORD*****LIBRARY HEADER RECORD!!!!!!00
```

CPORT プロシジャで移送ファイルが作成された場合、ファイルの先頭には次の ASCII テキストが含まれています。

```
**COMPRESSED** **COMPRESSED** **COMPRESSED** **COM
```

注: PROC CPORT で NOCOMPRESS オプションを設定した場合、圧縮は行われず、移送ファイルに前述のテキストは表示されません。

互換性のない方法で作成された移送ファイルを復元する場合、正しい方法を使用してその移送ファイルを復元します。



- このメッセージは、サイトでデフォルト以外の変換テーブルを使用している場合に表示される可能性もあります。この問題の修復操作については、“[適切でない移送ファイル](#)” (73 ページ) を参照してください。

### エントリの種類 `catalog-entry-type` は `CPORT` ではサポートされていません

このメッセージは、異なる SAS リリースのコンピュータ間でこのカタログエントリの種類の移送はサポートされていないことを意味します。

モジュール自体からは定義を取得できないため、エントリの種類を定義した SAS ステートメント (IML モジュールなど) を移送先コンピュータに移動してからモジュールを再作成できます。

### エントリの種類 `catalog-entry-type` は以前のリリースとの互換性がありません

このメッセージは、PROC CPORT を使用して、カタログエントリを SAS 9 から SAS 6 に移動しようとした場合に表示されます。SAS 9 では、このカタログエントリの低位互換性はサポートされていません。

### ファイル `library.member.DATA` のメンバ名は `XPORT` エンジンでは長すぎます

このメッセージは、XPORT エンジンと PROC COPY を使用して、名前が 8 文字を超えるデータセットを SAS 9 が実行されている移送元コンピュータから SAS 6 のライブラリに移動した場合に表示されます。このようなメッセージの例を次に示します。

エラー: ファイル `OUT.THIS_IS_LONG_NAMED_DATA.DATA` のメンバ名は XPORT エンジンでは長すぎます。

メンバ名 `THIS_IS_LONG_NAMED_DATA` は、XPORT エンジンが導入されたバージョン 5 の機能セットで適用される 8 文字のメンバ名の制限を超えています。

長い変数名の自動的な切り捨てを有効にする `VALIDVARNAME` システムオプションと V6 の割り当て値は、メンバ名をサポートしていません。回復するには、そのメンバを 8 文字以下の名前別のメンバにコピーし、移送処理を再試行します。

### ファイル `library.member.DATA` のメンバ名は `V6` エンジンでは長すぎます

このメッセージは、PROC COPY を使用して、名前が 8 文字を超えるデータセットを SAS 9 が実行されている移送元コンピュータから SAS 6 のライブラリに移動した場合に表示されます。このようなメッセージの例を次に示します。

エラー: ファイル `V6LIBMYDATABASE.DATA` のメンバ名は V6 エンジンでは長すぎます。

SAS 9 のデータセット名 `MYDATABASE` は、SAS 6 でサポートされている最大 8 文字のメンバ名の制限を超えています。SAS 6 では、データセット名 `MYDATABASE` には最大 8 文字の制限を超える 10 文字が含まれていると解釈されます。

長い変数名の自動的な切り捨てを有効にする `VALIDVARNAME` システムオプションと V6 の割り当て値は、メンバ名をサポートしていません。回復するには、そのメンバを 8 文字以下の名前別のメンバ名に変更し、移送を再試行します。

### ファイル `libref.ALL` は損傷しています。I/O 処理は完了していません

通常、このメッセージはファイルが破損していることを表します。サイトの通信ソフトウェアによって移送ファイルにキャリッジリターンが挿入されている可能性があります。

移送先コンピュータでコンピュータ固有のユーティリティ(UNIX の 16 進ダンプユーティリティ `xd` など)を使用し、移送ファイルを 16 進形式で表示してキャリッジリターンが挿入されているかどうかを確認できます。詳細については、UNIX `xd(1)` マニュアルページを参照してください。別の例として、z/OS の場合は、SPF 1 コマンドを使用して参照し、データセットを選択してコマンドラインに `hex on` と入力します。

最初のレコードの最後の方にキャリッジリターン文字(0D)とラインフィード文字(0A)が含まれている、移送ファイルの例を次に示します。最後の行の先頭の 2 つが 0D と 0A の 16 進値になっています。

```
48 45 41 44 45 52 20 52 45 43 4F 52 44 2A 2A 2A HEADER RECORD***
2A 2A 2A 2A 4C 49 42 52 41 52 59 20 48 45 41 44 ***LIBRARY HEAD
45 52 20 52 45 43 4F 52 44 21 21 21 21 21 21 21 ER RECORD!!!!
30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 00000000 000000
30 30 30 30 30 30 30 30 30 30 30 30 30 30 20 20 00000000 0000
0D 0A 53 41 53 20 20 20 20 20 53 41 53 20 20 20 ...SAS SAS
```

キャリッジリターン文字またはラインフィード文字が見つからない場合、すぐには判別できない別の形式の破損が発生している可能性があります。この可能性をテストするには、移送先コンピュータで同じ種類のメンバから別の移送ファイルを作成し、その 16 進表現を表示します。作成した破損していないファイルと、復元しようとしている破損の可能性があるファイルの表示内容を見比べます。視覚的比較で、復元しようとしている移送ファイルが破損していることを確認できる場合があります。この場合、移送元コンピュータで移送ファイルを再作成して転送し、移送先コンピュータで復元します。

移送元コンピュータで、移送ファイルの属性にキャリッジリターンが含まれているかどうかを確認します。ファイルの属性の表示と修正の詳細については、現在の動作環境向けの章を参照してください。

移送元コンピュータで、移送ファイルを再度移送先コンピュータに転送します。

移送ファイルに正しいファイルの属性が設定されていても復元できない場合、“[移送ファイルの再ブロック化](#)” (80 ページ) の再ブロック化プログラムを使用してください。

### 移送ファイルが損傷しています

回復操作については、“[適切でない移送ファイル](#)” (73 ページ) を参照してください。

### データ取得時の内部エラー

通常、このメッセージはファイルが BINARY 以外の形式で移送された場合、または移送先コンピュータへの転送中に移送ファイルの属性が変更された場合に表示されません。

回復操作については、“[転送形式と移送ファイルの属性の検証](#)” (79 ページ) を参照してください。

### 無効なデータ長

通常、このメッセージはファイルが BINARY 以外の形式で移送された場合、または移送先コンピュータへの転送中に移送ファイルの属性が変更された場合に表示されません。

回復操作については、“[転送形式と移送ファイルの属性の検証](#)” (79 ページ) を参照してください。

## ファイル filename★で使用できないメンバまたはライブラリ

通常、このメッセージはファイルが BINARY 以外の形式で移送された場合、または移送先コンピュータへの転送中に移送ファイルの属性が変更された場合に表示されません。

回復操作については、“[転送形式と移送ファイルの属性の検証](#)” (79 ページ) を参照してください。

別の考えられる原因として、移送元コンピュータが SAS 6.12 のセッションで移送先コンピュータが SAS 6.08 のセッションである可能性があります。データセットの並べ替え機能 (SORTEDBY=データセットオプションを使用して指定) は SAS 6.12 の CPORT プロシジャに含まれていますが、SAS 6.08 の CIMPORT プロシジャには含まれていません。

この問題を解決するには、次の操作のいずれかを実行します。

- SAS 6.12 の CPORT プロシジャで SORTINFO=オプションを使用して、並べ替え機能を無効にします。次に、例を示します。

```
proc cport data=grades.jr file='tranfile.jr' sortinfo=no;
```

- SAS 6.12 の CPORT プロシジャで V608 エンジン指定して、SAS 6.12 の並べ替え機能を無効にします。次に、例を示します。

```
proc cport data=grades.jr file='tranfile.j' v608;
```

SORTEDBY=データセットオプションの情報は、SAS 6.12 の PROC CPORT に含まれています。

## 入力ファイルにはもっと多くのライブラリメンバが存在します。PROC ステートメントに LIBRARY=libref パラメータを指定して、すべてのメンバを交換してください

この警告メッセージは、移送元コンピュータで LIBRARY=出力先メンバを含む PROC CPORT が使用され、移送先コンピュータで PROC CIMPORT に DATA=出力先メンバが含まれている場合に表示されます。ただし、移送先コンピュータでは 1 つのデータセットのみのインポートに成功します。このメッセージは、同時にインポート可能な他のメンバがライブラリ内に含まれていることを示します。次に、例を示します。

```
proc cport file=in library=out;
proc cimport infile=in data=new;
```

出力先ライブラリの内容全体を含めるようにインポート処理を拡大するには、PROC CIMPORT で DATA=オプションの代わりに LIBRARY=オプションを指定します。

## SQL プロシジャでは、V9 のビューを V6 ライブラリに保存しません

通常、このメッセージは XPORT エンジンを使用して、SAS 6 のライブラリの移送形式で SAS 9 の PROC SQL ビューを作成した場合に表示されます。ただし、XPORT エンジンを使用して SQL テーブルを作成することはできません。

回復するには、SQL テーブルを含むデータセットを移送先コンピュータに転送し、そこで PROC SQL ビューを再作成します。

### 指定した機能をサポートしていません

このメッセージは、バージョン間の互換性がないため、SAS 9 を実行している移送元コンピュータのライブラリから SAS 6 を実行している移送先コンピュータのライブラリへの移動に失敗したことを示します。たとえば、データセットの生成や一貫性制約などの SAS 9 の機能はサポートされていません。

回復するには、SAS 6 を実行しているコンピュータのライブラリに移動されるライブラリまたはメンバから SAS 9 の機能を削除し、移送処理を再試行する必要があります。ログ内の直前の注記で、問題となっているサポートされていない SAS 9 の機能についてのヒントが示されます。次に、例を示します。

NOTE: Integrity constraint mc defined.

このメッセージから、SAS 6 は一貫性制約をサポートしていないことが推測できます。

SAS 9 の機能の削除に関するヒントについては、“ファイル library.member.DATA のメンバ名は V6 エンジンでは長すぎます” (75 ページ) および “変数名 variable は Version-6-data-set では不正です” (79 ページ) のメッセージの回復操作を参照してください。

### レコードが切り捨てられました

通常、このメッセージはファイルが BINARY 以外の形式で移送された場合、または移送先コンピュータへの転送中に移送ファイルの属性が変更された場合に表示されません。

回復操作については、“転送形式と移送ファイルの属性の検証” (79 ページ) を参照してください。

このメッセージは、DOS、Macintosh または UNIX などの他の動作環境で、移送ファイルが仮想ディスクまたは共有ディスクに移動されたことを示している可能性があります。回復操作については、現在の動作環境向けの章を参照してください。

### libref.member-name は、異なるオペレーティングシステム向けに作成されているため更新できません

このメッセージは、アクセスしているコンピュータとは異なる形式のファイルを更新しようとしたときに表示されます。ファイルで PROC CONTENTS を使用して、ファイルのデータ表現を確認します。データ表現が FOREIGN の場合、そのファイル形式とアクセスしているコンピュータに互換性がありません。

### XPORT エンジンでは、UTILITY FILE OPEN 機能はサポートされていません

このメッセージは、XPORT エンジンで PROC COPY を使用して、MDDDB などのユーティリティファイルの移送ファイルを作成しようとしたときに表示されます。XPORT エンジンではユーティリティファイルはサポートされていません。

### 値 y コードは有効な SAS 名ではありません。エラーのためデータセットをスキップします

このエラーと警告メッセージは、SAS 9 の PROC CPORT を使用して作成された移送ファイルを SAS 8 の PROC CIMPORT を使用して読み込んだときに表示されます。

PROC CPORT と PROC CIMPORT は上位互換性がありますが(SAS 8 の CPORT を使用して作成された SAS ファイルを SAS 9 の CIMPORT で読み込み可能)、下位互

換性はありません(SAS 9 の CPORT を使用して作成された SAS ファイルは SAS 8 の CIMPORT では読み込み不可)。

移送ファイルの作成に使用された SAS のバージョンを特定するには、次の SAS プログラムを使用してその移送ファイルを指定します。

```
data _null_;
infile 'transport-file-path';
input @109 rel $7.;
put rel=;
stop;
run;
```

移送ファイルの作成に使用された SAS のバージョンが出力されています。

### 変数名 *variable* は *Version-6-data-set* では不正です

このメッセージは、PROC CIMPORT を使用して長い変数名を含む SAS 9 のデータセットを SAS 6 のデータセットに移動したときに表示されます。次に、例を示します。

```
ERROR: The variable name Region_Of_The_Country
is illegal for file V6LIB.CITY.DATA.
```

SAS 9 の変数名 `Region Of The Country` は、SAS 6 でサポートされている最大 8 文字の変数名の制限を超えています。回復するには、クライアントの SAS セッションで VALIDVARNAME システムオプションを V6 に設定し、長い変数名の自動的な切り捨てを有効にして移送処理を再試行します。次に、例を示します。

```
options validvarname=v6;
```

この例では、`Region Of The Country` が `Region O` に切り捨てられます。ただし、データセット内に最初の 8 文字が重複する複数の変数名がある場合、SAS 9 は切り捨てられた変数名が重複しないようにする切り捨てアルゴリズムを使用します。詳細については、“[SAS データセットを SAS 6 形式に戻す](#)” (28 ページ) を参照してください。

---

## 転送形式と移送ファイルの属性の検証

移送ファイルの転送に使用する通信ソフトウェアで、BINARY 形式が指定されていることを確認します。たとえば、FTP を使用している場合は FTP BINARY コマンドを指定します。FTP の呼び出し例を次に示します。

```
ftp
> open host
> binary
> get file file
> close
> quit
```

FTP の詳細については、“[ファイルの転送](#)” (39 ページ) を参照してください。

通信ソフトウェアがデフォルトで適切な形式を使用して移送ファイルをサブミットすることになっている場合でも、常にバイナリ形式を明示的に指定して確実にバイナリ形式が使用されるようにします。転送形式の指定方法の詳細については、通信ソフトウェアのドキュメントを参照してください。

また、移送ファイルの属性も確認します。この確認は、移送先コンピュータでファイルを復元するために必要です。移送先コンピュータによってはファイルの属性は不要な場

合がありますが、転送方式(テープおよびネットワーク)では常に属性が必要です。ファイルの属性が必要な動作環境のリストについては、“[移送ファイルの属性](#)”(40 ページ)を参照してください。移送先の動作環境に必要なファイルの属性と転送方式によって適用されたファイルの属性に互換性がない場合、問題が発生する可能性があります。

移送先コンピュータで必要なファイルの属性を確認します。ファイルの属性の表示方法と指定方法は、コンピュータによって異なります。現在の動作環境向けの章を参照してください。

また、転送方式によって設定されたファイルの属性も確認します。たとえば、FTP を使用している場合は FTP コマンドでファイルの属性を設定します。FTP の呼び出し例を次に示します。

```
ftp
> open host
> binary
> locsite recfm=fb blocksize=8000 lrecl=80
> get file file
> close
> quit
```

ネットワークを介して移送ファイルを転送する場合は、通信ソフトウェアのドキュメントを参照してください。テープを使用したファイルの転送の詳細については、現在の動作環境向けのトピックを参照してください。

問題を修正できる場合は、移送元コンピュータで移送ファイルを再作成して移送先コンピュータに転送し、その移送ファイルを復元します。

問題が解決しない場合、移送ファイルを再ブロック化して移送を再試行してください。詳細については、“[移送ファイルの再ブロック化](#)”(80 ページ)を参照してください。

## 移送ファイルの再ブロック化

移送ファイルに不正なブロックサイズが含まれ、移送先コンピュータで正しいブロックサイズを含む移送ファイルを取得できない場合、再ブロック化プログラムを使用して移送ファイルを再ブロック化します。

**注:** 再ブロック化プログラムを実行する移送ファイルは、破損していない必要があります。つまり、キャリッジリターンまたはラインフィードが追加挿入されないようにしてください。移送ファイルが破損していることが判明した場合、再ブロック化プログラムは失敗します。

次のプログラムは、移送ファイルをコピーし、80 バイトの固定ブロックレコードを含む新しい移送ファイルを作成します。

```
data _null_;

/* Note: the INFILE and FILE statements must */
/* be modified. Substitute your file names. */
infile 'your_transport.dat' eof=wrapup;
file 'new_transport.dat' recfm=f lrecl=80;

length irec $16 outrec $80 nullrec $80;
retain count 1 outrec nullrec;
input inrec $char16. @@;
substr(outrec, count, 16) = inrec;
```



```
count + 16;  
if (count > 80) then do;  
put outrec $char80.;  
count=1;  
end;  
return;  
  
wrapup;  
file log;  
nullrec = repeat('00'x,80);  
if outrec = nullrec then do;  
put ' WARNING: Null characters may have been'  
' added at the end of transport file by'  
' communications software or by a copy'  
' utility. For a data set transport file,'  
' this could result in extra null'  
' observations being added at the end'  
' of the last data set.';  
end;  
run;
```

この例では、元の移送ファイルのレコード形式は固定で、レコード長は 16 で割り切れ  
ます。

レコードの種類が固定でレコード長が 16 で割り切れない場合、80 と移送ファイルのレ  
コード長の両方で割り切れる最大公約数を見つけます。その後、前述のプログラム内  
のすべての 16 をこの数に置き換えます。

たとえば、80 は 1、2、5、8、10 で割り切れます。移送ファイルの 99 の固定レコード長  
は、1、3、9、11 で割り切れます。この唯一の公約数は 1 です。そのため、最小公倍数  
と最大公約数の両方が 1 になります。

**注:** 移送ファイルに変数の長さのレコードの種類がある場合、最大公約数として 16 の  
代わりに 1 を使用します。

**注意:**

データセットを含む移送ファイルでは、一部の通信ソフトウェアは最後のレコードを null 文  
字で埋め込みます。再ブロック化プログラムによって、ライブラリの最後のデータセ  
ットの末尾にすべて 0 値が含まれるオブザベーションが追加される可能性があります。





## 6 部

---

# サンプルとログ

13 章	SAS ファイルの移動の例 .....	85
------	---------------------	----



## 13 章

## SAS ファイルの移動の例

---

異なるコンピュータ間での SAS ファイルの移動例の概要	85
例: OpenVMS から UNIX へのファイル移送	86
移送元コンピュータで PROC COPY を使用して、移送ファイルを作成する	86
移送元コンピュータで SAS ログを表示する	87
移送ファイルの検証	88
移送先コンピュータへの移送ファイルの転送	89
移送先コンピュータで PROC COPY を使用して、移送ファイル をネイティブフォーマットに復元する	91
移送先コンピュータで SAS ログを表示する	92
例: z/OS から Windows へのファイル移送	93
移送元コンピュータで PROC CPORT を使用して、移送ファイルを作成する	93
移送元コンピュータで SAS ログを表示する	94
移送ファイルの検証	94
移送先コンピュータへの移送ファイルの転送	95
移送先コンピュータで PROC CIMPORT を使用して、移送フ ァイルをネイティブフォーマットにインポートする	96
移送先コンピュータで SAS ログを表示する	97
例: z/OS JCL Batch から UNIX へのファイル移送	98
z/OS JCL Batch プログラムの概要	98
PROC COPY を使用して、移送ファイルを作成する	98
ネットワークを経由した移送ファイルの転送	99
移送ファイルの精度の確認	100
PROC COPY を使用して、移送ファイルを復元する	101
データセットと移送ファイルの作成を SAS ログに記録する	101
移送ファイルの移送先コンピュータへの転送を SAS ログに記録する	103
移送ファイルの検証を SAS ログに記録する	104
移送ファイルの移送元コンピュータへの復元を SAS ログに記録する	105
移送ファイルの検証方法	106
移送元コンピュータで移送ファイルを復元する	106
移送ファイルのサイズの確認	106
元のデータセットと復元されたデータセットを比較する	107

## 異なるコンピュータ間での SAS ファイルの移動例の概要

以降の例では、異なる動作環境で稼働している 2 台のコンピュータ間の移送ファイルの作成、転送および復元を示します。次の表で、各例の基本的特徴について説明します。

表 13.1 SAS ファイルの移動の例の概要

移動するメンバ	移送元コンピュータと SAS バージョン	移送先コンピュータと SAS バージョン	方法
データセット	OpenVMS 6.12	UNIX 8	PROC COPY での XPORT エンジンの使用
データセットとカタログ	z/OS 6.09	Windows 8	CPORT プロシジャと CIMPORT プロシジャ
データセット	JCL Batch z/OS 6.09	UNIX 9	PROC COPY での XPORT エンジンの使用

各例は動作環境に固有のものですが、移送方式の基本的な SAS コマンド構文は、すべての動作環境で同じです。動作環境の種類によって構文が異なるのは、LIBNAME ステートメントで SAS ライブラリを指定する部分です。LIBNAME ステートメントの構文の詳細については、動作環境の付属ドキュメントを参照してください。

## 例: OpenVMS から UNIX へのファイル移送

### 移送元コンピュータで PROC COPY を使用して、移送ファイルを作成する

この例では、OpenVMS 形式で 3 つのデータセットを作成し、これらのデータセットを移送形式に変換する SAS プログラムを示します。

#### 例のコード 13.1 データセットと移送ファイルを作成する SAS プログラム

```
libname xptlib xport 'xptlib.dat'; 1
libname xptds xport 'xptds.dat'; 2

/* creates data set GRADES; contains numeric and */
/* character data */
data grades; 3
input student $ test1 test2 final;
datalines;
Fred 66 80 70
Wilma 97 91 98
;

/* creates data set SIMPLE; contains */
/* character data only */
data simple; 4
x='dog';
y='cat';
z='fish';
run;

/* creates data set NUMBERS; contains */
/* numeric data only */
data numbers; 5
```

```

do i=1 to 10;
output;
end;
run;

/* create a transport file for the entire library */
proc copy in=work out=xptlib; 6
run;

/* create a transport file for a data set */
proc copy in=work out=xptds; 7
select grades;
run;

```

- 1 LIBNAME ステートメントで、ライブラリ参照名 XPTLIB を物理的な場所 XPTLIB.DAT に割り当てます。ここには、作成されるライブラリ全体が保存されます。XPORT エンジンで XPTLIB.DAT が作成されます。
- 2 LIBNAME ステートメントで、ライブラリ参照名 XPTDS を物理的な場所 XPTDS.DAT に割り当てます。ここには、作成されるデータセットが 1 つ保存されます。XPORT エンジンで XPTDS.DAT が作成されます。
- 3 DATA ステップで、2 個のオブザベーションを含む最初のデータセット WORK.GRADES を作成します。各オブザベーションには、変数が 4 つ(文字値 1 つ、数値 3 つ)が含まれます。
- 4 DATA ステップで、1 個のオブザベーションを含む 2 つ目のデータセット WORK.SIMPLE を作成します。オブザベーションには文字値が 3 つ含まれます。
- 5 DATA ステップで、10 個のオブザベーションを含む 3 つ目のデータセット WORK.NUMBERS を作成します。各オブザベーションには数値が 1 つ含まれます。
- 6 PROC COPY で、デフォルトの WORK ライブラリから新しいライブラリ XPTLIB に、3 つのデータセットをすべてコピーします。WORK データセットが移送形式で出力ライブラリ XPTLIB に書き込まれます。
- 7 PROC COPY で、選択したデータセット GRADES を新しいライブラリ XPTLIB にコピーします。データセット GRADES が移送形式で出力ライブラリ XPTDS に書き込まれます。

## 移送元コンピュータで SAS ログを表示する

次の例に、例のコード 13.1 (86 ページ) の SAS ログを示します。

### 例のコード 13.2 移送元コンピュータの SAS ログ

```

NOTE: SAS (r) Proprietary Software Release 6.12 TS050 1
NOTE: Running on DEC Model 7000 MODEL 740 Serial Number 80000000. 2
NOTE: Libref XPTLIB was successfully assigned as follows: 3
Engine: XPORT
Physical Name: Device:system-specific file/pathname XPTLIB.DAT
NOTE: Libref XPTDS was successfully assigned as follows: 4
Engine: XPORT
Physical Name:system-specific file/pathname XPTDS.DAT
NOTE: The data set WORK.GRADES has 2 observations and 4 variables. 5
NOTE: The data set WORK.SIMPLE has 1 observations and 3 variables.
NOTE: The data set WORK.NUMBERS has 10 observations and 1 variables.
NOTE: Copying WORK.GRADES to XPTLIB.GRADES (MEMTYPE=DATA). 6

```

NOTE: BUFSIZE is not cloned when copying across dissimilar engines.  
System Option for BUFSIZE was used.  
NOTE: The data set XPTLIB.GRADES has 2 observations and 4 variables.  
NOTE: Copying WORK.NUMBERS to XPTLIB.NUMBERS (MEMTYPE=DATA).  
NOTE: BUFSIZE is not cloned when copying across dissimilar engines.  
System Option for BUFSIZE was used.  
NOTE: The data set XPTLIB.NUMBERS has 10 observations and 1 variables.  
NOTE: Copying WORK.SIMPLE to XPTLIB.SIMPLE (MEMTYPE=DATA).  
NOTE: BUFSIZE is not cloned when copying across dissimilar engines.  
System Option for BUFSIZE was used.  
NOTE: The data set XPTLIB.SIMPLE has 1 observations and 3 variables.  
NOTE: Copying WORK.GRADES to XPTDS.GRADES (MEMTYPE=DATA). 7  
NOTE: BUFSIZE is not cloned when copying across dissimilar engines.  
System Option for BUFSIZE was used.  
NOTE: The data set XPTDS.GRADES has 2 observations and 4 variables.

- 1 移送元コンピュータでは SAS 6.12 が実行されています。つまり、SAS セッションのデフォルトライブラリエンジンは V612 です。
  - 2 移送元コンピュータは DEC モデル 7000 です。これは AX7000 のことです。
  - 3 ライブラリ参照名 XPTLIB が物理デバイスに割り当てられます。この物理デバイスの指定はプラットフォームに依存します。XPORT エンジンで XPTLIB が作成されます。
  - 4 ライブラリ参照名 XPTDS が物理デバイスに割り当てられます。この物理デバイスの指定はプラットフォームに依存します。XPORT エンジンで XPTDS が作成されます。
  - 5 一連のメモのうち、最初の 3 つはデータセット WORK.GRADES、WORK.SIMPLE および WORK.NUMBERS の作成を示しています。
  - 6 それ以降のメモは、WORK.GRADES が XPTLIB.GRADES へ、WORK.NUMBERS が XPTLIB.NUMBERS へ、WORK.SIMPLE が XPTLIB.SIMPLE へそれぞれコピーされることを示しています。XPORT エンジンで、各データセットが OpenVMS 形式から移送形式に変換されます。
- 注: SAS システムオプションの BUFSIZE に関する次のメモは、エラー状態を示すものではありません。BUFSIZE では、出力データセットの永久バッファサイズを指定します。これは、システムの性能を向上させるために調整できます。XPORT エンジンでは BUFSIZE=オプションがサポートされないため、BUFSIZE オプションに割り当てられているシステム値が使用されます。詳細については、動作環境付属ドキュメントを参照してください。
- 7 WORK.GRADES が XPTDS.GRADES にコピーされます。XPORT エンジンで、データセットが OpenVMS 形式から移送形式に変換されます。

## 移送ファイルの検証

移送先コンピュータにファイルを転送する前に、移送元コンピュータで移送ファイルの整合性を検証する必要があります。移送元コンピュータで検証が正常に行われることで、不正な移送ファイルが作成された場合はそれを排除できます。移送先コンピュータに移送ファイルを転送した後に、移送元コンピュータから送信した移送ファイルと、移送先コンピュータで受信したファイルと比較します。詳細については、“[移送ファイルの検証方法](#)” (106 ページ) を参照してください。

## 移送先コンピュータへの移送ファイルの転送

移送ファイルを移送先コンピュータに転送する前に、ファイル属性を検証します。この例では、一般的な出力を示します。

**例のコード 13.3** DIR/FULL を使用して移送ファイルの属性を検証する

```
vms> DIR/FULL xptlib.dat
Directory HOSTVAX: [JOE.XPTTEST]

XPTLIB.DAT;1 File ID: (31223,952,0)
Size: 7/8 Owner: [HOSTVAX,JOE]
  Created: 25-APR 2008 16:47:31.34
Revised: 25-APR-2008 16:47:31.69 (1)
Expires: Effective: File organization: Sequential
Shelved state: Online
File attributes: Allocation: 8, Extend: 0, Global buffer count: 0
Version limit: 2
Record format: Fixed length 80 byte records 1
Record attributes: None 2
RMS attributes: None
Journaling enabled: None
File protection: System:RWED, Owner:RWED, Group:RE, World:
Access Cntrl List: None

Total of 1 file, 7/8 blocks.
$ dir/size xptlib.dat

Directory HOSTVAX: [JOE.XPTTEST]

XPTLIB.DAT;1 7

Total of 1 file, 7 blocks.
```

- 1 RECORD FORMAT 属性は、固定レコードの種類および 80 バイトのレコードサイズを示しています。ネットワーク経由でファイルを正常に転送するには、これらの値が必要です。
- 2 RECORD ATTRIBUTES フィールドの値には、値 NONE が指定されている必要があります。

### 注意:

このフィールドに **CARRIAGE RETURN CARRIAGE CONTROL** が指定されている場合は、**ファイルが破損します**。破損を防ぐには、移送ファイルを転送する前に RECORD ATTRIBUTES フィールドからこの値を削除してください。破損したファイルを転送しようとする、この状態について警告するエラーメッセージが表示されます。

移送ファイルの属性を検証した後、FTP を使用して移送ファイルを移送先コンピュータに転送します。

この例では、移送先コンピュータの動作環境ディレクトリへの書き込み権限が移送元コンピュータにないため、移送先コンピュータ側で移送ファイルを移送元コンピュータから取得します。移送ファイルを移送先コンピュータへ書き込む権限が移送元コンピュータに付与されていることはあまりありません。

移送先コンピュータで、移送ファイルのコピー先の場所にディレクトリを変更します。移送ファイルを取得するために使用する FTP コマンドの例を次に示します。

## 例のコード13.4 ファイル転送のFTP ダイアログ

```

hp> ftp ax7000.vms.sas.com 1
Connected to ax7000.vms.com.
220 ax7000.vms.com MultiNet FTP Server Process V4.0(15) at Thu-Sep 30-99
12:59PM-EDT
Name (ax7000.vms.com:): joe
331 User name (joe) ok. Password, please.
Password:
230 User JOE logged into HOSTVAX:[JOE] at Thu 30-Sep-99 12:59PM-EDT, job
27a34cef.
Remote system type is VMS.
ftp> cd [.xpttest] 2
250 Connected to system-specific file/pathname.
ftp> binary 3
200 Type I ok.
ftp> get xptds.dat xptds.dat 4
200 Port 14.83 at Host 10.26.2.45 accepted.
150 IMAGE retrieve of system-specific file/pathname XPTDS.DAT;1 started.
1360 bytes received in 0.02 seconds (87.59 Kbytes226 Transfer completed.
1360 (8) bytes transferred. 5/s)
ftp> get xptlib.dat xptlib.dat 6
200 Port 14.84 at Host 10.26.2.45 accepted.
150 IMAGE retrieve of system-specific file/pathname XPTLIB.DAT;1 started.
3120 bytes received in 0.04 seconds (85.81 Kbytes226 Transfer completed.
3120 (8) bytes transferred. 7/s)
ftp> quit 8

```

- 1 転送先の UNIX コンピュータから FTP を起動し、転送元の OpenVMS 動作環境 AX7000.VMS.SAS.COM に接続します。
- 2 接続が確立された後、FTP プロンプトでユーザーの JOE は移送ファイルが含まれている移送元コンピュータのサブディレクトリに変更します。
- 3 移送ファイル属性 BINARY は、OpenVMS 移送ファイルを BINARY 形式で転送元コンピュータから転送する必要があることを示しています。
- 4 FTP の `get` コマンドで、移送元コンピュータから XPTDS.DAT という移送ファイルを取得します。次に、この移送ファイルは、移送先コンピュータで稼働する移送先動作環境の現在のディレクトリにある、同じ XPTDS.DAT という名前の新しいファイルにコピーされます。
- 5 メッセージは、転送が正常に行われ、移送ファイルのサイズが 1360 バイトであることを示しています。移送元コンピュータと移送先コンピュータのそれぞれの移送ファイルのサイズを比較します。サイズが同じであれば、ファイルは正常に転送されています。ファイルサイズの一覧表示の詳細については、“[移送ファイルのサイズの確認](#)” (106 ページ) を参照してください。
- 6 FTP の `get` コマンドで、移送元コンピュータから XPTLIB.DAT という別の移送ファイルが取得され、移送先コンピュータの移送先動作環境の現在のディレクトリにある、同じ XPTLIB.DAT という名前の新しいファイルにコピーされます。
- 7 メッセージは転送が正常に行われたことを示しています。移送元コンピュータと移送先動作環境のそれぞれの移送ファイルのサイズを比較します。
- 8 ユーザーは FTP セッションを終了します。

ファイル転送ユーティリティの詳細については、FTP のドキュメントを参照してください。



## 移送先コンピュータで PROC COPY を使用して、移送ファイルをネイティブフォーマットに復元する

次の例では、ネイティブのファイル形式に移送ファイルを変換する SAS プログラムを示します。

### 例のコード 13.5 移送ファイルをネイティブのファイル形式に復元する SAS プログラム

```
libname xptlib xport 'xptlib.dat'; 1
libname xptds xport 'xptds.dat'; 2

libname natvlib v7 'natvlib' 3
libname natvds v7 'natvds'; 4

/* translate transport file for library */
/* to native format on target computer. */

proc copy in=xptlib out=natvlib; 5
run;

/* translate transport file for data set*/
/* to native format on target computer */

proc copy in=xptds out=natvds; 6
select grades;
run;
```

- LIBNAME ステートメントで、ライブラリ参照名 XPTLIB を物理的な場所 XPTLIB.DAT に割り当てます。ここには、移送先コンピュータに転送されたライブラリ全体が保存されます。XPORT エンジンで XPTLIB が読み込まれます。
- LIBNAME ステートメントで、ライブラリ参照名 XPTDS を物理的な場所 XPTDS.DAT に割り当てます。ここには、移送先コンピュータに転送されたデータセットが 1 つ保存されます。XPORT エンジンで XPTDS が読み込まれます。
- LIBNAME ステートメントで、ライブラリ参照名 NATVLIB を物理的な場所 NATVLIB に割り当てます。ここには、移送形式からネイティブ形式に変換されるライブラリ全体が保存されます。V7 エンジンで NATVLIB が作成されます。
- LIBNAME ステートメントで、ライブラリ参照名 NATVDS を物理的な場所 NATVDS に割り当てます。ここには、移送形式からネイティブ形式に変換されるデータセットが 1 つ保存されます。V7 エンジンで NATVDS が作成されます。
- PROC COPY で、ライブラリ参照名 XPTLIB から新しいライブラリ参照名 NATVLIB に、3 つのデータセットをすべてコピーします。XPORT エンジンで、XPTLIB からすべてのデータセットが移送形式で読み込まれます。V7 エンジンで、出力ライブラリ参照名 NATVLIB にネイティブの UNIX 形式でデータセットが書き込まれます。
- PROC COPY で、新しいライブラリ NATVDS にコピーするデータセット GRADES を選択します。XPORT エンジンで、データセット GRADES が移送形式で読み込まれます。V7 エンジンで、出力ライブラリ XPTDS がネイティブの UNIX 形式で書き込まれます。

## 移送先コンピュータでSAS ログを表示する

この例では、例のコード 13.5 (91 ページ)次に示す SAS プログラムの正常な実行を記録する SAS ログを示します。

### 例のコード13.6 移送先コンピュータの SAS ログ

```
NOTE: Copyright (c) 1999 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software Version 8 (TS00.00P1D090398) 1
Licensed to SAS Institute Inc., Site 0000000001.
NOTE: This session is executing on the UNIX B.10.20 platform. 2
NOTE: Running on HP Model 9000/715 Serial Number 2005516582.
libname xptlib xport 'xptlib.dat'; 3
NOTE: Libref XPTLIB was successfully assigned as follows:
Engine: XPORT
Physical Name: system-specific file/pathname/xptlib.dat
libname xptds xport 'xptds.dat'; 4
NOTE: Libref XPTDS was successfully assigned as follows:
Engine: XPORT
Physical Name:
system-specific file/pathname/xptds.dat
libname natvlib v7 'natvlib'; 5
NOTE: Libref NATVLIB was successfully assigned as follows:
Engine: V7
Physical Name:
system-specific file/pathname/natvlib
libname natvds v7 'natvds'; 6
NOTE: Libref NATVDS was successfully assigned as follows:
Engine: V7
Physical Name:
system-specific file/pathname/natvds

/* translate transport file for library to native */
/* format on target computer. */
proc copy in=xptlib out=natvlib;
run;
NOTE: Input library XPTLIB is sequential.
NOTE: Copying XPTLIB.GRADES to NATVLIB.GRADES (memtype=DATA). 7
NOTE: BUFSIZE is not cloned when copying across different engines.
System Option for BUFSIZE was used.
NOTE: The data set NATVLIB.GRADES has 2 observations and 4 variables.
NOTE: Copying XPTLIB.NUMBERS to NATVLIB.NUMBERS (memtype=DATA). 8
NOTE: BUFSIZE is not cloned when copying across different engines.
System Option for BUFSIZE was used.
NOTE: The data set NATVLIB.NUMBERS has 10 observations and 1 variables.
NOTE: Copying XPTLIB.SIMPLE to NATVLIB.SIMPLE (memtype=DATA). 9
NOTE: BUFSIZE is not cloned when copying across different engines.
System Option for BUFSIZE was used.
NOTE: The data set NATVLIB.SIMPLE has 1 observations and 3 variables.
/* translate transport file for data set to native */
/* on target computer */
proc copy in=xptds out=natvds;
select grades;
run;
NOTE: Input library XPTDS is sequential.
```

NOTE: Copying XPTDS.GRADES to NATVDS.GRADES (memtype=DATA). 10  
 NOTE: BUFSIZE is not cloned when copying across different engines.  
 System Option for BUFSIZE was used.  
 NOTE: The data set NATVDS.GRADES has 2 observations and 4 variables

- 1 移送先コンピュータでは SAS 8 が実行されています。つまり、移送先動作環境の SAS セッションでは、デフォルトライブラリエンジン V8 が使用されます。
- 2 移送先コンピュータでは UNIX が実行されています。
- 3 LIBNAME ステートメントで、ライブラリ参照名 XPTLIB を物理デバイスに割り当てます。この物理デバイスの指定はプラットフォームに依存します。この例では、物理デバイスは UNIX 動作環境を示しています。XPORT エンジンで XPTLIB が読み込まれます。
- 4 LIBNAME ステートメントで、ライブラリ参照名 XPTDS を物理デバイスに割り当てます。この物理デバイスの指定はプラットフォームに依存します。XPORT エンジンで XPTDS が読み込まれます。
- 5 LIBNAME ステートメントで、ライブラリ参照名 NATVLIB を物理デバイスに割り当てます。この物理デバイスの指定はプラットフォームに依存します。この例では、物理デバイスは UNIX 動作環境を示しています。V7 エンジンで NATVLIB に書き込まれます。
- 6 LIBNAME ステートメントで、ライブラリ参照名 NATVDS を物理デバイスに割り当てます。この物理デバイスの指定はプラットフォームに依存します。この例では、物理デバイスは UNIX 動作環境を示しています。V7 エンジンで NATVDS に書き込まれます。
- 7 PROC COPY で、XPTLIB.GRADES を NATVLIB.GRADES にコピーします。NATVLIB データセットが V7 形式で書き込まれます。
- 8 PROC COPY で、XPTLIB.NUMBERS を NATVLIB.NUMBERS にコピーします。NATVLIB データセットが V7 形式で書き込まれます。
- 9 PROC COPY で、XPTLIB.SIMPLE を NATVLIB.SIMPLE にコピーします。NATVLIB データセットが V7 形式で書き込まれます。
- 10 PROC COPY で、XPTDS.GRADES を NATVDS.GRADES にコピーします。NATVDS データセットが V7 形式で書き込まれます。

---

## 例: z/OS から Windows へのファイル移送

### 移送元コンピュータで PROC CPORT を使用して、移送ファイルを作成する

この例では、ライブラリから 2 つのデータセットと 2 つのカatalogを z/OS 形式でコピーし、移送形式のデフォルトの出カファイルに書き込む SAS プログラムを示します。

**例のコード 13.7** データセットとカatalogを移送ファイルにコピーする SAS プログラム

```
filename tport 'joe.mytest.data' disp=rep;
libname test 'joe.mytest.sas';
proc cport library=test file=tport;
run;
```

LIBNAME ステートメントで、ライブラリ参照名 TEST を物理的な場所 JOE.MYTEST.SAS に割り当てます。これは、移送されるライブラリを指しています。JOE は、移送処理を実行する SAS セッションに関連付けられているユーザー ID で

す。FILENAME ステートメントで、ファイル参照名 TPORT を移送ファイル JOE.MYTEST.DATA に割り当てます。DISP=REP で、新しいファイルが作成されるか、既存のファイルが置き換えられます。

### 移送元コンピュータで SAS ログを表示する

この例では、例のコード 13.7 (93 ページ) に示す SAS プログラムの正常な実行を記録する SAS ログを示します。

#### 例のコード 13.8 移送元コンピュータの SAS ログファイル

```
filename tport 'joe.mytest.data';
libname test 'joe.mytest.sas';
proc cport lib=test file=tport;
run;
WARNING: No output file is specified. Default output
file JOE.SASCAT.DATA is used.

NOTE: Proc CPORT begins to transport data set TEST.CITY
NOTE: The data set contains 7 variables and 72 observations.
NOTE: Transporting data set index information.

NOTE: Proc CPORT begins to transport catalog TEST.FORMATS
NOTE: The catalog has 3 entries
NOTE: Transporting entry REGFMT .FORMATC
NOTE: Transporting entry SALEFMT .FORMATC
NOTE: Transporting entry SIZEFMT .FORMATC

NOTE: Proc CPORT begins to transport catalog TEST.TEST
NOTE: The catalog has 11 entries
NOTE: Transporting entry ABOUT .CBT
NOTE: Transporting entry APPEND .CBT
NOTE: Transporting entry BOOKMENU.CBT
NOTE: Transporting entry DEFAULT .FORM
NOTE: Transporting entry HELP .HELP
NOTE: Transporting entry CLIST .LIST
NOTE: Transporting entry ENTRYTYP.LIST
NOTE: Transporting entry SPELLLALL.PMENU
NOTE: Transporting entry SPELLSUG.PMENU
NOTE: Transporting entry ADDON1 .PROGRAM
NOTE: Transporting entry ADDON2 .PROGRAM
NOTE: Proc CPORT begins to transport data set TEST.VARNUM
NOTE: The data set contains 10 variables and 100 observations.
```

注: デフォルトの出カファイル名は、動作環境に固有です。

PROC CPORT で、ライブラリ参照名 TEST で参照されるライブラリ全体の内容を読み込み、デフォルトの移送ファイルに書き込みます。残りのメモは、PROC CPORT でデータセット TEST.CITY、カタログ TEST.FORMATS、カタログ TEST.TEST、データセット TEST.VARNUM が移送ファイル JOE.MYTEST.DATA に移送されることを示しています。

### 移送ファイルの検証

移送先コンピュータにファイルを転送する前に、移送元コンピュータで移送ファイルの整合性を検証する必要があります。移送元コンピュータで検証が正常に行われることで、不正な移送ファイルが作成された場合はそれを排除できます。また、移送先コンピ

ュータにファイルを転送した後に、移送元コンピュータから送信した移送ファイルと、移送先コンピュータで受信したファイルを比較できます。詳細については、“[移送ファイルの検証方法](#)” (106 ページ) を参照してください。

## 移送先コンピュータへの移送ファイルの転送

移送先コンピュータに転送する前に、移送ファイルのファイル属性を検証します。この例では、TSO の一般的な出力を示します。

**例のコード 13.9** TSO LISTD コマンドを使用して移送ファイルの属性を検証する

```
listd "userid.mytest.data"
USERID.MYTEST.DATA
--RECFM-LRECL-BLKSIZE-DSORG
FB 80 8000 PS
--VOLUMES--
APP009
```

移送ファイルの属性を検証した後、FTP を使用して移送ファイルをネットワーク経由で転送できます。FTP ダイアログで、必要に応じてデフォルトの DCB 属性を変更します。この例では、移送元コンピュータのユーザーに移送先コンピュータへの書き込み権限があるため、FTP の `put` コマンドを使用して移送先コンピュータに移送ファイルを書き込みます。

この例では、移送先コンピュータに移送ファイルを書き込むために移送元コンピュータで指定する FTP コマンドを示します。

```
ftp mypc 1
EZA1450I MVS TCP/IP FTP V3R2
EZA1554I Connecting to SPIDER 10.24.2.32, port 21
220 spider FTP server (Version 4.162 Tue Nov 1
10:50:37 PST 1988) ready.
EZA1459I USER (identify yourself to the host):
userid password
EZA1701I >>>USER joe
331 Password required for joe.
EZA1701I >>>PASS *****
230 User joe logged in.
EZA1460I Command: 2
binary
EZA1701I >>>TYPE i
200 Type set to I.
EZA1460I Command: 3
put 'joe.mytest.data' c:\tport.dat
EZA1701I >>>SITE VARrecfm Lrecl=80 4
Recfm=FB BLKSIZE=8000
500 'SITE VARRECFM Lrecl=80 Recfm=FB BLKSIZE=23440':
EZA1701I >>>PORT 10,253,1,2,129,50
200 PORT command
EZA1701I >>>STOR c:\tport.dat 5
150 Opening BINARY mode data connection for c:\tport.dat
226 Transfer complete. 6
EZA2517I 6071600 bytes transferred in 13 seconds.
Transfer rate 466.18 Kbytes/sec.
EZA1460I Command: 7
quit
EZA1701I >>>QUIT
221 Goodbye.
```

READY

- 1 移送元の z/OS コンピュータから FTP を起動し、移送先の Windows コンピュータ MYPC に接続します。
- 2 移送ファイル属性 BINARY は、z/OS 移送ファイルを BINARY 形式で移送元コンピュータから転送する必要があることを示しています。
- 3 FTP の put コマンドで、移送元コンピュータから JOE.MYTEST.DATA という移送ファイルを、移送先コンピュータの物理的な場所 C:\TPORT.DAT にコピーします。
- 4 FTP のファイル属性コマンドは、レコード長 80 バイト、固定レコードの種類、ブロックサイズ 8000 を示しています。
- 5 TPORT.DAT がドライブ C に保存されます。
- 6 メッセージは転送が正常に行われたことを示しています。ファイルサイズの一覧表示の詳細については、“[移送ファイルのサイズの確認](#)” (106 ページ) を参照してください。
- 7 ユーザーは FTP セッションを終了します。

### 移送先コンピュータで PROC CIMPORT を使用して、移送ファイルをネイティブフォーマットにインポートする

この例では、移送形式からネイティブ形式に移送ファイルを変換する SAS プログラムを示します。

#### 例のコード 13.10 移送ファイルをネイティブ形式にインポートする SAS プログラム

```
libname newlib 'c:\mylib';
proc cimport infile='c:\tport.dat' library=newlib;
run;
```

この LIBNAME ステートメントで、ライブラリ参照名 NEWLIB を物理的な場所 c:\mylib に割り当てます。ここには、V7 ライブラリ全体が保存されます。PROC CIMPORT で、INFILE=オプションで指定する移送ファイルの内容全体を読み込み、LIBNAME=オプションで指定する出力先に書き込みます。

ライブラリの内容全体をネイティブの V7 形式にインポートする別の方法として、移送ライブラリから特定のエンティティを選択または除外できます。

次に、例を示します。

#### 例のコード 13.11 データセットを選択する

```
filename target 'c:\tport.dat';
libname newlib 'c:\mylib';
proc cimport infile=target library=newlib;
select varnum;
run;
```

前述の例では、ファイル参照名 TARGET は、移送先コンピュータ上の移送ファイルの転送先を指しています。ライブラリ参照名 NEWLIB は、選択したメンバの保存先を指します。PROC CIMPORT で、INFILE=オプションで指定する移送ファイルの内容全体を読み込み、SELECT ステートメントで指定するメンバのみを書き込みます。データセット VARNUM は、ライブラリ NEWLIB に Windows 形式で書き込まれます。

#### 例のコード 13.12 カタログエントリの種類を選択する

```
filename target 'c:\tport.dat';
libname newlib 'c:\mylib';
proc cimport infile=target library=newlib
```

```
memtype=catalog et=program;
run;
```

前述の例では、PROC CIMPORT で INFILE=オプションで指定する移送ファイルの内容全体を読み込み、メンバの種類 CATALOG、エントリの種類 PROGRAM のみをライブラリ NEWLIB に Windows 形式で書き込みます。

#### 例のコード 13.13 カタログエントリを選択する

```
filename target 'c:\tport.dat';
libname newlib 'c:\mylib';
proc cimport infile=target library=newlib memtype=cat;
select spellsug.pmenu addon1.program;
run;
```

前述の例では、PROC CIMPORT で INFILE=オプションで指定する移送ファイルの内容全体を読み込み、メンバの種類 CATALOG のエントリ SPELLSUG.PMENU および ADDON1.PROGRAM のみをライブラリ NEWLIB に Windows 形式で書き込みます。

### 移送先コンピュータで SAS ログを表示する

この例では、例のコード 13.10 (96 ページ) に示す SAS プログラムの正常な実行を記録する SAS ログを示します。

```
NOTE: Proc CIMPORT begins to create/update data set NEWLIB.CITY
NOTE: The data set index REGION is defined.
NOTE: Data set contains 7 variables and 72 observations.
NOTE: Proc CIMPORT begins to create/update catalog NEWLIB.FORMATS
NOTE: Entry REGFMT.FORMATC has been imported.
NOTE: Entry SALEFMT.FORMATC has been imported.
NOTE: Entry SIZEFMT.FORMATC has been imported.
NOTE: Total number of entries processed in catalog NEWLIB.FORMATS: 3
```

```
NOTE: Proc CIMPORT begins to create/update catalog NEWLIB.TEST
NOTE: Entry ABOUT.CBT has been imported.
NOTE: Entry APPEND.CBT has been imported.
NOTE: Entry BOOKMENU.CBT has been imported.
NOTE: Entry DEFAULT.FORM has been imported.
NOTE: Entry HELP.HELP has been imported.
NOTE: Entry CLIST.LIST has been imported.
NOTE: Entry ENTRYTYP.LIST has been imported.
NOTE: Entry SPELLALL.PMENU has been imported.
NOTE: Entry SPELLSUG.PMENU has been imported.
NOTE: Entry ADDON1.PROGRAM has been imported.
NOTE: Entry ADDON2.PROGRAM has been imported.
NOTE: Total number of entries processed in catalog NEWLIB.TEST: 11
```

```
NOTE: Proc CIMPORT begins to create/update data set NEWLIB.VARNUM
NOTE: Data set contains 10 variables and 100 observations.
```

PROC CIMPORT で、データセット NEWLIB.CITY、カタログ NEWLIB.FORMATS、カタログ NEWLIB.TEST、データセット NEWLIB.VARNUM を Windows 形式で移送先コンピュータに作成します。

## 例: z/OS JCL Batch から UNIX へのファイル移送

### z/OS JCL Batch プログラムの概要

4 つに分けて示していますが、次のプログラムは 1 つのプログラムとして設計されています。次のプロセスが実行されます。

1. PROC COPY を使用して、移送元の z/OS コンピュータに移送ファイルを作成します。
2. 移送ファイルは、ネットワークを經由して移送先の UNIX コンピュータに転送されます。
3. 移送ファイルの精度が検証されます。
4. PROC COPY を使用して、移送元の z/OS コンピュータに移送ファイルを復元します。

埋め込みコメントでプログラムが記録されます。

### PROC COPY を使用して、移送ファイルを作成する

例のコード 13.14 (98 ページ)では、z/OS 形式で 3 つのデータセットを作成し、これらのデータセットを移送形式に変換するプログラムの最初の部分を示します。このプログラム部分の実行に関する SAS ログの詳細については、“データセットと移送ファイルの作成を SAS ログに記録する” (101 ページ) を参照してください。

#### 例のコード 13.14 データセットと移送ファイルを作成する

```
//XPORTTST JOB job-card-information
/*-----
/* Run SAS step that creates a transport library
/* for the three SAS test data sets.

/*-----
//SASOUT EXEC SAS
/*-----
/* Allocate the SAS XPORTOUT library.
/* The XPORTOUT library should have the
/* following data set information:
/* Record format: FB
/* Record length: 80
/* Block size: 8000
/* Organization: PS
/*-----

//XPORTOUT DD DSN=userid.XPORTOUT.DAT, DISP=(NEW,CATLG,DELETE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000),
// SPACE=(TRK,(1,1))
//SYSIN DD *
/*-----*/
/* Assign the SAS test xport library */
/*-----*/
libname xportout xport;
```



```

/*-----*/
/* Creates data set GRADES which contains */
/* numeric and character data. */
/*-----*/
data grades;
input student $ test1 test2 final;
datalines;
Fred 66 80 70
Wilma 97 91 98
;

/*-----*/
/* Creates data set SIMPLE which */
/* contains character data only. */
/*-----*/
data simple;
x='dog';
y='cat';
z='fish';
run;

/*-----*/
/* Creates data set NUMBERS which */
/* contains numeric data only. */
/*-----*/
data numbers;
do i=1 to 10;
output;
end;
run;

/*-----*/
/* Copy the three test data sets to */
/* the XPORT library. */
/*-----*/
proc copy in=work out=xportout;
run;
/*

```

## ネットワークを経由した移送ファイルの転送

この例では、FTP コマンドファイルの生成および移送先コンピュータへのネットワーク経由の移送ファイルの転送を示します。プログラムのこの部分の実行に関する SAS ログの詳細については、“[移送ファイルの移送先コンピュータへの転送を SAS ログに記録する](#)” (103 ページ) を参照してください。

**例のコード 13.15** FTP を使用して移送ファイルを転送する

```

/*-----
/* Generate FTP command file for sending XPORTOUT
/* test library to the target computer.
/*-----
//FTPCMDO EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=userid.FTP.OUT,

```

```

// UNIT=DISK,DISP=(NEW,CATLG),
// SPACE=(TRK,(1,1)),DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//*-----
/* Ensure that the FTP commands specify a BINARY
/* mode transfer.
/*-----
//SYSUT1 DD *
userid password
cd mydir
binary
put 'userid.xportout.dat' xportout.dat
quit
/*
/*-----
/* FTP library XPORTOUT to the target computer.
/*-----
//FTPXEQO EXEC PGM=IKJEFT01,REGION=2048K,DYNAMNBR=50,COND=EVEN
//SYSPRINT DD SYSOUT=*
//SYSTSOUT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
ALLOC FI(input) DA('userid.FTP.OUT') SHR
FTP target-host (EXIT
/*

```

### 移送ファイルの精度の確認

移送先の UNIX コンピュータから移送元の z/OS コンピュータにネイティブ形式で転送して移送ファイルを検証する例を次に示します。移送形式からネイティブの z/OS 形式に正常に変換されれば、移送ファイルの精度が検証されたこととなります。プログラムのこの部分の実行に関する SAS ログの詳細については、“[移送ファイルの検証を SAS ログに記録する](#)” (104 ページ) を参照してください。

#### 例のコード 13.16 移送ファイルの検証

```

//*-----
/* The following steps retrieve the XPORTOUT library
/* from the target computer and read the three test
/* data sets back into the WORK library.
/*-----
/* Generates the FTP command file for getting
/* the test library XPORTOUT from the target computer.
/*-----
//FTPCMDI EXEC PGM=IEBGENER,COND=EVEN
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=userid.FTP.IN,
// UNIT=DISK,DISP=(NEW,CATLG),
// SPACE=(TRK,(1,1)),DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//*-----
/* The FTP commands specify a BINARY mode
/* transfer. Uses the LOCSITE command to define
/* the correct XPORT library data set information.
/*-----
//SYSUT1 DD *
userid password

```

```

cd mydir
locsite recfm=fb blocksize=8000 lrecl=80
binary
get xportout.dat 'userid.xportin.dat'
quit
/*
/*-----
/* Connects to the target computer and retrieves
/* the library XPORTOUT.
/*-----
/*FTPXEQI EXEC PGM=IKJEFT01,REGION=2048K,DYNAMNBR=50,COND=EVEN
//SYSPRINT DD SYSOUT=*
//SYSTSOUT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
ALLOC FI(input) DA('userid.FTP.IN') SHR
FTP target-host (EXIT
/*

```

### PROC COPY を使用して、移送ファイルを復元する

この例では、移送元の z/OS コンピュータで移送ファイルをネイティブ形式に復元します。プログラムのこの部分の実行に関する SAS ログの詳細については、“[移送ファイルの移送元コンピュータへの復元を SAS ログに記録する](#)” (105 ページ) を参照してください。

#### 例のコード 13.17 移送ファイルをネイティブ形式に復元する

```

/*-----
/* Runs SAS step that reads the transport library
/* and writes the three SAS test data sets to
/* library WORK.
/*-----
/*SASIN EXEC SAS
//XPORTIN DD DSN=userid.XPORTIN.DAT,DISP=SHR
//SYSIN DD *
/*-----*/
/* Assigns the SAS test library XPORTIN. */
/*-----*/
libname xportin xport;

/*-----*/
/* Reads the transport file and writes the test */
/* data sets to library WORK. */
/*-----*/

proc copy in=xportin out=work;
run;
/*

```

### データセットと移送ファイルの作成を SAS ログに記録する

この SAS ログは、データセットおよび対応する移送ファイルの作成を示しています。

**例のコード13.18** 移送元のz/OS コンピュータでSAS ログを表示する(1/4)

```

The SAS System
11:03 Monday, October 26, 1999

NOTE: Copyright (c) 1999 by SAS Institute Inc.,
Cary, NC, USA.
NOTE: SAS (r) Proprietary Software Version 6.09.0460P0304986
Licensed to SAS INSTITUTE INC., Site 0000000001.

NOTE: Running on IBM Model 9672,
IBM Model 9672,
IBM Model 9672.

NOTE: No options specified.

/*-----*/
/* Assigns the SAS test library XPORTOUT. */
/*-----*/
libname xportout xport;
NOTE: Libref XPORTOUT was successfully assigned
as follows:
Engine: XPORT
Physical Name: JOE.XPORTOUT.DAT

/*-----*/
/* Creates data set GRADES which contains */
/* numeric and character data. */
/*-----*/
data grades;
input student $ test1 test2 final;
datalines;

NOTE: The data set WORK.GRADES has 2 observations
and 4 variables.

/*-----*/
/* Creates data set SIMPLE which */
/* contains character data only. */
/*-----*/
data simple;
x='dog';
y='cat';
z='fish';
run;

NOTE: The data set WORK.SIMPLE has
1 observations and 3 variables.

/*-----*/
/* Creates data set NUMBERS which */
/* contains numeric data only. */
/*-----*/
data numbers;
do i=1 to 10;

```

```

output;
end;
run;
NOTE: The data set WORK.NUMBERS has
10 observations and 1 variables.
/*-----*/
/* Copies the three test data sets to */
/* the XPORTOUT library. */
/*-----*/
proc copy in=work out=xportout;
run;

NOTE: Copying WORK.GRADES to XPORTOUT.GRADES
(MEMTYPE=DATA).
NOTE: BUFSIZE is not cloned when copying across different engines.
System Option for BUFSIZE was used.
NOTE: The data set XPORTOUT.GRADES has
2 observations and 4 variables.
NOTE: Copying WORK.NUMBERS to XPORTOUT.NUMBERS
(MEMTYPE=DATA).
NOTE: BUFSIZE is not cloned when copying across different engines.
System Option for BUFSIZE was used.
NOTE: The data set XPORTOUT.NUMBERS has
10 observations and 1 variables.
NOTE: Copying WORK.SIMPLE to XPORTOUT.SIMPLE
(MEMTYPE=DATA).
NOTE: BUFSIZE is not cloned when copying across different engines.
System Option for BUFSIZE was used.
NOTE: The data set XPORTOUT.SIMPLE has 1 observations and 3 variables.

```

注: SAS システムオプションの BUFSIZE に関するメモは、エラー状態を示すものではありません。BUFSIZE では、出力データセットの永久バッファサイズを指定します。これは、システムの性能を向上させるために調整できます。XPORT エンジンでは BUFSIZE=オプションがサポートされないため、BUFSIZE オプションに割り当てられているシステム値が使用されます。詳細については、動作環境付属ドキュメントを参照してください。

### 移送ファイルの移送先コンピュータへの転送を SAS ログに記録する

この SAS ログは、移送先コンピュータへの移送ファイルの転送を示しています。

**例のコード 13.19** 移送元の z/OS コンピュータで SAS ログを表示する(2/4)

```

EZA1450I MVS TCP/IP FTP V3R2
EZA1772I FTP: EXIT has been set.
EZA1736I conn MYHOST.MYCOMPANY.COM
EZA1554I Connecting to MYHOST.MYCOMPANY.COM
10.26.11.235, port 21
220 myhost FTP server (Version 4.162 Tue Nov 1 10:50:37 PST 1988)
ready.
EZA1459I USER (identify yourself to the host):
EZA1701I >>>USER joe
331 Password required for joe.
EZA1701I >>>PASS *****
230 User joe logged in.
EZA1460I Command:

```

```

EZA1736I cd joe
EZA1701I >>>CWD joe
250 CWD command successful.
EZA1460I Command:
EZA1736I binary
EZA1701I >>>TYPE i
200 Type set to I.
EZA1460I Command:
EZA1736I put 'joe.xportout.dat'
xportout.dat
EZA1701I >>>SITE VARrecfm Lrecl=80
Recfm=FB BLKSIZE=8000
500 'SITE VARrecfm Lrecl=80 Recfm=FB
BLKSIZE=8000': command not understood
EZA1701I >>>PORT 10,253,1,2,33,182
200 PORT command.
EZA1701I >>>STOR xportout.dat
150 Opening BINARY mode data connection for
xportout.dat.
226 Transfer complete.
EZA1460I Command:
EZA1736I quit
EZA1701I >>>QUIT

```

### 移送ファイルの検証を SAS ログに記録する

この SAS ログは、転送した移送ファイルの精度を検証するプログラム部分を示しています。

#### 例のコード13.20 移送元の z/OS コンピュータで SAS ログを表示する(3/4)

```

EZA1450I MVS TCP/IP FTP V3R2
EZA1772I FTP: EXIT has been set.
EZA1736I conn MYHOST.MYCOMPANY.COM
EZA1554I Connecting to MYHOST.MYCOMPANY.COM
10.26.11.235, port 21
220 myhost FTP server (Version 4.162 Tue Nov 1 10:50:37 PST 1988)
ready.
EZA1459I USER (identify yourself to the host):
EZA1701I >>>USER joe
331 Password required for joe.
EZA1701I >>>PASS *****
230 User joe logged in.
EZA1460I Command:
EZA1736I cd joe
EZA1701I >>>CWD joe
250 CWD command successful.
EZA1460I Command:
EZA1736I locsite recfm=fb blocksize=8000 lrecl=80
EZA1460I Command:
EZA1736I binary
EZA1701I >>>TYPE i
200 Type set to I.
EZA1460I Command:
EZA1736I get xportout.dat 'joe.xportin.dat'
EZA1701I >>>PORT 10,253,1,2,33,184

```

```

200 PORT command
EZA1701I >>>RETR xportout.dat
150 Opening BINARY mode data connection for
xportout.dat(3120 bytes).
226 Transfer complete.
EZA1617I 3120 bytes transferred in 0.198 seconds. Transfer rate
9.12 Kbytes/sec.
EZA1460I Command:
EZA1736I quit
EZA1701I >>>QUIT

```

### 移送ファイルの移送元コンピュータへの復元を SAS ログに記録する

この SAS ログは、z/OS コンピュータのネイティブ形式に移送ファイルをコピーするプログラム部分を示しています。

#### 例のコード 13.21 移送元の z/OS コンピュータで SAS ログを表示する(4/4)

```

NOTE: SAS (r) Proprietary Software Release 6.09.0460P030498
Licensed to SAS INSTITUTE INC., Site 0000000001.
NOTE: Running on IBM Model 9672,
IBM Model 9672,
IBM Model 9672.

```

NOTE: No options specified.

```

/*-----*/
/* Assigns the SAS test library XPORTIN. */
/*-----*/
libname xportin xport;
NOTE: Libref XPORTIN was successfully assigned
as follows:
Engine: XPORT
Physical Name: JOE.XPORTIN.DAT
/*-----*/
/* Reads the transport file and writes the */
/* test data sets to the library WORK. */
/*-----*/
proc copy in=xportin out=work;
run;

```

```

NOTE: Input library XPORTIN is sequential.
NOTE: Copying XPORTIN.GRADES to WORK.GRADES
(MEMTYPE=DATA).
NOTE: BUFSIZE is not cloned when copying across
different engines. System Option for BUFSIZE was used.
NOTE: The data set WORK.GRADES has 2 observations
and 4 variables.
NOTE: Copying XPORTIN.NUMBERS to WORK.NUMBERS
(MEMTYPE=DATA).
NOTE: BUFSIZE is not cloned when copying across
different engines. System Option for BUFSIZE was used.
NOTE: The data set WORK.NUMBERS has 10 observations
and 1 variables.
NOTE: Copying XPORTIN.SIMPLE to WORK.SIMPLE
(MEMTYPE=DATA).

```

注: SAS システムオプションの BUFSIZE に関するメモは、エラー状態を示すものではありません。BUFSIZE では、出力データセットの永久バッファサイズを指定します。これは、システムの性能を向上させるために調整できます。XPORT エンジンでは BUFSIZE=オプションがサポートされないため、BUFSIZE オプションに割り当てられているシステム値が使用されます。詳細については、動作環境付属ドキュメントを参照してください。

---

## 移送ファイルの検証方法

### 移送元コンピュータで移送ファイルを復元する

適切な方法(PROC COPY または PROC CIMPORT)を使用して、移送元コンピュータに移送ファイルを復元します。移送ファイルが移送元コンピュータのネイティブ形式に正常に変換されれば、転送される移送ファイルの整合性が検証されたこととなります。

この例では、移送ファイルの作成を示します。

```
libname xptlib xport 'xptlib.dat';
/* create a transport file for the entire library */
proc copy in=work out=xptlib;
run;
```

PROC COPY で、ライブラリ参照名 WORK からライブラリを読み込み、移送ファイルをライブラリ参照名 XPTLIB に書き込みます。

この例では、作成したばかりの移送ファイルを移送元コンピュータに復元します。

```
libname test 'test';
/* restore the transport file at the source computer */
proc copy in=xptlib out=test;
run;
```

移送ファイルを作成する例の OUT=オプションの値は、移送元コンピュータに移送ファイルを復元する例の IN=オプションの値になります。WORK に作成された元のデータライブラリが上書きされないようにするには、ライブラリ TEST に直接出力します。PROC COPY で、移送ファイルがライブラリ参照名 XPTLIB から読み込まれ、ネイティブ形式でライブラリ参照名 TEST に復元されます。

これらプロシジャの構文の詳細については、*Base SAS プロシジャガイド*を参照してください。

移送元コンピュータで SAS ログを表示して、このテストの結果を検証します。移送処理が移送元コンピュータで正常に実行された場合は、移送ファイルの内容が正しいと見なされます。移送処理が失敗した場合は、移送ファイルが正しく作成されなかったと見なされます。この場合は、移送ファイルを再作成し、移送元コンピュータでもう一度復元します。

### 移送ファイルのサイズの確認

動作環境のリストコマンドを使用して、移送ファイルが正常に作成されたことを検証します。OpenVMS の例を次に示します。

```
vms> dir/size=all *dat
```

```
Directory HOSTVAX: [JOE.XPTTEST]
```



```
XPTDS.DAT;1 7/8
XPTLIB.DAT;1 7/8
```

両方のファイルのサイズはブロックの 7/8 です。これは 448 バイトに相当します。

UNIX の例を次に示します。

```
$ ls -l *dat
-rw-r--r-- 1 joe mkt 448 Oct 13 14:24 xptds.dat
-rw-r--r-- 1 joe mkt 890 Oct 13 14:24 xptlib.dat
```

XPTDS.DAT のサイズは 448 バイト、XPTLIB.DAT のサイズは 890 バイトです。

ファイルサイズを一覧表示する方法は、動作環境によって異なります。

移送元コンピュータの移送ファイルのサイズと、移送先コンピュータに転送された移送ファイルのサイズを比較します。移送ファイルのサイズが同じであれば、ファイルが正常に転送されたと見なされます。サイズが同じでない場合は、ネットワーク転送が失敗したと見なされます。この場合は、転送オプションを確認してからもう一度転送してください。

### 元のデータセットと復元されたデータセットを比較する

移送元コンピュータの元のデータセットと移送先コンピュータで復元されたデータセットの差異を明らかにするには、CONTENTS プロシジャを使用します。比較によって、転送されたデータについての誤解が明らかになることがあります。たとえば、データセットを調べてみると、意図したデータセットのみを転送する代わりに、データセットのライブラリ全体が誤って転送されたことがわかることがあります。

メンバの種類が DATA の内容を一覧表示するには、CONTENTS プロシジャまたは PRINT プロシジャを使用します。

この例では、PROC CONTENTS で、ライブラリ内の 1 つのデータセットの内容を表示します。

#### 例のコード 13.22 PROC CONTENTS を使用してデータセットの内容を表示する

```
proc contents data=xptds._all_;
CONTENTS PROCEDURE

Data Set Name: XPTDS.GRADES Observations: .
Member Type: DATA Variables: 4
Engine: XPORT Indexes: 0
Created: . Observation Length: 32
Last Modified: . Deleted Observations: 0
Protection: Compressed: NO
Data Set Type: Sorted: NO
Label:

-----Alphabetic List of Variables and Attributes-----

# Variable Type Len Pos
-----
4 FINAL Num 8 24
1 STUDENT Char 8 0
2 TEST1 Num 8 8
3 TEST2 Num 8 16
DATAPROG: Creates data sets for TRANSPORTING
```

CONTENTS PROCEDURE

-----Directory-----

Libref: XPTDS

Engine: XPORT

Physical Name: \$1\$DUA330:[HOSTVAX.JOE.XPTTEST]XPTDS.DAT

# Name Memtype Indexes

-----

1 GRADES DATA

問題が検出された場合は、移送ファイルを再作成し、移送元コンピュータでもう一度復元します。

# 用語集

---

**CEDA**

あらゆるディレクトリベース動作環境(Solaris、Windows、HP-UX、OpenVMS など)で作成された SAS データファイルを、別のディレクトリベースの環境で実行する SAS セッションで読み込めるようにする SAS ソフトウェアの機能。中間の変換ステップを使用せずに、SAS データファイルにアクセスできます。略称: CEDA。

**Cross-Environment Data Access(クロス環境データアクセス、複合シーダ)**

CEDA を参照してください。

**data set**

SAS データセットを参照してください。

**DCB**

データコントロールブロックを参照してください。

**Extensible Markup Language**

XML を参照してください。

**fileref**

外部ファイルまたは集合的な保存場所(ディレクトリやフォルダなど)に一時的に割り当てられる名前。ファイルの場所または保存場所を SAS で識別するために使用されます。

**JCL**

Job Control Language を参照してください。

**Job Control Language(ジョブコントロール言語)**

z/OS および OS/390 の動作環境で使用する言語。データセット、実行時間およびジョブに必要なメモリ量などを含む情報をオペレーティングシステムに伝えるために使用されます。略称: JCL。

**libref**

SAS ライブラリの場所に関連付けられる SAS 名。たとえば、MYLIB.MYFILE という名前の場合、MYLIB が libref で MYFILE が SAS ライブラリ内のファイルです。

**SAS カタログ**

カタログエントリと呼ばれる下位の単位に多くの種類の情報を保存する SAS ファイル。1 つの SAS カタログに、複数の種類のカタログエントリを保存できます。

**SAS カタログエントリ**

SAS カタログ内の個別の格納単位。各エントリのエントリの種類が SAS での用途を示します。

**SAS データセット**

いずれかのネイティブ SAS ファイル形式の内容を含むファイル。SAS のデータセットには、SAS データファイルと SAS データビューの 2 種類があります。SAS データファイルには、データに関連付けられたディスクリプタ情報の他に、データ値が含まれます。SAS データビューには、ディスクリプタ情報と、他のソフトウェアベンダのファイル形式の内容を含むファイルや、他の SAS データセットからデータ値を取得するために必要な情報のみが含まれます。

**SAS データビュー**

SAS データセットの一種。他のファイルからデータを取得します。SAS データビューに含まれる情報は、データ型と変数(列)長のディスクリプタ情報に加え、他の SAS データセットや他のソフトウェアベンダのファイル形式で保存されているファイルからデータ値を取得するために必要な情報のみです。略称: データビュー。

**SAS データファイル**

SAS データセットの一種。データ値と、データに関連付けられたディスクリプタ情報を含みます。ディスクリプタ情報に含まれるのは、データを作成するために使用されたエンジン名に加えて、データ型、変数の長さなどの情報です。

**SAS ファイル名拡張子**

標準ファイル名識別子。次のファイル属性に関する情報を伝えます。1)ファイルの作成に使用された SAS エンジン、2)ファイルが作成されたときの動作環境アーキテクチャ、3)メンバの種類。ファイル名拡張子によって、アクセスする適切なファイルが識別されます。

**SAS ファイルの移送**

SAS ファイルを移送形式に変換し、互換性のない動作環境間を移動するプロセス。移送プロセスは、移送元動作環境で移送ファイルを作成し、それを移送先動作環境に転送してから移送先動作環境のネイティブ形式に復元します。移送先動作環境と移送先動作環境で異なるバージョンの SAS を実行している場合、移送プロセスで SAS ファイルの変換(以前のバージョンから新しいバージョンへの変換のみ)が暗黙的に実行されます。

**SAS ファイルの移動**

磁気メディアまたはネットワークなどの手段を介して、SAS ファイルを 1 つの動作環境から別の動作環境に渡すプロセス。SAS ファイルの具体的な移動方法には、変換、コピーおよび移送の 3 つがあります。

**SAS ファイルの転送**

磁気メディアまたはネットワークなどの手段を介して、SAS ファイルを転送元動作環境から転送先動作環境に渡すプロセス。

**SAS ファイルを戻す**

特定バージョンの SAS から以前のバージョンの SAS(たとえば、SAS 9 から SAS 6.12)に SAS ファイルを移動するプロセス。新しいバージョンで作成したファイルに、以前のバージョンでサポートされていない機能(一貫性制約など)が含まれている場合は、ファイルを戻せません。戻す代わりに、新しいバージョンの SAS を実行する動作環境でファイルを再作成します。

**SAS ライブラリ**

1 つ以上のファイルを 1 つの単位として定義し、認識およびアクセスできるようにしたものの。この単位で SAS からファイルを参照および格納できます。ライブラリのメンバは、それぞれのファイルです。

**TCP/IP**

1 組のネットワークプロトコルを指す略称。Transmission Control Protocol(TCP)は、Ethernet などのローカルエリアネットワーク上で情報を転送するための標準プロトコルです。TCP は、プロセス対プロセス間の情報を適切な順序で配信する機能を果たします。Internet Protocol(IP)は、動作環境間の接続を管理するプロトコルです。IP は特定の動作環境宛てにネットワークを介して情報を経路指定します。また、転送する情報の分割(フラグメント)と再組み立て(リアセンブル)を行います。

**V7 エンジン**

SAS 7 のデフォルトエンジン。SAS 7 データライブラリの SAS ファイルにアクセスします。SAS 9、SAS 8 および SAS 7 のファイル形式は同一です。

**V8 エンジン**

SAS 8 のデフォルトエンジン。SAS 8 データライブラリの SAS ファイルにアクセスします。SAS 9、SAS 8 および SAS 7 のファイル形式は同一です。

**V9 エンジン**

SAS 9 のデフォルトエンジン。SAS 9 データライブラリの SAS ファイルにアクセスします。SAS 9、SAS 8 および SAS 7 のファイル形式は同一です。

**XML**

内容、意味、用途のタグを付けて、情報を構造化するマークアップ言語。構造化された情報は、内容(ワードや数字など)と内容の役割を示すインジケータから構成されます。たとえば、ヘッダーセクションの内容にはデータベーステーブルの内容とは異なる位置付けがされます。短い形式は XML です。

**XML LIBNAME エンジン**

XML ドキュメントを処理する SAS エンジン。SAS 独自のファイル形式を XML マークアップに変換して、SAS データセットを XML ドキュメントにエクスポートします。また、XML マークアップを SAS データセットに変換して、外部 XML ドキュメントをインポートします。

**XPORT エンジン**

SAS 移送エンジン。移送形式の SAS ファイルにアクセスします。

**アクセスメソッド**

コミュニケーションアクセスメソッドを参照してください。

**移送エンジン**

動作環境固有の内部表現から移送形式に SAS ファイルを変換する機能。

**移送形式**

SAS データセット、SAS データライブラリおよび SAS カタログを 1 つの動作環境から別の動作環境に移動するために使用される 2 つのファイル形式のいずれか。1 つは、COPY プロシジャで XPORT エンジンを使用すると生成される移送形式です。もう 1 つは、CPORT プロシジャと CIMPORT プロシジャで生成される移送形式です。両移送形式とも、すべての動作環境で同じ形式です。

**移送ファイルのインポート**

移送先の動作環境に適した形式の SAS 移送ファイルを元の形式(SAS ライブラリ、SAS カタログまたは SAS データセット)に戻すプロセス。'インポート'および'復

元'はいずれもこのプロセスを表しますが、通常'インポート'は CIMPORT プロシジャを使用することを指します。

### 移送ファイルの復元

移送先の動作環境に適した形式の SAS 移送ファイルを元の形式(SAS ライブラリ、SAS カタログまたは SAS データセット)に戻すプロセス。復元処理は 2 通りの方法があります。1)COPY プロシジャで XPORT エンジンを使って作成された SAS 移送ファイルを COPY プロシジャで復元する方法と、2)CPORT プロシジャを使って作成された SAS 移送ファイルを CIMPORT プロシジャで復元する方法のいずれかから、適切な方法を使用します。移送ファイルの復元は、移送ファイルの読み込みまたはインポートとも呼ばれます。

### 一貫性制約

SAS データファイルの変数に保存するデータ値を制限するために指定するデータ検証ルールのセット。一貫性制約は、データの有効性と整合性を維持するのに役立ちます。

### エンジン

ファイルの読み込みまたは書き込みを行う SAS ソフトウェアのコンポーネント。SAS ではさまざまなエンジンを使用して各種のファイル形式を扱います。

### エントリの種類

SAS カタログエントリの特長。カタログエントリの構造と属性を SAS で識別するために使用されます。SAS カタログエントリを作成すると、自動的にエントリの種類が名前の一部に割り当てられます。

### 下位互換性

あるバージョンの SAS(SAS 9 や SAS 8 など)を実行する SAS クライアントで、以前のバージョン(SAS 6 など)を使って作成した SAS ファイルの読み込み、書き込み、および更新が可能なこと(ただし、クライアントのアプリケーションで長い名前対応などの新機能を使用しない場合)。新しいバージョンを実行する SAS クライアントとアプリケーションから、以前のバージョンを使って作成した SAS ファイルを利用できる場合を下位互換と言います。

### 外部ファイル

ホスト OS または別のベンダのソフトウェアアプリケーションにより作成、管理されるファイル。データと保存済みの SAS ステートメントの両方を外部ファイルから読み込みます。

### 外部ファイル形式

内部データ表現を、特定の動作環境のファイルに使われている表現との対比で表す用語。内部形式が同じではないファイル形式は、その動作環境の外部形式とみなされます。たとえば、OS/390 または z/OS の動作環境で作成されたファイル形式は、Windows 動作環境の外部形式とみなされます。外部ファイル形式は非ネイティブ(ノンネイティブ)ファイル形式と呼ばれることもあります。

### カタログエントリ

SAS カタログエントリを参照してください。

### クライアントセッション

クライアントコンピュータで実行中の SAS セッション。クライアントセッションは SAS ステートメントを受け入れ、サブミットされたステートメントを処理のためにサーバーに渡します。クライアントセッションは、クライアントセッションとサーバーセッションの両方の出力およびメッセージを管理します。

**項目ストア**

個別にアクセス可能な情報群で構成される SAS データセット。項目ストアの内容は、USS(UNIX System Services)または Windows で使用されているディレクトリ構造に似たディレクトリのツリー構造に整理されます。たとえば、特定の値の保存と検索にディレクトリパス(root\_dir/sub\_dir/value)を使用できます。SAS Registry は項目ストアの例です。

**コミュニケーションアクセスメソッド**

SAS とネットワークプロトコルの間のインターフェイス、または 2 つの動作環境間の接続に使用されるインターフェイス。SAS/SHARE および SAS/CONNECT は、動作環境に応じて TCP/IP または XMS をコミュニケーションアクセスメソッドを使用します。

**上位互換性**

あるバージョンの SAS を実行する SAS クライアントで、新しいバージョンを使って作成した SAS ファイルの読み込み、書き込みおよび更新が可能なおこと(ただし、SAS ファイルで、新しいバージョン固有の長い名前対応などの新機能を使用しない場合)。以前のバージョンを実行する SAS クライアントとアプリケーションから、新しいバージョンを使って作成した SAS ファイルを利用できる場合を上位互換と言います。

**精度**

データ精度を参照してください。

**データコントロールブロック**

z/OS などの IBM メインフレームオペレーティングシステムにおける、オペレーティングシステムのデータセットの物理特性に関する情報を保持するストレージ領域。略称: DCB。

**データ精度**

動作環境間で交換される SAS ファイルの数値データの信頼性。互換性のある動作環境(浮動小数値データの格納に同一の内部表現を使用する動作環境)間では、精度の劣化なしで数値データを交換できます。互換性のない動作環境間で数値データをやり取りすると、精度が劣化します。

**データビュー**

SAS データビューを参照してください。

**データ表現**

特定の動作環境でデータを保存するときの形式。動作環境が異なると、使用される標準や規則も異なります。たとえば、浮動小数点数の保存には IEEE や IBM 390、文字エンコーディングには ASCII や EBCDIC、メモリ内のバイト順にはビッグエンディアンまたはリトルエンディアン、ワード配置には 4 バイト境界または 8 バイト境界、データ型の長さには 16 ビット、32 ビットまたは 64 ビットなどの標準や規則があります。

**長い名前**

SAS 7 で実装された機能強化の 1 つ。使用できる名前の最大長が SAS 6 から拡張されました。拡張対象は、変数名、データセット名、プロシジャ名、オプション名、ステートメントのラベル名、libref の名前、fileref の名前です。長い名前で使用できる最大長は、名前の種類によって異なります。SAS 7 以降で作成されたファイルは SAS 6 動作環境で使用するときは、切り捨てルールが長い名前に適用されます。

**ネイティブファイル形式**

内部データ表現を、特定の動作環境のファイルに使われている表現との対比で表す用語。内部形式が同一のファイル形式は、その動作環境から見てネイティブフ

イル形式です。たとえば、Windows 動作環境で作成されたファイル形式は、Windows 動作環境から見るとネイティブ形式です。

#### ファイル参照

fileref を参照してください。

#### ファイル破損

ファイルのデータまたはファイルのヘッダーに対する変更操作の結果、ファイルの構造または内容にアクセスできなくなる。ファイルの移送時に破損が発生する一般的な原因としては、レコードの最後を示すキャリッジリターンまたはラインフィードが移送ファイル内で 1 つ以上不正に配置されている場合があります。このようなファイルがネットワークを介して転送されると、ファイル全体が読み込み不能になります。コミュニケーションソフトウェアによって論理レコード長やブロックサイズ、レコード形式などが変更される場合にも、ファイルが破損する可能性があります。

#### 変換テーブル(translation table)

動作環境に固有の SAS カタログエントリ。ある文字の値を別の文字の値に変換するために使用されます。アプリケーションで複数言語の使用をサポートするには、変換テーブルが必要となることが少なくありません。変換テーブルの例として、文字を EBCDIC から ASCII-ISO に変換するものがあります。

#### メンバの種類

SAS ファイルに保存されている情報の種類を識別するための SAS 名。メンバの種類には、ACCESS、AUDIT、DMBD、DATA、CATALOG、FDB、INDEX、ITEMSTOR、MDDb、PROGRAM、UTILITY、VIEW があります。

#### ライブラリ参照

libref を参照してください。



# キーワード

---

- 1**  
 16 進データ  
   z/OS 移送ファイルの読み込み 56
- B**  
 BINARY コマンド(FTP) 41  
 BLOCKSIZE=オプション  
   FILENAME ステートメント 41
- C**  
 CATALOG=オプション  
   CIMPORT プロシジャ 24, 25  
   CPORT プロシジャ 21  
 CATALOG=パラメータ  
   CIMPORT プロシジャ 74  
 CC=オプション  
   FILENAME ステートメント 50  
   LIBNAME ステートメント 50  
 CEDA(クロス環境データアクセス) 11  
   制限事項 12  
   非ネイティブファイルの読み込みと書き込み 17  
   ファイル形式の特定 16  
   ファイル形式の変更 13  
   ファイルの転送 16  
   利点 12  
 CIMPORT プロシジャ 19  
   CATALOG=オプション 24, 25  
   CATALOG=パラメータ 74  
   DATA=オプション 24, 77  
   EET=オプション 74  
   ET=オプション 74  
   EXCLUDE ステートメント 23, 25  
   INFILE=オプション 23  
   LIBNAME=オプション 24, 74  
   LIBRARY=オプション 23  
   LIBRARY=パラメータ 74  
   MEMTYPE=オプション 23, 74  
   SELECT ステートメント 23, 25  
   移送ファイルからのデータセットのインポート 23  
   移送ファイルの一貫性の検証 72  
   移送方法の併用 71  
   旧バージョンへの変換不可 20  
   ファイルの移動, z/OS から Windows へ 96  
 CONTENTS ステートメント  
   DATASETS プロシジャ 16  
 CONTENTS プロシジャ 16, 107  
   SAS エンジンの特定 64  
   SAS バージョンの特定 49, 53, 58, 60  
 COPY プロシジャ 28  
   EXCLUDE ステートメント 31, 32  
   IN=オプション 30, 32  
   JCL Batch から UNIX への移送 101  
   MEMTYPE=オプション 30  
   OUT=オプション 30, 32  
   SELECT ステートメント 30, 32, 35  
   移送ファイルの一貫性の検証 72  
   移送ファイルの作成 30  
   データセットからの XML ドキュメントの作成 35  
   データセットの復元 32, 36  
   ファイルの移動, OpenVMS から UNIX へ 86, 91  
   ファイルの移動, z/OS から Windows へ 98  
 CPORT プロシジャ 19  
   CATALOG=オプション 21  
   DATA=オプション 21  
   EET=オプション 22  
   ET=オプション 21  
   EXCLUDE ステートメント 21, 22  
   FILE=オプション 21  
   LIBNAME=オプション 74  
   LIBRARY=オプション 21, 77  
   MEMTYPE=オプション 21  
   NOCOMPRESS オプション 65, 74  
   SELECT ステートメント 22  
   SORTINFO=オプション 73

移送ファイル 20  
 移送ファイルの一貫性の検証 72  
 移送方法の併用 71  
 旧バージョンへの変換不可 20  
 サポートされていないエントリの種類  
     75  
 ファイルの移動, z/OS から Windows へ  
     93  
 ファイルヘッダー 65

**D**

DATA=オプション  
     CIMPORT プロシジャ 24, 77  
     CPORT プロシジャ 21  
 DATA=パラメータ  
     CIMPORT プロシジャ 74  
 DATA=パラメータまたは LIBRARY=パ  
     ラメータの指定 74  
 DATASETS プロシジャ  
     CONTENTS ステートメント 16  
 DATA ステップ  
     XML ドキュメントからデータセットを復  
     元する 36  
     移送ファイルの作成 30, 86, 91, 98,  
     101  
     移送ファイルの復元 32  
     データセットからの XML ドキュメントの  
     作成 34  
 dd コマンド(UNIX) 44, 61  
 DOWNLOAD プロシジャ 28

**E**

EET=オプション  
     CIMPORT プロシジャ 74  
     CPORT プロシジャ 22  
 ENCODING=システムオプション 7  
 ET=オプション  
     CIMPORT プロシジャ 74  
     CPORT プロシジャ 21  
 EXCLUDE ステートメント  
     CIMPORT プロシジャ 23, 25  
     COPY プロシジャ 31, 32  
     CPORT プロシジャ 21, 22

**F**

FILE=オプション  
     CPORT プロシジャ 21  
 FILENAME ステートメント  
     BLOCKSIZE=オプション 41  
     CC=オプション 50  
     FTP オプション 41  
     HOST=オプション 41  
     LRECL=オプション 41

PASS=オプション 41  
 RCMD=オプション 41  
 RECFM=オプション 41  
 SMTP オプション 41  
 SOCKET オプション 41  
 UMASK=オプション 41  
 URL オプション 41  
 USER=オプション 41  
 移送ファイルの属性の指定 41  
 テープへの移送ファイルの作成 61  
 FTP(ファイル転送プロトコル) 41  
     z/OS 54  
     非ネイティブファイルの転送 42  
     ファイルの転送 99  
 FTP オプション  
     FILENAME ステートメント 41

**G**

GET コマンド(FTP) 42  
 GRADES データセット(例) 7

**H**

HOST=オプション  
     FILENAME ステートメント 41

**I**

I/O 処理は完了されません 75  
 IN=オプション  
     COPY プロシジャ 30, 32  
 INFILE=オプション  
     CIMPORT プロシジャ 23  
 INITIALIZE コマンド(DCL) 49

**J**

JCL Batch から UNIX への移送(z/OS)  
     98

**L**

LIBNAME=オプション  
     CIMPORT プロシジャ 24, 74  
     CPORT プロシジャ 74  
 LIBNAME ステートメント  
     CC=オプション 50  
     OUTREP=オプション 14  
     テープへの移送ファイルの作成 61  
     ファイルの転送, 磁気媒体 44  
 LIBRARY=オプション  
     CIMPORT プロシジャ 23  
     CPORT プロシジャ 21, 77  
 LIBRARY=パラメータ  
     CIMPORT プロシジャ 74

- LISTD コマンド(TSO) 53, 95  
 LOCALE=システムオプション 7  
 LRECL=オプション  
   FILENAME ステートメント 41
- M**  
 MEMTYPE=オプション  
   CIMPORT プロシジャ 23, 74  
   COPY プロシジャ 30  
   CPORT プロシジャ 21  
 MSGLEVEL=システムオプション 16
- N**  
 NFTCOPY コマンド(DCL) 50  
 NLS(各国語サポート) 6  
 NOCOMPRESS オプション  
   CPORT プロシジャ 65, 74
- O**  
 OpenVMS 47  
   UNIX へのファイルの移動 86, 91  
   移送ファイル 49  
   移送ファイルの属性 47  
   エラーメッセージ 49  
   使用された SAS バージョンの特定 48  
   テープデバイスのマウント 49  
   ファイル属性の指定 48  
   ファイル属性の表示 47  
 OUT=オプション  
   COPY プロシジャ 30, 32  
 OUTREP=オプション  
   LIBNAME ステートメント 14  
 OUTREP=データセットオプション 13
- P**  
 PASS=オプション  
   FILENAME ステートメント 41  
 PATHWORKS 70  
 PUT コマンド(FTP) 42
- R**  
 RCMD=オプション  
   FILENAME ステートメント 41  
 RECFM=オプション  
   FILENAME ステートメント 41  
 rename コマンド(DOS) 58
- S**  
 SAS エンジン  
   使用されたバージョンの特定 64  
   特定 63  
 SAS 名 78  
 SAS バージョン, 特定  
   OpenVMS 48  
   UNIX 59  
   Windows 57  
   z/OS 53  
 SAS ファイルではありません 74  
 SAS ファイルのアクセス 3  
   CEDA 11  
   異なる言語環境 6  
   トラブルシューティング 70  
   非ネイティブファイルの更新 17  
   非ネイティブファイルの読み込みと書き込み 17  
   方法 4, 5  
 SAS ファイルの転送 3, 16, 39  
   CPORT プロシジャと CIMPORT プロシジャ 19  
   FILENAME ステートメント 41  
   FTP 41  
   OpenVMS から UNIX へ 86, 91  
   XPORT エンジン 27  
   z/OS JCL Batch から UNIX へ 98  
   z/OS から Windows へ 93  
   異なる言語環境 6  
   磁気媒体 43, 49, 61, 72  
   データセットからの XML ドキュメントのエクスポート 33  
   トラブルシューティング 70  
   ネットワークを経由した XML ドキュメント 35  
   方法 4, 5  
   例 85  
 SELECT ステートメント  
   CIMPORT プロシジャ 23, 25  
   COPY プロシジャ 30, 32, 35  
   CPORT プロシジャ 22  
 SEQUENTIAL\_FIXED 属性 70  
 SMTP オプション  
   FILENAME ステートメント 41  
 SOCKET オプション  
   FILENAME ステートメント 41  
 SORTINFO=オプション  
   CPORT プロシジャ 73  
 SQL プロシジャ 77  
 SQL プロシジャでは、V9 のビューを保存しません 77
- T**  
 TRANFILE コマンド(DCL) 49  
 type コマンド(DOS) 58

**U**

- UMASK=オプション
  - FILENAME ステートメント 41
- UNIX
  - JCL Batch から UNIX への移送 98
  - OpenVMS からのファイルの移動 86, 91
  - 移送ファイルのコピー 61
  - 使用された SAS バージョンの特定 59
  - テープへの移送ファイルの作成 61
  - ファイル属性の指定 59
- UNIX System Services Directory 54
- UPLOAD プロシジャ 28
- URL オプション
  - FILENAME ステートメント 41
- USER=オプション
  - FILENAME ステートメント 41
- UTILITY FILE OPEN 機能はサポートされていません 78

**V**

- V6 エンジン
  - 長すぎるメンバ名 75
- V9 ビュー 77
- VALIDVARNAME システムオプション 29, 75, 79

**W**

- Windows
  - z/OS からのファイルの移動 93
  - 暗号化されたデータ 58
  - 移送ファイル 58
  - エラーメッセージ 58
  - 使用された SAS バージョンの特定 57
  - ファイル属性の指定 57

**X**

- XML エンジン 33
  - 制限事項 34
  - 利点 34
- XML ドキュメント
  - 移送元マシンでの作成 34
  - データセットからのエクスポート 33
  - データセットからの作成 34, 35
  - データセットに復元する 36
  - ネットワーク経由の転送 35
- XML ドキュメントのエクスポート
  - データセット 33
- XML ドキュメントを復元する
  - データセット 36
- XPORT エンジン 27
  - UTILITY FILE OPEN 機能はサポートされていません 78

- 移送ファイルの作成 30
- 移送ファイルの復元 31
- 長すぎるメンバ名 75
- カタログファイルを開く機能 74
- 制限事項 28
- データセットの復元 32
- データセットを戻す 28
- ネットワークを経由した移送ファイルの転送 31
- ファイルの移動, OpenVMS から UNIX へ 86, 91, 98
- ファイルヘッダー 65
- 変数名の切り捨て 28
- 利点 28

**Z**

- z/OS
  - 16 進表現の移送ファイル 56
  - FTP 54
  - JCL Batch から UNIX への移送 98
  - UNIX System Services Directory 54
  - Windows へのファイルの移動 93
  - 移送ファイルの属性 53
  - 移送ファイルの転送 54
  - 移送ファイルの読み込み 55
  - 使用された SAS バージョンの特定 53
  - ファイル移送に使用するバッチステートメント 54
  - ファイル属性の表示 53
  - レコード長 54

**あ**

- 値 y コードは有効な SAS 名ではありません 78
- 暗号化したデータが無効です(Windows) 58
- 移送形式 27
- 移送ファイル 19, 22, 31, 70
  - 16 進データとして読み込む 56
  - OpenVMS 49
  - Windows 58
- XPORT エンジン 27
  - z/OS 54
  - z/OS での読み込み 55
  - 移送先マシンでの復元 22, 31
  - 移送元マシンでの作成 20
  - 一貫性の検証 72
  - 大きなファイルの分割 72
  - カタログとエントリの作成 20, 93
  - 形式とファイル属性の検証 79
  - 検証, 方法 106
  - サイズ, 検証 106
  - 再ブロック化 80
  - 作成 30

精度 100  
 データセットのインポート 23, 32  
 データセットの作成 20, 30, 93  
 データセットの復元 23, 32  
 テープへの作成 61  
 適切でない移送ファイル 73, 76  
 トラブルシューティング 70  
 内容の特定 22, 31  
 バイナリ形式 70  
 破損, 確認 75  
 ファイルヘッダー 65  
 移送ファイルが損傷している(OpenVMS)  
   49  
 移送ファイルの一貫性 72  
 移送ファイルの一貫性の検証 72  
 移送ファイルの検証 106  
 移送ファイルのサイズ 106  
 移送ファイルの再ブロック化 80  
 移送ファイルの属性 40  
   **関連項目:** 移送ファイルの属性  
   FILENAME ステートメントを使用して  
     指定する 41  
   OpenVMS 47  
   UNIX 59  
   Windows 57  
   z/OS 53  
   検証 79  
   通信ソフトウェアによる変更 71  
 移送ファイルの復元  
   CIMPORT プロシジャ 23  
   JCL Batch から UNIX への移送 101  
   XPORT エンジン 91  
   移送先マシン 22, 31  
   検証 106  
   トラブルシューティング 70  
   ファイルの内容の特定 22, 31  
 移送ファイルの読み込み  
   z/OS 55  
 移送方法 4, 5  
   互換性 71  
   使用された方法の特定 65  
 エラーメッセージ 73  
   OpenVMS 49  
   Windows 58  
 エンジン  
   使用されたエンジンの特定 63  
   バージョンの特定 64  
 エントリの種類がサポートされていま  
   せん 75  
 エントリの種類に互換性がありません  
   75  
 大きな移送ファイル  
   テープ用に小さなファイルに分割する  
   72

## か

カタログ, 移送ファイル  
   CPORT プロシジャ 20, 93  
   トラブルシューティング 75  
 カタログエントリ, 移送ファイル  
   CPORT プロシジャ 93  
 カタログエントリ, 移送ファイル  
   CIMPORT プロシジャ 25  
   CPORT プロシジャ 21  
 カタログファイルを開く機能はサポートさ  
   れていません 74  
 各国語サポート(NLS) 6  
 キャリッジリターン 70  
 クロス環境データアクセス  
   **参照項目:** CEDA(クロス環境データ  
   アクセス)  
 警告メッセージ 73  
 更新できません 78  
 異なる言語環境 6

## さ

磁気媒体 43  
   OpenVMS へのマウント 49  
   UNIX 61  
   大きなファイルの分割 72  
   ラベルなしテープ 43, 72  
 指定した機能をサポートしていません 78  
 相違, 元のデータセットと復元されたデー  
   タセット 107  
 損傷ファイル 75

## た

通信ソフトウェア 71  
 データ取得時の内部エラー 51, 76  
 データセット  
   SAS 6 形式に戻す 28  
   XML ドキュメントのエクスポート 33  
   XML ドキュメントの作成 34, 35  
   XML ドキュメントを復元する 36  
   移送ファイル 20, 30  
   移送ファイルからのインポート 23, 32,  
     96  
   移送ファイルからの復元 32  
   元のファイルと復元されたファイルとの  
   相違 107  
   例 7  
   データセットのインポート 23, 32  
   データセットの復元 32  
   データ破損, 確認 70, 75  
 テープ  
   **参照項目:** 磁気媒体  
 適切でない移送ファイル 73  
 転送形式とファイルの属性の検証 79  
 動作環境

- 通信ソフトウェアの起動 71
- 特定 63
- トラブルシューティング
  - 移送ファイルの再ブロック化 80
  - 移送ファイルの属性の検証 79
  - 移送ファイルの転送 70
  - 移送ファイルの復元 70
  - エラーメッセージと警告メッセージ 73
  - 形式の検証 79
- な**
- 長い変数名
  - 切り捨て 28
- 長い変数名の切り捨て 28, 75, 79
- 入力ファイルにはもっと多くのライブラリメンバが存在します 77
- は**
- バイナリ形式 70
- 破損, 確認 70, 75
- バッチステートメント
  - z/OS のファイル移送 54
- 非ネイティブファイル
  - 更新 17
  - 読み込みと書き込み 17
- 非ネイティブファイルの書き込み 17
- 非ネイティブファイルの更新 17
- 非ネイティブファイルの読み込み 17
- ビュー 77
- ファイルが損傷しています 75
- ファイル形式
  - 移送ファイル, 作成 27
  - 移送ファイル, 転送 19
  - 検証 79
  - 磁気媒体 43, 49, 61
  - 特定 16
  - バイナリ形式 70
  - 非ネイティブファイルの更新 17
  - 非ネイティブファイルの読み込みと書き込み 17
  - 変更 13
- ファイル名拡張子
  - 使用された SAS エンジンの特典 63
  - 使用された動作環境の特典 63
- ファイルに長すぎるメンバ名があります 75
- ファイルに使用できないメンバまたはライブラリ 51, 77
- ファイルの移動
  - 参照項目: SAS ファイルの転送
- ファイルの転送
  - データセットの作成 30
- ファイルヘッダー 65
- ブロックサイズ 40
  - 移送ファイルの再ブロック化 80
- 変数名
  - 切り捨て 28
- 方法
  - 移送ファイルの検証 106
  - 互換性 71
  - 使用された方法の特定 65
  - ファイルの移動とアクセス 4, 5
- ま**
- 無効なデータ長 76
- 無効な変数名です 79
- 命名規則 7, 78
- 戻す 20, 78
  - SAS 6 形式のデータセット 28
- や**
- ユーザー補助機能 8
- ら**
- ラベルなしテープ 43, 72
- 例**
- データセット 7
  - 命名規則 7
- レコード長
  - z/OS 54
- レコードの切り捨て 78
- レコードの切り捨てエラー 51, 78
- ログ**
- OpenVMS から UNIX への移送 87, 92
  - z/OS JCL Batch から UNIX へ 101, 103, 104, 105
  - z/OS から Windows への移送 94, 97
- 移送元マシンで表示する 15
- 論理レコード長 40