# Windows 版 SAS® 9.4

第 3 版

SAS® ドキュメント

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2014. *Windows 版 SAS® 9.4 第 3 版*. Cary, NC: SAS Institute Inc.

**Windows 版 SAS® 9.4 第 3 版**

Copyright © 2014, SAS Institute Inc., Cary, NC, USA

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

**For a hard-copy 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.

August 2014

SAS provides a complete selection of books and electronic products to help customers use SAS® software to its fullest potential. For more information about our offerings, visit **support.sas.com/bookstore** 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 trademarks of their respective companies.

# 目次

*本書について* . . . . . *ix*
*Windows 版 SAS 9.4 の新機能* . . . . . *xv*
*ユーザー補助* . . . . . *xix*

1 部 Windows 版 SAS の実行 1

***1 章 • Windows 版入門ガイド*** . . . . . **3**
SAS:Windows の機能を活用する . . . . . 4
SAS の起動 . . . . . 5
SAS で使用されるファイル . . . . . 14
SAS コードのサブミット . . . . . 25
SAS セッションを中断する . . . . . 28
SAS から Windows または MS-DOS のコマンドを実行する . . . . . 28
SAS プロセスを終了する . . . . . 32
SAS セッションを終了する . . . . . 33

***2 章 • Windows 版の SAS との通信*** . . . . . **35**
SAS インターフェイスの概要 . . . . . 36
SAS セッションでの作業 . . . . . 39
SAS セッションのカスタマイズ . . . . . 57
オンラインヘルプとドキュメントの利用 . . . . . 73

***3 章 • Windows 版での SAS エディタの使用*** . . . . . **79**
拡張エディタの使用 . . . . . 79
プログラムエディタの使用 . . . . . 107

***4 章 • Windows 版での SAS ファイルの使用*** . . . . . **117**
SAS ファイルについて . . . . . 118
マルチエンジンアーキテクチャ . . . . . 122
データライブラリの使用 . . . . . 125
複数の SAS セッションから SAS ファイルにアクセスする . . . . . 135
Windows 版 SAS 9.4 で他のバージョンのファイルを使用する . . . . . 136
SAS 9.4 のファイルを以前のリリースで使用する . . . . . 140
SAS 9.4 でリモートホストの SAS ファイルを使用する . . . . . 141
BMDP ファイル、OSIRIS ファイル、SPSS ファイルの読み取り . . . . . 141
SAS ファイルを動作環境間で転送する . . . . . 144
SAS/ACCESS を使用し、データベースファイルにアクセスする . . . . . 144
SAS ODBC Driver を使用し、他のアプリケーションから SAS データにアクセスする . . . . . 145

***5 章 • Windows 版での外部ファイルの使用*** . . . . . **147**
外部ファイルについて . . . . . 147
外部ファイルの参照 . . . . . 148
SAS ステートメントを使用し、外部ファイルにアクセス . . . . . 157
SAS コマンドを使用し、外部ファイルにアクセスする . . . . . 160
高度な外部 I/O の操作法 . . . . . 162

***6 章 • Windows 版での SAS 出力の管理*** . . . . . **165**
印刷 . . . . . 165
プロシジャ出力の Web ブラウザへの表示 . . . . . 180
ファイルへのプロシジャの出力と SAS ログの送付について . . . . . 183
SAS ログ機能を使用したログメッセージの書き込み . . . . . 185
グラフィックの作成 . . . . . 185

***7 章 • Windows 版でのパフォーマンスに関する注意点*** . . . . . **197**
ハードウェアに関する注意点 . . . . . 197
パフォーマンスを最適化する Windows の機能 . . . . . 199
パフォーマンスを最適化する SAS の機能 . . . . . 203
ネットワークのパフォーマンスに関する注意点 . . . . . 204
sasiotest ユーティリティ . . . . . 204
高度なパフォーマンス調整メソッド . . . . . 205

2 部 他の Windows アプリケーションでの SAS の使用 211

***8 章 • Windows 版 SAS での Windows システムツールの使用*** . . . . . **213**
SAS での Windows システムツールの使用について . . . . . 213
イベントビューアアプリケーションログ . . . . . 214
パフォーマンスツール . . . . . 217
SAS を Windows のサービスとして起動する . . . . . 222

***9 章 • Windows 版 SAS/AF ソフトウェアでの OLE の使用*** . . . . . **233**
OLE について . . . . . 234
SAS/AF カタログの互換性 . . . . . 234
OLE オブジェクトを FRAME エントリに挿入する . . . . . 234
FRAME エントリ内での OLE オブジェクトの編集 . . . . . 238
OLE Verb の起動 . . . . . 239
リンクされた OLE オブジェクトを使用する . . . . . 240
OLE オブジェクトの変換 . . . . . 241
OLE オブジェクトとアプリケーションの自動化 . . . . . 243
SAS/AF アプリケーションで OLE カスタムコントロール(OCX)を使用する . . . . . 250

***10 章 • OLE を使用した他のアプリケーションからの SAS 制御*** . . . . . **259**
SAS の自動化について . . . . . 259
SAS のインスタンスの作成 . . . . . 260
SAS セッションからのフィードバックの取得 . . . . . 260
OLE を用いた SAS の自動化の例 . . . . . 261
SAS OLE オートメーションオブジェクトで使用するメソッドとプロパティ . . . . . 262
ディクショナリ . . . . . 263

***11 章 • Windows 版動的データ交換の使用*** . . . . . **267**
動的データ交換(DDE)の概要 . . . . . 267
SAS での DDE 構文 . . . . . 268
DDE 外部ファイルの参照 . . . . . 269
DDE の例 . . . . . 270

***12 章 • Windows 版の名前の付いていないパイプと名前付きパイプの使用について*** . . . . . **279**
パイプの概要 . . . . . 279
名前の付いていないパイプの使用 . . . . . 280
名前付きパイプの使用 . . . . . 282

**13 章 • Windows 版 SAS から外部 DLL にアクセスする** . . . . . **291**
SAS のダイナミックリンクライブラリ(DLL)の概要 . . . . . 291
SASCBTBL 属性テーブル . . . . . 292
外部 DLL を使用する場合に特に注意すべき点 . . . . . 297
例 . . . . . 307

**14 章 • Windows 版での SAS/AF プログラマーのための特記注意事項** . . . . . **313**
SAS の表示画面と機能の制御 . . . . . 313
メイン SAS ウィンドウの制御 . . . . . 314
SAS から外部 DLL にアクセスする . . . . . 318
カスタムツールバーコントロールの設計、保存、ロード . . . . . 318
SAS/AF アプリケーションの自動起動 . . . . . 319
独自のロゴとアイコンを SAS/AF アプリケーションに関連付ける . . . . . 319

3 部 Windows 版の SAS 言語機能 321

**15 章 • Windows 版に固有のデータセットオプション** . . . . . **323**
Windows 版に固有の SAS データセットオプション . . . . . 323
ディクショナリ . . . . . 323

**16 章 • Windows 版に固有の SAS コマンド** . . . . . **325**
Windows 版に固有の SAS コマンド . . . . . 326
ディクショナリ . . . . . 327

**17 章 • Windows 版に固有の SAS 出力形式** . . . . . **377**
Windows 版に固有の SAS 出力形式 . . . . . 377
バイナリデータの書き込み . . . . . 377
以前のリリースのユーザー定義入力形式を SAS 9.4 で利用する . . . . . 378
ディクショナリ . . . . . 378

**18 章 • Windows 版に固有の SAS 関数と CALL ルーチン** . . . . . **389**
Windows 版に固有の SAS 関数と CALL ルーチン . . . . . 389
ディクショナリ . . . . . 390

**19 章 • Windows 版に固有の SAS 入力形式** . . . . . **415**
Windows 版に固有の SAS 入力形式 . . . . . 415
バイナリデータの読み取り . . . . . 416
以前のリリースのユーザー定義入力形式を SAS 9.4 用に変換する . . . . . 416
ディクショナリ . . . . . 417

**20 章 • Windows 版に固有の SAS プロシジャ** . . . . . **425**
Windows 版に固有の SAS プロシジャ . . . . . 425
ディクショナリ . . . . . 425

**21 章 • Windows 版に固有の SAS ステートメント** . . . . . **443**
Windows 版に固有の SAS ステートメント . . . . . 443
ディクショナリ . . . . . 443

**22 章 • Windows 版に固有の SAS システムオプション** . . . . . **471**
Windows 版に固有の SAS システムオプション . . . . . 473
SAS システムオプション設定の表示 . . . . . 474
SAS システムオプション設定の変更 . . . . . 475
複数の場所に設定されたシステムオプションの処理 . . . . . 477
カテゴリごとの SAS システムオプション . . . . . 477

ディクショナリ . . . . . 486

***23 章 • Windows 版での変数の長さと精度*** . . . . . **595**
Windows 版での変数の長さと精度 . . . . . 595
数値変数 . . . . . 595
文字変数 . . . . . 596

***24 章 • Windows 版に固有の SAS マクロ機能*** . . . . . **599**
Windows 版に固有の SAS マクロ機能 . . . . . 599
自動マクロ変数 . . . . . 599
マクロステートメント . . . . . 601
マクロ関数 . . . . . 602
自動呼び出しライブラリ . . . . . 602

4 部 付録 605

***付録 1 • Windows 版で OLE オブジェクトを自動化するための SCL メソッド*** . . . . . **607**
OLE クラスメソッドの要約 . . . . . 607
ディクショナリ . . . . . 608

***付録 2 • Windows 版に固有の SAS エラーメッセージ*** . . . . . **617**
SAS エラーメッセージの概要 . . . . . 617
リターンコードと完了ステータス . . . . . 617
ファイルへのアクセス . . . . . 618
SAS 機能の使用 . . . . . 619
OLE の使用 . . . . . 620
ネットワークの使用 . . . . . 621
内部エラーの解決 . . . . . 622
オペレーティングシステムと Windows のエラーメッセージの解決 . . . . . 623
初期化と終了のエラーメッセージ . . . . . 623

***付録 3 • Windows 版でのグラフィックに関する注意点*** . . . . . **625**

***付録 4 • インタラクティブ SAS セッションにおける Windows 版でのデフォルトキー設定*** . . . . . **627**
Windows 版でのデフォルトキー定義 . . . . . 627
メイン SAS ウィンドウでのショートカットキー . . . . . 629
拡張エディタでのショートカットキー . . . . . 631
印刷プレビュー内でのショートカットキー . . . . . 635

***付録 5 • Cleanwork ユーティリティ*** . . . . . **637**
Cleanwork ユーティリティ . . . . . 637
Microsoft タスクスケジューラを使用した Cleanwork ユーティリティのスケジュール設定 . . . . . 638
ディクショナリ . . . . . 638

***付録 6 • Sasiotest ユーティリティ*** . . . . . **641**
Sasiotest ユーティリティ . . . . . 641
ベストプラクティスと考慮事項 . . . . . 642
許容可能な SAS I/O パフォーマンス . . . . . 643
ディクショナリ . . . . . 643

***付録 7 • ASCII システムでの EBCDIC データの使用*** . . . . . **647**
About EBCDIC and ASCII Data . . . . . 647
Moving Data from EBCDIC to ASCII Systems . . . . . 649

*推奨資料* . . . . . 657
*用語集* . . . . . 659
*キーワード* . . . . . 669

# 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。複数の引数を含む構文で区切り記号を使用する場合と使用しない場合を説明するために、引数の構文の形式が複数示されています。

キーワード
プログラムの作成ときに使用する 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。

これらの例の SAS 構文では、キーワードには太字が使用されています。

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

**CALL RANBIN** (*seed, n, p, x*);

**ALTER** (*alter-password*)

**BEST** *w.*

**REMOVE** <*data-set-name*>

この例では、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**

プロシジャステートメントによっては、ステートメント構文中に複数のキーワードが含まれます。

**CREATE** <UNIQUE> **INDEX** *index-name* **ON** *table-name* (*column-1* <, *column-2*, …>)

*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、オプションの引数は山かっこ( < > )で囲まれます。

この例では、*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。この例の SAS コードでは、引数 *string* の値は 'summer'、引数 *position* の値は 4 です。

```
x=char('summer', 4);
```

この例では、*string* および *substring* は必須引数ですが、*modifiers* と *startpos* はオプションです。

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

*argument(s)*

引数は必ず 1 つ必要であり、複数の引数が許可されます。引数の間はスペースで区切ります。カンマ( , )などの区切り記号は、引数間に必要ありません。

たとえば、MISSING ステートメントは、この形式で複数の引数を含みます。

**MISSING** *character(s)*;

<LITERAL_ARGUMENT> *argument-1* <<LITERAL_ARGUMENT> *argument-2 ...* >

引数は必ず 1 つ必要であり、リテラル引数がこの引数に関連付けられます。リテラルと引数のペアは複数指定できます。リテラルと引数の間に区切り記号は必要ありません。省略記号(...)は、追加のリテラルと引数が許可されることを示します。

たとえば、BY ステートメントはこの引数を含みます。

**BY** <DESCENDING> *variable-1* <<DESCENDING> *variable-2* …>;

*argument-1 <option(s)> <argument-2 <option(s)> ...>*

引数は必ず 1 つ必要であり、1 つ以上のオプションがこの引数に関連付けられます。複数の引数と関連するオプションを指定できます。引数とオプションの間に区切り記号は必要ありません。省略記号(...)は、追加の引数と関連するオプションが許可されることを示します。

たとえば、FORMAT プロシジャの PICTURE ステートメントは、この形式で複数の引数を含みます。

**PICTURE** name <(*format-option(s)*)>
<*value-range-set-1* <(*picture-1-option(s)*)>
<*value-range-set-2* <(*picture-2-option(s)*)> …>>;

*argument-1=value-1 <argument-2=value-2 ...>*
引数には値を割り当てる必要があり、複数の引数を指定できます。省略記号(...)は、追加の引数が許可されることを示します。引数間に区切り記号は必要ありません。

たとえば、LABEL ステートメントは、この形式で複数の引数を含みます。

**LABEL** *variable-1=label-1 <variable-2=label-2 …>*;

*argument-1 <, argument-2, ...>*
引数は必ず 1 つ必要であり、カンマまたは別の区切り記号で区切って複数の引数を指定できます。省略記号(...)は、カンマで区切られた引数が続くことを示します。SAS ドキュメントでは両方の形式が使用されます。

次に、この形式で指定された複数の引数の例を示します。

**AUTHPROVIDERDOMAIN** (*provider-1:domain-1 <, provider-2:domain-2, …>*

**INTO** :*macro-variable-specification-1 <, :macro-variable-specification-2, …>*

*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。この例では、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。

この 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。

- 非リテラル引数。この LINK ステートメントの例では、引数 *label* はユーザー指定値のため、斜体で表示されます。

  **LINK** *label*;

- 引数に割り当てられる非リテラル値。

  この 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(s)* <*format* > <DEFAULT = *default-format*>;

## *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。

この MAPS システムオプションの例では、等号により MAPS の値が設定されます。

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

< >
山かっこはオプションの引数を示します。必須引数は山かっこで囲みません。

この 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-2*, …>)

|
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。

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

**CMPMODEL**=BOTH | CATALOG | XML

...
省略記号は、引数の繰り返しが可能なことを示します。引数と省略記号が山かっこで囲まれている場合、その引数はオプションです。繰り返される引数には、その引数の前や後ろに、区切り記号を入れる必要があります。

この CAT 関数の例では、複数の *item* 引数が許可され、カンマで区切る必要があります。

**CAT** (*item-1* <, *item-2*, …>)

'*value*'または"*value*"
一重引用符や二重引用符付きの引数は、その値にも一重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。

この FOOTNOTE ステートメントの例では、引数 *text* に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。

この例では、各ステートメントが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、通常、参照の作成に 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* を引用符で囲んで使用します。

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

# Windows 版 SAS 9.4 の新機能

## 概要

Windows 版 SAS では、デスクトップや企業内に配置されている Windows アプリケーションを利用して、データタスクや計算タスクを完了します。Windows のツールや技術を使用して情報を共有できます。

## 全般的な拡張

次の拡張が 9.4 では実装されています。

- ソート操作の説明をより適切にするため、“ソートされるファイルの場所の選択” (206 ページ)のコンテンツが修正されました。
- GIMPORT プロシジャへの参照が 削除されました。GIMPORT は新しい TrueType フォントでは機能しません。
- 付録として、JRE (Java Runtime Environment)エラーのトラブルシューティングが削除されました。説明は http://support.sas.com/kb/48/548.html の SAS ノートにあります。
- 拡張属性の説明は、“SAS データセット(メンバタイプ:Data または View)” (120 ページ)に記載されています。
- LRECL= オプションのデフォルト値は 32767 です。
- “Windows 版での SAS データ配信のための Lotus Notes の使用”という章が削除されました。Lotus Notes は WX6 ではサポートされません。
- Windows XP、Windows Server 2003、および Windows Vista はもうサポートされません。サポートされる Windows エディションのリストは、“エンタープライズ環境での SAS の実行” (4 ページ)を参照してください。
- SAS 9.4 の最初のメンテナンスリリースでは、Windows 32 ビットアーキテクチャのサポートの詳細が追加されました。
- また、SAS 9.4 の最初のメンテナンスリリースでは、ユーザー補助機能のトピックが削除されました。変更された ユーザー補助機能のトピックは、Base 製品ページで参照できます。
- SAS 9.4 の最初のメンテナンスリリースでは、付録 “Sasiotest ユーティリティ” (641 ページ)が追加されました。

- SAS 9.4 の 2 番目のメンテナンスリリースでは、付録 7, “ASCII システムでの EBCDIC データの使用” (647 ページ)が追加されました。

## ドキュメント機能改善

TOOLSMENU および VIEWMENU システムオプションのドキュメントは、*Windows 版 SAS* に配置されました。ドキュメントは *SAS システムオプション: リファレンス* から削除されました。

## Cleanwork ユーティリティ

SAS 9.4 メンテナンスリリース第 2 版では、Cleanwork ユーティリティが更新されました。コンソールベースのアプリケーションになり、Base SAS の一部として、SASROOT ディレクトリにインストールされます。詳細については、“Cleanwork ユーティリティ” (637 ページ)を参照してください。

## 関数

SAS 9.4 の最初のメンテナンスリリースでは、PEEKLONG 関数に次の手順が追加されました。「SAS サーバーがロックダウン状態の場合は、PEEKLONG 関数は実行されない」詳細については、“PEEKLONG 関数:Windows” (409 ページ).を参照してください。

## ステートメント

SAS 9.4 の 2 番目のメンテナンスリリースでは、次のステートメントに PERMISSION オプションが追加されました。FILE ステートメント (445 ページ)および FILENAME ステートメント (448 ページ)

SAS 9.4 の 2 番目のメンテナンスリリースでは、SAS がロックダウン状態の場合に、次の FILENAME ステートメントのアクセスメソッドは使用できません。詳細については、FILENAME ステートメント (448 ページ)を参照してください。

- EMAIL
- FTP
- Hadoop
- SOCKET
- URL

ただし、サーバー管理者は、ロックダウン状態でアクセスできるように、これらのアクセスメソッドを再度有効にできます。

# システムオプション

システムオプションの概要表 は、カテゴリ別 SAS システムオプション表 (477 ページ) と置き換えられました。

次のシステムオプションは 新しいものです。

ALIGNSASIOFILES (p. 486)
SAS データセットでデータのページを調整します。

FILELOCKWAIT (p. 506)
ロックされたファイルが使用できるようになるまで SAS が待機する秒数を設定します。

HOSTINFOLONG (p. 520)
SAS 起動時に SAS ログの追加動作環境情報の記述を指定します。

SAS 9.4 の 2 番目のメンテナンスリリースでは、RTRACELOC システムオプションに次のヒントが追加されました。「%p (pid)、%d (date)、%t (time)が指定されている場合は、RTRACELOC ファイル名を拡張できます。」詳細については、“RTRACELOC システムオプション:Windows” (555 ページ)を参照してください。

SAS 9.4 の 2 番目のメンテナンスリリースでは、SORTSIZE オプションのデフォルト値が 1G に変更されました。詳細については、“SORTSIZE システムオプション:Windows” (569 ページ)を参照してください。

# ユーザー補助

この製品のユーザー補助の詳細については、SAS 9.4 の Windows 環境のユーザー補助機能(support.sas.com)を参照してください。を参照してください。

# 1 部

# Windows 版 SAS の実行

1 章
**Windows 版入門ガイド** . . . . . 3

2 章
**Windows 版の SAS との通信** . . . . . 35

3 章
**Windows 版での SAS エディタの使用** . . . . . 79

4 章
**Windows 版での SAS ファイルの使用** . . . . . 117

5 章
**Windows 版での外部ファイルの使用** . . . . . 147

6 章
**Windows 版での SAS 出力の管理** . . . . . 165

7 章
**Windows 版でのパフォーマンスに関する注意点** . . . . . 197

*1 章*

# Windows 版入門ガイド

**SAS:Windows の機能を活用する** . . . . . **4**
エンタープライズ環境での SAS の実行 . . . . . 4
Windows 環境での統合要素 . . . . . 4
互換性と保守容易性 . . . . . 4
Java の起動 . . . . . 5

**SAS の起動** . . . . . **5**
SAS を対話モードまたはバッチモードで使用 . . . . . 5
スタートメニューからの起動 . . . . . 8
カスタムショートカットまたはプログラムアイテムからの起動 . . . . . 8
[ファイル名を指定して実行]ダイアログボックスまたはコマンドプロンプトからの起動 . . . . . 8
SAS ファイルからの起動 . . . . . 9
SAS バッチジョブのサブミット . . . . . 10
SAS 起動時にプログラムエディタを開始 . . . . . 10
SAS 起動時の作業フォルダの設定 . . . . . 11
サンプル SAS セッション . . . . . 11
SAS が起動しない場合について . . . . . 14

**SAS で使用されるファイル** . . . . . **14**
SAS で使用されるファイルについて . . . . . 14
SAS 構成ファイル . . . . . 15
SAS 自動実行ファイル . . . . . 20
Profile カタログ . . . . . 22
Work データライブラリ . . . . . 23
SAS レジストリファイル . . . . . 23
SAS デフォルトフォルダの構造 . . . . . 24

**SAS コードのサブミット** . . . . . **25**
SAS コードのサブミットについて . . . . . 25
拡張エディタまたはプログラムエディタからコードをサブミット . . . . . 25
SAS NOTEPAD テキストエディタからコードをサブミット . . . . . 25
バッチモードでの SAS の実行 . . . . . 26
クリップボードからコードをサブミット . . . . . 26
ドラッグアンドドロップを使用したコードのサブミット . . . . . 27
登録された SAS ファイルタイプに保存されたコードをサブミット . . . . . 28

**SAS セッションを中断する** . . . . . **28**

**SAS から Windows または MS-DOS のコマンドを実行する** . . . . . **28**
SAS からの Windows コマンドまたは MS-DOS コマンド実行の概要 . . . . . 28
X ステートメントまたは X コマンドを使用して Windows コマンド実行 . . . . . 29

DATA ステップを使用して、条件付きオペレーティングシステムコマンドを条件付きで発行 . . . . . 30
XWAITEMAILDLG システムオプション: . . . . . 30
XSYNC システムオプション . . . . . 31
XWAIT システムオプションと XSYNC システムオプションの比較 . . . . . 31
**SAS プロセスを終了する** . . . . . **32**
**SAS セッションを終了する** . . . . . **33**

# SAS:Windows の機能を活用する

## *エンタープライズ環境での SAS の実行*

Windows エンタープライズ環境では、柔軟で使いやすい操作環境が実現されています。また、SAS をご利用のエンタープライズソリューションに統合することができます。次に、SAS 9.4 でサポートされる Window 環境を示します。

http://support.sas.com/supportos/list

次の URL にアクセスすると、動的データ交換(DDE)によってサポートされるアプリケーションにアクセスできます。Word および Excel アプリケーションがサポートされています。PowerPoint アプリケーションはサポートされていません。

http://support.microsoft.com/kb/142821

## *Windows 環境での統合要素*

Windows 版 SAS は、すでにデスクトップや企業で使用されている Windows アプリケーションを利用しながら、データ集約型の処理や演算集約型の処理を完了するように設計されています。SAS では、Windows が提供しなければならない最も強力なツールと技術を使用した情報共有がサポートされています。これらのツールと技術には、次が含まれます。

- OLE
- 動的データ交換(DDE:Dynamic Data Exchange)
- ODBC (Open Database Connectivity )
- 電子メールシステム
- パイプと名前付きパイプ
- Windows クリップボード

## *互換性と保守容易性*

### *以前のリリースからの SAS データセットの読み込み/書き込み*

SAS では、以前のリリースで作成された SAS データセットの読み込み、書き込みが可能です。

ただし、SAS カタログアーキテクチャの更新や動作環境構造の相違に対応するため、移送プロシジャ CPORT と CIMPORT を使用して、以前の形式(Windows 版のリリース 9.1 など)のカタログを SAS 9.4 の形式に変換する必要があります。

### グラフィカルインターフェイスまたはコマンド行インターフェイスの使用

これまでのリリースと同じように、コマンド行を使用することができます。しかし、グラフィカルユーザーインターフェイス(GUI)を使用して、コマンドを発行することも可能です。GUI からは、既存のほとんどの SAS コマンドとウィンドウを使用することができます。操作の選択時に、ダイアログボックスや他のさまざまな GUI コントロールを使用する場合もあります。

## Java の起動

SAS では JRE (Java Runtime Environment)を使用して Java を起動し、Java コードが実行されるようにします。SAS の一部は Java で書かれています。JRE には Java を実行するのに必要なライブラリが含まれています。

SAS 9.3 では、ODS グラフィックプロシジャは SAS/GRAPH から Base SAS へ移行しました。Java に依存するこれらのプロシジャは独立した SAS 製品としてではなく、Base SAS の一部として提供されます。その結果として、SAS 9.4 ウィンドウ環境での主要な相違点は、SAS では起動の際に Java 環境が初期化されるようになりました。

HTML がデフォルトの SAS 9.4 ウィンドウ環境出力となり、リストの出力先を置き換えます。SAS ウィンドウ環境のバッチモードで生成されるデフォルトの HTML ODS 出力先のデフォルトスタイルは、Styles.Default から Styles.HTMLBlue に変更されました。

Graph Template Language および SGPLOT、SGPANNEL、SGSCATTER、SGRENDER のような SAS/GRAPH プロシジャは、SAS/GRAPH プロシジャから Base SAS へと移行したため、Java は SAS 9.4 で Base SAS の一部として実装されます。デフォルトでは、Windows および UNIX オペレーティングシステムの SAS ウィンドウ環境でプロシジャが実行される際に ODS グラフをサポートする SAS プロシジャには、ODS GRAPHICS ステートメントが ON に設定されます。ODS グラフはこれらのプロシジャに対して有効化されていますが、グラフィック環境を制御するには ODS GRAPHICS ステートメントを使用してください。

SAS 9.4 で SAS のウィンドウ環境 (双方向 SAS) を使う場合、プログラム(PROC javainfo; run;)を実行するときに **SAS ログ** ログウィンドウ内で次のようなエラーメッセージが表示される場合があります。

```
ERROR:Unable to attach current thread" ERROR:Unable to load the java virtual machine" ER
```

これらのエラーは Java Runtime Environment (JRE)に関連する典型的な問題です。これらの問題を解決するには、次のステップに従います。http://support.sas.com/kb/48/548.html

# SAS の起動

## SAS を対話モードまたはバッチモードで使用

Windows 版 SAS の実行時、対話型セッションを起動してプログラムのサブミットや出力結果の表示を行ったり、SAS バッチジョブを実行して後で出力を表示したりできます。

デフォルトでは、SAS を起動すると対話型 SAS セッションが開始されます。バッチジョブとしてサブミットしたい SAS プログラムがある場合は、SAS 起動時に、SYSIN システムオプションに SAS プログラムファイルの名前を指定します。

初めて SAS を対話型セッションとして起動すると、入門ガイドを利用して基本的なタスクの学習をするかどうかが尋ねられます。入門ガイドを開始する場合は、**入門ガイドの開始**をクリックします。入門ガイドの利用を確認するダイアログが表示されないように

するには、**次回からこのダイアログを表示しない**を選択します。**ヘルプ** ⇨ **SAS ソフトウェア入門ガイドを選択すると、いつでも入門ガイドを開始できます。**

次の例では、Window 7 での SAS のバッチジョブの処理を示します。

- SAS が適切に登録されている場合は、エクスプローラを開いて、実行するプログラムを右クリックします。初めて SAS をインストールしたときに、SAS は正常に登録される必要があります。SAS が正常に登録されない場合、次のページの例を参照してください。バッチサブミットオプションを選択します。このオプションはプログラムをバッチジョブとして実行します。.デフォルトでは、LOG ファイルと LST ファイルは、プログラムと同じフォルダにあります。

  LOG ファイルは実行されたプログラムのログです。

  LST ファイルは実行されたプログラムの出力ファイルです。プログラムへの出力がある場合にのみ、このファイルは作成されます。

  エクスプローラ内でダブルクリックして、バッチモードで.sas ファイルを実行することもできます。sas ファイルへのデフォルト操作は**バッチサブミット**に設定する必要があります。

  .sas ファイルへのデフォルト操作を変更するには、次の操作を実行してください。

  1. **Windows エクスプローラ**を開きます。
  2. .sas ファイルを選択します。
  3. **プログラムから開く** を右クリックし、**既定のプログラム**を選択します。
  4. **Windows 用 SAS システム**を選択します。
  5. .sas ファイルをダブルクリックして、バッチモードで SAS プログラムを実行します。LOG ファイルおよび LST ファイルは.sas ファイルと同じフォルダに作成されます。

- この例では、-SYSIN オプション付きの SAS 実行コマンドを使います。

  1. **スタート** ⇨ **ファイル名を指定して実行**を選択します。
  2. 次に示すものと似たコマンドを入力します。**`"c:\Program Files\SASHome\SASFoundation\V9.4\Sas.exe" -sysin c:\test.sas`**。プログラムの場所を指定します。この例では、C ドライブの root にある TEST.SAS プログラムを実行します。スプラッシュスクリーンを表示しないようにするには、–NOSPLASH オプションを使います。-ICON オプションを加えて、プログラムが起動したときに **DOS** ウィンドウ を最小化します。.LOG ファイルと LST ファイルは SAS root ディレクトリに出力されます。

     -SYSIN オプションはバッチモードで実行される SAS プログラムを指定します。パスには、有効な Windows のパス名を指定します。

- 次の例では BAT ファイルを使用します。.BAT ファイルはオペレーティングシステムが実行します。

  1. ノートパッドや同じようなテキストエディタでファイルを作成します。
  2. 次に示すものと似たコマンドを入力します。**`"c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe" -sysin c:\test.sas`**.
  3. To execute the BAT file, select BAT ファイルを実行するには、**スタート** ⇨ **ファイル名を指定して実行**を選択するか、または、BAT ファイルをダブルクリックします。

     LOG ファイルと LST ファイルは.BAT ファイルのあるフォルダに出力されます。.LOG ファイルと LST ファイルの出力先を変更するには、-LOG オプションと-PRINT オプションを使用します。

-PRINT オプションを使用して、プログラムの出力先フォルダを変更します。-LOG オプションを使用して、実行されたプログラムのログの出力先フォルダを変更します。**"C:\Program Files\SASHome\SASFoundation\9.4\Sas.exe" -SYSIN c:\testprog.sas -NOSPLASH -ICON -PRINT c:\test.lst -LOG c:\test.log**

*注:* BAT プログラムを実行すると **DOS** ウィンドウが開き、ジョブが終了するまで閉じられません。

BAT ファイルで複数のプログラムを処理することが可能です。次の例では、同時に 5 つのプログラムが実行されています。

```
"c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe" –sysin c:\test1.sas
"c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe" –sysin c:\test2.sas
"c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe" –sysin c:\test3.sas
"c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe" –sysin c:\test4.sas
"c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe" –sysin c:\test5.sas
```

- この例では、Windows で同時にバッチジョブを実行するための START コマンドの使用方法について示します。START コマンドの使用には、タイトルが必須です。タイトルは二重引用符で囲むことができますが、この例では Job#を使用しています。

```
Start "Job1" "c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe"
–sysin c:\test1.sas
Start "Job2" "c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe"
–sysin c:\test2.sas
Start "Job3" "c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe"
–sysin c:\test3.sas
Start "Job4" "c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe"
–sysin c:\test4.sas
Start "Job5" "c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe"
–sysin c:\test5.sas
```

- また、5 つのプログラムを連続して実行するバッチ(TEST.BAT)ファイルを作成することもできます。コマンドラインで–PRINT オプションと–LOG オプションを使用しない場合は、ログ(.LOG)と出力ファイル(.LST)は BAT ファイルがあるディレクトリに置かれます。

  この例は Windows に適用されます。.BAT ファイルで次のコマンドを入力します。

```
"c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe"
–sysin c:\test1.sas
"c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe"
–sysin c:\test2.sas
"c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe" –sysin c:\test3.sas
"c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe"
–sysin c:\test4.sas
"c:\Program Files\SASHome\SASFoundation\9.4\Sas.exe"
–sysin c:\test5.sas
```

- スケジューラーを使用して、指定した時間に SAS ジョブを実行することができます。スケジュールに入れられる場合、プログラムは操作者の補助なし実行されます。**スケジュールされたタスク**プログラムは、Windows の**コントロールパネル** にあります。

  1. **スケジュールされたタスク**アイコンをダブルクリックして、**スケジュールタスクウィザード**を起動します。
  2. **参照**ボタンを選択して、TEST.BAT ファイルを選択します。

3. このタスクがどれくらいの頻度で実行されるのかを指定します。**次へ**を選択します。
4. プログラムを実行する日付と時間を選択します。**次へ**を選択します。
5. このプログラムに適用されるユーザー名とパスワードを入力します。**次へ**を選択します。確認のメッセージが表示されます。
6. **完了**を選択します。

### *スタートメニューからの起動*

Windows の**スタート**メニューから SAS を起動するには、次の操作を行います。

1. **スタート**をクリックします。
2. **プログラム**を選択します。
3. **SAS** を選択します。
4. **SAS 9.4 (言語)**を選択します。

### *カスタムショートカットまたはプログラムアイテムからの起動*

インストール時に、セットアッププログラムにより、SAS を起動できるプログラムアイコンが**スタート**メニューに自動的に作成されます。1 つのフォルダ内に、異なる構成の SAS セッションを起動する複数のプログラムアイコンを作成できます。また、Windows を起動するたびに SAS を起動したい場合は、スタートアップフォルダにプログラムアイコンまたはショートカットを置きます。ショートカットの作成に関する詳細は、Windows のドキュメントを参照してください。

SAS を起動するショートカットを作成した後、SAS コマンドにシステムオプションを追加することができます。システムオプションを追加するには、次の操作を行います。

1. **SAS のプロパティ**ウィンドウを開いて**ショートカット**タブをクリックします。
2. **リンク先**フィールドで、SAS コマンドにシステムオプションを追加します。パス名は、二重引用符で囲む必要があります。次の例では、二重引用符を使用します。

```
"c:\program files\SASHome\SASFoundation\9.4\sas.exe" -config "c:\mydir\sasv9.cfg"
```

*注:* システムオプションの値にスペースが含まれる場合、コマンドラインまたは config.ファイルでは引用符で値を囲む必要があります。次の例では正しい構文が示されます。

```
-bottommargin '2 in';
```

### *[ファイル名を指定して実行]ダイアログボックスまたはコマンドプロンプトからの起動*

#### *SAS 構成ファイルの指定*

SAS を起動するときに、コマンドラインを使用して(**ファイル名を指定して実行**ダイアログボックスまたは**コマンド** ウィンドウ)を使用する場合、CONFIG システムオプションで SAS 構成ファイルの場所を指定してください。デフォルトの構成ファイル SASV9.CFG を使用する場合も、意図した構成ファイルが間違いなく使用されるように、この SASV9.CFG を指定してください。構成ファイルがどのように検索されるかについての

詳細については、“SAS での構成ファイルの検索法と処理法” (17 ページ) を参照してください。

WORK システムオプションと SASUSER システムオプションを設定すると、SAS の起動パスにかかわらず、WORK データライブラリと SASUSER データライブラリが指定したパスに常駐します。Sasuser データライブラリの詳細については、“Profile カタログ ” (22 ページ) を参照してください。Work データライブラリの詳細については、“Work データライブラリ ” (23 ページ) を参照してください。

### *実行ダイアログボックスの使用*

**ファイル名を指定して実行**ダイアログボックスを使用して対話型セッションを開始するには、次のメニューパスを選択します。

1. **スタート** ⇨ **ファイル名を指定して実行**を選択します。
2. **ファイルを開く**フィールドに、パスとプログラムファイルの正確な名前(拡張子、オプションを含む)を入力します。
3. **OK** をクリックします。

たとえば、デフォルトのフォルダ **`c:\Program Files\SASHome\SASFoundation\9.4`** に SAS がインストールされる場合、**`c:\program files\SASHome\SASFoundation\9.4\sas.exe`** および、指定したいオプションを入力します。

*注:* Windows では、**プログラムとファイルの検索スタートメニュー** エントリフィールドに sas.exe への正確なパスを指定する必要があります。

### *SAS コマンドのコマンドプロンプトからの使用*

コマンドプロンプトに SAS コマンドを入力すると、対話型 SAS セッションや SAS バッチジョブを開始できます。たとえば、次のコマンドは、対話型セッションを開始し、行数、文字数、および SAS 構成ファイルの場所を指定しています。

```
c:\program files\SASHome\SASFoundation\9.4\sas.exe -ls 80 -ps 60
   -config c:\program files\SASHome\SASFoundation\9.4\sasv9.cfg
```

同様の方法で、次のコマンドは SAS バッチジョブを開始します。

```
c:\program files\SASHome\SASFoundation\9.4\sas.exe
   -sysin c:\mysas\programs\prog1.sas
   -config c:\program files\SASHome\SASFoundation\9.4\sasv9.cfg
```

*注:* これらの例は、スペースが限られているために、複数の行に表示されています。コマンドプロンプトからコマンドを入力するときには、コマンドは 1 行に入力してください。

## *SAS ファイルからの起動*

Windows エクスプローラで、SAS プログラムファイルから SAS を起動するには、2 つの方法があります。

- SAS プログラムファイルをダブルクリックします。
- SAS プログラムファイルを右クリックして、適切な操作を選択します。

## SAS バッチジョブのサブミット

### バッチ SAS ジョブのサブミット法

次に、SAS バッチジョブをサブミットする方法をいくつか示します。

- コマンドプロンプトまたは**ファイル名を指定して実行**ダイアログボックスから発行される SAS コマンドに、SYSIN システムオプションを指定し、サブミットする SAS プログラムを指定します。
- ファイル拡張子が.sas または sasv7bpgm のファイルを右クリックします。ポップアップメニューから、**バッチサブミット**を選択します。
- Windows エクスプローラで、SAS プログラムファイルのアイコン(SAS コードを含むファイル)を SAS.EXE ファイルアイコンまたはショートカットにドラッグします。

### ステータスウィンドウ

SAS をバッチモードで実行すると、サブミットする SAS ジョブの**ステータス**ウィンドウが表示されます。このウィンドウには、実行されている SAS バッチジョブ名と、ログファイルとプロシジャ出力ファイルが書き出される場所が表示されます。このウィンドウは、SAS バッチジョブが完了するまで開いています。

SAS バッチジョブの実行中に**ステータス**ウィンドウを表示する必要がない場合、ICON システムオプションを使用して SAS を起動し、ジョブの実行中にステータスウィンドウがアイコンで表示されるようにできます。また、ステータスウィンドウの表示時に**アイコン**ボタンをクリックしても、ウィンドウを最小化することができます。SAS ジョブの実行中は、アイコンにビジー状態を示すカーソル(通常は砂時計)が表示されます。ジョブが完了すると、アイコンは表示されなくなります。

### バッチジョブの取り消し

バッチジョブを取り消すには、キーボードまたはマウスを使用します。

- Ctrl+Break を押します。
- **Status** ウィンドウで **キャンセル**をクリックします

### バッチジョブでのウィンドウプロシジャ

SAS バッチジョブ中に、SAS/GRAPH、SAS/INSIGHT、SAS/ACCESS などのソフトウェアと関連するウィンドウ操作プロシジャを実行することができます。プログラム実行中に、対話が必要なポイントに来ると、メイン SAS ウィンドウが開かれます。

## SAS 起動時にプログラムエディタを開始

拡張エディタは、SAS の起動時に開始されるデフォルトのエディタです。プログラムエディタを使用したい場合は、次のいずれかの方法で、SAS 起動時にプログラムエディタを起動します。

- SAS 起動時に NOENHANCEDEDITOR システムオプションを指定します。

```
sas.exe -noenhancededitor
```

- **プリファレンス**ダイアログボックスで**編集**タブの拡張エディタを無効にします。

追加情報については、“拡張エディタとプログラムエディタの切り替え” (107 ページ) 、“編集プリファレンス” (60 ページ) および “ENHANCEDEDITOR システムオプション:Windows” (506 ページ) を参照してください。

### SAS 起動時の作業フォルダの設定

デフォルトでは、SAS によって作業フォルダが決定されます。SAS ファイルの読み込みと書き込みを行う場所のパスが明示的に指定されなかった場合に、この作業フォルダが使用されます。

また、SAS では、次に示すステートメントに基づき、AUTOEXEC.SAS ファイルまたは INITSTMT ファイルの作業フォルダを検索します。この場合、SASINITIALFOLDER システムオプションで指定されたパスは無視されます。

しかし、作業フォルダに使用するパス名を指定するには、SAS の起動時に SASINITIALFOLDER システムオプションを使用します。あるいは、次のルールに基づき、作業フォルダを特定することもできます。

1. プログラムアイコンまたはショートカットを使用して SAS を起動した場合や、**Windows のプロパティのショートカット**タブ(**開始**フィールド)でパスを指定した場合、そのパスを作業フォルダとして使用します。
2. **ファイル名を指定して実行**ダイアログボックスやコマンド行でコマンドを使用して SAS を起動する場合や、コマンドに SAS.EXE ファイルへのパスが含まれる場合、作業フォルダは、Windows 上の SAS.EXE ファイルの実際の格納場所にかかわらず、SAS コマンドの一部として指定するパスになります。
3. コマンドを使用して SAS を起動する場合、SAS コマンドの一部としてパスを指定しないと、作業フォルダはコマンドを実行したパスで指定されます。

指定されたフォルダに SAS.EXE ファイルがない場合でも、SAS コマンドで指定されたフォルダが作業フォルダになり、SAS.EXE ファイルは Windows PATH 環境変数を使用して検索されます。

たとえば、次のコマンドを指定する場合、SAS.EXE ファイルが実際にそのフォルダにあるかどうかにかかわらず、**C:\MYSAS** が作業フォルダになります。

```
c:\mysas\sas.exe -config c:\mysas\sasv9.cfg
```

詳細については、“SAS の現在のフォルダの変更” (41 ページ) および “SASINITIALFOLDER システムオプション:Windows” (558 ページ) を参照してください。

*注:* 作業フォルダと Work データライブラリを混同しないでください。Work データライブラリの詳細については、“Work データライブラリ ” (23 ページ) を参照してください。

### サンプル SAS セッション

このセクションでは、次の項目について説明します。

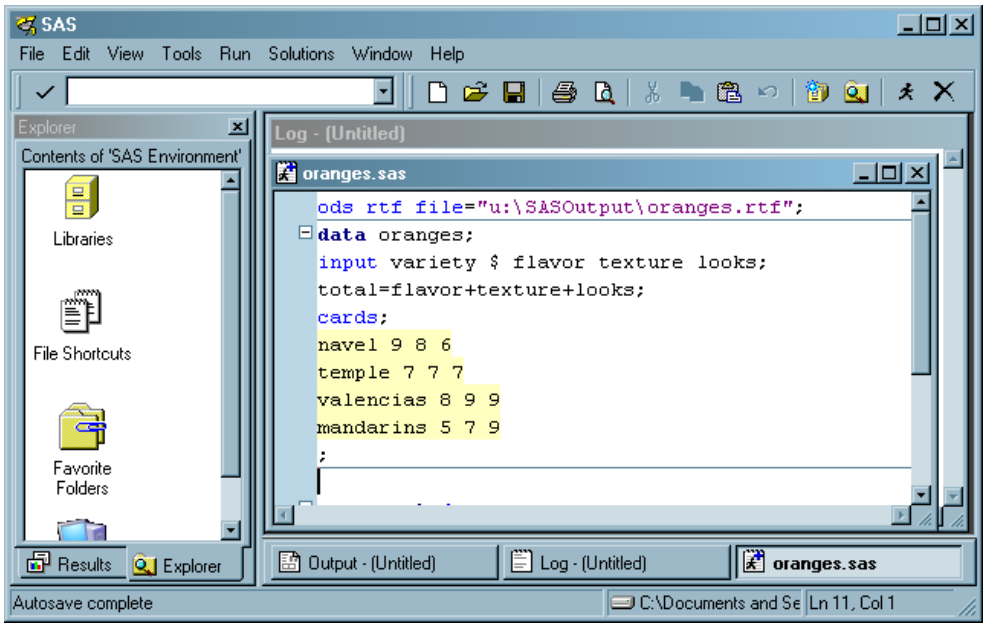
- **スタート**メニューからの SAS の起動
- サンプル SAS プログラムのサブミット
- プログラム出力の検証
- SAS セッションの終了

スタートメニューから SAS を起動するには、**スタート**メニューから SAS を起動するには、**すべてのプログラム** ⇨ **SAS** ⇨ **9.4** を選択します。

次に、サブミット可能なサンプル SAS プログラムが表示された拡張エディタとログウィンドウを示します。このプログラムは、ORANGES という 4 種類のオレンジの試食テス

トの結果の SAS データセットを作成します。データセットはテストの総スコアによって並べ替えられ、出力されます。

*図 1.1　サンプル SAS プログラムのサブミット*

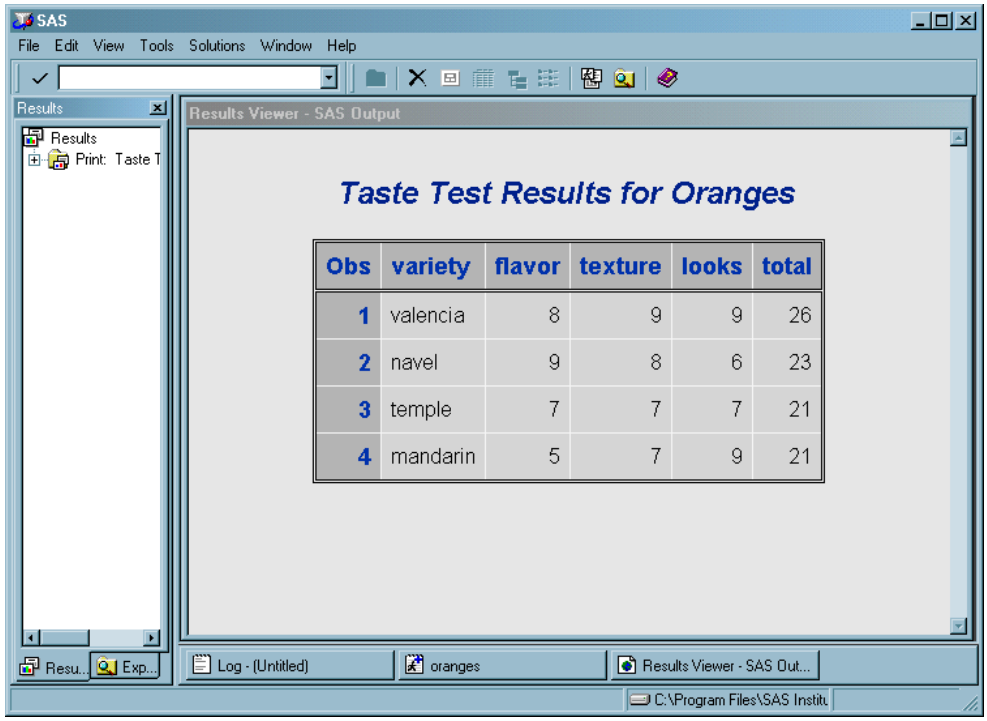



**拡張エディタ**ウィンドウに、次の SAS コードが表示されます。

```
ods rtf file="c:\em\oranges.rtf";
data oranges;
   input variety $ flavor texture looks;
   total=flavor+texture+looks;
   cards;
navel 9 8 6
temple 7 7 7
valencia 8 9 9
mandarin 5 7 8
;
proc sort data=oranges;
   by descending total;
run;
proc print data=oranges;
   title 'Taste Test Results for Oranges';
run;
ods rtf close;
```

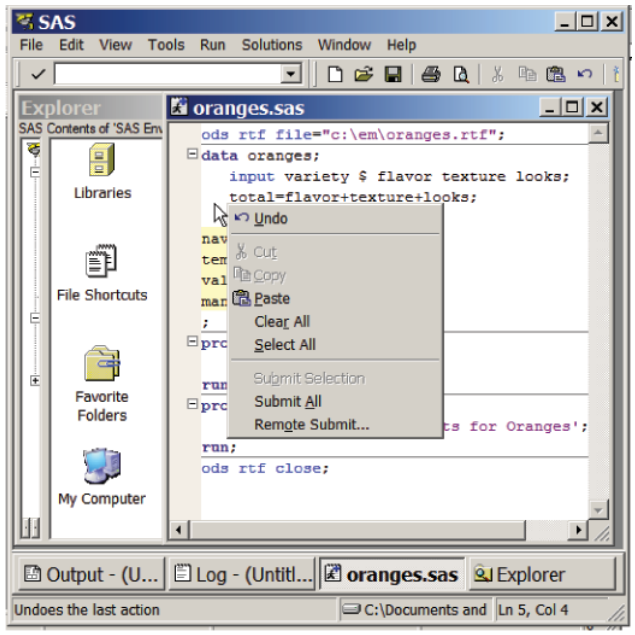
プログラムをサブミットすると、次のように**結果ビューア**ウィンドウに出力が表示されます。

**図 1.2** プログラム出力の表示




メイン SAS ウィンドウの上部にある **SAS メニュー**バーの項目は、SAS セッションで、どのウィンドウがアクティブかによって変わります。また、各ウィンドウ固有のポップアップメニューを使用して、メニューバーと同様のメニューを選択できます(同じメニューを表示できないウィンドウもあります)。**拡張エディタ**ウィンドウで右クリックすると、次に示すポップアップメニューが表示されます。

**図 1.3** [拡張エディタ]ウィンドウのポップアップメニュー



SAS セッションを終了するには、SAS のコントロールメニュー(メイン SAS ウィンドウの左上隅にある小さいアイコン)をダブルクリックするか、または **X**(右上隅)をクリックし、ダイアログボックスに確認メッセージが表示されたら **OK** をクリックします。

*注:* **プリファレンス** ダイアログボックスで **SAS の終了を確認する**オプションを無効にしている場合は、セッションを終了するかどうかを確認せずに SAS を終了します。SAS セッションのカスタマイズの方法については、“セッションプリファレンスについて ” (58 ページ) を参照してください。

### *SAS が起動しない場合について*

SAS が起動しない場合は、SAS ログにエラーの内容を説明するエラーメッセージが含まれる可能性があります。SAS ログが初期化される前に SAS から発行されたエラーメッセージは、**MSG** ウィンドウか、可能であれば、Windows ファイルである SAS コンソールログに書き込まれます。Windows では、SAS コンソールログは通常、`c:\Users\`*user ID*`\AppData` に格納されます。SAS コンソールログの格納場所とファイル名は、アプリケーションイベントログから取得できます。アプリケーションイベントログを開くには、`eventvwr` を、[**ファイル名を指定して実行** ]ダイアログボックスから、[**アプリケーション**]をクリックします。

SAS が起動しない場合や、画面が表示されてから消えた場合、SAS の起動に長い時間がかかる場合は、プリンタが見つからない、プリンタドライバが損傷している、ネットワークプリンタに接続できないなどの問題が発生している可能性があります。この問題を解決するには、次の操作を行います。

1. プリンタが有効なネットワークサーバーに接続されていることを確認します。プリンターが無効なサーバーとリンクされている場合、**スタート** ⇨ **設定** ⇨ **コントロールパネル** ⇨ **プリンター**を選択します。
2. とアクセスすることでプリンターを削除します。
3. 次のパスで、profile2.SAS7bcat の名前を profile2.old に変更し、profile.SAS7bcat の名前を profile.old に変更します。`c:\Users\`*user-ID*`\Documents\My SAS Files\9.4\`
4. SAS を起動します。

## SAS で使用されるファイル

### *SAS で使用されるファイルについて*

SAS では起動中に多くのファイルが使われます。ただし、次のようなファイルのいくつかは、ユーザーにとって特に重要です。

- SAS 構成ファイル(デフォルトのファイル名は、SASV*x*.CFG です。*x* はリリース番号を示します。)
- SAS autoexec ファイル(デフォルト名は、AUTOEXEC.SAS です)
- ユーザープロファイルカタログ(Profile.sas7bcat)
- ユーザープリンタプロファイルカタログ(Profile2.sas7bcat)
- WORK データライブラリ(利用しているシステムに指定された TEMP 領域の SAS Temporary Files フォルダ)
- SAS レジストリファイル

## SAS 構成ファイル

### 構成ファイルの用途

SAS 構成ファイルでは、SAS セッションの開始に使用する SAS システムオプションの指定が可能です。特に、これらのシステムオプションによって、SAS ヘルプとドキュメントファイルの保存場所、メッセージファイルの保存場所、SAS 実行可能ファイルのパス名が指定されます。SAS 構成ファイルの重要な役割として、SAS プロダクトの各種コンポーネントを検索するフォルダを指定することがあります。SAS の初期化には、少なくとも 1 つの構成ファイルが必要です。複数の構成ファイルを保有することもでき、これらの構成ファイルは SAS セッションの開始時にすべて処理されます。システム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。

### デフォルトの構成ファイル

以前のリリースの SAS では、デフォルトの構成ファイルは **!SASROOT** フォルダに保存されています。[**!SASROOT** ]フォルダは、SAS がインストールされたフォルダです。SAS 9 からは、インストール時に 2 つのデフォルトの構成ファイルが作成されるようになりました。どちらの構成ファイルも、SASV9.CFG という名前が付けられます。

SAS 9.4 では、SAS ソフトウェアのインストールに選択された場所は、SASHOME と呼ばれ作成されます。!SASROOT は **!SASHOME/SASFoundation/9.4** に格納されます。デフォルトのフォルダ **!SASHOME** は **c:\program files\SASHome** に格納されます。デフォルト構成の場所は、**c:\program files\SASHome\SASFoundation \9.4\nls\en\sasv9.cfg** に格納されます。

[**!SASROOT** ]フォルダ内の SASV9.CFG ファイルには、SAS デフォルト言語の構成ファイルの保存場所を指定する CONFIG システムオプションが含まれます。SAS を起動するのに使用するデフォルトシステムオプションは**!SASROOT\nls***language-code* **\SASV9.CFG** ファイルで指定されます。たとえば、SAS がデフォルトフォルダにインストールされ、デフォルト言語が英語の場合は、**!SASROOT** フォルダの SASV9.CFG ファイルには
**-config "c:\program files\SASHOME\SASFoundation\9.4\nls\en\sasv9.cfg"**が含まれ

SAS には構成ファイルが必要なため、対話モードかバッチモードかにかかわらず、SAS 構成ファイルを使用する必要があります。

登録されている SAS ファイルのタイプ(.sas など)をダブルクリックして SAS を起動する際には、デフォルトの構成ファイルが使用されます。

フォルダの詳細については、**!SASHOME** “SAS デフォルトフォルダの構造” (24 ページ) を参照してください。

### 構成ファイルのシステムオプションの指定

SAS 起動時に、あらゆるシステムオプションが指定できます。よく使用するシステムオプションを構成ファイルに指定すると大変便利です。SAS 構成ファイルのシステムオプション指定の構文についての詳細は、“SAS 起動または SAS 構成ファイルのシステムオプションの構文” (475 ページ) を参照してください。

デフォルトの構成ファイルを編集して、システムオプション設定を追加または変更することができます。また、独自の構成ファイルを作成することができます。“カスタム構成ファイルの作成” (16 ページ) では構成ファイルの変更方法について記述されています。

構成ファイルは、2 つのセクションに分かれています。最初のセクションには、SAS Setup アプリケーションによって更新されないシステムオプションが示されます。2 番目のセクションは、Setup アプリケーションによって編集され、SAS のインストール場所の情報が更新されます。2 つのセクションの間には、次の警告が挿入されます。

```
WARNING:   INSTALL Application edits below this line. User
           options should be added above this box comment.
           INSTALL Application maintains and modifies the
           following options; -SASAUTO, -SASHELP, -SASMSG
           -PATH, and -MAPS. It also maintains and modifies
           the following CONFIG variables with the -SET option;
           INSTALL, USAGE, LIBRARY, SAMPSIO, SAMPSRC, SASCBT,
           and SASEXT01-SASEXT50. It preserves all lines above
           the line containing 'DO NOT EDIT BELOW THIS LINE'.
```

Setup アプリケーションはこの警告より下にあるすべてのデータを削除しますが、この警告より上で指定したオプションには影響しません。SET システムオプションは、次の SAS 環境変数を定義します。SASROOT、SASEXT0、SASFOLDER、MYSASFILES、SASCFG、SASAUTOS、SAMPSIO、SAMPSRC、EISIMAGE、INSTALL。Setup アプリケーションは、この警告の下に次のシステムオプションを追加します。SASUSER、WORK、HELPLOC、DMSEXP、APPLETLOC、TEXTURELOC、RESOURCESLOC、JREOPTIONS、SASSCRIPT、SASHELP、MSG、PATH。

***注意:***

**構成ファイルの編集には、ファイルの破損を防ぐため、SAS テキストエディタか ASCII テキストエディタを使用してください。** 構成ファイルの編集にどのテキストエディタを選択するかは、ファイルに特殊文字フォーマットが保存されるかに関連するため重要です。拡張エディタなどの SAS テキストエディタを使用して構成ファイルを編集し、**名前を付けて保存**ダイアログボックスを使用して保存することをお勧めします。SAS テキストエディタを使用しない場合は、必ず他の ASCII テキストエディタ(Windows のメモ帳など)を使用してください。ワードパッドアプリケーションや Microsoft Word などの専用エディタは使用しないでください。それらの専用エディタを使用すると、構成ファイルにキャリッジコントロール文字を挿入したり、文字が破損する場合があります。

### *カスタム構成ファイルの作成*

SAS をインストールするときに、**!SASROOT\nls***language-code* フォルダに SASV9.CFG ファイルが作成されます。*language-code* は、SAS デフォルト言語を示す、2 文字の言語コードです。独自のファイルを、デフォルトのファイルである SASV9.CFG より優先される構成ファイルとして機能するように指定することもできます。

ファイル名を定義する際、Windows 動作環境の命名規則に従う必要があります。ファイルの拡張子には、.CFG を使用してください。

デフォルトの構成ファイルの代わりに独自の構成ファイルを使用する場合は、複数の必須システムオプションを追加する必要があります。たとえば、SET システムオプションを使用して環境変数!SASROOT を定義するか、Windows 環境変数として SASROOT を定義する必要があります。

すべての必須システムオプションが構成ファイルで定義されるように、独自のファイルを作成する代わりに、デフォルトのファイル(**!SASROOT\nls***language-code* **\SASV9.CFG**)をコピーし、そのコピーを修正します。

Windows ユーザープロファイルフォルダ内に、カスタマイズされた構成ファイルを作成し、ファイル名として SASV9.CFG または.SASV9.CFG を指定することができます。SAS 起動時に、-CONFIG オプションが指定されない場合は、SAS では Windows ユーザープロファイルフォルダ内で SASV9.CFG または.SASV9.CFG が検索されます。Windows では、ユーザープロファイルフォルダのパスは **c:\Users***user-ID* **\Documents** です。

### 代替構成ファイルでの SAS の起動

デフォルトの構成ファイルとはフォルダが異なるファイルや、名前が異なるファイルを使用する場合は、SAS に構成ファイルの保存場所を指定する必要があります。CONFIG システムオプションを使用して、この構成ファイルの保存場所を指定します。たとえば、**SAS のプロパティ**ダイアログボックスの**リンク先フィールド**に次の内容が含まれています。

```
"c:\program files\SASHome\SASFoundation\9.4\sas.exe -config c:\mysas\mysasconfig.CFG"
```

SAS で構成ファイルが見つからない場合は、エラーメッセージが表示され、SAS は初期化されません。

CONFIG システムオプションの詳細については、“追加 CONFIG オプションで指定したオプションの処理” (20 ページ) および “CONFIG システムオプション Windows” (501 ページ) を参照してください。

### SAS での構成ファイルの検索法と処理法

SAS を起動すると、自動的に、SAS セッションに影響する構成オプションが複数の場所から検索されます。SAS では、次の場所を表示された順序で処理します。

SAS_SYS_CONFIG オペレーティングシステム環境変数
: この環境変数が定義されている場合は、この環境変数は有効な構成ファイルに割り当てられている必要があります。Multi-user Windows システムでは、この環境変数はユーザー環境変数ではなくシステム環境変数として定義されるため、この環境変数はシステムの全ユーザーに対して処理されます。ユーザー固有の構成ファイルを指定するには、SAS_USER_CONFIG ユーザー環境変数を使用します。

CONFIG システムオプションに指定されるファイル
: SAS 起動コマンドに、使用する構成ファイルの名前を示す 1 つまたは複数の-CONFIG オプションを指定することができます。指定するファイルごとに、別の-CONFIG オプションを使用する必要があります。

SAS.EXE が配置されたフォルダの SASVx.CFG
: SAS 起動時に-CONFIG オプションが指定されない場合のみ、SAS.EXE ファイルが格納されたフォルダから SASVx.CFG (*x* は SAS のバージョン番号)ファイルが検索されます。この構成ファイルには、SAS デフォルト言語の構成ファイルを指定する CONFIG システムオプションのみが含まれます。

Windows ユーザープロファイルフォルダの.SASVx.CFG
: SAS 起動時に-CONFIG オプションが指定されない場合のみ、Windows ユーザープロファイルフォルダから.SASV*x*.CFG (*x* は SAS のバージョン番号)ファイルが検索されます。Windows では、ユーザープロファイルフォルダのパスは **c:\Users** *user-ID***Documents** です。

Windows ユーザープロファイルフォルダの.SASVx.CFG
: SAS 起動時に-CONFIG オプションが指定されない場合のみ、Windows ユーザープロファイルフォルダから.SASV*x*.CFG (*x* は SAS のバージョン番号)ファイルが検索されます。Windows では、ユーザープロファイルフォルダのパスは **c:\Users** *user-ID***Documents**\です。

作業フォルダの SASVx.CFG
: SAS 起動時に-CONFIG オプションが指定されない場合のみ、作業フォルダから SASVx.CFG (x は SAS のバージョン番号)ファイルが検索されます。

SAS_USER_CONFIG オペレーティングシステム環境変数
: この環境変数が定義されている場合、この環境変数は有効な SAS 構成ファイルへのパスにする必要があります。Multi-user Windows システムでは、この環境変数はシステム環境変数ではなくユーザー環境変数として定義されるため、この環境変数はシステムの現ユーザーにのみ処理されます。システム共通の構成ファイルを指定するには、SAS_SYS_CONFIG システム環境変数を使用します。

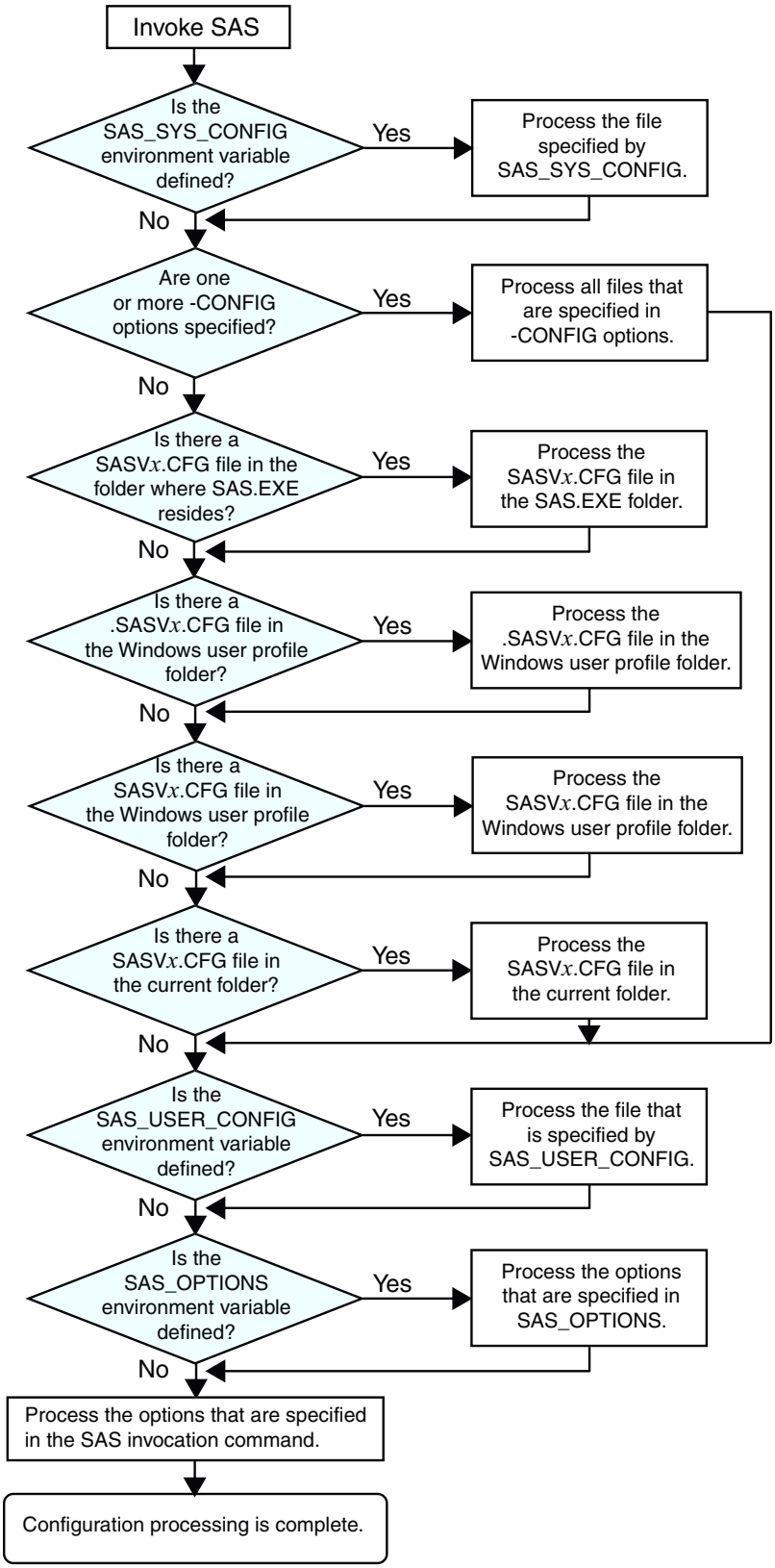
SAS_OPTIONS オペレーティングシステム環境変数
: この環境変数が定義されている場合、この環境変数には、SAS 起動ごとに処理されるあらゆる他の SAS システムオプションのオプションを指定する文字列が含まれます。たとえば、次の環境変数は**-obs 2m -sasinitialfolder c:\myfolder -linesize max** を含んでいる場合があります。

SAS 起動コマンド行
: SAS 起動時に使用するコマンドに、追加システムオプションを指定することができます。このシステムオプションは、常に構成ファイルで設定されたオプション値より優先されます。

Windows の**スタート**メニューから SAS を起動する場合、SAS への**スタート**メニューショートカットに指定された構成ファイルが、最後に処理されます。この構成ファイルに指定されたオプションは、それまでに処理されたオプションより優先されます。デフォルトでは、**スタート**メニューショートカットにより、
**-config !SASROOT\nls***language-code***\SASV***x***.CFG** が指定されます。

その種類が登録されている SAS ファイル(.sas や.sas7bpgm など)をダブルクリックして SAS を起動する際には、デフォルトの構成ファイルが使用されます。

**図 1.4** SAS 構成ファイルの処理順序



### 追加 CONFIG オプションで指定したオプションの処理

構成ファイルに、追加の-CONFIG オプションを指定することもできます。SAS では、-CONFIG オプションが検索されると、ただちに指定されたファイルでそのオプションを処理し、現在のファイルに戻って処理を続けます。後で処理された SAS オプションは、前に指定されたオプションより優先されます。たとえば、SAS_SYS_CONFIG 環境変数によって指定されたファイルに-ICON を指定する場合、–NOICON は SAS_USER_CONFIG 環境変数に指定されたファイルにあり、–NOICON オプションが使用されます。SAS 起動コマンドで指定されるオプションは必ず最後に処理されるため、これらのオプションの値は常に設定ファイルで指定されるオプション値をオーバーライドします。図 1.4 (19 ページ) は、SAS 構成ファイル処理の流れを示します。

CONFIG システムオプションの詳細については、“CONFIG システムオプション Windows” (501 ページ) を参照してください。

## SAS 自動実行ファイル

### SAS 自動実行ファイルについて

SAS 自動実行ファイルには、SAS が初期化されると、ユーザーからの入力を受け入れる前に、すぐに実行される SAS ステートメントが格納されています。これらの SAS ステートメントは、SAS プログラムの自動起動、SAS セッション中に使用する特定の変数の設定、システムオプションの設定などに使用されます。

SAS テキストエディタを使用し、自動実行ファイルを作成します。自動実行ファイルを作成する際、どのテキストエディタを使用するかが重要になります。**拡張 エディタ**ウィンドウなどの SAS テキストエディタを使用してファイルを作成し、**名前を付けて保存**ダイアログボックスを使用して保存することをお勧めします。SAS テキストエディタを使用しない場合は、必ず、Windows のメモ帳など、ASCII テキストエディタを使用します。ワードパッドアプリケーションや Microsoft Word などの専用エディタは使用しないでください。このようなエディタを使用すると、自動実行ファイル内のステートメントを実行する際に SAS では解釈できないキャリッジコントロール特殊文字が、自動実行ファイルに挿入される可能性があります。

### デフォルトの自動実行ファイル

構成ファイルとは異なり、SAS 自動実行ファイルは、SAS の実行に必ずしも必要ではありません。自動実行ファイルを使用する場合、デフォルトの名前は AUTOEXEC.SAS になります。SAS では、次の順序で AUTOEXEC.SAS ファイルが検索されます。

1. 作業フォルダを検索します。

   *注:* 作業フォルダの決定に関する詳細については、“SAS 起動時の作業フォルダの設定” (11 ページ) を参照してください。

2. Windows PATH 環境変数によって指定されたパスを検索します。
3. 現在のドライブのルートフォルダを検索します。
4. SAS.EXE ファイルが格納されたフォルダを検索します。

これらのフォルダに AUTOEXEC.SAS ファイルが存在せず、コマンド行またはいずれの構成ファイルでも-AUTOEXEC オプションが指定されなかった場合は、処理が必要な自動実行ファイルは存在しないと見なされます。詳細については、“AUTOEXEC システムオプション:Windows” (491 ページ) を参照してください。

### 名前が変更された自動実行ファイルの検索

自動実行ファイル名に AUTOEXEC.SAS と命名する必要はありません。たとえば、SAS ウィンドウショートカットの**リンク先**フィールドの SAS.EXE ファイルのパスの指定の後に、次のオプションを指定できます。

```
-autoexec c:\mysasfiles\init.sas
```

指定した autoexec ファイルが見つからない場合は、エラーメッセージが表示され、SAS は終了します。

### 自動実行ファイルの使用

自動実行ファイルを使用すると、SAS の起動時に標準的な SAS プログラムステートメントを簡単に実行することができます。SAS セッションの起動時に自動的に実行する必要がある OPTIONS、LIBNAME、FILENAME などのステートメントやその他の SAS ステートメント、システムオプションをこのファイルに含めます。たとえば、SAS/CONNECT ソフトウェアのスクリプトファイルを指定する場合、AUTOEXEC.SAS ファイルに次のステートメントを挿入できます。

```
filename rlink 'c:\program files\SASHome\SASFoundation\9.4\connect\saslink\
startSession.scr';
```

または次の例のように、OPTIONS ステートメントを使用して SAS の出力のページサイズと行サイズを設定し、複数の FILENAME ステートメントを使用して、よくアクセスするネットワークドライブのファイル参照名を設定することもできます。

```
options linesize=80 pagesize=60;
filename saledata 'f:\qtr1';
filename custdata 'l:\newcust';
filename invoice 'o:\billing';
```

AUTOEXEC システムオプションに加えて、他のシステムオプションを使用することで、SAS 初期化時に情報を送信することができます。次に、これらのオプションを、処理される順序で示します。

1. CONFIG (SAS 起動時のみ)
2. AUTOEXEC
3. INITCMD
4. INITSTMT
5. SYSIN

CONFIG、AUTOEXEC、INITSTMT、SYSIN の各システムオプションについての詳細は、“Windows 版に固有の SAS システムオプション” (473 ページ) を参照してください。INITCMD システム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。

### 自動実行ファイルの非表示

作業フォルダに AUTOEXEC.SAS ファイルが存在する場合に、SAS で使用しないようにするには、次の例のように、SAS コマンドに NOAUTOEXEC オプションを指定します。

```
c:\program files\SASHome\SASFoundation\9.4\sas.exe -noautoexec
```

## Profile カタログ

### プロファイルカタログについて

SAS セッションを開始するたびに、SASUSER データライブラリから、Sasuser.Profile と呼ばれるユーザープロファイルカタログがチェックされます。このプロファイルカタログでは、SAS セッションの起動プロファイル(キー定義、表示構成、その他のユーザーが指定したカスタマイズなど)が定義されます。

Sasuser.Profile が存在せず、Sashelp.Profile (Sashelp データライブラリ内)が存在する場合、SAS セッションが起動する前に Sashelp.Profile は Sasuser.Profile にコピーされます。

すでにプロファイルカタログが存在する場合は、再作成されません。その後のセッションで定義されたキー定義や色の変更などのカスタマイズは、指定されたフォルダのプロファイルカタログに格納されます。

### デフォルトの Profile カタログ

デフォルトの SAS 構成ファイルでは、SASUSER システムオプションは次のように指定されています。

***表 1.1*** *Windows 動作環境での SASUSER のデフォルトの保存場所*

| Windows 7 および 8、Windows Server 8 および 2012 |
| --- |
| `-sasuser "c:\Users\user-ID\Documents\My SAS Files\9.4"` |

### Profile カタログの場所の変更

SASUSER システムオプションを使用して、プロファイルカタログの場所をデフォルト(`\My SAS Files\9.4`フォルダ)以外に指定できます。このオプションは、他のユーザーとマシンを共有する場合や、ネットワークから SAS にアクセスしている場合に SAS セッションをカスタマイズするうえで便利です。

SASUSER システムオプションの形式は、次のとおりです。

-SASUSER ("*library-specification*")

複数の library-specification を指定する場合はカッコ()を使用します。library-specification に特殊文字や空白が含まれる場合は引用符(")を使用します。

有効な Windows パスを指定する *library-specification* が存在しない場合は、SAS によって作成されます。たとえば、次のオプションを指定すると、C ドライブのルートフォルダ内の MYUSER というフォルダにプロファイルカタログが作成されます。

```
-sasuser "c:\myuser"
```

詳細については、“SASUSER システムオプション n:Windows” (559 ページ) を参照してください。

### Profile カタログの削除

プロファイルカタログを削除すると、それまでに定義したキー定義、ウィンドウ構成、オプション設定は、プロファイルカタログに保存したその他のエントリとともに失われます。さらに、[NOTEPAD]ウィンドウに保存したすべてのテキストも消去されます。したがって、SAS セッションの設定を変更した後には、プロファイルカタログのバックアップコピーをとることをお勧めします。

## Work データライブラリ

### WORK データライブラリについて

SAS では、SAS セッション中に一時ディスク領域が必要になります。この一時ディスク領域を WORK データライブラリと呼びます。デフォルトでは、1 レベル名の SAS ファイルが WORK データライブラリに格納され、これらのファイルは SAS セッションの終了時に削除されます。1 レベル名の SAS ファイルが格納される WORK データライブラリは、変更することができます。詳細については、“ユーザーライブラリ参照名の使用” (134 ページ) を参照してください。

### デフォルトの Work フォルダ

デフォルトの SAS 構成ファイルでは、WORK システムオプションには、システムに指定された(TEMP 環境変数で定義された)一時領域内のフォルダが指定されます。例: `!TEMP\SAS Temporary Files`

TEMP 環境変数を定義するには、コントロールパネルから**システムプロパティ**ダイアログボックスにアクセスします。

Work データライブラリの使用と、デフォルトの保存場所のオーバーライドについては、“WORK データライブラリの使用” (133 ページ) を参照してください。

### Work データライブラリの場所の指定

WORK システムオプションは、WORK データライブラリの保存場所を制御します。WORK オプションの指定は、SAS 構成ファイルで行うか、SAS 起動時に行います。通常は、デフォルトの構成ファイルに指定された WORK オプションが使用されます。

### 一時サブフォルダ

SAS では一度に複数の SAS セッションを実行できるため、WORK オプションで指定されたフォルダの下に、一時サブフォルダが作成されます。これらの一時サブフォルダは、_TD*nnnnnnnnnn* という一意の形式で作成されます。TD は一時フォルダを意味し、*nnnnnnnnnn* はプロセス ID で、nodename はフォルダを作成したマシンのホスト名です。サブフォルダを使用することによって、同一の構成ファイルを使用して複数の SAS セッションを起動することができます。また、WORK フォルダの共有を防ぐことができます。SAS では、それぞれの一時フォルダに必要な一時ファイルを作成します。SAS セッション中に WORK データライブラリ内に作成される一時ファイルと同じように、これらの一時フォルダも SAS セッションが終了すると削除されます。SAS が異常終了した場合、一時ファイルを削除する必要があります。

### Work フォルダの削除

SAS が異常終了する場合は、Work ライブラリが削除されたかどうかを確認します。削除されていない場合は、Windows コマンドを使用して削除してください。

*注:* SAS の実行中に、Work フォルダを削除しようとしないでください。

作業フォルダの保存場所を確認するには、**SAS のエクスプローラ**ウィンドウでライブラリフォルダを開きます。Work フォルダの上でマウスの右ボタンをクリックし、ポップアップメニューから**プロパティ**を選択します。

## SAS レジストリファイル

SAS レジストリファイルは、SAS セッションアプリケーションの情報を格納するために使用されます。レジストリのエントリをカスタマイズする方法として、SAS レジストリエディタを使用する方法や、レジストリファイルをインポートする方法があります。SAS レジス

トリエディタを起動するには、**ソリューション** ⇨ **アクセサリー** ⇨ **レジストリエディタ**を選択します。

***注意:***

**レジストリの入力を誤ると、SAS レジストリが破損する可能性があります。**レジストリのカスタマイズは、SAS と動作環境に精通した上級ユーザーが行ってください。

## *SAS デフォルトフォルダの構造*

SAS Setup プログラムでは、インストール実行時に多数のサブフォルダが作成されます。SAS フォルダの構造を理解することによって、より効率的に SAS を使用することができます。

SAS のルートフォルダは、SAS のインストールフォルダです。SAS での、このフォルダの論理名は、!SASHOME です。SAS Foundation 製品は SASROOT と呼ばれる SASHOME 下のフォルダ構造にインストールされます。SAS が指定したデフォルトのフォルダを使う場合、SASHOME は **c:\Program Files\SASHOME\SASFoundation\9.4** です。このドキュメントの例では、!SASHOME フォルダは **c:\Program Files\SASHOME\SASFoundation\9.4** であるものとします。

X64 システム上の 32 ビットのコンテンツは **c:\program files\SASHome\x86\SASFoundation\9.4 にインストールされます。**

他の SAS プロダクトでも使用される共有コンポーネント(拡張エディタや画像など)のフォルダが作成されます。SAS 8 では、共有コンポーネントのデフォルトのパスは **c:\Program Files\SAS Institute\Shared Files** です。SAS 9 から 9.4 までは、共有コンポーネントがインストールされていれば、共有ファイルのデフォルトのパスは **c:\Program Files\SAS\Shared Files** です。共有コンポーネントがインストールされていない場合、デフォルトのパスは **c:\Program Files\SAS\SharedFiles** です。SharedFiles のスペルにはスペースは含まれません。

!SASROOT フォルダの重要なサブフォルダの 1 つは、CORE サブフォルダです。CORE サブフォルダには、多数のサブフォルダが含まれますが、ここでは 3 つのサブフォルダについて説明します。

!SASROOT\CORE\RESOURCE
フォントや画像などの SAS リソースを格納します。

!SASROOT\CORE\SAMPLE
SAS サンプルプログラムを格納します。

!SASROOT\CORE\SASINST
インストールプロセスソフトウェアを格納します。

インストールする SAS プロダクトごとに、次のサブフォルダが作成されます(プロダクトによって次に示すフォルダのすべてを要しないものもあります。)

!SASROOT*product*\SASEXE
SAS 実行ファイルを格納します。

!SASROOT*product*\SASHELP
多数の専用カタログとファイルを格納します。

!SASROOT*product*\SASMACRO
SAS 自動呼び出しマクロファイルを格納します。

!SASROOT*product*\SASMSG
SAS メッセージファイルを格納します。

!SASROOT*product*\SAMPLE
サンプルライブラリプログラムを格納します。

!SASROOT*product*\SASTEST
テストストリームプログラムプログラムを格納します。

!SASROOT*product*\SASMISC
製品とともに出荷される各種外部ファイルを格納します。

SAS/CONNECT ソフトウェアなどの一部の製品では、関連する他のサブフォルダも作成されます。各プロダクトのフォルダ構造の詳細については、それぞれの SAS プロダクトのドキュメントを参照してください。

各サイトでの SAS フォルダの構成方法については、SAS サポートコンサルタントに問い合わせてください。

# SAS コードのサブミット

## *SAS コードのサブミットについて*

Windows 版 SAS では、複数の方法を使用して、SAS プログラムをサブミットし、処理することができます。多様な操作環境をサポートするため、SAS は対話モードでもバッチモードでも実行でき、また、他の Windows プログラムと一緒に実行することもスタンドアロンのアプリケーションとして実行することも可能です。

## *拡張エディタまたはプログラムエディタからコードをサブミット*

**拡張 エディタまたはプログラムエディタ**ウィンドウに入力した SAS コードをサブミットするには、SUBMIT コマンドを発行します。この操作には、次の複数の方法が用意されています。

- エディタウィンドウをアクティブのときに、F8 キーを押します。
- **サブミットツールバー**ボタンをクリックします。
- コマンドバーに `submit` と入力します。
- **実行**メニューから、**サブミット**を選択します。

SAS コード内の 1 行または複数の行をサブミットするには、コマンド行または**実行**メニューから SUBTOP コマンドを使用します。詳細については、“SUBTOP コマンド:Windows” (352 ページ) を参照してください。

## *SAS NOTEPAD テキストエディタからコードをサブミット*

SAS NOTEPAD テキストエディタに入力した SAS コードを、サブミットすることができます。NOTEPAD を呼び出すには、拡張エディタを無効にして、**ツール** ⇨ **テキストエディタ**を選択します。NOTEPAD に入力したコードをサブミットするには、次の複数の方法が用意されています。

- **サブミットツールバー**ボタンをクリックします。
- コマンドバーに `submit` と入力します。
- メニューから、**ファイル名を指定して実行** ⇨ **サブミット**を選択します。

## バッチモードでの SAS の実行

### 概要:SAS のバッチモード実行

Windows 動作環境では、バッチモードで SAS ジョブを実行できます。ファイルに SAS ステートメントを挿入して、サイトで必要な制御ステートメントやシステムコマンドとともにサブミットして実行します。

SAS バッチジョブ中に、SAS/GRAPH、SAS/INSIGHT、SAS/ACCESS などのソフトウェアと関連するウィンドウ操作プロシジャを実行することができます。

### SAS バッチジョブのサブミット

バッチモードで SAS を実行すると、ステータスウィンドウが表示されます。ステータスウィンドウには、実行中のジョブと、ログとプロシジャ出力ファイルの場所が表示されます。ステータスウィンドウは、SAS ジョブが完了するまで開いています。

SAS バッチジョブの実行中にステータスウィンドウを表示する必要がない場合、ICON システムオプションを使用して SAS System を起動し、ジョブの実行中にステータスウィンドウがアイコン表示されるようにできます。また、ステータスウィンドウの表示時に**アイコン**ボタンをクリックしても、ウィンドウを最小化することができます。SAS ジョブの実行中は、アイコンにビジー状態を示すポインタ(通常は砂時計)が表示され、ジョブが完了すると表示されなくなります。バッチジョブをキャンセルするには、Ctrl キーを押しながら Break キーを押すか、またはステータスウィンドウの**キャンセル**をクリックします。

行の最大長は 32767 バイトです。

永久ライブラリ参照名を割り当てるには、バッチジョブを開始する際、STARTLIB システムオプションを有効化します。

SAS バッチジョブをサブミットするには、次の方法を使用します。

- コマンドプロンプトまたは**ファイル名を指定して実行**ダイアログボックスから発行される SAS コマンドに、SYSIN システムオプションを指定し、サブミットする SAS プログラムを指定します。例を次に示します。`C:\SAS\SAS.EXE -SYSIN C:\SAS\PROGRAMS\PROG1.SAS -CONFIG C:\SAS\SASV9.CFG`.
- ファイル拡張子が.SAS、.SS2、SS7、.sas7bpgm のファイルを右クリックして選択します。ポップアップメニューから、**SAS 9.4 でバッチサブミット**を選択します。
- Windows エクスプローラで SAS プログラムファイル(SAS コードが含まれるファイル)アイコンを選択、ドラッグして、SAS.EXE ファイルアイコンまたはショートカットにファイルをドロップします。

## クリップボードからコードをサブミット

拡張エディタまたはプログラムエディタを使用すると、別の Windows アプリケーション(エディタやワードプロセッサなど)や、SAS ヘルプとドキュメントからコピーした SAS コードをサブミットすることができます。別の Windows アプリケーションのテキストをコピーすると、そのテキストは Windows クリップボードに保存されます。

拡張エディタまたは**プログラムエディタ**の**実行**メニューから、**クリップボードをサブミット**を選択します。コードがクリップボードから SAS に直接サブミットされます。このコードは、拡張エディタや**プログラムエディタ**ウィンドウには表示されません。注釈と結果が、それぞれ **SAS ログウィンドウと出力**ウィンドウに送信されます。RECALL コマンドを発行するか F4 キーを押すと、サブミットされたプログラムを**拡張エディタ**または**プログラムエディタ**ウィンドウに呼び戻すことができます。

また、GSUBMIT コマンドを使用して、クリップボードに格納されている SAS コードをサブミットすることもできます。詳細については、“GSUBMIT コマンド:Windows” (347 ページ) を参照してください。

## ドラッグアンドドロップを使用したコードのサブミット

### ドラッグ・ドロップによるサブミットについて

開いている SAS セッションに他の Windows アプリケーションから SAS プログラムをドラッグして、サブミットすることができます。また、SAS コードを含むファイルをドラッグし、開いている SAS セッションにドロップして、サブミットすることもできます。

### 他のウィンドウからのテキストのドラッグ

他の Windows アプリケーションや SAS ウィンドウから拡張エディタまたは**プログラムエディタ**ウィンドウにテキストをドラッグすると、デフォルトではテキストはドラッグ先のウィンドウに移動します。テキストは、F8 キーを押すか SUBMIT コマンドを発行するまでサブミットされません。

ただし、アプリケーションでデフォルト以外のドラッグがサポートされている場合は、マウスの右ボタンを使用してテキストを選択、ドラッグすると、デフォルトの操作が無効になります。選択したテキストを拡張エディタまたは**プログラムエディタ**ウィンドウにドロップすると、コードの移動またはコピーを選択するメニューが表示されます。プログラムエディタのメニューを使用しても、コードをサブミットすることができます。

### 対話型セッションでのファイルのドラッグ

**お気に入りフォルダ**ウィンドウを使用すると、SAS 環境の外にあるファイルにアクセスすることができます。SAS コードを含むファイルを対話型 SAS セッションにドラッグして、実行することができます。**表示**メニューを使用して、**お気に入りフォルダ**ウィンドウにアクセスします。

SAS コードを含むファイルを**拡張エディタ**または**プログラムエディタ**ウィンドウにドロップすると、コードはウィンドウに挿入されますが、サブミットはされません。ファイルを**ログまたは出力**ウィンドウ、あるいは最小化された SAS セッションにドロップすると、コードが自動的にサブミットされます。

SAS セッションを最小化すると、Windows タスクバーにアイコンが表示されます。タスクバーにファイルをドロップすることはできません。ただし、マウスボタンを離さず、ファイルをタスクバー上の SAS アイコンにドラッグし、そこで固定することはできます。約 1 秒後に、SAS ウィンドウが通常のサイズに戻ります。この開いている SAS セッションにファイルをドロップすることができます。

開いている SAS セッションのウィンドウ(拡張エディタまたはプログラムエディタウィンドウ以外)にファイル **C:\MYPROG.SAS** をドロップすることは、次のコマンドの発行と同じです。

```
gsubmit "%include 'c:\myprog.sas'";
```

一度に複数のファイルをサブミットするには、SAS プログラムを含むファイル群をまとめて選択して、開いている SAS セッションにドロップします。複数のファイルをサブミットした場合の実行順序は、Windows によって決定されます。したがって、実行順序が重要な場合は、プログラムファイルを 1 つずつドロップする必要があります。

SAS プログラムアイコンをドロップする際に SAS がビジーであると、ドロップしたファイルは無視されます。ドロップしたファイルが無視されたことは、警告音でのみ通知されます。

### *登録された SAS ファイルタイプに保存されたコードをサブミット*

インストール時に、SAS セットアッププロシジャによって特定のファイルの種類が Windows に登録されます。その種類のオブジェクトをダブルクリックすると、指定された操作が呼び出されます。たとえば、ファイル拡張子が.SAS のファイルは、SAS プログラムとして登録されます。これらの登録されたファイルの種類は次に示すような特殊なアイコンで表示されます。

この拡張子のファイルや、このアイコンで示されるファイルをダブルクリックすると、SAS が呼び出され、ファイルの内容が**拡張エディタ**または**プログラムエディタ**ウィンドウに挿入されます。ファイルに含まれる SAS コードは、F8 キーを押すか**サブミット**ツールをクリックするまではサブミットされないため、処理されません。すでに SAS セッションが実行されている場合は、ファイルをダブルクリックすると、2 番目の SAS セッションが開始されます。

すでに実行されているセッションは使用されません。.sas や.sas7bpgm などの登録された SAS ファイルの種類をダブルクリックして SAS を開始する際には、デフォルトの構成ファイルが使用されます。

## SAS セッションを中断する

SAS セッションの処理を中断するには、ツールバー上で円に囲まれた感嘆符(!)をクリックするか、Ctrl と Break キーを同時に押します。中断時に実行されていたタスクによっては、サブミットされたステートメントを取り消したり、アップロードやダウンロードの要求を取り消したりできます。中断の続行や中断の取り消しなどの、さまざまな選択ができるダイアログボックスが表示されます。

*注:* セッションを中断するときに、何のタスクが処理中であるかによっては、SAS が処理を中断するのに数秒かかる場合があります。SAS では、Windows 共通の方法を使用して、セッションを中断することもできます。

コントロールメニューのアイコンをクリックしてアプリケーションの終了を選択する、Windows のタスクバー上の SAS のポップアップメニューから **閉じる**を選択する、または Windows のタスクマネージャから**タスクの終了**を選択することもできます。いずれの方法を使用しても、選択を確認するダイアログボックスが表示されます。SAS が処理を完了するまで、タスクは終了しないことに注意してください。

## SAS から Windows または MS-DOS のコマンドを実行する

### *SAS からの Windows コマンドまたは MS-DOS コマンド実行の概要*

SAS から Windows または MS-DOS のコマンドを実行するには、X ステートメントか X コマンドを使用します。あるいは、DATA ステップ内で CALL SYSTEM ステートメントか SYSTASK ステートメントを使用します。Windows または MS-DOS のコマンドは、非同期タスクとして発行することも、同期タスクとして発行することもできます。非同期タスクとしてコマンドを実行すると、コマンドは、現在実行されている他のタスクとは無関係に実行されます。同期タスクとしてコマンドを実行すると、コマンドが完了するまでは、他のタスクは実行できません。

非同期タスクとしてコマンドを発行するには、SYSTASK ステートメントに NOWAIT オプションを指定するか、NOXSYNC システムオプションを指定します。同期タスクとしてコマンドを発行するには、SYSTASK ステートメントに WAIT オプションを指定するか、XSYNC システムオプションを指定します。SYSTASK ステートメントを使用した非同期コマンドの実行に関する詳細については、“SYSTASK ステートメント:Windows” (463 ページ) を参照してください。

## X ステートメントまたは X コマンドを使用して Windows コマンド実行

X ステートメントまたは X コマンドを使用して、Windows コマンドを実行することができます。X ステートメントは、DATA ステップ以外で実行します。X コマンドは、コマンドバーか、SAS コマンド行に入力します。

X ステートメントと、SAS ウィンドウ環境の X コマンドは類似しています。両者の大きな違いは、X ステートメントは SAS ステートメントと同様にサブミットされることです。ただし、X のコマンドはウィンドウ環境コマンドとして発行されます。このセクションでは、例の中で X ステートメントを使用していますが、この内容は X コマンドにもあてはまります。

X ステートメントをサブミットするときは、SAS セッションを一時的に終了し、Windows コマンドプロセッサにアクセスします。X ステートメントの構文は次のとおりです。

**X** < '*command* ' >;

*command* 引数の使用は任意です。この引数は、オペレーティングシステムコマンドを発行する場合、またはメモ帳などの Windows アプリケーションを起動する場合に使用します。ここでは、X のステートメントを使用してオペレーティングシステムコマンドを発行することに関して詳述されています。なお、X ステートメントを使用して Windows アプリケーションも起動できます。

実行するオペレーティングシステムコマンドが 1 つだけの場合は、X ステートメントの引数として、そのコマンドを指定します。X ステートメントをサブミットすると、そのコマンドが実行されます。追加のコマンドを発行することはできません。

複数のオペレーティングシステムコマンドを実行する場合は、引数を指定せずに X ステートメントをサブミットします。コマンドプロンプトが表示されたら、オペレーティングシステムコマンドを発行します。発行するコマンドの数に制限はありません。定義した環境変数は、SAS では利用できないことに留意してください。*command* 引数を指定せずに X ステートメントまたは X コマンドをサブミットする場合は、EXIT と入力して SAS セッションに戻ります。

X コマンドは、グローバル SAS ステートメントです。したがって、条件付きで X コマンドを実行できないので注意してください。たとえば、次のコードをサブミットするには、X ステートメントを実行します。

```
data _null_;
   answer='n';
   if upcase(answer)='y' then
      do;
         x 'md c:\extra';
      end;
run;
```

この場合、ANSWER の値が**'n'**または**'y'**と等しいかどうかにかかわらず、**C:\EXTRA** フォルダが作成されます。

### DATA ステップを使用して、条件付きオペレーティングシステムコマンドを条件付きで発行

オペレーティングシステムコマンドを条件付きで発行するには、次の例のように、CALL SYSTEM ルーチンを使用します。

```
options noxwait;
data _null_;
   input flag $ name $8.;
   if upcase(flag)='Y' then
      do;
         command='md c:\'||name;
         call system(command);
      end;
   datalines;
Y mydir
Y junk2
N mydir2
Y xyz
;
```

この例では、変数 FLAG の値を使用して、条件付きでフォルダが作成されます。DATA ステップが実行されると、3 つのフォルダが作成されます。**C:\MYDIR**、**C:\JUNK2**、および **C:\XYZ**.**C:\MYDIR2** フォルダは、オブザベーションの FLAG の値が **Y** ではないため、作成されません。

CALL SYSTEM ルーチンの詳細については、“CALL SYSTEM ルーチン:Windows” (391 ページ) および、*SAS システムオプション: リファレンス*の CALL SYSTEM ルーチンに関するセクションを参照してください。

### XWAITEMAILDLG システムオプション:

XWAIT システムオプションは、X ステートメントまたは X コマンドで MS-DOS コマンドを実行した後、SAS セッションに戻るために、EXIT を手入力する必要があるかどうかを制御します。*command* 引数を指定せずに X ステートメントを発行する場合、または X ステートメントでメモ帳などの Windows アプリケーションを起動する場合は、XWAIT システムオプションは使用しません。このオプションとその反対のオプションは、次のように機能します。

XWAIT
: SAS セッションに戻るために、EXIT の入力を必要とするのを指定します。この値はデフォルトの設定です。

NOXWAIT
: 指定したコマンドの実行後に、コマンドプロセッサによって自動的に SAS セッションに戻るのを指定します。EXIT を入力する必要はありません。

*command* 引数を指定せずに X ステートメントまたは X コマンドを発行する場合は、NOXWAIT が有効であっても、SAS セッションに戻るには EXIT を入力する必要があります。

X ステートメントによって作成されたウィンドウがアクティブになっているときに、コマンドプロセッサを終了せずに SAS を再度アクティブにしようとすると、次のようなメッセージを含むメッセージボックスが表示されます。

```
The X command is active. Enter EXIT at
the prompt in the X command window to
reactivate this SAS session.
```

このメッセージボックスが表示されたら、Windows タスクバーの**コマンドプロンプト**をクリックします。プロンプトから EXIT コマンドを入力し、ウィンドウを閉じて、SAS セッションに戻ります。

### *XSYNC システムオプション*

XSYNC システムオプションは、サブミットされたオペレーティングシステムコマンドの実行を、SAS セッションと同期とするか非同期とするかを指定します。このオプションとその反対のオプションは、次のように機能します。

XSYNC
: オペレーティングシステムコマンドの実行を SAS セッションと同期とする場合に指定します。つまり、コマンドが完了するまでは、SAS に制御が戻りません。CALL SYSTEM ステートメント、X コマンド、または X ステートメントによって作成されたコマンドプロンプトセッションが閉じるまでは、SAS セッションに戻れません。これはデフォルトの設定です。

NOXSYNC
: オペレーティングシステムコマンドの実行を SAS セッションと非同期とする場合に指定します。つまり、制御はただちに SAS に戻され、コマンドの実行中も SAS セッションは中断されません。NOXSYNC を有効にすると、CALL SYSTEM ステートメント、X コマンド、または X ステートメントを実行し、X コマンドまたは X ステートメントによって作成されたウィンドウを閉じずに SAS セッションに戻ることができます。

SAS セッションからメモ帳や Excel などのアプリケーションを起動するときに、NOXSYNC を指定すると便利です。たとえば、次の X ステートメントをサブミットするとします。

```
x notepad;
```

XSYNC が有効な場合は、メモ帳を閉じるまでは SAS セッションに戻れません。しかし、NOXSYNC が有効な場合は、SAS セッションとメモ帳の間で切り替えを行うことができます。NOXSYNC オプションは、SAS セッションと他のアプリケーションとの関係を断ちます。SAS セッションを終了しても、他のアプリケーションはそれ自体を終了するまで開かれています。

### *XWAIT システムオプションと XSYNC システムオプションの比較*

XWAIT システムオプションと XSYNC システムオプションは、まったく異なる効果をもたらします。次に、これらの違いをわかりやすく説明します。

XWAIT
: コマンドプロンプトセッションで EXIT を入力するまでは、SAS セッションに戻れません。

XSYNC
: 他のアプリケーションを終了するまでは、SAS セッションに戻れません。

さまざまなオプションの組み合わせが 表 1.2 (31 ページ) で要約されています。

***表 1.2*** *XWAIT システムオプションと XSYNC システムオプションの組み合わせ*

| 有効なオプション | 結果 |
|---|---|
| XWAIT<br>XSYNC | コマンドプロンプトウィンドウは EXIT を入力するまで閉じられません。アプリケーションが終了するまでは SAS に戻れません。 |

| 有効なオプション | 結果 |
|---|---|
| XWAIT<br>NOXSYNC | コマンドプロンプトウィンドウは EXIT を入力するまで閉じられません。アプリケーションが終了しなくても SAS と切り替えられます。 |
| NOXWAIT<br>XSYNC | アプリケーションが終了するとコマンドプロンプトウィンドウは自動的に閉じられます。アプリケーションが終了するまでは SAS に戻れません。 |
| NOXWAIT<br>NOXSYNC | アプリケーションが終了するとコマンドプロンプトウィンドウは自動的に閉じられます。アプリケーションが終了しなくても SAS と切り替えられます。 |

## SAS プロセスを終了する

SAS プロセスを終了する方法には、複数の方法があります。SAS Server は、特定の種類の SAS プロセスです。

*注:* 次のいずれかの方法で SAS を終了させる前に、次に記述されているいずれかの方法を使用してプロセスの終了を試してください。"SAS セッションを終了する" (33 ページ) や SAS サーバーのドキュメントにもプロセスを終了させる方法が記載されています。

Windows サービスとして SAS プロセスのインスタンスが作成されていた場合は、次のいずれかの方法でプロセスを終了することができます。

- コマンドプロンプトで、次のいずれかのコマンドをサブミットします。
  - **`net stop <service name>`** (*service name* には、Windows サービス名を指定します。)
  - **`sc <server> stop <service name>`** (*server* は、"\\ServerName"形式で指定します。*service name* には、Windows サービス名を指定します。)
- Microsoft 管理コンソールのサービススナップインで、終了するサービスを選択し、**停止**を選択します。

SAS プロセスを停止するには、次のいずれかの方法を使用します。

- コマンドプロンプトでをサブミットします。

  ```
  taskkill/pid <process ID>
  ```

  *プロセス ID* は SAS プロセス ID です。このプロセス ID は、**`tasklist`** コマンドの出力から得られます。
- Windows タスクマネージャでプロセスを選択し、**プロセスの終了**をクリックします。

***注意:***

**taskkill コマンドまたは Windows タスクマネージャを使用して SAS プロセスを終了すると、データ損失またはデータ破損が発生する場合があります。**

# SAS セッションを終了する

SAS セッションを終了するには、次の複数の方法があります。

- メイン SAS ウィンドウのコントロールメニューから**閉じる**を選択する。
- **Status** ウィンドウで **キャンセル**を選択する。このウィンドウは、バッチモードでの実行時に表示されます。
- メイン SAS ウィンドウのコントロールメニューをダブルクリックするか、メイン SAS ウィンドウの右上隅にある **X** をクリックする。
- SAS コマンドラインから BYE または ENDSAS コマンドを発行する。
- ENDSAS ステートメントを発行する。
- タスクリストから SAS セッションを閉じる。セッションプロセス(プロセス名は、SAS の開始方法によって異なる)を選択し、**タスクの終了**を選択する。
- **SAS の終了**をメイン SAS ウィンドウの**ファイル**メニューバーから選択する。
- **終了**を**ファイル**ポップアップメニューから選択する。
- Alt と F4 を同時に押下する。このショートカットキーは、Windows で定義された組み合わせです。

SAS の終了時にエラーが発生した場合は、SAS ログに障害の内容を説明するエラーメッセージが書き込まれた可能性があります。SAS から発行されたエラーメッセージは、SAS ログが初期化される前に、**MSG** ウィンドウ(使用可能な場合)や、Windows ファイルである SAS コンソールログに書き込まれます。Windows では、SAS コンソールログは通常、`c:\Users\`*user ID*`\AppData`.SAS コンソールログの格納場所とファイル名は、アプリケーションイベントログから取得できます。アプリケーションイベントログを開くには、`eventvwr` を、**ファイル名を指定して実行**ダイアログボックスからサブミットします。Windows とそれ以降の動作環境下では、**Windows ログ**カテゴリを展開して、**アプリケーション**をクリックします。

*2 章*

# Windows 版の SAS との通信

**SAS インターフェイスの概要** . . . . . **36**
SAS ウィンドウ環境 . . . . . 36
メイン SAS ウィンドウの構成要素について . . . . . 36
メイン SAS ウィンドウのヘルプの利用 . . . . . 38

**SAS セッションでの作業** . . . . . **39**
ドッキングビューの使用 . . . . . 39
ウィンドウバーの使用 . . . . . 40
メニューの使用 . . . . . 41
SAS の現在のフォルダの変更 . . . . . 41
SAS コマンドの発行 . . . . . 42
SAS を使用して電子メールの送信 . . . . . 44
ウィンドウを外部ファイルに保存する . . . . . 53
ウィンドウとファイル名のクリア . . . . . 54
キーの定義 . . . . . 54
Microsoft IntelliMouse を利用した操作 . . . . . 55
クリップボードの使用 . . . . . 55
テキストの強調表示と特殊文字の作成 . . . . . 56

**SAS セッションのカスタマイズ** . . . . . **57**
フォントの選択 . . . . . 57
セッションプリファレンスについて . . . . . 58
コマンドを使用した SAS ウィンドウ環境のカスタマイズ . . . . . 62
システムオプションを使用したウィンドウ環境のカスタマイズ . . . . . 64
メイン SAS ウィンドウ . . . . . 65
ツールバーのカスタマイズ . . . . . 67

**オンラインヘルプとドキュメントの利用** . . . . . **73**
Microsoft HTML ヘルプの使用 . . . . . 73
コマンドバーからのヘルプの利用 . . . . . 73
ダイアログボックスのヘルプの利用 . . . . . 74
SAS プロダクトのヘルプの利用 . . . . . 74
ヘルプメニューからのヘルプの利用 . . . . . 74
SAS リモートブラウザでの出力とヘルプの表示 . . . . . 75

# SAS インターフェイスの概要

## SAS ウィンドウ環境

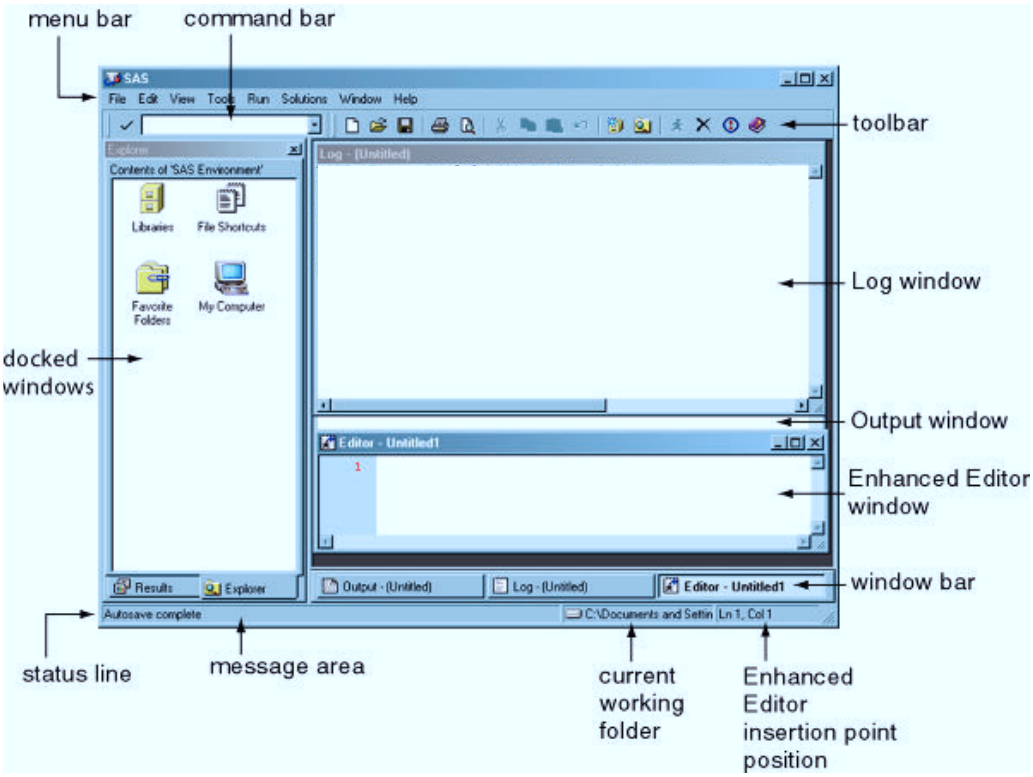
SAS ウィンドウ環境とは、メイン SAS ウィンドウ内に開かれるウィンドウを指します。PC またはターミナルエミュレータから SAS を起動すると、メイン SAS ウィンドウが開きます。

Enterprise Guide など、メイン SAS ウィンドウにアクセスしないクライアントソフトウェアのウィンドウについては、各プロダクトのドキュメントを参照してください。Enterprise Guide の詳細については、Enterprise Guide のヘルプを参照してください。

## メイン SAS ウィンドウの構成要素について

メイン SAS ウィンドウには、他のすべての SAS アプリケーションウィンドウが含まれています。メイン SAS ウィンドウは、自由に設定を変更することができ、生産性の向上のために使用できる機能を備えています。次の図は、最初に SAS システムを起動したときに表示される、メイン SAS ウィンドウです。このセクションでは、ウィンドウの機能を簡単に説明します。

*図 2.1 メイン SAS ウィンドウ*



次に、メイン SAS ウィンドウの主な構成要素を示します。

メニューバー
アクティブな SAS アプリケーションウィンドウで使用できるメニューを表示します。アプリケーションウィンドウを切り替えると、メニューバーの内容も変わります。

同様に、アプリケーションウィンドウ内で右マウスボタンをクリックしたときに表示されるポップアップメニューも、ウィンドウによって異なります。

コマンドバー
SAS コマンドをすばやく入力できます。コマンドバーには、入力したコマンドのリストが保持されます。

発行したことのあるコマンドを繰り返して指定するには、コマンドバーに当該のコマンドが表示されるまで、コマンド文字を入力するか、または、リストボックスからコマンドを選択し、チェックマークボタンをクリックします。Enter キーを押して、コマンドをサブミットします。

キーボードのフォーカスをコマンドバーに切り替えるには、F11 キーを押します。F11 キーは、COMMAND FOCUS として定義されています。

ツールバー
頻繁に実行するコマンドにすばやくアクセスできるようにします。ツールバーは、自由に設定を変更でき、30 個までのツールを含むことができます。

SAS アプリケーションのウィンドウごとに、異なるツールの組み合わせを割り当てることができます。ツールを作成するときに、ツールボタン、ツールに割り当てられるコマンド、ステータス行に表示されるヘルプテキスト、ヒントテキストを指定できます。ビットマップブラウザには、ツールバー上でコマンドを表すために使用することができるさまざまなイメージが用意されています。

ウィンドウ環境
メイン SAS ウィンドウ内でウィンドウを開く作業領域が含まれます。**プリファレンス**ダイアログボックスから**ドッキングビュー**を有効にすると、一部のウィンドウ(ナビゲーションに使用するウィンドウなど)を、メイン SAS ウィンドウの左側にドッキングできます。メイン SAS ウィンドウにドッキングできないウィンドウは、ドッキングエリアの右側に表示されます。上の図の**ログ**ウィンドウ、**アウトプット**ウィンドウ、**拡張エディタ**ウィンドウ、およびドッキングされたウィンドウは、すべてウィンドウ環境の一部です。ドッキング可能なウィンドウの使用に関する詳細な情報は “ドッキングビューの使用 ” (39 ページ) を参照してください。

ウィンドウバー
メイン SAS ウィンドウの下部に配置されます。メイン SAS ウィンドウ内の各ウィンドウに簡単にアクセスするために使用します。

ウィンドウが開くと、そのウィンドウを表すボタンがウィンドウバーに配置されます。ウィンドウにアクセスするには、そのウィンドウのボタンをクリックします。クリックしたウィンドウが、アクティブウィンドウになります。

ファイルをアプリケーションにロードするには、アプリケーションに対応するウィンドウバーボタンにファイルをドラッグし(そのアプリケーションをアクティブウィンドウにして)、そのファイルをアプリケーションウィンドウにドラッグします。

**プリファレンス**ダイアログボックスまたはウィンドウバーのポップアップメニューから有効または無効にできます。

ステータスバー
メッセージ領域、SAS の現在のフォルダ、拡張エディタのカーソル位置を含みます。

メッセージ領域には、メニューおよびツールのヘルプテキスト、SAS アプリケーションウィンドウに固有なメッセージが表示されます。

現在のフォルダには、現在の作業フォルダ名が表示されます。現在のフォルダを変更するには、現在のフォルダをダブルクリックします。詳細については、“SAS の現在のフォルダの変更” (41 ページ) を参照してください。

拡張エディタウィンドウがアクティブウィンドウのとき、カーソル位置には、カーソルの行と列が表示されます。

ステータスバーは、**プリファレンス** ダイアログボックスから有効または無効にできます。メッセージ行、現在のフォルダ、カーソル位置は、**プリファレンス**ダイアログボックスまたはステータスバーのポップアップメニューから有効または無効にできます。メッセージ行が表示されない場合、メッセージはアクティブウィンドウに表示されます。

## *メイン SAS ウィンドウのヘルプの利用*

メイン SAS ウィンドウのヘルプは、スクリーンヒントやステータスバーのメッセージ行に表示されます。

メニューまたはメニュー項目の説明を表示するには、次の操作を行います。

1. メニューまたはメニュー項目を選択します。
2. マウスポインタをメニューまたはメニュー項目の上に置くと、説明がステータスバーのメッセージ領域に表示されます。

ツールバーのヘルプを表示するには、ツールバーボタン上にマウスポインタを置きます。マウスポインタの下にポップアップスクリーンチップが表示され、詳細がメッセージ行に示されます。

タブ、ボタン、ステータスバーなど、メイン SAS ウィンドウのほかの部分でヘルプを表示するには、次の操作を行います。

1. 目的の項目の上にマウスポインタを置きます。
2. 項目の上にマウスポインタを置いたまま、数秒待ちます。マウスポインタの下にポップアップスクリーンチップが表示されます。マウスポインタをウィンドウバーのボタンの上に置くと、スクリーンチップにウィンドウ名が表示されます。

スクリーンヒントやツールバーで使用できるコマンドのステータスバーヘルプ文章を変更する方法は “ツールバーのカスタマイズ” (67 ページ) を参照してください。

コマンドバーまたはツールバーのスクリーンチップを有効または無効にするには、**カスタマイズ**ダイアログボックスの**ツールバーにスクリーンヒントの表示**オプションを使用するか、コマンドバーに TOOL TIP コマンドを入力します。

その他のすべてのスクリーンチップは、**プリファレンス**ダイアログボックスの**表示**タブの**スクリーンチップ**オプションを使用するか、WSCREENTIPS コマンドを使用して、有効または無効にできます。スクリーンチップを有効または無効にする詳細については、“セッションプリファレンスについて ” (58 ページ) , “ツールバーのカスタマイズ” (67 ページ) , “WSCREENTIPS コマンド:Windows” (370 ページ) 、および “TOOLTIPS コマンド:Windows” (355 ページ) を参照してください。

# SAS セッションでの作業

## ドッキングビューの使用

### ドッキングビューについて

ドッキングビューを使用すると、メイン SAS ウィンドウ内を簡単に移動できます。ドッキングビューを有効にすると、エクスプローラウィンドウや結果ウィンドウなどのドッキング可能な(メイン SAS ウィンドウに組み合わせ可能な)ウィンドウが、メイン SAS ウィンドウの左側に表示されます。ドッキングされたウィンドウのオブジェクトを開くと、開いたオブジェクトはドッキングエリアの右側に表示されます。

ドッキングされたウィンドウには、ドッキングエリアの下部にタブが表示されるため、簡単にウィンドウにアクセスできます。ドッキングされたウィンドウの数が多すぎてタブを識別できないときは、左向きの矢印と右向きの矢印が表示され、これらの矢印を使用してドッキングされたウィンドウ間を移動できます。

### ウィンドウのドッキングとドッキング解除

個々のウィンドウをドッキングまたはドッキング解除するには 、

1. ウィンドウを選択して、アクティブウィンドウにします。
2. **ドッキング**されているメニュー項目を切り替えるには、**ウィンドウ** ⇨ **ドッキング**を選択します。

ドッキング画面の好みの設定に関しては、“表示プリファレンス” (60 ページ) を参照してください。画面のドッキングとドッキング解除をするコマンドの使用については、“WDOCKVIEW コマンド:Windows” (359 ページ) を参照してください。

### ドッキングビューのサイズ変更

ドッキングされたウィンドウは、個々に移動したりサイズ調整したりできません。

ドッキングエリアを拡大/縮小するには 、

1. ドッキングエリアとメイン SAS ウィンドウのその他の部分の間にある分割バーの上にマウスポインタを置きます。
2. マウスの左ボタンを押したままにします。
3. マウスを左右に移動して、ドッキングエリアのサイズを調整します。

WDOCKVIEWRESIZE コマンドを使用してドッキングビューのサイズを調整することもできます。詳細については、“WDOCKVIEWRESIZE コマンド:Windows” (360 ページ) を参照してください。

### ドッキングビューの最小化および復元

ドッキングされたウィンドウを最小化するには、次のいずれかの操作を行います。

- ウィンドウタイトルを右クリックして、**最小化**を選択します。
- コマンドバーに `wdockviewminimize` と入力します。

ドッキングされたウィンドウのサイズを元に戻すには、次のいずれかの操作を行います。

- ウィンドウバーの **ドッキングビュー** ボタンをクリックします。

- コマンドバーに **wdockviewrestore** と入力します。

詳細については、“WDOCKVIEWMINIMIZE コマンド:Windows” (360 ページ) および “WDOCKVIEWRESTORE コマンド:Windows” (361 ページ) を参照してください。

### *ドッキングビューの有効と無効の切り替え*

ドッキングビューの有効、無効を切り替えるには、次のいずれかの操作を行います。

- コマンドバーに WDOCKVIEW と引数を指定せずに入力して、ドッキングビューを切り替えます。
- **プリファレンス**ダイアログボックスの**表示**タブを使用します。
  1. **ツール** ⇨ **オプション** ⇨ **プリファレンス** ⇨ **表示**を選択します。
  2. **ドッキングビュー**チェックボックスを選択(有効化)または選択解除(無効化)します。

## *ウィンドウバーの使用*

ウィンドウバー(Windows のタスクバーに相当)は、開いているウィンドウに迅速にアクセスできるように、SAS 内で開いているウィンドウのボタンを表示するために予約されたメイン SAS ウィンドウの下部のスペースです。ウィンドウバーのボタンをクリックすると、そのウィンドウがアクティブになり、その他のすべてのウィンドウより手前に表示されます。アクティブウィンドウのボタンをクリックすると、ウィンドウが最小化されます。

ウィンドウバーの各ボタンには、関連するメニューがあります。メニューを表示するには、ボタン上にマウスポインタを置き、右マウスボタンをクリックします。

ウィンドウの有効、無効を切り替えるには、次のいずれかを使用します。

- **プリファレンス**ダイアログボックスの**表示**タブ
- ステータスバーのポップアップメニュー
- ウィンドウバーのポップアップメニュー
- コマンドバーへの **wwindowbar** の入力

ウィンドウバーのボタンの上にカーソルを置くと、スクリーンチップがポップアップし、ウィンドウ(エディタ)名、開いているファイル名が表示されます。スクリーンチップは、**プリファレンス**ダイアログボックスを使用するか、コマンドバーに **wscreentips** を入力することで、有効化します。スクリーンヒントの有効に関しての詳細については、“表示プリファレンス” (60 ページ) および “WSCREENTIPS コマンド:Windows” (370 ページ) を参照してください。

ドラッグアンドドロップ編集を利用することで、アプリケーションのウィンドウバーのボタンを使用して、ファイルを入力として受け入れる拡張エディタなどのアプリケーションにファイルをロードできます。ファイルをロードするには、次の操作を行います。

1. ファイルをアプリケーションのウィンドウバーのボタン上にドラッグして、アプリケーションをアクティブウィンドウにします。
2. ファイルをアプリケーションウィンドウにドラッグします。
3. マウスボタンを離して、ファイルをアプリケーションにロードします。

ファイルを **windowbar** ボタンにドロップしようとすると、エラーメッセージが発行されます。

## メニューの使用

SAS のコマンド、ツール、オプションをアクセスするには、メイン SAS ウィンドウの上部にあるメニューから選択するか、またはアプリケーションウィンドウ内でポップアップメニューを使用します。メニューには、アクティブウィンドウで使用できるオプションが表示されます。特定のウィンドウのポップアップメニューにアクセスするには、該当するウィンドウ内の任意の場所でマウスの右ボタンをクリックします。ポップアップメニューには、そのウィンドウで使用できるメニュー項目が表示されます。

メイン SAS ウィンドウと同様、**エクスプローラ**ウィンドウなどの一部の SAS ウィンドウには、オブジェクトを右クリックすると独自のポップアップメニューが表示されるオブジェクトがあります。たとえば、コマンドバー、ツールバー、ステータス行には、それぞれのポップアップメニューがあります。これらのウィンドウでは、ポップアップメニューは選択されたオブジェクトに固有です。

## SAS の現在のフォルダの変更

### 現在のフォルダについて

現在のフォルダは、SAS コマンドやアクションがすべてではありませんが、多く適用される動作環境フォルダです。現在のフォルダは、メイン SAS ウィンドウの下部にあるステータス行に表示されます。デフォルトでは、SAS セッションの開始時に、SAS 構成ファイルの SASUSER システムオプションで現在のフォルダとして指定されているフォルダが、現在のフォルダとして使用されます。デフォルトの現在のフォルダは、SAS プログラムショートカットの**プロパティ**タブにある**開始**フィールドを変更するか、SAS 起動中に SASINITIALFOLDER システムオプションを選択すると、別のフォルダを指定できます。詳細については、“SASINITIALFOLDER システムオプション:Windows” (558 ページ) を参照してください。

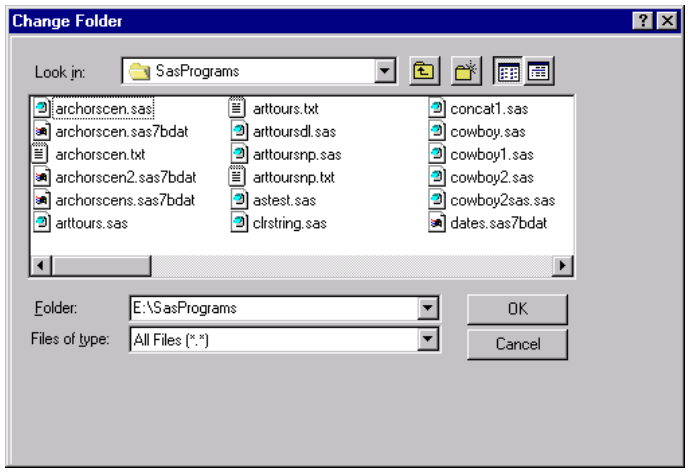
### 新しい現在のフォルダの対話型の選択

新しい現在のフォルダを対話的に選択するには、2 つの方法があります。1 つ目の方法を次に示します。

1. 次のメニューパスを選択します。**ツール** ⇨ **オプション** ⇨ **現在のフォルダの変更**ダイアログボックスからフォルダを選択します。
2. **フォルダの変更**ダイアログボックスからフォルダを選択します。

SAS セッション中の現在のフォルダを変更するもう一つの方法は、ステータス行で現在のフォルダをダブルクリックします。**フォルダの変更**ダイアログボックス(次の図を参照)を使用して、新しい現在のフォルダを選択します。

*図 2.2 フォルダの変更ダイアログボックス*




各プロジェクトが独自のフォルダを持つようにファイルを整理しておけば、この**フォルダの変更** ダイアログボックスを使用して、プロジェクトをすばやく切り替えられます。別のプロジェクトを変更すると、ダイアログボックスは選択したディレクトリの**フォルダ**リストボックスに保存されます。

### *SAS ステートメントを使用して現在フォルダの変更*

現在のドライブとフォルダを変更するには、SAS 内で X ステートメントを使用して、ディレクトリ変更(CD または CHDIR)コマンドをサブミットします。SAS ではディレクトリ変更コマンドを受信すると、ドライブコマンドを変更し、現在のフォルダを変更します。

たとえば、次のステートメントはそれぞれ、SAS セッションの現在のフォルダを MYDATA フォルダと `G:\SALES\JUNE` フォルダにそれぞれ変更します。

```
x 'cd \mydata';
x 'cd g:\sales\june';
```

現在のドライブを変更するには、次のようにドライブ変更コマンド(ドライブレターの後ろにコロン)をサブミットします。

```
x 'a:';
```

## *SAS コマンドの発行*

### *メニューを使用してコマンドの発行*

多くのコマンドは、適用先のウィンドウのメニュー項目に割り当てられています。たとえば、**実行**メニューを選択して、**サブミット**を選択する操作は、コマンドバーに `submit` を入力する操作と同じです。

メニューバーとポップアップメニューの項目は、アクティブウィンドウごとに異なります。

### *ツールバーを使用してコマンドの発行*

デフォルトでは、SAS を起動するとメイン SAS ウィンドウの上部にツールバーが表示されます。ツールバーでは、頻繁に使用するコマンドを簡単に発行できます。ツールバーのコマンドは、アクティブになっている SAS ウィンドウによって異なります。

ツールバーを使用してコマンドをサブミットするには、必要なコマンドを表すツールボタンをクリックします。

マウスポインタをツール上に置くとそのツールのスクリーンチップが表示され、ツールが実行するコマンドを確認できます。

別ウィンドウで使用するためにツールバーのドッキングを解除するまたはツールバーウィンドウをメイン SAS ウィンドウにドッキングするには 、

1. ツールではなく、ツールバー上にマウスポインタを置きます。
2. 右マウスボタンを押したままにします。
3. **ドッキング**を選択します。

ツールバーに定義されたツールを追加、変更したり、アプリケーションのツールバーをカスタマイズしたりできます。詳細については、“ツールバーのカスタマイズ” (67 ページ) を参照してください。

### *コマンドバーを使用してコマンドの発行*

図 2.1 (36 ページ) に示すコマンドバーは任意の SAS コマンドを入力できる中央場所を提供する統合コマンドバーです。ただし、アクティブウィンドウにて有効なコマンドでなければなりません。コマンドバーのドッキングを解除し、独立したウィンドウに表示させることも可能です。また、デスクトップの任意の場所に移動することもできます。アクティブウィンドウに対して無効なコマンドを入力すると、ステータス行に、エラーメッセージが表示されます。

カーソル位置をコマンドバーに移動するには、F11 を押します。

SAS は、コマンドバーに入力したコマンドをセッション間を通して保存するため、前に発行したコマンドをコマンドリストから選択して簡単に取り込めます。デフォルトでは、15 個のコマンドが保存されます。この数は、0-50 の範囲内で変更できます。コマンドバーに保存するコマンドの数を変更するには、次の操作を行います。

1. **ツール** ⇨ **カスタマイズ** ⇨ **ツールバー**を選択します。
2. **コマンドの保存数**で上または下矢印を押します。

デフォルトでは、使用頻度の高い順にコマンドが保存されます。最後に使用したコマンド順に保存するには、次の操作を行います。

1. **ツール** ⇨ **カスタマイズ** ⇨ **ツールバー**を選択します。
2. **最後に使用した順にコマンドをソート**を選択します。

オートコンプリート機能を使用して、前に発行したコマンドを取り込むこともできます。コマンドバーに先頭の数文字を入力すると、入力中のコマンドと最も一致度の高いコマンドが取得されます。必要なコマンドがコマンドバーに表示されたら、Enter キーを押します。オートコンプリートを有効にするには、

1. **ツール** ⇨ **カスタマイズ**を選択します。
2. **オートコンプリートの使用**を選択します。
3. **OK** をクリックします。

コマンドバーをドッキングまたはドッキング解除するには 、

1. コマンドバー上にマウスポインタを置きます。テキストフィールド内に置かないようにします。
2. 右マウスボタンを押したままにします。
3. **ドッキング**を選択します。

コマンドを使ってコマンドバーをカスタマイズするには、“COMMAND コマンド:Windows” (330 ページ) を参照してください。**ツールバーのカスタマイズ**タブの詳細については、“全般的なツールバーのプリファレンスの設定” (68 ページ) を参照してください。

### *コマンド行を使用してコマンドの発行*

ウィンドウぞれぞれにコマンド行が存在するように、コマンド行を切り替えられます。コマンドを適用するウィンドウのコマンド行に、コマンドを入力します。たとえば、INCLUDE コマンドは、**拡張エディタ**ウィンドウに適用されますが、**ログ**ウィンドウには適用されません。

コマンド行を切り替えるには、COMMAND コマンドを引数を指定せずに使用します。**コマンド行**を、**プリファレンス**ダイアログボックスの**表示**タブから選択することもできます。

詳細については、“セッションプリファレンスについて ” (58 ページ) および “COMMAND コマンド:Windows” (330 ページ) を参照してください。

## *SAS を使用して電子メールの送信*

### *電子メールの送信の概要*

SAS では、電子メールを対話形式(ダイアログボックスの使用)またはプログラム(DATA ステップの SAS ステートメントまたは SCL を使用)で送信できます。SAS には、3 種類の電子メールインターフェイスが備わっています。

- MAPI (Microsoft Exchange などのメール API)
- VIM (Vendor Independent Mail)
- SMTP (Simple Mail Transfer Protocol).

これらのプロトコルのいずれかをサポートする電子メールクライアントをインストールしないと、SAS 電子メールサポートを使用できません。SAS では、メッセージの送信はできますが、メッセージを表示して読むには電子メールアプリケーションを使用する必要があります。

対話形式でメールを送信する場合、アクティブウィンドウの内容が電子メールに自動的に添付されます。アクティブウィンドウの内容によって、テキストファイル(.TXT)、ビットマップ(.BMP)、HTML ファイル(.HTML)あるいは RTF ファイル(.RTF)の形式が選択され添付されます。

SMTP は、電子メールに FILENAME ステートメントを使用する場合のみ利用できます。EMAILSYS システムオプションで SMTP を電子メールシステムとして指定し、かつ電子メールダイアログボックスを使用する場合、MAPI 電子メールシステムが使用されます。SMTP の使用に関する詳細な情報については、次を参照してください。*SAS 言語リファレンス: 解説編*次と同じ “FILENAME Statement, EMAIL (SMTP) Access Method” (*SAS Statements: Reference*) 、と “EMAILHOST= System Option” (*SAS System Options: Reference*) 、と “EMAILPORT System Option” (*SAS System Options: Reference*) を参照してください。

### *電子メールの初期化*

SAS から電子メールを送信するには、各自の電子メールシステムに適した次のシステムオプションを SAS 構成ファイルに使用するか、または SAS セッションの起動時に使用します。SAS 呼び出し時や SAS セッション中に、これらの電子メールオプションは設定ファイルにいつでも設定することができます。

-EMAILSYS MAPI | VIM | SMTP
使用する電子メールインターフェイスを指定します。デフォルトでは、MAPI が使用されます。SMTP インターフェイスは、電子メールを自動的に送信する場合にのみ利用できます。電子メールアプリケーションのダイアログボックス、または SAS による電子メールのダイアログボックスを使用する場合、SMTP は使用できません。

*注:* Windows の PATH 環境変数に、MAPI32.DLL や VIM32.DLL などの電子メール DLL ファイルが格納されているディレクトリが指定されている必要があります。SAS では、指定されたインターフェイスに対して、最初に検出された電子メール DLL が使用されます。

-EMAILDLG NATIVE | SAS
電子メールアプリケーションのネイティブ電子メールの対話形式のダイアログボックスまたは SAS 提供の電子メールインターフェイスのどちらを使用するかを指定します。デフォルトでは、電子メールアプリケーションのダイアログボックスが使用されます。

-EMAILHOST *SMTP サーバー*
電子メールアクセスをサポートする SMTP サーバーのドメイン名を指定します。このオプションは、SMTP 電子メールインターフェイスを使用する場合のみ指定する必要があります。複数の SMTP サーバーが次のように指定できます。**EMAILHOST SMTP server | ('server1' 'server2' <...'server-n'>)**

EMAILHOST オプションと追加サブオプションの詳細は、次のドキュメントの EMAILHOST オプションを参照してください: *SAS システムオプション: リファレンス*

-EMAILAUTHPROTOCOL *認証プロトコル*
SMTP 電子メールで使用する認証プロトコルを指定します。このオプションのデフォルトは NONE です。サポートされているプロトコルは、LOGIN と PLAIN のみです。

-EMAILID *VIM 電子メールログイン ID | MAPI プロファイル | 電子メールアドレス*
VIM 電子メールのログイン ID、電子メールシステムへのアクセスに使用する MAPI プロファイルまたは完全に認証済みの電子メールアドレス(SMTP を使用する場合)を指定します。これらの値にスペースが含まれる場合は、二重引用符で囲む必要があります。

-EMAILPORT *ポート番号*
SMTP サーバーを割り当てるポート番号を指定します。このオプションは、SMTP 電子メールインターフェイスを使用する場合のみ指定する必要があります。デフォルトポートは 25 です。

-EMAILPW "*パスワード*"
電子メールのログインパスワードを指定します。*password* は、使用するログイン名のログインパスワードです。*password* にスペースが含まれる場合は、二重引用符で囲む必要があります。パスワードは記号でも可能です。PROC PWENCODE.を使って記号化されたパスワードを作成することができます。

### *電子メールソフトウェアを使用して電子メールの送信*

EMAILDLG システムオプションのデフォルト値は、NATIVE です。SAS から対話形式で電子メールを送信する際、各自の電子メールソフトウェアを使用できます。任意の電子メールソフトウェアを使用して電子メールを送信するには、次のいずれかを実行します。

- **ファイル** ⇨ **メールの送信**を選択します。
- コマンドバーに、DLGSMAIL と入力します。

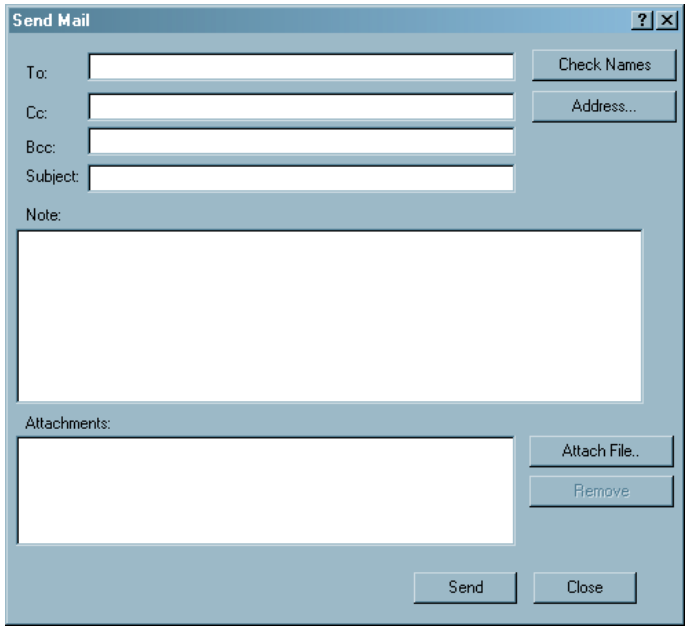
電子メールソフトウェアに備わっているインターフェイスを使用して、電子メールを送信できます。

### *SAS のメールの送信ダイアログボックスの使用*

EMAILDLG システムオプションの値を SAS に設定すると、図のように、SAS によって提供される**メールの送信**ダイアログボックスを使用して電子メールを送信できます。

**メールの送信**ダイアログボックスを使用して電子メールシステムを送信するには、**ファイル** ⇨ **メールの送信**を選択します。

*図 2.3 メールの送信ダイアログボックス*




**メールの送信**ダイアログボックスには、次のフィールドがあります。

**宛先**
宛先電子メールの主受信者。電子メールを送信する前に、電子メールシステムで有効な、1 つ以上の電子メールアドレスを指定する必要があります。複数の宛先を指定する場合は、セミコロン(;)で区切ります。

**Cc**
送信するメールのコピーを受信するユーザーの電子メールアドレス。このフィールドは、空白のままにしておくこともできます。複数の宛先を指定する場合は、セミコロン(;)で区切ります。

**Bcc**
電子メールのコピーを受け取る受信者を指定します。このアドレスは、宛先や CC に指定した受信者には表示されません。

**件名**
メッセージの件名。このフィールドは、空白のままにしておくこともできます。

**本文**
SAS アプリケーションのウィンドウ、または他の Windows アプリケーションからテキストをコピーして、ここに貼り付けることができます(Ctrl キーと C キー、Ctrl と V キーを使用します)。入力したテキストが、表示可能なウィンドウ領域を越える場合、矢印キーを使用して、スクロールアップしたりスクロールダウンしたりできます。また、PageUp キーと PageDown キーを使用することもできます。

現在、一部の電子メールシステムでは、本文の長さが 32K (32,768 文字)に制限されています。**本文**に入力したテキストは、右側で自動的に折り返されます。

テキストのサイズが大きい場合は、以下のようにテキストファイルを添付します。

**添付**

メッセージと一緒に送信するファイルのアイコンと名前。ファイルの拡張子がファイルマネージャで Windows アプリケーションに関連付けられている場合は、自身または受信者がアイコンをダブルクリックして添付ファイルを開くことができます(たとえば、.TXT 拡張子はメモ帳に関連付けられています)。

添付するファイルを選択できるファイル選択ダイアログボックスを開くには、**添付ファイル**をクリックします。

添付ファイルを削除するには、**添付** フィールドでファイルのアイコンを選択し、**削除**をクリックします。

*注:* ハードディスクに存在したままの状態で添付ファイルは送られます。つまり、電子メールメッセージに添付する前にファイルを変更した場合、保存されたファイルがメッセージとともに送られます。

**宛先**および **Cc** フィールドで指定したアドレスが有効かどうかを確認するには、**名前の確認**をクリックします。SAS では、不明なアドレスが存在すると(つまり、メールアプリケーションのアドレス帳に登録されていない場合)、エラーメッセージを表示し、最初の不明なアドレスを強調表示します。

不明なアドレスが無効であるとは限りません。**名前の確認**を使用してアドレスが解決できない場合でも、ゲートウェイを使用して直接接続する LAN(ローカルエリアネットワーク)の外部の受信者にメールを送信できます。

使用している電子メールプログラムやネットワークの構成によっては、アドレスが無効または不明とみなされることがあります。たとえば、インターネットで関係者に電子メールを送信する場合を考えてみます。LAN にインターネットへのゲートウェイが設定され、メールのアドレスを `JBrown@rhythm.com at Internet` と指定できるとします。この場合、`at` はゲートウェイディレクティブキーワードを表し、`Internet` は LAN 上のゲートウェイの名前を表します。メールプログラムでは、`at` キーワードを使用してメッセージを `Internet` ゲートウェイに送るため、このアドレスは有効なアドレスとみなされます。ただし、**名前の確認**をクリックすると、最終的な宛先アドレスをローカルのアドレス帳を使用して解決できないので、不明なアドレスとみなされます。その場合でも、**送信**をクリックすることで、エラーなしでメッセージを送信できます。

**アドレス**をクリックすると、電子メールプログラムの電話帳機能が起動します(アクセス可能な場合)。

### *ウィンドウコンテンツを電子メールで送信する*

SAS 内からメールを送信すると、アクティブウィンドウの内容が電子メールに自動的に添付されます。SAS が作成するファイルの種類は、アクティブウィンドウによって異なります。

***表 2.1*** *電子メールメッセージのファイルの種類(アクティブウィンドウ別)*

| ファイルの種類 | アクティブウィンドウ |
|---|---|
| .txt | ログ<br>出力<br>拡張エディタ<br>プログラムエディタ |
| .bmp | エクスプローラ<br>グラフィック<br>結果 |

| ファイルの種類 | アクティブウィンドウ |
|---|---|
| .htm | **結果ビューア**ウィンドウと出力の種類が HTML |
| .rtf | **結果ビューア**ウィンドウと出力の種類が RTF |

### *DATA ステップまたは SCL を使用して電子メールを送る*

EMAIL アクセスメソッドでは、DATA ステップや SCL を使用して SAS から電子メールを送信できます。このメソッドには、次の複数の利点があります。

- DATA ステップや SCL のロジックを使用して、サイズが大きい電子メールアドレスデータセットに基づき、電子メールの配信をサブセットできます。
- バッチプロセスにサブミットした SAS プログラムの完了時に、電子メールを自動的に送信できます。
- 処理の結果に基づいて、出力を直接電子メールで送信できます。
- ユーザーインターフェイスをカスタマイズして、SAS/AF FRAME アプリケーションから電子メールメッセージを送信できます。

通常、電子メールを送信する DATA ステップや SCL コードには、次のコンポーネントが含まれます。

- EMAIL デバイスタイプキーワードを含む FILENAME ステートメント
- FILENAME ステートメントや FILE ステートメントに指定する、電子メール受信者、件名、添付ファイルを示すオプション
- メッセージ本文を含む PUT ステートメント
- 電子メール属性(TO、CC、BCC、SUBJECT、ATTACH)より優先される、またはアクション(SEND、ABORT、NEWMSG など)を実行する特殊な電子メールディレクティブ(!EM_*directive*!形式)を含む PUT ステートメント

DATA ステップや SCL を使用して電子メールを送信するには、電子メールプログラムにサインオンする必要があります。

電子メールを送信する FILENAME ステートメントの構文は、次のとおりです。

**FILENAME** *fileref* EMAIL '*address*' <*電子メールオプション*>;

ここで

*fileref*
有効な fileref です。

EMAIL
電子メールを使用することを示すデバイスタイプキーワード。

'*アドレス*'
有効な電子メールの送信先アドレス。アドレスは、引用符で囲む必要があります。TO=電子メールオプションや PUT ステートメントの!EM_TO!ディレクティブを指定する場合は、指定したアドレスが変更されるため、FILENAME ステートメントの引数としてのアドレスの指定は必須ではありません。

*電子メールオプション*
次のいずれかを指定できます。

*注:* 電子メールオプションは、FILENAME ステートメントでのみ指定できます。電子メールオプションを指定すると、SAS の対応するシステムオプションが変更されます。

EMAILID=*電子メールログイン ID* | *MAPI プロファイル* | *電子メールアドレス*
電子メールシステムにアクセスする際に使用する電子メールログイン ID、MAPI プロファイル名または SMTP 電子メールアドレスを指定します。EMAILSYS システムオプションに MAPI を指定する場合は、プロファイル名を指定してください。値にスペースが含まれる場合は、二重引用符で囲む必要があります。この電子メールオプションは FILENAME ステートメントに指定され、SAS システムオプションより優先されます。

EMAILPW="*パスワード*"
電子メールのログインパスワードを指定します。*password* は、使用するログイン名のログインパスワードです。*password* にスペースが含まれる場合は、二重引用符で囲む必要があります。この電子メールオプションは FILENAME ステートメントに指定され、SAS システムオプションより優先されます。

EMAILSYS=MAPI | VIM | SMTP
SAS では、次の 3 種類の電子メールインターフェイスを使用できます。

MAPI メール API は、Windows 動作環境でサポートされ、Microsoft Exchange で使用されるインターフェイスです。MAPI は、デフォルトの設定です。

VIM (Vendor Independent Mail)

SMTP Simple Mail Transfer Protocol

TO=*送付アドレス*
電子メールの主受信者を指定します。複数の単語から成るアドレスは、二重引用符で囲む必要があります。複数のアドレスを指定する場合は、アドレスをまとめてかっこで囲み、各アドレスに二重引用符を付ける必要があります。たとえば、有効な TO の値は **to="John Smith" to=("J.Callahan" "P.Sledge")**

CC=*cc アドレス*
電子メールのコピーを受け取る受信者を指定します。複数の単語から成るアドレスは、二重引用符で囲む必要があります。複数のアドレスを指定する場合は、アドレスをまとめてかっこで囲み、各アドレスに二重引用符を付ける必要があります。たとえば有効な CC の値は **cc="John Smith" cc=("J.Callahan" "P.Sledge")**

BCC=*bcc アドレス*
電子メールのコピーを受け取る受信者を指定します。このアドレスは、宛先や CC に指定した受信者には表示されません。複数の単語から成るアドレスは、二重引用符で囲む必要があります。複数のアドレスを指定する場合は、アドレスをまとめてかっこで囲み、各アドレスに二重引用符を付ける必要があります。たとえば有効な BCC の値は **bcc="John Smith" bcc=("J.Callahan" "P.Sledge")**

SUBJECT=*件名*
メッセージの件名を指定します。件名のテキストが、複数の単語から成る場合 (つまり、1 つ以上のスペースが含まれる場合)、そのテキストを二重引用符で囲む必要があります。件名に特殊文字が含まれる場合も、引用符を使用する必要があります。たとえば、**subject=Sales**、**subject="June Report"** は、それぞれ有効な件名です。

ATTACH=*filename.ext*ATTACH=(*filename.ext* <LRECL=*record-length*> <RECFM=*record-format*>)
*filename.ext* はメッセージに添付する1つまたは複数のファイルのフルパスとファイル名を指定します。

LRECL=*record-length* はレコード長をバイト単位で指定します。Windows ではデフォルト値は 32767 です。*record-length* の値は、1 から 1,073,741,823 (1 GB) の範囲から指定できます。

RECFM=*record-format* はレコードフォーマットを設定します。Windows の場合、次の値を指定できます。

F は固定フォーマットを意味します。

N はバイナリ形式を意味し、ファイルはバイトストリームとして処理されます。

P は印刷フォーマットを意味します。

V | D は変数フォーマットを意味します。この値はデフォルトの設定です。

複数のファイルを添付したり、レコード長やレコード形式を指定するには、ファイル名をまとめてかっこで囲み、各ファイル名に二重引用符を付ける必要があります。たとえば添付ファイルの有効な値に次があります。

```
attach=opinion.txt attach=("june2004.txt" "july2004.txt")
attach=("home.html" recfm=v lrecl=372);
```

補足の ATTACH 引数に関して、電子メールシステムが SMTP の場合、"FILENAME Statement, EMAIL (SMTP) Access Method" (*SAS Statements: Reference*) を参照してください。

FILE ステートメントに指定したオプションは、FILENAME ステートメントに指定した同じオプションを無効にします。DATA ステップで、FILE ステートメントを使用して、出力先に電子メールのファイル参照名を定義した場合は、PUT ステートメントでメッセージの本文を定義してください。FILE ステートメントにて電子メールオプションの使用の例に関しては、例のコード 2.2 (51 ページ) を参照してください。

PUT ステートメントを使用して、電子メールメッセージの属性を変更したり、アクションを実行したりする電子メールディレクティブを指定することもできます。1 つの PUT ステートメントには、1 つのディレクティブのみ指定できます。つまり、各 PUT ステートメントには、指定するディレクティブに割り当てられたテキストのみが存在します。メッセージの属性を変更させるディレクティブを次に示します。

!EM_TO!*addresses*
現在の主受信者のアドレスを*複数アドレス*に置き換えます。複数の単語から成るアドレスは、引用符で囲む必要があります。複数のアドレスを指定する場合は、アドレスをまとめてかっこで囲み、各アドレスに引用符を付ける必要があります。

!EM_CC!*addresses*
現在のメールのコピーの受信者のアドレスを*複数アドレス*に置き換えます。複数のアドレスを指定する場合は、アドレスをまとめてかっこで囲み、各アドレスに引用符を付ける必要があります。

!EM_BCC!*addresses*
現在のメールのコピーの受信者のアドレスを*複数アドレス*に置き換えます。受信者は、!EM_TO!アドレスや!EM_CC!アドレスには表示されません。複数のアドレスを指定する場合は、アドレスをまとめてかっこで囲み、各アドレスに引用符を付ける必要があります。

!EM_SUBJECT!'*subject*'
現在のメッセージの件名を*件名*に置き換えます。

!EM_ATTACH!*filename.ext*
添付ファイル名を *filename.ext* に置き換えます。複数のファイルを指定する場合は、ファイルをまとめてかっこで囲み、各ファイルに引用符を付ける必要があります。

次に、アクションを実行するディレクティブを示します。

!EM_SEND!
現在の属性でメッセージを送信します。デフォルトでは、ファイル参照の終了時にメッセージが送信されます。ファイル参照は、次の FILE ステートメントが検出されるか、または DATA ステップが終了すると終了します。このディレクティブを使用した

場合、ディレクティブを検出した時点でメールが送信されます。また、DATA ステップが終了時点で再度メールが送信されます。このディレクティブは、メッセージを条件付きで送信したり、ループを使用して複数メッセージを送信する DATA ステッププログラムを記述する場合に役立ちます。

!EM_ABORT!
現在のメッセージを中断します。このディレクティブを使用すると、DATA ステップの最後に行われるメッセージの自動送信を停止できます。デフォルトでは、SAS は、各 FILE ステートメントごとにメッセージを送信します。

!EM_NEWMSG!
PUT ステートメントディレクティブを使用して設定した現在のメッセージのすべての属性を解除します。

### *DATA ステップからの電子メールの送信の例*

SAS 構成ファイルのコピーを同僚の Jim (ユーザー ID は **JBrown**)と共有するとします。次のコードは、DATA ステップでファイルを送信する方法を示します。

***例のコード 2.1*** *DATA ステップでファイルを送信する*

```
filename mymail email "JBrown" subject="My SASV9.CFG file" attach="c:\sas\sasV9.cfg"; da
```

次のコードの例では、複数の受信者にメッセージにファイルを添付して送信する方法を説明します。ここでは、FILENAME ステートメントではなく、FILE ステートメントに電子メールオプションを指定します。

***例のコード 2.2*** *ファイルの添付と FILE ステートメントでのオプション設定*

```
filename outbox email "Ron B"; data _null_; file outbox /* Overrides value in */ /* file
```

DATA ステップで条件付きロジックを使用することで、複数のメッセージを送信したり、どのメッセージをどの受信者を受け取るかを設定できます。たとえば、2 つの部門の従業員に対して、カスタマイズしたレポートを送信する場合を考えてみます。次のコードは、この場合の DATA ステップを示します。

***例のコード 2.3*** *DATA ステップを使ったカスタマイズされたメッセージの送信*

```
filename reports email "Jim"; data _null_; file reports; length name dept $ 21; input na
```

これを実行すると、受信者が所属する部門ごとに異なる電子メールメッセージと添付ファイルが送信されます。

*注:* !EM_NEWMSG!を使用する必要があります。受信者間のメッセージ属性を解除するディレクティブ!EM_ABORT! ディレクティブを使用すると、DATA ステップの終了時にメッセージが自動的に送信されません。

次のコードの例では、複数の受信者にメッセージにファイルを添付して送信する方法を説明します。ここでは、FILE ステートメントではなく、FILENAME ステートメントに電子メールオプションを指定します。この方法は、SAS システムオプションの EMAILID、EMAILPW、EMAILSYS の値より優先されます。

```
filename outbox email "Ron B" emailsys=VIM
 emailpw="mypassword" emailid="myuserid";
data _null_;
  file outbox
        /* Overrides value in */
        /* filename statement */
     to=("Ron B" "Lisa D")
     cc=("Margaret Z" "Lenny P")
     subject="My SAS output"
     attach="c:\sas\results.out"
```

```
    ;
  put 'Folks,';
  put 'Attached is my output from the SAS';
  put 'program I ran last night.';
  put 'It worked great!';
run;
```

### SCL コードを使ったメールの送信の例

電子メールに使用する FRAME エントリを生成する SCL コードの例を示します。

FRAME エントリには、ユーザーが次の情報を入力できるオブジェクトが含まれます。

MAILTO
メールの送信先ユーザー ID

COPYTO
メールのコピー(CC)の送信先ユーザー ID

ATTACH
添付するファイル名

SUBJECT
メールの件名

LINE1
メッセージの本文

次のコードの例は、**send** ラベルでマーク付けしたこの SCL コードを実行する SEND ボタンも含む FRAME エントリです。

***例のコード 2.4*** *FRAME エントリからの SCL コードの実行*

```
send:/* set up a fileref */ rc = filename('mailit','userid','email'); /* if the fileref
```

### SMTP コードを使った電子メールの送信の例

次に、SMTP を使用して DATA ステップにより電子メールを送信する方法と、ODS HTML 出力を電子メールの添付ファイルとしてではなく HTML 形式で送信する方法を示します。

SMTP を使用するには、アクセス可能な SMTP 電子メールサーバーが必要です。

SMTP を使用するように SAS を構成するには、次のシステムオプションを構成ファイルに追加します。

- -emailsys SMTP
- -emailhost *your.email.server.com*
- -emailport 25.

*your.email.server.com* の場所は、システム管理者に問い合わせてください。ポート 25 が、最も一般的に使用されます。

次のコードは、FILENAME ステートメントと DATA ステップを使用して電子メールを送信します。

```
filename mymail email from="yourid@email.com"
                      to=("id1@.emailaddr.com" "id2@emailaddr.com")
                      subject="Put Subject Here"
                      content_type="text/plain";
data _null_;
   file mymail;
   put 'hello world';
```

```
run;
quit;
```

FILENAME ステートメントに ATTACH 電子メールオプションを使用して、添付ファイルを送信することもできます。SAS データセット、ビットマップファイル、HTML ファイルなどの非テキストの添付ファイルを圧縮してから、ATTACH 電子メールオプションを使用してください。

SMTP を使用して、添付ファイルを使用せずに、HTML 出力を送信することもできます。

```
filename temp1 email to=("yourid@email.com")
                     from="wileycoyote@acme.com"
                     subject="HTML OUTPUT"
                     content_type="text/html";
ods html body=temp1 style=default;
proc print data=sashelp.class;
run;
ods html close;
```

### *ウィンドウを外部ファイルに保存する*

**拡張エディタ**ウィンドウや**プログラムエディタ**ウィンドウなどのテキストエディタウィンドウ、あるいはログ、アウトプット、結果ビューアなどの SAS ウィンドウを外部ファイルに保存できます。

アクティブウィンドウの内容をファイルに保存するには 、

1. 保存ボタン(ディスク)をクリックするか、またはファイルメニューで**保存**を選択します。以前にこのウィンドウの内容をファイルに保存したことがあり、ファイル名がウィンドウタイトルの一部である場合は、以前に指定されたファイルに内容が保存されます。セッションでウィンドウの内容をファイルに保存していない場合は、**名前を付けて保存**ダイアログボックスから有効または無効にできます。

   すでにウィンドウの内容を保存済みで、今回、別のファイルにウィンドウの内容を保存する場合、コマンドバーに **dlgsave** と入力するか、または**ファイル** ⇨ **名前を付けて保存**を選択します。

   ***注意:***
   **ファイルメニューから、名前を付けて保存ではなく上書き保存を選択してファイルを保存すると、その名前のファイルにウィンドウの内容が上書き、追加されます。** 新しいファイルにエディタの内容を保存する場合は、名前を付けて保存を使用します。テキストファイルをエディタウィンドウに開く際に、**ファイルを開く**ダイアログボックスまたは INCLUDE コマンドのどちらを使用しても、エディタのタイトルバーには、開いたファイル名が表示されます。ファイルメニューで**保存**項目を選択すると、その名前のファイルにエディタの現在の内容が上書きまたは追加されます。

2. ウィンドウの内容を保存するファイルを選択するか、名前を指定します。**保存するファイルの種類**リストからファイルの種類を選択することもできます。大部分のファイルの種類はテキストとして保存され、選択した種類に基づいてファイル拡張子が割り当てられます。例外は、RTF テキスト形式で、リッチテキスト形式(RTF)で保存されます。

   リストからファイルの種類を選択した場合、SAS によってそのファイルの種類が記憶され、そのウィンドウで次に新しいファイルを保存するときにデフォルトの種類として表示されます。

### ウィンドウとファイル名のクリア

SAS ウィンドウの内容と保存されているファイル名(存在する場合)を消去するには、次のいずれかの操作を行います。

- Ctrl + E を押します。
- **すべてクリア**を**編集**メニューから選択します。
- **新規作成(空白のページ)** ボタンをクリックします。
- [ファイル]メニューを選択し、**新規作成**または**新規プログラム**を選択します。
- コマンドバーに CLEAR と入力して、**ENTER** キーを押します。

ウィンドウの内容が保存されていない場合、ウィンドウが消去される前に保存するかどうかを確認するメッセージが表示されます。

### キーの定義

SAS セッションのアクティブなキー定義(SASUSER.PROFILE カタログの DMKEYS エントリ)を表示するには、コマンドバーに KEYS と入力するか、または**ツール** ⇨ **オプション** ⇨ **キー**を選択します。これらのキー定義は、拡張エディタウィンドウ、アウトプットウィンドウ、およびログウィンドウなどの SAS の基本的なウィンドウに適用されます。デフォルトキーのリストは “Windows 版でのデフォルトキー定義” (627 ページ) および “拡張エディタでのショートカットキー” (631 ページ) を参照してください。

SAS でキーを定義または再定義するには 、

1. キーの反対側の定義列のマウスポインタまたはマウスボタンをクリックします。
2. そのキーまたはボタンに割り当てるコマンドを入力します。

定義は、有効な SAS コマンドまたはコマンドシーケンスである必要があります。複数のコマンドを指定するときは、各コマンドをセミコロン( ; )で区切ります。たとえば、ウィンドウを最大化して最後にサブミットされたプログラムをリコールするように Ctrl + H ショートカットキーを定義する場合は、**CTL H** の隣にある定義列で次のコマンドを指定します。

```
zoom; recall
```

コマンドの構文は、使用されるとき(キーが押されたとき)までチェックされません。コマンドのスペルを間違えたり、正しくないコマンドを入力した場合、実際にキーを使用して、コマンドが認識されなかったことを示すエラーメッセージを受け取るまで、ユーザーはエラーを発見できません。

キー定義は、SASUSER.PROFILE カタログに保存されます。SAS を起動するたびに、SASUSER オプションの別の値を使用してプロファイルカタログが作成されます。1 つのプロファイルカタログに加えた変更は、その他のカタログには反映されません。ただし、**KEYS** ウィンドウで COPY コマンドを使用するか、または CATALOG プロシジャを使用すると、キー定義のメンバを他のプロファイルカタログにコピーできます。詳細については、CATALOG プロシジャに関する説明を *Base SAS プロシジャガイド*で参照してください。

SAS では、**KEYS** ウィンドウに表示されるすべてのキーを定義できますが、Windows では、Windows アプリケーション間の整合性をとるために一部のキーが Windows 用に予約されています。これらの予約キーは、**KEYS** ウィンドウには表示されません。

他の SAS プロダクトには、それぞれ独自のキー定義があります。特定のプロダクトのキー定義にアクセスするには、そのプロダクトのウィンドウのメニューを使用します。

## Microsoft IntelliMouse を利用した操作

SAS では、Microsoft IntelliMouse がサポートされています。IntelliMouse は、回転ホイール(ホイールコントロール)が内蔵された改良型マウスです。このマウスを利用すると、従来とは異なる方法での操作が可能になります。IntelliMouse は、垂直スクロールバーを使用してウィンドウの内容をスクロールする SAS ウィンドウで機能します。

IntelliMouse を使用すると、SAS ウィンドウのナビゲーションコントロールを操作することなく、マウスを使用してスクロールできます。IntelliMouse を使用してスクロールするには、ホイールコントロールを前方や後方に回転させます。この動作は、スクロールバーで上矢印や下矢印を押す動作に相当します。

IntelliMouse では、自動スクロールもサポートされています。自動スクロールを開始するには、マウスホイールをクリックしてから、マウスを任意の方向に移動します。ウィンドウの内容が、マウスの移動方向と同じ方向にスクロールされます。マウスの移動開始位置からの移動量が大きいほど、内容のスクロール速度も速くなります。自動スクロールモードを終了するには、任意のキーを押すか、マウスボタンをクリックするか、またはマウスホイールを回します。

Windows のコントロールパネルのマウスの設定にて、IntelliMouse の設定を変更できます。IntelliMouse の詳細については、Microsoft のドキュメントを参照してください。

## クリップボードの使用

### テキストの選択とコピー

拡張エディタ、ノートパッド、ログ、アウトプット、KEYS などのテキストを含むウィンドウでは、左マウスボタンか、または Alt + 左マウスボタンを押したままマウスをドラッグして、切り取る領域またはコピーする領域を指定できます。左マウスボタンを押したまま複数の行をドラッグすると、それらの行のテキスト全体が選択されます。Alt キーを押しながら左マウスボタンを押すと、テキストが矩形ブロックまたは列状に選択されます。マウスをドラッグすると、テキスト領域がただちに反転表示されます。テキストウィンドウでは、スクロールしたい方向にウィンドウの枠を越えてマウスポインタを移動することで、マウスをドラッグしながらスクロールできます。テキストの選択領域を拡大するには、Shift キーと左マウスボタンを使用します。コピーするテキストをすべて選択したら、マウスボタンを離します。

指定したテキストをクリップボードにコピーするには、次のいずれかの操作を行います。

- Ctrl + C を押します。
- **コピー**ボタン(コピー元ドキュメント)をクリックします。
- 編集メニューで**コピー**を選択します。

クリップボードに格納されているテキストを貼り付けるには、ウィンドウのテキスト領域に挿入ポインタを合わせ、次の操作のいずれかを行います。

- Ctrl + V を押します。
- **貼り付け**ボタン(クリップボードとドキュメント)をクリックします。
- 編集メニューで**貼り付け**を選択します。

クリップボード内のテキストが指定した領域に貼り付けられます。ターゲットウィンドウ内ですでにテキストを選択している場合は、選択したテキストがクリップボードの内容に置き換わります。テキストを貼り付けることができるのは、拡張エディタまたは NOTEPAD など、テキスト入力が可能な SAS ウィンドウだけです。

### 非テキストウィンドウでの選択とコピー

SAS/GRAPH などのウィンドウでは、領域を反転表示した画像ではなくボックスでマークして表示します。このボックスは、選択している領域がビットマップ形式であることを示します。領域の選択が完了すると、その領域をクリップボードにコピーできます。作業しているウィンドウに編集ポップアップメニューがない場合は、次のキーの組み合わせを使用してコピーと貼り付けを実行できます。

CTRL+C
　選択した領域をクリップボードにコピーします。

CTRL+V
　クリップボードの内容を貼り付けます。

### SAS セッションにビットマップ形式の情報を貼り付ける

SAS/AF ソフトウェアの FRAME エントリの BUILD:**DISPLAY** ウィンドウなどの一部のウィンドウでは、ビットマップをウィンドウに貼り付けることができます。詳細な情報については *Windows 版 SAS* の「SAS/AF ソフトウェアでの OLE の使用」の章の「クリップボードからの OLE オブジェクトを貼り付け」を参照してください。

また、ビットマップを **SAS GRAPH** ウィンドウに貼り付けて、グラフィックをインポートすることもできます。詳細については、“他のアプリケーションからのグラフィックのインポート” (190 ページ) を参照してください。

### クリップボードから SAS コードをサブミットする

SAS では、Windows クリップボードを使用して SAS コードをサブミットできます。この機能を使用すると、Windows のメモ帳やほかのテキストエディタなどのアプリケーションからコピーした(または切り取った)SAS コードを、SAS にサブミットして実行することができます。この機能は、SAS ヘルプとドキュメントで利用できるサンプルプログラムをサブミットする際にも便利です。

クリップボードに格納されている SAS コードをサブミットするには、実行メニューで**クリップボードをサブミット**を選択します(**拡張エディタ**ウィンドウまたは**プログラムエディタ**ウィンドウがアクティブな場合)。コマンド行から GSUBMIT コマンドを使用することもできます。

```
gsubmit buf=default
```

構文は次のとおりです。GSUBMIT コマンドを使用すると、エディタウィンドウがアクティブウィンドウでない場合(閉じている場合)でも、クリップボードに格納されている SAS コードをサブミットできます。GSUBMIT コマンドを頻繁に使用する場合、ツールバーにこのコマンドのアイコンを定義したり、このコマンドをファンクションキーに割り当てることができます。ボタンの定義の方法についての詳細な情報は “ツールバーのカスタマイズ” (67 ページ) を参照してください。

RUN グループのプロシジャがアクティブな場合に Windows クリップボードから SAS コードをサブミットすると、サブミットは失敗します。この場合は、コードを新しい**拡張エディタ**ウィンドウにコピーしてコードをサブミットする必要があります。

## テキストの強調表示と特殊文字の作成

### 特殊文字の属性

SAS ノートパッドや SAS/AF のアプリケーションでは、テキストの拡張表示色と強調表示属性を使用できます。これらの属性にアクセスするには、Esc キーと適切な文字キーまたは番号キーを押して、色と属性のオン/オフを切り替えます。この機能を使用すれば、行全体、各単語、各文字の色と属性を変更することができます。使用できる色と属性およびキーとその設定は、表 2.2 (57 ページ) および 表 2.3 (57 ページ) を参照してください。色を示す文字は、大文字と小文字のどちらでも入力できます。

*表 2.2 表示色のショートカットキー*

| キー | 色 | キー | 色 |
|---|---|---|---|
| ESC+A | 灰色 | ESC+B | 青色 |
| ESC+C | 青緑色 | ESC+G | 緑色 |
| ESC+K | 黒色 | ESC+M | 赤紫色 |
| ESC+N | 茶色 | ESC+O | オレンジ色 |
| ESC+P | ピンク色 | ESC+R | 赤色 |
| ESC+W | 白色 | ESC+Y | 黄色 |

*表 2.3 属性の拡張ショートカットキー*

| キー | 説明 |
|---|---|
| ESC+0 | すべての強調表示属性を無効にします。 |
| ESC+2 | 下線属性を有効にします。 |
| ESC+3 | 反転表示属性を有効にします。 |

### *代替 ASCII 文字*

外国語文字などの代替 ASCII 文字を作成するには、Alt キーと ASCII 文字コードとを組み合わせて使用します。文字コードを入力するには、NumLock をオンにした状態でテンキーを使用します。ASCII 文字コードの一覧と、Alt ショートカットキーの使用方法については、Microsoft のドキュメントを参照してください。

## SAS セッションのカスタマイズ

### *フォントの選択*

ボタンのテキストと説明テキストの要素のフォントを変更するには、構成ファイルまたは SAS 起動時のコマンドプロンプトのいずれかに SYSGUIFONT システムオプションを使用します。

SAS ウィンドウのテキストに別のフォントまたはポイントサイズを選択するには、DLGFONT コマンドを使用して**フォント**ダイアログボックスを開くか、**ツール** ⇨ **オプション** ⇨ **フォント**を選択します。

拡張エディタのフォントを変更するには**ツール** ⇨ **オプション** ⇨ **拡張エディタ** を選択して**外観**タブをクリックします。

SAS ウィンドウに使用できるフォントは、Windows 環境にインストールしたモノスペースフォントに依存します。たとえば、Courier フォントや Lucida Console フォントなどがあります。

フォントまたはポイントサイズを選択すると、**フォント**ダイアログボックスと**拡張エディタオプション**ダイアログボックスに、選択したフォントのプレビューが表示されます。拡張エディタのフォント選択の詳細については、**ヘルプ** ⇨ **このウィンドウの使い方**を選択するか、拡張エディタがアクティブウィンドウである場合は F1 キーを押します。

SAS のインストールでは、SAS での使用を目的として設計された TrueType フォント SAS Monospace が、セットアッププログラムにより自動的にインストールされます。このフォントと表示フォントの Sasfont と組み合わせて、**アウトプット**ウィンドウでの表示、プリント、他の Windows アプリケーションへのコピーなどで、表形式の出力に使用されます。

デフォルトでは、プリント出力の生成に SAS Monospace フォントが使用されます。また、テキストを SAS ウィンドウから切り取り、コピー、またはドラッグして、別の Windows アプリケーションに貼り付ける場合、テキストはすべて SAS Monospace フォントで表示されます。

低解像度ディスプレイでは、ディスプレイ特性の変更には注意が必要です。

***注意:***

**モニタによっては、大きなフォントサイズを選択すると、すべてのテキストを一度に SAS ウィンドウに表示できない可能性があります。** ウィンドウにスクロールバーがない場合は、フォントサイズを大きくすると、いくつかの選択肢が表示できなくなることがあります。このようなディスプレイでは、大きなフォントサイズの使用はお勧めしません。デザインプロパティを変更し、ウィンドウの境界を太いサイズに変更した場合も、同じ問題が発生します。低解像度ディスプレイでは、太いウィンドウ境界を使用しないでください。プリファレンス

## *セッションプリファレンスについて*

### *セッションプリファレンスの概要*

SAS セッションは、操作しやすいように構成できます。例:

- 移動可能な別ウィンドウのコマンドバーを使用できます。
- スクロール操作とウィンドウ表示の環境設定ができます。
- インターネットの Web ページや HTML 出力の表示時に使用する標準 Web ブラウザを設定できます。

次のセクションでは、**プリファレンス**ダイアログボックスと、これらの設定を使用した SAS セッションの制御方法について説明します。


- “プリファレンスダイアログボックスの使用” (58 ページ)
- “一般プリファレンス” (59 ページ)
- “表示プリファレンス” (60 ページ)
- “編集プリファレンス” (60 ページ)
- “結果プリファレンス” (60 ページ)
- “Web プリファレンス” (61 ページ)
- “詳細プリファレンス” (61 ページ)


### *プリファレンスダイアログボックスの使用*

SAS セッションをカスタマイズするには、次のいずれかの方法で**プリファレンス**ダイアログボックスを開きます。

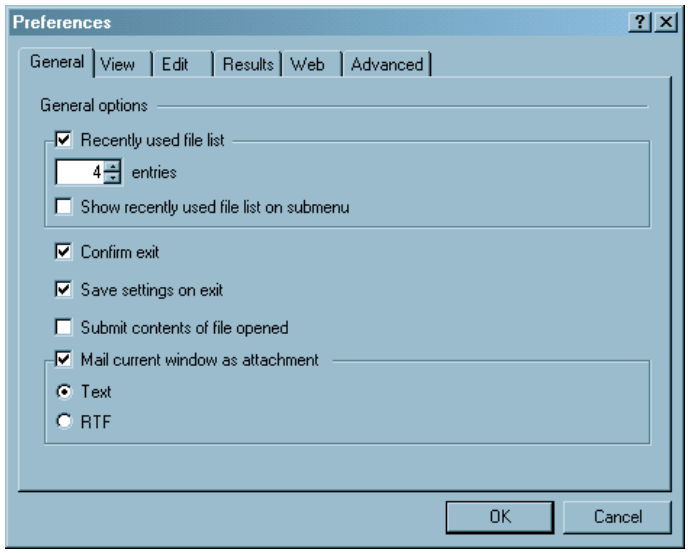
- コマンドバーに `dlgpref` を入力します

- **ツール** ⇨ **オプション** ⇨ **プリファレンス**を選択します。

**プリファレンス**ダイアログボックスには、セッションの設定がカテゴリ分類されたタブがあります。各シートのタブをクリックして、変更する設定を表示し、目的のオプションを選択します。終了したら **OK** をクリックします。

選択した設定は、セッション間を通して SASUSER.PROFILE カタログに、それぞれのページ別に**結果**タブを除き保存されます。各タブに対応する SASUSER.PROFILE のエントリは、GENWSAVE、VIEWWSAVE、EDITWSAVE、WEBWSAVE、ADVWSAVE です。**結果**タブの設定内容は、SAS レジストリに保存されます。このため、別のコンピュータに SASUSER.PROFILE カタログをコピーしても、このタブの設定は移動されません。

***図 2.4*** *プリファレンスダイアログボックス(全般タブ)*




### *一般プリファレンス*

**全般**タブでは、SAS セッションの動作を制御する全般オプションを指定できます。全般タブには、次のオプションがあります。

**最近使ったファイルの一覧**
アクセスしたファイルのリストを記憶するかどうかを指定します。このオプションを選択すると、記憶するファイル数を**入力**フィールドで指定できます。ファイル数の上限は 30 個です。**サブメニューに最近使ったファイルの一覧を表示する**に最近使ったファイルの一覧を表示するには、ファイルメニューからアクセスする**最近使ったファイル**サブメニューからファイルを表示するかどうかを指定します。**サブメニューに最近使ったファイルの一覧を表示する**をオフにした場合は、ファイルはファイルメニューに表示されます。エディタウィンドウでファイルを開くたびに、そのファイル名がリストに追加されます。

**終了の確認をする**
SAS セッションの終了前に確認プロンプトを表示するかどうかを指定します。

**終了時の設定の保存**
SAS セッションの終了時に設定を自動的に保存するかどうかを指定します。

**開いたファイルの内容をサブミットする**
SAS に開いたすべてのファイルの内容をサブミットするかどうかを指定します。

**ウィンドウの内容を電子メールに添付する**
SAS 内から電子メールを送信する際、アクティブウィンドウを電子メールの添付ファイルとして自動的に挿入するかどうかを指定します。このオプションを選択する

と、添付ファイルの形式をテキストにするか、RTF(リッチテキスト形式:フォントと色の情報を保持)にするかを指定できるようになります。

### *表示プリファレンス*

**表示**タブでは、SAS セッションの表示を制御するオプションを指定します。表示で設定できるオプションは、次のとおりです。

**ウィンドウ**
SAS ウィンドウにスクロールバーとコマンド行を含めるかどうかを指定します。また、スクリーンチップ(ウィンドウの各種コントロールの上にマウスポインタを置いた時に表示される便利なヒント)を有効にするかどうかも指定できます。

**表示**
次の設定で、SAS インターフェイスの要素を表示するかどうかを指定します。

**ドッキングビュー**
ドッキング領域を有効化して、メイン SAS ウィンドウの左側にドッキングウィンドウを表示するかどうかを指定します。

**ウィンドウバー**
メイン SAS ウィンドウ下部にウィンドウバーを表示するかどうかを指定します。

**ステータス行**
セッション時に表示するステータスバーを指定します。**メッセージ行を表示する**を選択すると、ステータスバーのメッセージ領域が表示されます。**現在のフォルダを表示する**を選択すると、ステータス行に現在の SAS フォルダを示す領域が表示されます。**カーソル位置を表示する**を選択すると、拡張エディタのカーソルが位置する行と列が表示されます。

### *編集プリファレンス*

**編集**タブでは、SAS テキストエディタに影響する次のオプションを制御します。

**上書き入力する**
SAS アプリケーションウィンドウでテキストを入力する際、テキストを挿入するか、既存のテキストを上書きするかを指定します。上書きモードのオン/オフの切り替えは、キーボードの Insert キーでも行えます。上書きモードは、拡張エディタでは使用できません。

**自動保存する間隔時間 n**
エディタの内容を自動保存するかどうか、および保存の頻度を指定します。

拡張エディタの内容は **`Autosave of filename.$AS`** としてオペーレーティング環境アプリケーションデータフォルダに保存されます。Windows Vista では、アプリケーションデータフォルダのパス名は、**`c:\Users\`** *user ID* **`\AppData`** です。

プログラムエディタの内容は、現在のアクティブフォルダの **`pgm.asv`** に保存されるので、エディタの内容を保存せずに SAS セッションが終了した場合でも修復できます。

**ナビゲーションキーによるマーク解除を有効にする**
上下左右の方向キーを使用して、テキストのマーク解除ができます。

**拡張エディタの使用**
拡張エディタが主エディタであるかどうかを指定します。このチェックボックスがオフの場合、SAS の起動時にはプログラムエディタが開きます。

### *結果プリファレンス*

**結果**タブでは、プログラムの出力結果の表示方法を設定できます。**結果**タブには、次のオプションがあります。

**リスト**
プログラム出力を**アウトプット**ウィンドウに表示する場合に指定します。

*注:* リスト形式を使用して出力を表示する場合は、**プリファレンス**ウィンドウからリストオプションを選択する必要があります。

**HTML**

**HTML を作成する**
デフォルトにてプログラム出力を HTML 形式で表示する場合に指定します。

**フォルダ**
HTML 出力ファイルの保存フォルダを指定します。フォルダ名を入力するか、**参照**をクリックしてフォルダを検索します。この設定は、**WORK フォルダを使用する**がオフの場合にのみ使用できます。

**WORK フォルダを使用する**
は、`Work` フォルダに HTML 出力ファイルを保存するように指示します。[ `Work` フォルダとは、SAS の終了時に削除される一時フォルダです。

**スタイル**
プログラム出力の表示設定を選択できます。スタイルの詳細については、10 章: “TEMPLATE Procedure: Overview” (*SAS 9.4 Output Delivery System: Procedures Guide*) を参照してください。

**すぐに結果ビューアを表示する**
HTML 出力の生成時にブラウザを更新するかどうかを指定します。

**結果を見る**
ここでは、内部ブラウザ、インタネットエクスプローラ、ファイアフォックスなどのサポートするブラウザを指定します。

**ODS グラフィックを使用**
ODS グラフィックをサポートするプロシジャが動作中にグラフを自動的に生成することを指定します。

### *Web プリファレンス*

**Web** タブは SAS セッション内で使用したい Web ブラウザを指定します。WBROWSE コマンドを発行するときは、常にここでの設定が使用されます。WBROWSE コマンドは、直接発行するか、ヘルプメニュー項目またはツールバーボタンを選択して発行します。詳細については、“WBROWSE コマンド:Windows” (358 ページ) を参照してください。Web タブでは、次のオプションを指定できます。

**標準システムブラウザ**
SAS 内からの Web 情報アクセスに使用する標準システムブラウザを指定します。デフォルトでは、システムにインストールされ、デフォルトのブラウザとして Windows に登録されたブラウザが使用されます。デフォルト以外のブラウザを使用するには、**その他**ラジオボタンを選択して Web ブラウザのパスを入力するか、**参照**をクリックして Web ブラウザのパスを検索します。

**スタートページ**
SAS 内で Web ブラウザを起動すると表示されるデフォルトの Web ページを指定します。デフォルトでは、ブラウザは http://www.sas.com/en_us/home.html を表示します。

### *詳細プリファレンス*

**詳細**タブでは、スクロールポリシや他の動作など、SAS セッションに影響するオプションを指定できます。詳細で設定できるオプションは、次のとおりです。

**スクロールオプション**

ログウィンドウとアウトプットウィンドウに情報が書き込まれる際のスクロール行数を指定します。**ログ**ウィンドウのデフォルト値は 12 です。

**ページスクロール**を選択すると、ページ全体が書き込まれるまで、**アウトプット**ウィンドウには行が表示されません。

**最大スクロール**を選択すると、プロシジャが終了するまで出力はウィンドウに書き込まれません。

**スクロール行**を選択した場合で、**アウトプット**ウィンドウにスペースがないときは、**アウトプット**ウィンドウは、**スクロール行**ボックスで指定された行数スクロールします。このデフォルト値は 0 です。この場合、ステートメントの実行中には出力がウィンドウに書き込まれず、パフォーマンスが向上します。

スクロールにより、プログラムの実行時間が長くなる可能性があります。ログウィンドウとアウトプットウィンドウのスクロール頻度を少なくすると、プログラムの実行速度が上がります。

これらの値は、**エディタのオプション**ウィンドウまたは AUTOSCROLL コマンドを使用して設定することもできます。詳細な情報は “AUTOSCROLL コマンド:Windows” (327 ページ) と SAS ヘルプ及びドキュメントを参照してください。

**その他**

その他のオプション設定は、次のとおりです。

**非入力ウィンドウのカーソルは表示しない**

SAS/AF プログラムなど、テキスト入力の必要がないウィンドウに挿入位置を表示しない場合に指定します。

**スクロールバーのフォーカスを無効にする**

スクロールバーをクリックしても、選択されないように指定します。この設定により、一部の SAS Applications で発生する点滅の問題が解決されます。

## *コマンドを使用した SAS ウィンドウ環境のカスタマイズ*

### *ウィンドウ位置のカスタマイズ*

対話型セッションのデフォルトの表示構成( 図 2.1 (36 ページ) を参照)では、メイン SAS ウィンドウにエクスプローラウィンドウと結果ウィンドウがドッキング表示され、残りの SAS ワークスペースにログ、拡張エディタ、アウトプットのウィンドウが表示されます。

他の Windows アプリケーションと同様、次の Windows **メニュー**を使用して、SAS ウィンドウを配置できます。

- **全ウィンドウを最小化(元に戻す)**
- **カスケード**
- **上下に並べて表示**
- **左右に並べて表示**
- **サイズの変更**

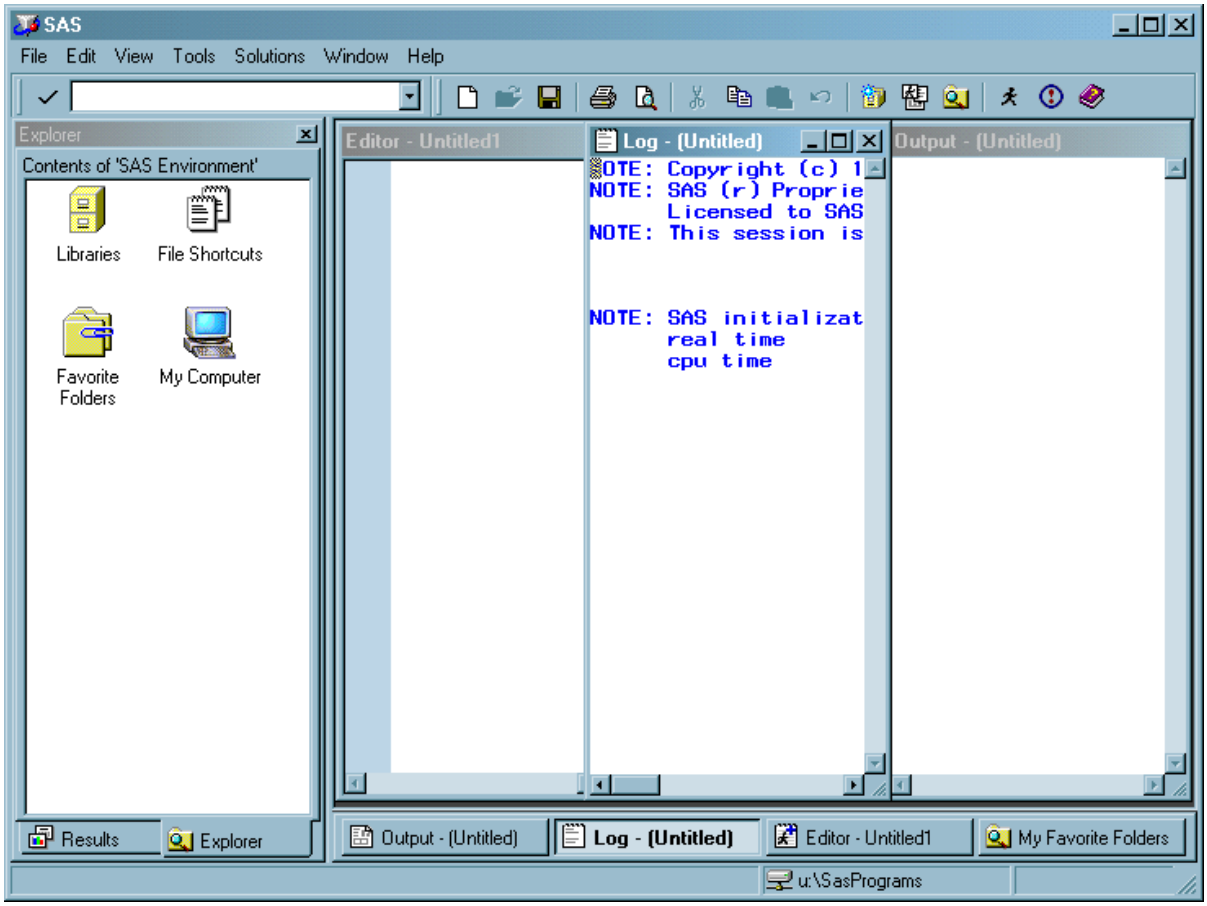
デフォルトの構成でも効率的に SAS を使用できます。簡単にアクセスできるように別のウィンドウも開いて、メイン SAS ウィンドウ内に再配置できます。たとえば、**お気に入りフォルダ**ウィンドウを開いて最小化にして、すべてのウィンドウを一度に表示できるようにモザイク状に配置できます。

この構成を

1. 実行するには、**表示** ⇨ **お気に入りフォルダ**を選択して、お気に入りフォルダを開きます。
2. **お気に入りフォルダ**ウィンドウのウィンドウタイトルバーの最小化ボタンを選択します。
3. 次のメニューパスを選択します:**ウィンドウ** ⇨ **上下に並べて表示**

この結果、メイン SAS ウィンドウは、次のように表示されます。

***図 2.5*** *カスタマイズされた SAS セッション*




また、ウィンドウのドッキングを解除してすべてのウィンドウを任意の場所に配置したり、ドッキングビューを最小化したりできます。ドッキング表示の詳細については、“ドッキングビューの使用 ” (39 ページ) を参照してください。

メイン SAS ウィンドウの外観の設定に使用する SAS コマンドのリストは、“メイン SAS ウィンドウを制御する SAS コマンド” (315 ページ) を参照してください。

### *ウィンドウの色の変更*

ウィンドウコンポーネントの色の変更は、Windows と SAS で共有されています。標準的なウィンドウの各部分の色は、ほとんどの場合、Windows デスクトップのプロパティで変更します。ログウィンドウのエラーメッセージの文字色など、ウィンドウ要素の中には、SAS により制御されるものもあります。

SAS により制御されるウィンドウコンポーネントを変更するには、次のいずれかの操作を行います。

- コマンドバーに `sascolor` を入力します。

- **ツール** ⇨ **オプション** ⇨ **色**を選択します。

**SASCOLOR** ウィンドウで、特定の要素の色を選択します。

アクティブウィンドウを閉じて開き直すと、新しい色の設定が有効になります。

詳細については、**SASCOLOR** ウィンドウの SAS ヘルプとドキュメントを参照してください。

### *システムオプションを使用したウィンドウ環境のカスタマイズ*

SAS 内のデフォルトのウィンドウ環境を制御できる SAS システムオプションが用意されています。最もよく使用されるオプションを次に示します。

AWSDEF
: SAS が初期化された際の、メイン SAS ウィンドウの位置と大きさを指定します。“メイン SAS ウィンドウのサイズと配置の変更” (65 ページ) を参照してください。

AWSTITLE
: メイン SAS ウィンドウのタイトルバーのテキストを指定します。“SAS セッションのタイトルの変更” (65 ページ) を参照してください。

HELPREGISTER
: メイン SAS ウィンドウのヘルプメニューに、カスタムヘルプにアクセスするヘルプを追加できます。“HELPREGISTER システムオプション:Windows” (517 ページ) を参照してください。

ICON
: SAS 初期化時の SAS ウィンドウを最小化します。“SAS セッションの最小化” (65 ページ) を参照してください。

REGISTER
: メイン SAS ウィンドウのツールメニューにアプリケーションを追加して、名前をクリックして実行できるようにします。“ツールメニューにアプリケーションの追加” (66 ページ) を参照してください。

SASINITIALFOLDER
: SAS が起動したときの現在のフォルダと**開くと名前を付けて保存**ダイアログボックスのデフォルトフォルダのパス名を設定します。“SASINITIALFOLDER システムオプション:Windows” (558 ページ) を参照してください。

SPLASHLOC と NOSPLASH
: SAS セッションの起動時に表示されるロゴ画面のダイナミックリンクライブラリのパス名を指定するか、ログ画面を表示しないことを指定します。“SAS 起動時のカスタムロゴ画面の表示” (66 ページ) を参照してください。

USERICON
: SAS/AF アプリケーションに組み込むユーザー定義アイコンを指定します。“ユーザー定義のアイコンを SAS に追加する” (67 ページ) を参照してください。

WEBUI
: オブジェクトを選択するためのポインタの使用を有効にし、1 回のクリックでオブジェクトのデフォルト実行を行うよう指定します。“SAS で Web 拡張を使用可能にする” (67 ページ) を参照してください。

これらのシステムオプションは、SAS 構成ファイル、または SAS 起動時にコマンドプロンプトの SAS コマンドで指定できます。システムオプションの中には、OPTIONS ステートメントで有効なものもあります。これらのオプションの構文と指定する場所に関する情報は、“Windows 版に固有の SAS システムオプション” (473 ページ) を参照してください。これらのオプションの全てのリストは、“メイン SAS ウィンドウを制御する SAS システムオプション” (314 ページ) を参照してください。

## メイン SAS ウィンドウ

### メイン SAS ウィンドウのサイズと配置の変更

AWSDEF システムオプションを使用して、SAS 初期化時のメイン SAS ウィンドウの位置とサイズを制御できます。たとえば、SAS セッションが常に画面の左上 4 分の 1 を占めるようにしたい場合は、SAS 構成ファイルで AWSDEF オプションを次のように指定します。

```
-awsdef 0 0 50 50
```

詳細については、“AWSDEF システムオプション:Windows” (493 ページ) を参照してください。

### SAS セッションのタイトルの変更

デフォルトでは、メイン SAS ウィンドウのタイトルバーには、テキスト **SAS** を選択します。別のタイトルを表示する場合は、AWSTITLE システムオプションを使用できます。たとえば、タイトルを **My SAS Session** に設定する場合は、SAS 構成ファイルで次のオプションを指定します。

```
-awstitle "My SAS Session"
```

詳細については、“AWSTITLE システムオプション:Windows” (495 ページ) を参照してください。

### ヘルプメニューにヘルプを追加

HELPREGISTER システムオプションを使用して、メイン SAS ウィンドウのヘルプメニューから、カスタマイズしたヘルプにアクセスできます。**ヘルプ**メニューには、WinHelp (.hlp)、HTML (.htm)、または Microsoft HTML Help (.chm)ファイルを最大 20 個まで追加できます。HELPREGISTER システムオプションの引数では、次の操作が可能です。

- ヘルプファイル内のトピックにリンクできます。
- ヘルプメニューに表示されるテキストをカスタマイズできます。
- ヘルプメニュー項目上にポインタを置いた場合にメッセージ行に表示されるテキストをカスタマイズできます。

*注:* WinHelp (WinHlp32.exe)は、Microsoft 社によるサポートが終了しています。Microsoft HTML ヘルプ(.chm)の使用をお勧めします。

**ヘルプ**メニューに複数のヘルプファイルを追加するには、構成ファイルまたは SAS 起動時のコマンドプロントのいずれかで、複数の HELPREGISTER システムオプションを使用します。

次の例では、**ヘルプ**メニューにヘルプファイル My Help.htm を追加します。

```
sas -helpregister "My Help" c:\mysashelp\myHelp.htm html
```

詳細については、“HELPREGISTER システムオプション:Windows” (517 ページ) を参照してください。

### SAS セッションの最小化

ICON システムオプションを使用して、起動時に SAS を最小化できます。バッチジョブの実行中は、このシステムオプションを使用して画面スペースを節約することができます。詳細については、“ICON システムオプション:Windows” (521 ページ) を参照してください。

### ツールメニューにアプリケーションの追加

REGISTER システムオプションを使用して、メイン SAS ウィンドウの**ツール**メニューにアプリケーション名を追加できます。追加した名前をクリックすると、そのアプリケーションを実行できます。REGISTER システムオプションには、引数として、メニュー名、動作環境のコマンドまたは実行可能ファイルへのパスを指定します。また、作業フォルダも指定できます。

次の例では、SAS フォルダの内容をプリントするコマンドを追加します。

```
-register "Contents of SAS"
          "dir c:\program files\sas"
```

**ツール**メニューの **SAS の内容**をクリックすると、Windows DIR コマンドの出力が、コマンドプロンプトウィンドウに表示されます。

次の例では、メニューに.EXE ファイルを追加し、その作業フォルダとして **C:\EXDATA** を指定します。

```
-register "Excel" "excel.exe" "c:\exdata"
```

この操作によって、**Excel** がメニューに追加されます。ツールメニューの **Excel** をクリックすると、EXCEL.EXE が起動します。

REGISTER システムオプションは、起動オプションとしてのみ有効です。つまり、SAS 構成ファイルまたは SAS 起動コマンド内でのみ有効です。詳細については、“REGISTER システムオプション:Windows” (552 ページ) を参照してください。

### 現在のフォルダの初期パスと、開くと保存のダイアログボックスに示すパスの設定

デフォルトの現在のフォルダではなく、別の現在のフォルダを指定して SAS を起動する場合、SAS の起動時に SASINITIALFOLDER システムオプションを使用します。SASINITIALFOLDER オプションに指定したパス名は、初期の現在のフォルダと**開くと名前を付けて保存**ダイアログボックスの初期パス名を設定します。

構成ファイルまたは SAS 起動時のコマンド行に SASINITIALFOLDER オプションを指定できます。たとえば、SAS 起動時に **sas -sasinitialfolder "c:\mySasFiles"**と指定します。

詳細については、“SASINITIALFOLDER システムオプション:Windows” (558 ページ) を参照してください。

### SAS 起動時のカスタムロゴ画面の表示

SAS 起動時に独自のロゴを表示するには 、

1. 表示するロゴを作成して、Windows ビットマップ(ファイル拡張子が.bmp)として保存またはリソースとしてコンパイルして DLL (ダイナミックリンクライブラリ)に組み込みます。
2. SAS の起動時、-SPLASHLOC システムオプションに、ビットマップが含まれるファイルのフルパス名を指定します。ビットマップが DLL の場合は、リソース番号も指定する必要があります。デフォルトのリソース番号は 1 です。

たとえば独自のロゴが **C:\MYBMPS\SPLASH.BMP** に保存されている場合、次の SPLASHLOC システムオプションを指定します。

```
-splashloc c:\mybmps\splash.bmp
```

たとえば独自のロゴが **C:\MYDLLS\OPENING.DLL** にリソース 101 として保存されている場合、次の SPLASHLOC システムオプションを指定します。

```
-splashloc c:\mydlls\opening.dll 101
```

詳細については、“SPLASHLOC システムオプション:Windows” (571 ページ) を参照してください。

### ユーザー定義のアイコンを SAS に追加する

USERICON システムオプションを使用して、SAS に独自のアイコンを追加できます。追加したアイコンは、SAS/AF アプリケーションおよび SAS/EIS アプリケーションで使用できます。USERICON システムオプションの構文は次のとおりです。

-USERICON *icon-resource-file number-of-icons*

*icon-resource-file* 引数には、ユーザーアイコンが含まれるダイナミックリンクライブラリ(DLL)ファイルのフルパスを指定します。*number-of-icons* 引数には、リソースファイル中にあるアイコンの数を指定します。たとえば、次のシステムオプションでは、c:\junk フォルダの中の ICONS.DLL というアイコンリソースファイルに、4 つのアイコンがあることを指定します。

```
-usericon c:\junk\icons.dll 4
```

アイコンリソースファイルとして使用される DLL は、Win32 Software Development Kit (32 ビット)を使用して作成する必要があります。リソースファイルを作成する方法の詳細については、Microsoft Win32 Software Development Kit のマニュアルを参照してください。

FRAME エントリを使用して、SAS/AF と SAS/EIS アプリケーションにアイコンを組み込むことができます。詳細な情報は “USERICON システムオプション:Windows” (585 ページ) と SAS/AF ソフトウェアと SAS/EIS ソフトウェアの SAS ヘルプ及びドキュメントを参照してください。

### SAS で Web 拡張を使用可能にする

Microsoft Internet Explorer (IE) 5.0 以降をインストールしている場合、WEBUI システムオプションを使用すると、**SAS エクスプローラ**ウィンドウなどの SAS ウィンドウを IE Web ページのように操作できます。

オブジェクトをポイントするとそのオブジェクトを選択でき、シングルマウスクリックでデフォルトのアクションを起動できます。オブジェクトを範囲選択するには、Shift キーを押しながら、グループの先頭オブジェクトと最終オブジェクトをクリックします。

複数の項目を選択するには、Ctrl キーを押しながら、グループ内の各項目をクリックします。

## ツールバーのカスタマイズ

### ツールバーのカスタマイズについて

よく使用するコマンドは、簡単に使用できるようにボタンに割り当てられています。ツールバー設定をカスタマイズして、よく使用するコマンドにアクセスしたり、特定のアプリケーションウィンドウのツールバーを作成したりできます。このセクションでは、そのような場合にツールバー設定をカスタマイズする方法について説明します。


- “ツールのカスタマイズダイアログボックスの使用” (68 ページ)
- “全般的なツールバーのプリファレンスの設定” (68 ページ)
- “ツールバーのカスタマイズ” (68 ページ)
- “ツールバーへのツールの追加” (70 ページ)
- “ツールバーからのツールの削除” (71 ページ)
- “特定のアプリケーションまたはウィンドウで使用するようにツールバーをカスタマイズし保存する” (71 ページ)

- “ツールをデフォルト設定にリセットする” (72 ページ)
- “作成可能な便利なツール例” (72 ページ)


### *ツールのカスタマイズダイアログボックスの使用*

すべてのツールバー設定のカスタマイズには、ツールのユーザー設定ダイアログボックスを使用します。**ツールのユーザー設定** ダイアログボックスを開くには、次のいずれかの操作を行います。

- コマンドバーで TOOLEDIT を入力します。
- **ツール** ⇨ **カスタマイズ**を選択します。

全体的なツールバー設定には**ツールバー**タブを使用して、**カスタマイズ**タブを使用してツールバーのツールを定義します。

### *全般的なツールバーのプリファレンスの設定*

**ツールバー**タブには、ツールバーの動作と外観を制御する設定があります。ツールで設定できるオプションは、次のとおりです。

**全般**
ボタン表示とヘルプオプションを指定します。これらのオプションには次のものが含まれます。

**大きいアイコン**
ツールバーに大きいボタンのセットを使用するかどうかを指定します。この設定は、高解像度ディスプレイで有用です。

**ツールバーにスクリーンヒントの表示**
ボタン上にポインタを置いた場合、ボタンの短い説明を表示するかどうかを指定します。

**ツールバー**
ツールバーとコマンドバーを表示するかどうかを指定します。これらのオプションには次のものが含まれます。

**アプリケーションツールバー**
アクティブなアプリケーションのツールバーを表示するかどうかを指定します。

**コマンドバー**
コマンドバーを表示して、コマンドバーを使用するオプションを有効化する場合に指定します。

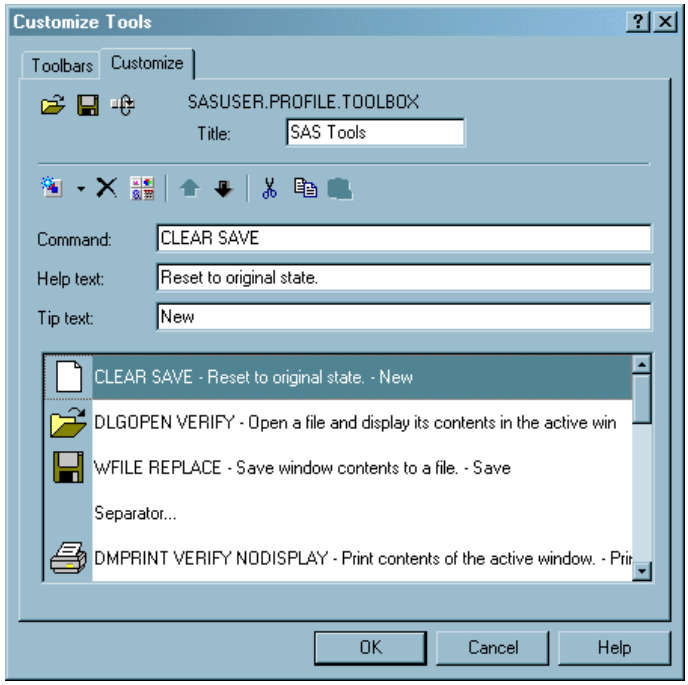
- **オートコンプリートの使用**を選択すると、前に入力したコマンドが保存され、コマンドの入力を開始したときにコマンド全体が補完入力されます。
- **最後に使用した順にコマンドをソート**を選択すると、最後に入力したコマンドからコマンドバーリストに表示されます。この設定が選択されていない場合、ドロップダウンリストのコマンドは、使用頻度の順に並べられます。
- **コマンドの保存数**で、コマンドバーリストに保存するコマンド数を入力します。有効な値の範囲は、0-50 です。デフォルトは 15 です。

**ツールバー**タブを構成している場合は、**カスタマイズ**をクリックしてカスタマイズを完了するか、**OK** をクリックしてダイアログボックスを閉じます。

### *ツールバーのカスタマイズ*

**カスタマイズ**タブでは、ツールバーのコマンドを追加、削除、変更できます。

*図 2.6 ツールのユーザー設定ダイアログボックスのユーザー設定タブ*




次に、各ボタン(コマンド)とフィールドを説明します。

開く
　ツールバーファイルを開きます。

保存
　ツールバーファイルを保存します。

元に戻す
　ツールバーをデフォルト設定に戻します。

タイトル
　ツールバーがドッキング解除された場合にタイトルバーに表示されるタイトルテキストが表示されます。

ツールの追加
　ツールバーにツールまたは空白区切りを追加します。このツールは 2 つに分かれています。左のボタンをクリックすると、ツールバーに空白のツールが追加されます。下向きの矢印をクリックすると、**空白ツール**または**区切り**を追加できます。エクスプローラなど、アクションセットを定義するウィンドウでは、選択肢に**操作**を選択します。

ツールの削除
　ツールバーリストから選択したツールを削除します。

アイコンの変更
　ビットマップブラウザを開いて、選択したツールの新しいアイコンを選択します。

ツールを上へ移動
　ツールバーリストのツールを 1 つ上へ移動します。

ツールを下へ移動
　ツールバーリストのツールを 1 つ下へ移動します。

削除
　ツールバーリストから選択したアイコンを削除して、クリップボードに置きます。

コピー
選択したアイコンのコピーをクリップボードに置きます。

貼り付け
クリップボードからツールバーリストの選択したツールにアイコンをコピーします。

コマンド
ツールバーリストで選択したツールのコマンドが表示されます。**コマンド**ボックスのコマンドを変更できます。

ヘルプテキスト
ポインタがツールバーのボタン上に置かれた場合、ステータスバーのメッセージ領域に表示されるヘルプテキストが示されます。**ヘルプテキスト**ボックスのヘルプテキストを変更できます。

ヒントテキスト
ポインタがツールバーのボタン上に置かれた場合に、ボタンの下にスクリーンチップが表示されます。**ヒントテキスト**ボックスで、ヒントテキストを変更することができます。

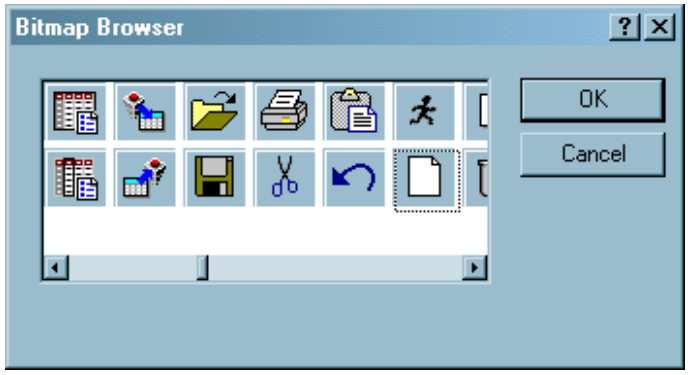
ツールバーリスト
に定義されたボタン、コマンド、ヘルプテキスト、区切りが示されます。

### *ツールバーへのツールの追加*

ツールバーにツールを追加するには 、

1. 次のいずれかの操作を実行します。
   - **ツールの追加**ボタンをクリックしてツールバーリストに空白ツールを追加します。**コマンド**ボックスに SAS コマンドを入力します。
   - **エクスプローラ**ウィンドウや**お気に入りフォルダ**ウィンドウなど、ツールセットが事前に定義されているウィンドウでは、ツールの追加の下向きの矢印をクリックして**操作**を選択します。**アクションの追加**ダイアログボックスから、アクションを 1 つ選択します。ツールバーに新しいアクションが追加されます。複数のコマンドをセミコロンで区切って入力することもできます。
   - ツールの追加の下向きの矢印をクリックして、**区切り**を選択してツールバーリストに区切りを追加します。
2. **ビットマップ ブラウザ** ボタンをクリックします。ツールのアイコンを 1 つ選択し、**OK** をクリックします。
3. **ヘルプテキスト**ボックスにヘルプテキストを入力します。
4. **ヒントテキスト**ボックスのスクリーンチップのテキストを入力できます。
5. **ツールを上へ移動**またはツールを下へ移動ボタンをクリックして、ツールバーリスト内でツールを配置します。
6. 終了したら**保存**をクリックします。**ツールの保存**ダイアログボックスで、ライブラリ名、カタログ名、ツールボックス名を入力します。次に **OK** をクリックします。

*図 2.7 ビットマップブラウザダイアログボックス*




### *ツールバーからのツールの削除*

ツールバーからツールを削除するには 、

1. ツールバーから削除するツールを選択します。
2. **ツールの削除**をクリックします。
3. 終了したら**保存**をクリックします。

### *特定のアプリケーションまたはウィンドウで使用するようにツールバーをカスタマイズし保存する*

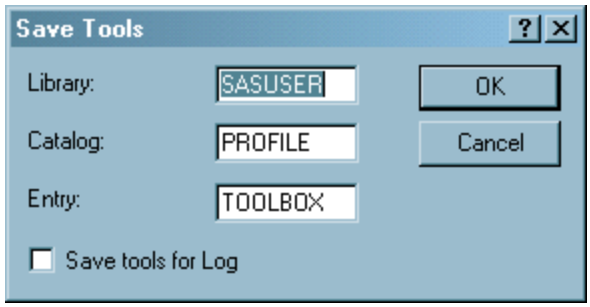
ツールバーにコマンドを追加する前に、そのコマンドがメニューから使用できることを確認します。常に有効な印刷コマンドとコピーコマンドを除き、メニューから使用できるコマンドにのみボタンを使用できます。

次の手順で、特定のアプリケーションまたはウィンドウで使用するツールバーをカスタマイズします。

1. アプリケーションまたはウィンドウをクリックし、アクティブウィンドウにします。
2. 前述のセクションの説明を参照して、ツールを追加、削除して、ツールバーをカスタマイズします。
3. ツールバーのカスタマイズが終了したら、**保存**をクリックします。**ツールの保存**ダイアログボックスが表示されます( 図 2.8 (72 ページ) を参照)。
4. 参照ライブラリ、カタログ、エントリのフィールドは、SAS により設定されます。ダイアログボックスで**ツールの保存** ***window*** のチェックボックスを選択します。ここで、*window* はアクティブなウィンドウです。次に **OK** をクリックします。

   **ツールの保存** ***window*** のチェックボックスを選択すると、特定のアプリケーションまたはウィンドウの PMENU エントリと同じライブラリ、カタログ、エントリ名を使用してツールバーがアプリケーションまたはウィンドウに関連付けられます。SAS では、アプリケーションカタログを検索する前に、SASUSER.PROFILE のツールボックスエントリが検索されます。

*図 2.8 ツールの保存ダイアログボックス*




特定のアプリケーションに関連付けられるようにツールバーを保存すると、そのアプリケーションウィンドウがアクティブな場合、SAS によりツールが自動的にロードされます。

TOOLLOAD コマンドを使用して、カスタムツールバーを手動でロードできます。詳細については、"TOOLLOAD コマンド:Windows" (354 ページ) を参照してください。

### *ツールをデフォルト設定にリセットする*

ツールバーを標準の設定に戻すには、**デフォルトに戻す**をクリックします。デフォルトのツール設定に戻すかどうかが確認されます。**Yes** をクリックすると、ツールは元の設定にリセットされます(SAS がインストールされたときの設定)。

SAS アプリケーションでアプリケーションウィンドウのデフォルトのツールバーが定義されている場合は、**デフォルトに戻す**ボタンをクリックすると、ツールバーの設定が復元されます。

### *作成可能な便利なツール例*

拡張エディタがアクティブウィンドウの場合に Base SAS サンプルプログラムの SAS Web ページを開くツールを作成するとします。その場合、次の手順に従います。

1. 拡張エディタをアクティブウィンドウにします。
2. **カスタマイズ**タブ(**ツールのカスタマイズ**ダイアログボックス)で、**ツールの追加**ツールバーボタンをクリックします。リストボックスに新しいツールのテンプレートが作成されます。
3. **コマンド**フィールドに `http://support.sas.com/techsup/sample/base_samples.html` を入力します。

   **ヘルプテキスト**フィールドに `Sample programs for Base SAS on sas.com` を入力します。

   **ヒントテキスト**フィールドに `sas.com sample programs` を入力します。
4. **アイコンの変更**ボタンをクリックします。**ビットマップ ブラウザ**ダイアログボックスから、SAS Web サイトのサンプルプログラムに適したビットマップを選択してから **OK** をクリックします。
5. **ツールを上へ移動**および**ツールを下へ移動**ボタンを使用して、ツールバーのツールの位置を決定します。
6. **ツールバーの保存**ボタンをクリックして、ユーザーのデフォルトのツール構成を持つツールを保存します。

次に、作成できる便利なツールの例を示します。

WPGM; CLEAR; INCLUDE C:\SAS\MYPROGRAM.SAS
**拡張エディタ**ウィンドウに、編集によく使用するプログラムを挿入します。

WPGM; FILE C:\SAS\MYPROGRAM.SAS; CLEAR
SAS プログラムの編集後、保存して、**拡張エディタ**ウィンドウをクリアします。

WPGM; CLEAR; INCLUDE C:\SAS\MYPROGRAM.SAS; SUBMIT
よく使用する SAS プログラムを挿入して、サブミットします。

WPGM; CLEAR; INCLUDE C:\SAS\SIGNON.SAS; SUBMIT
リモートシステムにサインオンする SAS プログラムを挿入して、サブミットします。たとえば、MVS セッションにサインオンするには、SIGNON.SAS プログラムに次を含めます。

```
options comamid=tcp remote=mytso; libname remtdata 'mylib.mydata.monthly'; signon;
```

リモートセッションへのサインオンに関する詳細な情報は *SAS/CONNECT User's Guide* を参照してください。

WPGM; CLEAR; INCLUDE C:\SAS\DOWNLOAD.SAS; SUBMIT
リモートセッションからデータセットをダウンロードする SAS プログラムを挿入して、サブミットします。仮にリモートセッションに既にサインオンしているとすると、DOWNLOAD.SAS は次を含みます。

```
proc download data=remtdata.june; /* where libname 'remtdata' is */
    /* already defined*/
run;
```

リモートセッションへのサインオンに関する詳細な情報は *SAS/CONNECT User's Guide* を参照してください。

TOOLLOAD BAR SASUSER.PROFILE.MORTOOLS
別のツールがまとめられたツールバーをロードします。

# オンラインヘルプとドキュメントの利用

## *Microsoft HTML ヘルプの使用*

SAS のヘルプとドキュメントでは、Microsoft HTML ヘルプ形式が採用されているため、簡単な操作、インデックス、検索機能が実現されています。その利用には、Microsoft Internet Explorer (IE) 5.00 および Microsoft HTML Help 1.3 以降がインストールされている必要があります。Microsoft HTML ヘルプを使用するために SAS を設定する必要はありません。

## *コマンドバーからのヘルプの利用*

コマンドバーに HELP コマンドを入力すると、アクティブウィンドウおよび SAS 言語要素に関するヘルプを利用できます。次の表は、HELP コマンドの引数と、その結果表示される SAS ヘルプとドキュメントの内容を示します。

***表 2.4*** *コマンドバーで使用できるヘルプの種類*

| ヘルプの引数 | SAS ヘルプとドキュメントの表示 | 例 |
|---|---|---|
| なし | アクティブウィンドウのヘルプ | **ヘルプ** |

| ヘルプの引数 | SAS ヘルプとドキュメントの表示 | 例 |
|---|---|---|
| 言語要素の名前と種類 | 指定した言語要素のヘルプ | **ライブラリ名ステートメントのヘルプ** |
| HELP | HELP コマンドの使用方法 | **ヘルプ ヘルプ** |

## *ダイアログボックスのヘルプの利用*

ダイアログボックスでヘルプにアクセスするには、ダイアログボックスの上にある?をクリックします。そして情報を知りたい項目をクリックします。ポップアップウィンドウを閉じるには、ダイアログボックスの任意の場所をクリックします。

ダイアログボックスに?ボタンがない場合、ヘルプボタンをクリックするか F1 キーを押します。

## *SAS プロダクトのヘルプの利用*

現在のアクティブウィンドウに関連付けられた SAS プロダクトのヘルプ情報にアクセスするには、次のいずれかの操作を実行します。

- **ヘルプ**ボタン(疑問符付きの本マーク)をクリックします。
- F1 キーを押します。
- ヘルプメニューの**このウィンドウの使い方**を選択します。たとえば、ヘルプボタンをクリックした時に、アクティブウィンドウが **SAS/GRAPH** ウィンドウであれば、SAS/GRAPH ソフトウェアに関する SAS ヘルプとドキュメント情報が表示されます。

インストールされた SAS プロダクトの全てのドキュメントは、SAS ヘルプとドキュメントの目次の **SAS プロダクト**エントリに用意されています。

## *ヘルプメニューからのヘルプの利用*

### *概要: ヘルプメニューからヘルプを利用する*

SAS セッションでは、常に**ヘルプ**メニューを使用できます。次に、ヘルプメニューから使用できるヘルプトピックについて説明します。

**このウィンドウの使い方**
アクティブウィンドウに関連するヘルプ情報。このトピックを選択すると、ヘルプボタンをクリックまたは F1 キーを押した場合と同じトピックが表示されます。

**SAS ヘルプとドキュメント**
SAS の使い方の学習を支援するチュートリアルとサンプルプログラム、サイトにインストールされたすべてのプロダクトに関する全てのドキュメント、追加サポートを依頼するための SAS 連絡先情報が含まれます。

**SAS 入門ガイド**
SAS を初めて使用するときに役立つチュートリアルが開始されます。

**SAS プログラミングの学習**
インストールされていれば、SAS プログラミングスキルの向上を支援する SAS Online Tutor が開始されます。SAS Online Tutor は、別途ライセンスが必要なプロダクトです。

**SAS Web ページ**
SAS Web サイト上の、有益なページへのリンクを提供します。テクニカルサポート、FAQ、SAS へのフィードバック送信、SAS ホームページなどにアクセスできます。

**SAS システムについて**
**SAS システムについて**ダイアログボックスが開き、SAS および Windows のソフトウェアレベルと、ハードウェア情報が表示されます。SAS の著作権情報とサイト情報にもアクセスできます。**システム情報**ボタンをクリックすると、**Microsoft システム情報**ウィンドウが開きます。

### *SAS から SAS Institute (と他の Web サイト)にアクセスする*

SAS は、ローカルの Web ブラウザを起動して HTML ファイルを表示するように設定されています。Web ブラウザは、次のいずれかの方法で起動できます。

- コマンドバーに URL (Uniform Resource Locator)を入力します。**Web** ダイアログボックスの**プリファレンス**で指定したブラウザが起動します。
- コマンドバーに **wbrowse** を入力します。ブラウザが起動して、SAS ホームページまたは**プリファレンス** ダイアログボックスの Web タブで指定したデフォルトの URL が表示されます。詳細については、“WBROWSE コマンド:Windows” (358 ページ) を参照してください。

SAS ホームページなど、インターネット上の Web ページにアクセスするには、インターネットアクセスを許可しているネットワークにワークステーションが接続されている必要があります。

## *SAS リモートブラウザでの出力とヘルプの表示*

### *リモートブラウジングについて*

リモートブラウジングを使用して、ローカルコンピュータの Web ブラウザに次の種類の HTML ドキュメントを表示できます。

- WBROWSE コマンドに指定された URL
- ODS 出力

リモートブラウジングを使用してローカルに ODS 出力を表示することで、Windows 64 ビットサーバーで使用できないブラウザプラグインを必要とする出力にアクセスできます。

リモートブラウザサーバーと呼ばれるソフトウェアエージェントは、ローカルコンピュータで実行されます。HTML コンテンツを表示する必要がある場合、SAS は Remote Browser Server に接続して、コンテンツの参照先の URL を Remote Browser Server に送信します。URL は、Remote Browser Server から Web ブラウザに渡され、表示されます。Remote Browser Server がコンピュータ上で起動中でない場合、Remote Browser Server のダウンロードに必要な URL を示すダイアログボックスが表示されます。

リモートブラウジングの構成には、HELPHOST と HELPPORT の 2 つのシステムオプションが用意されています。これらのオプションには、HTML コンテンツが表示されるローカルコンピュータのホスト名とポート番号を指定します。ほとんどの場合、これらのオプションの設定は必要ありません。デフォルトでは、HELPHOST は X11 DISPLAY 環境変数で指定されたホスト名、HELPPORT はリモートブラウザサーバーの標準ポートに設定されます。

リモートブラウジングは、Windows 64 ビットサーバーでサポートされています。

### 一般ユーザーのリモートブラウジングとファイアウォール

デスクトップコンピュータと SAS ホストコンピュータ間にネットワークのファイアウォールが存在する場合、SAS セッションから Web ブラウザに Web ページを表示することはできません。通常、Web ブラウザからタイムアウトまたは接続エラーによりこの問題が発生します。タイムアウトまたは接続エラーが通知された場合、システム管理者に問い合わせてください。

### システム管理者のリモートブラウジングとファイアウォール

デスクトップコンピュータと SAS ホストコンピュータ間にファイアウォールが存在する場合に Web ページを表示するには、Web ブラウザから SAS への接続を許可するファイアウォールルールを追加する必要があります。ファイアウォールルールには、SAS リモートブラウジング接続を許可するネットワークポート範囲を指定します。リモートブラウジングのネットワークポート範囲を選択、構成できるシステム管理者に依頼します。範囲は、同時に使用する SAS ユーザー数によって異なります。同時に使用する SAS ユーザー数の約 3 倍の値をネットワークポート数として用意することをお勧めします。

ファイアウォールルールの追加後、そのネットワークポート範囲内でネットワーク接続をリッスンするように SAS を構成する必要があります。通常は SAS が空いているネットワークポートを選択しますが、HTTPSERVERPORTMIN と HTTPSERVERPORTMAX のシステムオプションを使用して SAS が選択するネットワークポートを制限します。SAS 構成ファイルにこれらのシステムオプションを追加します。HTTPSERVERPORTMIN には、ネットワーク範囲の最小ポート番号を設定します。HTTPSERVERPORTMAX には、ネットワーク範囲の最大ポート番号を設定します。たとえば、システム管理者がネットワークポート範囲を 8000 から 8200 に設定した場合、次のようにシステムオプションを設定します。

```
httpserverportmin=8000
httpserverportmax=8200
```

これらのシステムオプションの設定後、デスクトップコンピュータは Web ページを表示できます。ポート番号が適当ではない、またはシステムオプションの指定が正しくない場合、SAS ログにメッセージが表示されます。

これらのオプションに関する詳細な情報は、次を参照してください。
“HTTPSERVERPORTMIN= System Option” (*SAS System Options: Reference*)および
“HTTPSERVERPORTMAX= System Option” (*SAS System Options: Reference*)。

### ODS 出力のリモートブラウジングの使用

SAS Output Delivery System (ODS)は、SAS データのグラフレポートの生成に使用されます。リモートブラウジングを使用することで、出力の生成と同時に、または**結果**ウィンドウで要求した際に、SAS セッションから出力を直接表示できます。

リモートブラウジングは、PDF 形式と RTF 形式で ODS 出力を表示します。非 HTML データタイプの適切なプラグインがブラウザにない場合、ブラウザは出力を表示せずに、ダイアログボックスを表示します。このダイアログボックスを使用して、PC にレポートをダウンロードして、XLS ファイルには Excel を使用するなど、ローカルプログラムを使用してレポートを表示できます。

ODS 出力(PDF 形式と RTF 形式)の自動表示は、デフォルトでは無効化されています。**結果**ウィンドウで AUTONAVIGATE コマンドを発行することで、ODS 出力の自動表示を有効化できます。

### リモートブラウザサーバーのインストール

Remote Browser Server は、SAS セッションから直接インストールできます。SAS からリモートブラウジング接続ができない場合、インストーラのダウンロード URL を示すダイアログボックスが表示されます。この URL を使用して、Remote Browser Server をダウ

ンロードしてインストールします。SAS を終了しないようにします。Remote Browser Server をインストールするには、次の操作を行います。

1. ダイアログボックスに表示された URL を Web ブラウザに入力して、Enter キーを押します。
2. ダウンロードページが表示されるので、コンピュータに適切なインストーラをダウンロードします。
3. インストーラを実行します。
   - Windows 環境では、リモートブラウザサーバーはスタートアップ項目に追加され、ログオンするたびにサーバーが起動します。Remote Browser Server が起動中であることが示すアイコンが、システムトレイに表示されます。
   - Linux 環境では、ウィンドウ環境のスタートアップスクリプトにコマンド **`rbrowser`** を手動で追加します。Remote Browser Server は実行されますが、初期状態では最小化されます。

### *リモートブラウジングのシステムオプション*

コンピュータ上に Remote Browser Server を起動した後、HELPHOST と HELPPORT のシステムオプションを指定することで、リモートブラウジングを実行できます。

- HELPHOST システムオプションには、ホストコンピュータ名を指定します。このオプションを指定しない場合、X ウィンドウディスプレイで指定されたホスト名が使用されます。詳細については、“HELPHOST システムオプション:Windows” (515 ページ) を参照してください。
- HELPPORT システムオプションには、コンピュータにインストールされた Remote Browser Server のポート番号を指定します。詳細については、*SAS システムオプション: リファレンス*の HELPPORT システムオプションを参照してください。

これらのオプションは、SAS 起動時に構成ファイルで設定することも、または SAS セッション中に OPTIONS ステートメントまたは **SAS システムオプション**ウィンドウで設定することもできます。

*3 章*

# Windows 版での SAS エディタの使用

**拡張エディタの使用** . . . . . **79**
拡張エディタの機能 . . . . . 79
拡張エディタウィンドウの使用 . . . . . 80
独自のキーワードの作成 . . . . . 100
ファイルタイプへのファイル拡張子の割り当て . . . . . 101
拡張エディタオプションの設定 . . . . . 102
拡張エディタでショートカットキーをカスタマイズする . . . . . 106
拡張エディタの有効と無効の切り替え . . . . . 107
**プログラムエディタの使用** . . . . . **107**
プログラムエディタの使用の概要 . . . . . 107
拡張エディタとプログラムエディタの切り替え . . . . . 107
ファイルを開く . . . . . 108
行番号の使用 . . . . . 109
挿入位置の移動 . . . . . 109
タブの使用 . . . . . 110
改行について . . . . . 110
テキストの選択 . . . . . 110
テキストの削除 . . . . . 111
テキストの検索と置換 . . . . . 111
テキストのドラッグアンドドロップ . . . . . 112
ドラッグ時のスクロール . . . . . 113
リッチテキストフォーマットのテキストの使用 . . . . . 114
ファイルの保存 . . . . . 114
自動保存によるプログラムエディタファイルの保存 . . . . . 114
SAS テキストエディタ独自の機能について . . . . . 115

## 拡張エディタの使用

### *拡張エディタの機能*

拡張エディタでは、使い慣れた一部のプログラムエディタ機能を保持すると同時に、次の操作を実行できます。

- 色分けを使用して、SAS と SCL のプログラム要素や HTML と XML のドキュメント要素を識別できます。色分けの設定をカラースキーマに保存できます。
- 独自のキーワードを作成し、書式を設定します。

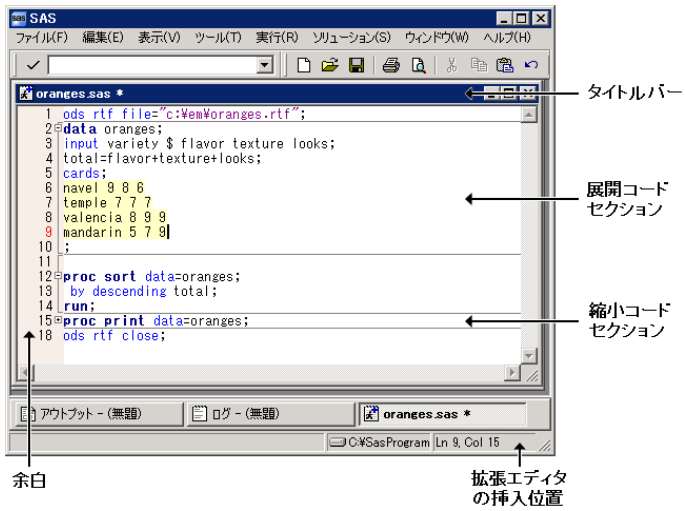
- Enter キーを押すと、次の行が自動的にインデントされます。
- SAS プロシジャ、DATA ステップ、マクロのセクションを展開または折りたたむことにより、SAS プログラムの高レベルのフローを表示したり、各ステートメントの詳細を表示します。
- キーボードマクロの記録機能を使用して、プログラム編集コマンドを記録および再生するマクロを作成できます。
- 省略形を使用してテキストを入力できるように、ショートカットを作成できます。
- コード行にブックマークを付けることにより、プログラムまたはドキュメントの異なるセクションに簡単にアクセスできます。
- ほとんどの拡張エディタコマンドのショートカットキーをカスタマイズできます。
- ファイルの複数のビューを開くことができます。複数の EE ウィンドウで複数のファイル(.sas プログラムファイル)を開きます。
- 言語要素名にカーソルを置いて F1 キーを押すと、SAS 言語のヘルプを表示できます。

## 拡張エディタウィンドウの使用

### 拡張エディタウィンドウの概要

**拡張エディタ**ウィンドウの各部分を次の画面に示します。

*図 3.1 拡張エディタウィンドウ*




タイトルバー
: タイトルバーには、拡張エディタアイコンとファイル名が表示されます。If the file is new, the filename is **Editor Untitled***x*, whre *x* is a window number.タイトルバーのアスタリスク( * )は、ファイルの変更内容が保存されていないことを示します。

展開コードセクション
: 展開コードセクションには、コードセクション内のすべてのコードが表示されます。詳細コードは、余白にマイナス記号( - )が表示されます。

折りたたまれたコードセクション
: 折りたたまれたコードセクションには、シグネチャコード行(キーワードを含むコード行)のみが表示されます。詳細コードは、余白にプラス記号( + )が表示されます。

拡張エディタの挿入位置
主な SAS ウィンドウでは、拡張エディタの挿入位置は、挿入位置のある行および列の位置を表示します。

余白
**拡張エディタ**ウィンドウの左余白は、次の目的に使用されます。

- 1 つ以上のテキスト行を選択します。
- コードセクションの展開および折りたたみを行います。
- 行番号、コードセクションのかっこ、およびブックマークを表示します。

拡張エディタウィンドウ間を移動するには、次の操作のいずれかを実行します。

- **拡張エディタ**ウィンドウを選択します。
- コマンドバーに **wnextedit** または **wpgm** と入力します。

## *ファイルを開く*

拡張エディタでファイルを開くさまざまな方法を次の表に示します。

***表 3.1*** *ファイルを開く*

| アクション | 実行手順 |
|---|---|
| 新規ファイルを開きます。 | 次のうちいずれかを実行します。<br>• コマンドバーに **wedit** と入力します。<br>• 次のメニューパスを選択します:**表示** ⇨ **拡張エディタ**<br>• **新規**ツールバーボタンをクリックします。<br>• 次のメニューパスを選択します:**ファイル** ⇨ **プログラムの新規作成** |
| **ファイルを開く**ダイアログボックスを使用して、既存のファイルを開きます。 | 次の方法の 1 つを使用してダイアログボックスを開きます。<br>• **開く**ツールバーボタンをクリックします。<br>• 次のメニューパスを選択します:**ファイル** ⇨ **プログラムを開く**<br>• コマンドバーに *fileopen* と入力します。*<br>ファイルを選択します。<br>**開く**をクリックします。<br>*注:* **最近使ったファイルの一覧**を使用して既存のファイルを開くこともできます。最近使ったファイルの一覧オプションを設定するには、**ツール** ⇨ **オプション** ⇨ **プリファレンス**にアクセスします。このオプションを設定すると、作成したファイルや開いたファイルが**ファイル**メニューの下部に一覧表示されます。 |
| ダイアログボックスを使用して、複数のファイルを開きます。 | 1. Ctrl キーを押したままにします。<br>2. ファイルを選択します。<br>3. **開く**をクリックします。 |

| アクション | 実行手順 |
|---|---|
| **ファイルを開く**ダイアログボックスを使わずに既存のファイルを開きます。 | 次のうちいずれかを実行します。<br>• コマンドバーに **wedit** "*filename*"を入力します。<br>• コマンドバーに **fileopen** "*filename*"を入力します。* |
| 開かれているファイルの複数のビューを開きます。 | 1. ファイルをアクティブウィンドウにします。<br>2. **ウィンドウ**を選択します。<br>3. **新しいウィンドウ**を選択します。<br>1 つのファイルについて複数ビューを開くと、任意のビューに加えられた変更が、すべてのビューに同時に反映されます。 |
| 開かれたファイルにファイルを追加します。 | 1. 追加するファイルをアクティブウィンドウにします。<br>2. **ファイル**を選択します。<br>3. **追加**を選択します。<br>4. 追加するフォルダおよびファイルを選択します。<br>5. **開く**をクリックします。 |
| *拡張エディタウィンドウがアクティブウィンドウになっている必要があります。 | |

*注:* **開く**ダイアログボックスのデフォルトのディレクトリを変更するには、SASINITIALFOLDER システムオプションを使用して SAS を起動するか、現在の作業ディレクトリを変更します。詳細については、“SASINITIALFOLDER システムオプション:Windows” (558 ページ) および “SAS の現在のフォルダの変更” (41 ページ) を参照してください。

### *ファイルの保存*

**拡張エディタ**ウィンドウのタイトルバーにアスタリスク( * )が表示される場合は、エディタ内のテキストがディスクに保存されていないことを示します。**拡張エディタ**のタイトルバーに**無題 x** が表示されている場合は、まだ保存されていない新しいファイルです。*x* は、ウィンドウ番号を示します。

**拡張エディタ**ウィンドウの内容を保存するには、**上書き保存**ツールバーボタン(記憶装置の形)をクリックします。初めて保存するファイルの場合は、**名前を付けて保存**ダイアログボックスが表示され、ファイル名の入力を求められます。

新しいファイル名を付けてファイルを保存するには、次の操作を行います。

1. 次のメニューパスを選択します:**ファイル** ⇨ **名前を付けて保存**
2. **保存する場所**フィールドでフォルダを選択します。
3. **ファイル名**フィールドにファイル名を入力します。
4. **ファイルの種類**フィールドからファイルの種類を選択します。
5. **名前を付けて保存**をクリックします。

*注:* **名前を付けて保存**ダイアログボックスのデフォルトのディレクトリを変更するには、SASINITIALFOLDER システムオプションを使用して SAS を起動するか、現在の作業ディレクトリを変更します。詳細については、“SASINITIALFOLDER システ

ムオプション:Windows" (558 ページ) および "SAS の現在のフォルダの変更" (41 ページ) を参照してください。

**プリファレンス**ダイアログボックスで自動保存機能を選択すると、SAS システムが予期しない時に終了した場合にも拡張エディタウィンドウの内容を復旧できます。拡張エディタでは、動作環境の Application Data フォルダに **Autosave of filename.$AS** のファイル名で自動保存されます。filename は、元のファイル名です。Windows では、Application Data フォルダのパス名は **c:\Users***user-ID***\AppData\Roaming** です。たとえば、フォルダ C:\TEMP 内の MYPROGRAM.SAS 自動保存ファイルのパスは、次のようになります。

```
C:\Users\my-user-ID\AppData\Roaming\SAS\EnhancedEditor\

Autosave of myprogram.$AS
```

自動保存ファイルは、ファイルを保存したとき、**拡張エディタ**ウィンドウを閉じたとき、または SAS セッションが正常に終了したときに、SAS により削除されます。ファイル名を変更したときに SAS が異常終了した場合には、自動保存ファイルは元のファイル名のままになります。

オペレーティングシステムでは、アプリケーションデータフォルダはデフォルトで非表示になります。フォルダを表示するには、次の操作を行います。

1. **スタート** ⇨ **コンピューター** ⇨ **整理** ⇨ **フォルダーと検索のオプション** ⇨ **表示**を選択します。
2. **すべてのファイルとフォルダーを表示**を選択します。
3. **登録されている拡張子は表示しない**の選択を解除します。
4. **適用**および **OK** を選択します。

自動保存機能の設定の詳細は、"編集プリファレンス" (60 ページ) および "WAUTOSAVE コマンド:Windows" (357 ページ)を参照してください。

*注:* 既存ファイルの上書き時にデータが失われないようにするために、拡張エディタではその内容を一時ファイル *SASuuuu.tmp* に保存します。*uuuu* は一意の 16 進値です。元のファイルは一時ファイルで置換されますが、元のファイルの作成日は保持されます。ファイルの保存および置換後に、一時ファイルが存在していてはなりません。一時ファイルがまだ存在している場合は、元のファイルの内容が更新されていることを確認して、一時ファイルを削除します。

### *同一ファイルの複数ビューの使用*

同一ファイルの複数ビューを開くことにより、同じファイルの異なる部分を同時に表示できます。複数のビューで作業していても、同じファイルの複数のコピーではなく、1 つのファイルで作業していることになります。

同一ファイルの複数ビューを開くには、次の操作を行います。

1. ファイルをアクティブウィンドウにします。
2. 次を選択します。**ウィンドウ** ⇨ **新しいウィンドウ**

タイトルバーのファイル名にコロンとビュー番号が追加されます。例を示します。**myfile.sas:1** および **myfile.sas:2**。

あるファイルの 1 つのビューに加えられた、テキストの変更または行のブックマーキングなどの変更は、すべてのビューに同時に反映されます。スクロールバーの移動、テキスト選択、コードセクションの展開または縮小などのアクションは、アクティブウィンドウのみで行われます。

### スクロールコマンドおよび行番号コマンド

拡張エディタでは、プログラムエディタでも使用されているスクロールコマンドおよび行番号コマンドがいくつかサポートされています。すべての拡張エディタコマンドは、コマンドバーからのみ入力可能です。

拡張エディタでは、次のスクロールコマンドがサポートされています。

*表 3.2 スクロールコマンド*

| コマンド | 説明 |
|---|---|
| UP | ファイルの先頭方向に 1 ページ移動する。 |
| DOWN | ファイルの末尾方向に 1 ページ移動する。 |
| LEFT | 左に 1 ページ移動する。 |
| RIGHT | 右に 1 ページ移動する。 |

行番号を表示するには、**拡張エディタオプション**ダイアログボックスの**行番号を表示する**を選択するか、コマンドバーに `nums` と入力します。行番号を非表示にするには、**行番号を表示する**の選択を解除するか、コマンドバーにもう一度 `nums` と入力します。

すべての行番号コマンドは、コロン( :)で始まります。コマンドと番号の間にスペースを入れる必要はありません。次の行番号コマンドがサポートされています。

*表 3.3 行番号コマンド*

| コマンド | 説明 | *n* のデフォルト値 | 許容可能な最大値 | 例 |
|---|---|---|---|---|
| :I*n* | 現在の行の後に、*n* 行を追加します。 | 1 | 9999 | :I4<br>現在の行の後に、4 行を挿入します。 |
| :IA*n* | 現在の行の後に、*n* 行を追加します。 | 1 | 9999 | :IA4<br>現在の行の後に、4 行を挿入します。 |
| :IB*n* | 現在の行の前に、*n* 行を追加します。 | 1 | 9999 | :IB2<br>現在の行の前に、2 行を挿入します。 |
| :D*n* | 現在の行から始まる *n* 行を削除します。 | 1 | 9999 | :D3<br>現在の行から始まる 3 行を削除します。 |

| コマンド | 説明 | *n* のデフォルト値 | 許容可能な最大値 | 例 |
|---|---|---|---|---|
| :R*nm* | 現在の行から始まる *m* 行のブロックを、*n* 回繰り返します。*n* と *m* の間には、スペースが必要です。 | 1 | 9999 | :R1 6<br>現在の行から始まる 6 行を、1 回繰り返します。 |

## *挿入位置の移動*

拡張エディタには、次の表に示されるように挿入位置を移動するショートカットキーが多数あります。Ctrl キーを押しながら G キーを押すと、**行番号へ移動**ダイアログボックスを開くことができます。他のショートカットキーは、すべて、定義に従って挿入位置を移動します。

***表 3.4*** *挿入位置を移動するショートカットキー*

| 使用するショートカットキー | 挿入位置の移動先 |
|---|---|
| 上矢印 | 上方向に 1 行移動する |
| 下矢印 | 下方向に 1 行移動する |
| 左矢印 | 左方向に 1 文字分移動する |
| 右矢印 | 右方向に 1 文字分移動する |
| Page Down | 下方向に 1 ページ分移動する |
| Page Up | 上方向に 1 ページ分移動する |
| Home | 現在行の先頭に移動する |
| Ctrl+Home<br>Ctrl+Page Up | 文書の先頭に移動する |
| End | 現在行の末尾に移動する |
| Ctrl+End<br>Ctrl+Page Down | 文書の末尾に移動する |
| Ctrl+上矢印 | 上方向にスクロールしながら、ファイルの先頭方向に移動する |
| Ctrl+下矢印 | 下方向にスクロールしながら、ファイルの末尾方向に移動する |
| Ctrl+右矢印 | 次の単語の先頭に移動する |
| Ctrl+左矢印 | 前の単語の先頭に移動する |

| 使用するショートカットキー | 挿入位置の移動先 |
|---|---|
| Ctrl+] | 対応するかっこまたは角かっこに移動する |
| Ctrl+G | 特定の行番号に移動する |
| Alt+上矢印 | 表示されている最初の行まで移動する |
| Alt+下矢印 | 表示されている最後の行まで移動する |
| Alt+右矢印 | 次の、大文字と小文字の切り替わる位置、またはワード境界まで移動する |
| Alt+左矢印 | 前の大文字と小文字の切り替わる位置、またはワード境界まで移動する |
| Shift+Tab | 前のタブストップまで戻る |

ショートカットキーを使用するほかに、コマンドバーで UP、DOWN、LEFT、RIGHT コマンドを使用して、挿入位置を 1 ページ上、1 ページ下、1 ページ左、1 ページ右に移動することもできます。

デフォルトでは、行末を越えた位置でマウスボタンをクリックすると、挿入位置は行の最後にある文字の後に置かれます。

拡張エディタで、行末を越えた位置に挿入位置を置くには、次の操作を行います。

1. **ツール** ⇨ **オプション** ⇨ **拡張エディタ** ⇨ **全般**を選択します。
2. **行端でカーソルを折り返さない**チェックボックスを選択します。
3. **OK** をクリックします。

### *テキストの選択と編集*

テキストを選択および編集するには、次に示すマウスおよびショートカットキーの操作を実行します。

***表 3.5*** *テキストの文字と行の選択*

| テキスト選択 | 実行手順 |
|---|---|
| 余白を使用してテキストから 1 行または複数行を選択する。 | 1. 選択するテキストの最初の行の余白で、マウスの左ボタンをクリックして押したままにします。<br>2. マウスの左ボタンを押したまま、選択する最後の行の余白までマウスポインタをドラッグします。<br>3. マウスの左ボタンを離します。 |

| テキスト選択 | 実行手順 |
| --- | --- |
| 1 つまたは複数の文字や、テキスト行全体を選択する。 | 1. 選択する最初の文字の前で、マウスの左ボタンをクリックして押したままにします。<br>2. マウスの左ボタンを押したまま、選択する最後の文字までマウスポインタをドラッグします。<br>3. マウスの左ボタンを離します。 |

次に示すショートカットキーを使用して、テキストを選択することもできます。

***表 3.6*** *ショートカットキーを使用したテキストの選択*

| 目的 | 実行手順 |
| --- | --- |
| 選択範囲を特定方向に拡大 | Shift キーを押しながら任意の方向の矢印キーを押します。 |
| 選択範囲を 1 文字ずつ拡大 | Shift キーを押しながら、左右の矢印キーを押します。 |
| テキストの選択を解除 | 任意の矢印キーを押します。 |
| 選択したテキストのコピー | Ctrl キーを押しながら C キーを押すか、次のメニューパスを選択します:**編集** ⇨ **コピー** |
| 選択したテキストの切り取り | Ctrl キーを押しながら X キーを押すか、次のメニューパスを選択します:**編集** ⇨ **切り取り** |
| クリップボードの内容のコピー | Ctrl キーを押しながら V キーを押すか、次のメニューパスを選択します:**編集** ⇨ **貼り付け** |
| 選択したテキストの移動 | 1. 選択したテキストの上にマウスポインタを置きます。<br>2. マウスの左ボタンをクリックして押したままにします。マウスポインタは垂直線で表示されます。<br>3. マウスの左ボタンを押したまま、選択したテキストをドラッグして、テキストを移動する場所にマウスポインタを置きます。<br>4. マウスの左ボタンを離します。 |
| コードまたはデータのカラムを反転表示するには | Alt キーを押しながら、マウスの左ボタンを押します。ドラッグして選択します。 |

選択されたテキストは、色が反転表示されます。

ショートカットキーの完全なリストについては、“拡張エディタでのショートカットキー” (631 ページ) の選択カテゴリを参照してください。

*注:* **編集**メニューからコマンドを使用するほかに、**拡張エディタ**ウィンドウでマウスの右ボタンをクリックすると、ポップアップメニューから編集コマンドを使用することもできます。

### *テキストのドラッグ*

テキストを移動またはコピーするには、次の操作を行います。

1. テキストを選択し、選択したテキストの上にポインタを置いて、マウスの左ボタンを押したままの状態にします。
2. テキストを移動するには、次の操作を行います。
   a. テキストを目的の位置へドラッグします。
   b. マウスの左ボタンを離します。
3. テキストをコピーするには、次の操作を行います。
   a. Ctrl キーを押します。
   b. テキストを目的の位置へドラッグします。
   c. マウスの左ボタンを離します。

ドラッグアンドドロップ編集を無効にするには、次の操作を行います。

1. **ツール** ⇨ **オプション** ⇨ **拡張エディタ** ⇨ **全般**を選択します。
2. **テキストのドラッグアンドドロップ編集を行う**チェックボックスを選択しないようにして **OK** をクリックします。

### *テキストの検索と置換*

テキストを検索するには、次の操作を行います。

1. 次のメニューパスを選択して、**検索**ダイアログボックスを開きます: **編集** ⇨ **検索**
2. 次の情報を入力します。

**検索する文字列**
検索するテキスト文字列を入力します。このフィールドの初期値は、検索に使用された最後のテキスト文字列です。

**検索範囲**
**検索範囲**ボックスをクリックして、コードのみを検索するか、コメントのみを検索するか、コードおよびコメントの両方を検索するかを指定します。

**検索する方向**
**上へ**オプションまたは**下へ**オプションを選択します。**上へ**は、カーソル位置からファイルの先頭方向に向けて検索することを指定します。**下へ**は、カーソル位置からファイルの末尾方向に向けて検索することを指定します。

**完全に一致するワードだけを検索する**
検索条件とテキストが部分的に一致するワードではなく、完全に一致するワードを検索するときにチェックボックスを選択します。

**大文字と小文字を区別する**
大文字と小文字を区別して検索する必要があるときに、チェックボックスを選択します。

**正規表現を使用する**
テキスト文字列を正規表現として指定する場合に、このチェックボックスを選択します。正規表現では、ワイルドカードとして特殊文字を使用することにより、文字列またはサブストリングの検索を行います。正規表現で使用できる特殊文字を選択するには、**検索する文字列**フィールドの右側にある矢印をクリックします。

3. 次のボタンをクリックします:**次を検索**

テキストを検索および置換するには、次の操作を行います。

1. テキストのサブセットのみを検索するには、そのテキストを選択します。
2. 次のメニューパスを選択して、**置換**ダイアログボックスを開きます:**編集** ⇨ **置換**
3. 次の情報を入力します。

**検索する文字列**
検索および置換するテキスト文字列を入力します。このフィールドの初期値は、検索に使用された最後のテキスト文字列です。

**置換後の文字列**
置き換えた後の文字列を入力します。

**検索範囲**
**検索範囲**ボックスをクリックして、コードのみを検索するか、コメントのみを検索するか、コードおよびコメントの両方を検索するかを指定します。

**検索する方向**
**上へ**オプションまたは**下へ**オプションを選択します。**上へ**は、カーソル位置からファイルの先頭方向に向けて検索することを指定します。**下へ**は、カーソル位置からファイルの末尾方向に向けて検索することを指定します。

**完全に一致するワードだけを検索する**
検索条件とテキストが部分的に一致するワードではなく、完全に一致するワードを検索するときにこのチェックボックスを選択します。

**大文字と小文字を区別する**
大文字と小文字を区別して検索する必要があるときに、このチェックボックスを選択します。

**正規表現検索**
テキスト文字列に正規表現が含まれることを指定する場合に、このチェックボックスを選択します。正規表現では、ワイルドカードとして特殊文字を使用することにより、文字列またはサブストリングの検索を行います。正規表現で使用できる特殊文字を選択するには、**検索する文字列**フィールドの右側にある右矢印をクリックします。

4. **次を検索**をクリックします。
5. テキストが検索された場合、次の項目の 1 つをクリックします。

- **置換して次に**をクリックすると、検索された 1 つのテキストが置換文字列と置き換えられます。
- **すべて置換**をクリックすると、検索されたファイル内のすべてのテキストが、置換文字列と置き換えられます。
- **選択内で置換**をクリックすると、選択されたテキストの範囲内にあるすべての検索されたテキストが、置換文字列と置き換えられます。

### *コーディングエラーのチェック*

拡張エディタでコーディングエラーを検出するには、次の操作を行います。

- プログラム要素、引用符に囲まれた文字列およびコメントに対してカラーコードを設定します。
- Ctrl キーを押しながら]キーを押すと、右角かっこ(])または右かっこ())を検索します。
- Alt キーを押しながら[キーを押すと、一致する DO-END ペア単位を検索します。

コーディングエラー検出のヒントについては、次の表を参照してください。

***表 3.7*** *コーディングエラー検索のヒント*

| コードエラーの種類 | 実行手順 |
|---|---|
| 未定義またはスペルミスのあるキーワード | **拡張エディタオプション**ダイアログボックスのデザインタブで、**定義済みキーワード**、**ユーザー定義のキーワード**および**未定義のキーワード**のファイル要素を、それぞれ固有のカラーの組み合わせに設定します。<br>キーワードが認識されると、キーワードは定義された色に変更されます。未定義のキーワードに選択されたカラーを検索することによって、未定義のキーワードを簡単に探し当てることができます。 |
| 不当な引用文字列 | プログラムの行の中から、同じ色のものを検索します。<br>引用符に続くテキストは、対になっている引用符によって文字列が閉じられるまで同じ色で表示されます。 |
| 不当なコメント | プログラムの行の中から、同じ色のものを検索します。<br>コメント開始記号( /* )に続くテキストは、コメント終了記号( */)でコメントが閉じられるまで同じ色で表示されます。 |
| 対応する DO-END ペア単位 | DO-END ブロックにカーソルを置き、Alt キーを押しながら[キーを押します。<br>カーソルは、最初に DO キーワードに移動します。キーワードのいずれかが見つからない場合、カーソルは置かれた場所に表示されます。<br>両方のキーワードが見つかった場合、Alt キーを押しながら[キーを押すと、カーソルを DO-END キーワード間で移動できます。 |
| 対応するかっこまたは角かっこ | カーソルをかっこまたは角かっこの片側に置きます。Ctrl キーを押しながら]キーを押します。<br>カーソルが対応するかっこまたは角かっこに移動します。対応するかっこが見つからなかった場合、カーソルは移動しません。 |
| セミコロン( ; )の欠落 | 標準のテキストで表示されるキーワードを検索します。 |

カラーコーディングできる構成要素のリストを参照するには、**拡張エディタオプション**ダイアログボックスを開きます。**デザイン**タブを選択します。構成要素は**ファイルの要素**

ボックスに表示されます。プログラム構成要素のカラー定義の詳細は、“デザインオプションの設定” (104 ページ) を参照してください。

### 自動インデントおよびタブの使用

Enter キーを押すと、現在の行がインデントされているスペースだけ、次の行が自動的にインデントされます。自動インデントを使用しない場合は、次の操作を行います。

1. **ツール** ⇨ **オプション** ⇨ **拡張エディタ** ⇨ **全般**を選択します。
2. **インデント**ボックスで**なし**を選択します。

自動インデントのほかに、Tab キーを使用してインデントすることもできます。Tab キーを押すと、挿入位置および挿入位置の右側にあるすべてのテキストが、**拡張エディタオプション**ダイアログボックスの**タブ幅**で指定されたスペースだけ移動します。

Tab キーを押すかファイルを開くと、タブ文字はスペースに置換されます。Tab キーを押したときにタブ文字ではなくスペースを挿入するには、**タブを空白に変換する**チェックボックスを選択します。ファイルを開いたときにタブ文字をスペースに置き換えるには、**ファイルを開くときにタブを空白で置換**チェックボックスを選択します。

*注:* タブ幅を変更すると、すべての拡張エディタウィンドウで、タブ設定が新しい値に変更されます。

### 行のブックマーキング

行にブックマークを付けると、ラインマーカーが作成されてその行に簡単にアクセスできるようになります。余白にある縦長の長方形は、行にブックマークが付いていることを示します。次の表に、ブックマーキングに使用できるショートカットキーを示します。

*注:* ブックマークはファイルとともに保存されるのではありません。特定の拡張エディタが開いている間は使用可能です。

***表 3.8*** *行のブックマーキングのショートカットキー*

| アクション | ショートカットキー |
|---|---|
| 行にブックマークを付ける | Crtl+F2(マークの付いていない行) |
| 行からブックマークを削除する | Ctrl+F2(マークの付いている行) |
| 次のブックマークに移動する | F2 |
| 前のブックマークに移動する | Shift+F2 |

### 省略形の使用

省略形を使用することにより、Tab キーまたは Enter キーを押したときにより長い文字列に展開される文字列を定義できます。たとえば、C:\Users\user-ID\Documents\My SAS Files\9.4';に展開される、my-v9-sas-files という省略形を定義できます。省略形とは、実際には 1 行以上のテキスト行を挿入するマクロです。

省略形を作成するには、次の操作を行います。

1. Ctrl キーと Shift キーと A キーを同時に押すか、次のメニューパスを選択します:**ツール** ⇨ **省略形の追加**
2. **省略キー**フィールドに、省略形の名前を入力します。

3. **省略するテキスト**フィールドに、省略形の展開後のテキストを入力します。
4. **OK** をクリックします。

省略形を使用するには、その省略形を入力します。省略形が認識されると、ツールチップに、展開されたテキストが表示されます。Tab キーまたは Enter キーを押して、省略形を確定します。

省略形を変更するには、次の操作を行います。

1. Ctrl キーと Shift キーと M キーを同時に押すか、**ツール** ⇨ **キーボードマクロ** ⇨ **マクロ**を選択します。
2. 現在のマクロのリストから、省略形を選択します。
3. **編集**をクリックします。
4. **キーボードマクロの内容**フィールドで文字列を選択します。
5. **変更**をクリックします。
6. **文字列の入力**ダイアログボックスに修正候補を入力し、**OK** をクリックします。
7. **キーボードマクロの編集**ダイアログボックスで **OK** をクリックします。
8. **マクロ**ダイアログボックスで**閉じる**をクリックします。

省略形を削除するには、次の操作を行います。

1. Ctrl キーと Shift キーと M キーを同時に押すか、**ツール** ⇨ **キーボードマクロ** ⇨ **マクロ**を選択します。
2. **現在のマクロ**のリストから、省略形を選択します。
3. **削除**をクリックします。
4. **閉じる**をクリックします。

### *ワードチップの使用*

ワードチップは、追加した省略形の理解に役立ちます。

ワードチップを使用するには、まず省略形を追加する必要があります。

1. Ctrl キーと Shift キーと A キーを同時に押します。
2. 省略形および対応するテキストを追加します。
3. 確定するには **OK** をクリックし、破棄するには**キャンセル**をクリックします。

入力時に自動的にワードチップが表示されます。

ワードチップを確定するには Enter キーを押します。ワードチップを無視するには、Esc キーを押すか、入力を続行します。

ワードチップを呼び戻すには、次の操作を行います。

- ポインタを短縮形の直後に合わせます。
- Alt キーを押しながら F1 キーを押します。

### *プログラムのサブミット*

プログラム全体またはプログラム内の指定された行番号を、最初の行からサブミットできます。

*注:* 行の最大長は 6KB です。

***表 3.9*** *プログラム全体のサブミット*

| プログラム全体のサブミット | 実行手順 |
|---|---|
| エディタでプログラムを開く場合 | **開く**ダイアログボックスで**サブミット**チェックボックスを選択します。 |
| 拡張エディタからプログラムを開く場合 | 次のうちいずれかを実行します。<br>• **サブミット**ツールバーボタンをクリックします。<br>• F3 キーまたは F8 キーを押します。<br>• 次のメニューパスを選択します:**実行** ⇨ **サブミット**<br>• コマンドバーに **submit** と入力します。 |

プログラムの最初の行または指定された行番号をサブミットするには、SUBTOP コマンドを使用します。

***表 3.10*** *プログラムの一部のサブミット*

| プログラムの一部のサブミット | 実行手順 |
|---|---|
| プログラムの最初の行だけサブミットする。 | 次のうちいずれかを実行します。<br>• コマンドバーに **subtop** と入力します。<br>• 次のメニューパスを選択します:**実行** ⇨ **先頭行をサブミント** |
| 最初の行から数えて、任意の行数をサブミットする。 | • コマンドバーに **subtop** *n* と入力します。**n** は、サブミットする行数です。<br>• **実行** ⇨ **N 行をサブミット**を選択し、サブミットする行数を入力します。 |

詳細については、“SUBTOP コマンド:Windows” (352 ページ) を参照してください。

**拡張エディタオプション**ダイアログボックスの**サブミットしたテキストをクリアする**を設定することにより、プログラムをサブミットした後に**拡張エディタ**ウィンドウの内容をクリアするかどうかを指定できます。詳細については、“拡張エディタオプションの設定” (102 ページ) を参照してください。

DATALINES または CARDS ステートメント内のデータ行が 256 文字より長い場合、データは 1 つの情報として読み込まれます。プログラムエディタを使用して長い行のデータをサブミットするときと同じように、FILENAME ステートメントに LRECL オプションを指定する必要はありません。

### *サブミットするプログラムまたはカタログエントリのファイル名およびフルパスの設定*

拡張エディタからコードまたはカタログエントリをサブミットするとき、ファイル名またはカタログエントリ名、およびそれらの各フォルダは、次の環境変数に設定されます。

SAS_EXECFILEPATH
　サブミットするプログラムまたはカタログエントリのフルパスを含みます。フルパスには、フォルダおよびファイル名が含まれます。

SAS_EXECFILENAME
　サブミットするプログラム名またはカタログエントリ名のみが含まれます。

SAS プログラムで使用する場合には、ファイル名およびフルパスを抽出できます。

次の DATA ステップを実行してデータを並べ替えると、PRINT プロシジャにはタイトルのファイル名およびプロシジャ出力のフットノートのフルパスが取り込まれます。結果は画面に表示されます。

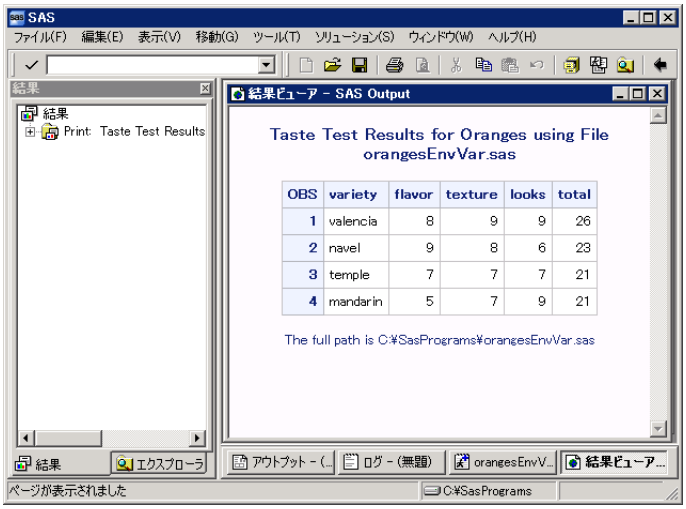
```
data oranges;
   input variety $ flavor texture looks;
   total=flavor+texture+looks;
cards;
navel 9 8 6
temple 7 7 7
valencias 8 9 9
mandarins 5 7 9
;

proc sort data=oranges;
   by descending total;
run;

proc print data=oranges;
title 'Taste Test Results for Oranges using File ' %sysget(SAS_EXECFILENAME);
footnote 'The full path is ' %sysget(SAS_EXECFILEPATH);
run;
```

出力結果には、タイトルのファイル名およびフットノートのフルパスが表示されます。

***図 3.2*** *DATA ステップ出力にファイル名を設定するための環境変数の使用*




これらの環境変数は、Windows 環境で拡張エディタを使用してコードをサブミットする場合のみに設定されます。これらの環境変数は、SCL コードをサブミットするときや、バッチセッションでコードをサブミットするときには設定されません。

ただし、SAS がバッチモードで実行されているときは、
**`%sysfunc(getoption(SYSIN))`**をサブミットすることによりフルパス(ファイル名を

含む)を設定できます。次のマクロにより、拡張エディタを使用してバッチセッションおよび対話型セッションの両方のフルパスを設定できます。

```
%let execpath=" ";
%macro setexecpath;
   %let execpath=%sysfunc(GetOption(SYSIN));
   %if %length(&execpath)=0
      %then %let execpath=%sysget(SAS_EXECFILEPATH);
%mend setexecpath;

%setexecpath;
%put &execpath;
```

次に示す%PUT マクロステートメントを使用して、SAS ログにファイル名およびフルパスを表示することもできます。

```
%put Submitted file path is %sysget(SAS_EXECFILEPATH).;
%put Submitted file name is %sysget(SAS_EXECFILENAME).;
```

***注意:***

**対話型プロシジャがアクティブであるのに次のプログラムがサブミットされると、これらの環境変数の値が上書きされることがあります。** 環境変数は、最後にサブミットされたプログラムに設定されます。対話型プロシジャがアクティブであるのにプログラムが対話型プロシジャを開始して、次のプログラムがサブミットされる場合、環境変数は最後にサブミットされたプログラムのファイル名およびフルパスに設定されます。対話型プロシジャをサブミットしたプログラムのファイル名およびフルパスは、その後使用することはできません。

## *ショートカットキーの使用*

拡張エディタでは、多数のショートカットキーを使用できます。“拡張エディタでのショートカットキー” (631 ページ)には、コマンドの完全なリストとデフォルトのショートカットキーが記載されています。次の表に、便利なショートカットキーをいくつか示します。

***表 3.11*** *実用的なショートカットキー*

| ショートカットキー | タスク |
|---|---|
| SAS プロシジャのヘルプを利用する | マウスのボタンを押してプロシジャ名にカーソルを置き、F1 キーを押します。 |
| 新しい省略形を追加する | CTRL+Shift+A |
| 現在の行の展開と折りたたみを切り替える | Alt+Num * |
| すべてのコードセクションを折りたたむ | Alt+Ctrl+テンキーの- |
| すべてのコードセクションを展開する | Alt+Ctrl+テンキーの+ |
| 現在の行のマーカーを切り替える | Ctrl+F2 |
| 次のマークされた行に移動する | F2 |
| 前のマークされた行に移動する | Shift+F2 |

| ショートカットキー | タスク |
|---|---|
| 行に移動する | Ctrl+G |
| ファイルの先頭に移動する | Ctrl+Page Up |
| ファイルの末尾に移動する | Ctrl+Page Down |
| 選択されたテキストを大文字に変換する | Ctrl+Shift+U |
| 選択されたテキストを小文字に変換する | Ctrl+Shift+L |

ショートカットキーの定義については、“拡張エディタでショートカットキーをカスタマイズする” (106 ページ) を参照してください。

### *キーボードマクロの使用*

キーボードマクロとは、タスクを自動的に実行するために 1 つのコマンドとしてまとめられた一連の拡張エディタコマンドおよび命令です。時間のかかる繰り返しの操作を手動で行うかわりに、1つのマクロを作成して実行できます。**ツール**メニューから、またはショートカットキーを使用して、マクロを実行します。マクロを使用するためのショートカットキーの定義については、“拡張エディタでショートカットキーをカスタマイズする” (106 ページ) を参照してください。

マクロを作成するには、**拡張エディタ**ウィンドウでマクロを記録します。

1. Alt キーと Shift キーと R キーを同時に押すか、次のメニューパスを選択して、記録を開始します:**ツール** ⇨ **キーボードマクロ** ⇨ **新しいマクロの記録**
2. 一連の操作を実行して、タスクを完了します。
3. Alt キーと Shift キーと R キーを同時に押すか、次のメニューパスを選択して、記録を停止します。**ツール** ⇨ **キーボードマクロ** ⇨ **レコードの停止**
4. 必要に応じて、マクロを実行するショートカットキーを定義します。詳細については、“拡張エディタでショートカットキーをカスタマイズする” (106 ページ) を参照してください。

マクロを作成する別の方法として、**キーボードマクロの作成**ダイアログボックスを使用してコマンドを追加できます。マクロを追加するには、次の操作を行います。

1. Ctrl キーと Shift キーと M キーを同時に押すか、次のメニューパスを選択して、**キーボードマクロ**ダイアログボックスを開きます。**ツール** ⇨ **キーボードマクロ** ⇨ **マクロ**
2. **作成**をクリックして、**キーボードマクロの作成**ダイアログボックスを開きます。
3. **キーボードマクロの名前**フィールドに、マクロ名を入力します。
4. **キーボードマクロの説明**フィールドに、マクロの説明を入力します。
5. **キーボードマクロの内容**リストボックスには、コマンドが、実行される順に一覧表示されます。

   コマンドを追加するには、**コマンド**ボックスからコマンドを選択し、選択したコマンドの内容挿入ボタン(二重矢印)をクリックします。**キーボードマクロの内容**リストボックスにすべてのコマンドが表示されるまで、この手順を繰り返します。

   **キーボードマクロの内容**リストボックスでコマンドを選択し、上矢印マークまたは下矢印マークをクリックすると、コマンドの順序を変更できます。

コマンドを削除するには、コマンドを選択して**削除**をクリックします。

6. すべてのコマンドが正しい順序でリストボックスに表示されたら、**OK** をクリックします。
7. **キーボードマクロ**ダイアログボックスで**閉じる**をクリックします。

マクロを実行するには、ショートカットキーを使用するか、次の操作を行います。

1. 次のメニューパスを選択します。**ツール** ⇨ **キーボードマクロ** ⇨ **マクロの実行**
2. **マクロ**の**実行**ダイアログボックスでマクロを選択し、**実行**をクリックします。

マクロを編集するには、次の操作を行います。

1. Ctrl キーと Shift キーと M キーを同時に押すか、次のメニューパスを選択して、**キーボードマクロ**ダイアログボックスを開きます。**ツール** ⇨ **キーボードマクロ** ⇨ **マクロ**
2. マクロを選択し、**編集**をクリックして**キーボードマクロの編集**ダイアログボックスを開きます。
3. コマンドを追加するには、**コマンド**リストボックスからコマンドを選択し、選択したコマンドの内容挿入ボタンをクリックします。

   コマンドを変更するには、最初に**キーボードマクロの内容**リストボックスからコマンドを選択して、コマンドを変更できることを確認します。コマンドを変更できる場合は、**変更**ボタンがアクティブになります。**変更**をクリックすると、コマンドを変更するためのダイアログボックスが表示されます。

   コマンドを削除するには、**キーボードマクロの内容**リストボックスでコマンドを選択し、**削除**をクリックします。

   コマンドの順序を変更するには、コマンドを選択して上へボタンまたは下へボタンをクリックします。
4. コマンドが正しい順序になったら、**OK** をクリックします。

マクロを削除するには、次の操作を行います。

1. Ctrl キーと Shift キーと M キーを同時に押すか、**ツール** ⇨ **キーボードマクロ** ⇨ **マクロ**を選択します。
2. マクロを選択して**削除**をクリックします。
3. **キーボードマクロの削除**ダイアログボックスで**はい**をクリックします。
4. **閉じる**をクリックします。

この例では RSUBMIT ステートメント、ENDRSUBMIT ステートメント、これらのステートメント間のブランク行を作成するステップをリストし、**拡張エディタオプション**ダイアログボックスの**タブ幅**フィールドに指定されたタブ幅を設定します。

1. 次のメニューパスを選択します:**表示** ⇨ **拡張エディタ**
2. **ツール** ⇨ **キーボードマクロ** ⇨ **新しいマクロの記録**を選択し、メッセージボックスで **OK** をクリックします。
3. **拡張エディタ**ウィンドウで、次の操作を行います。
   a. Enter キーを押します。
   b. `rsubmit;`と入力します。

c. Enter キーを押します。

d. Enter キーを押します。

e. **endrsubmit;**と入力します。

f. 上矢印キーを押します。

g. Tab キーを押します。

h. 次のメニューパスを選択します:**ツール** ⇨ **キーボードマクロ** ⇨ **記録終了**

結果のマクロには、次のコマンドが含まれます。

```
Insert carriage return
Insert character ['r']
Insert character ['s']
Insert character ['u']
Insert character ['b']
Insert character ['m']
Insert character ['i']
Insert character ['t']
Insert character [';']
Insert carriage return
Insert carriage return
Insert character ['e']
Insert character ['n']
Insert character ['d']
Insert character ['r']
Insert character ['s']
Insert character ['u']
Insert character ['b']
Insert character ['m']
Insert character ['i']
Insert character ['t']
Insert character [';']
Move cursor up
Insert character ['|']
```

キーボードマクロは、複数のユーザーで共有できます。共有するには、**キーボードマクロ**ダイアログボックスを使用してフォルダへのインポートまたはフォルダからのエクスポートを行います。ダイアログボックスを開くには、次のメニューパスを選択します。**ツール** ⇨ **キーボードマクロ** ⇨ **マクロ**

キーボードマクロをインポートするには、**インポート**をクリックし、**ファイルの場所**リストボックスからフォルダを、**ファイル名**ボックスからファイル名を選択します。次に **OK** をクリックします。

キーボードマクロの場合のみ、**キーの割り当て**をクリックし、マクロごとに新しいショートカットキーを選択します。省略形にはショートカットキーを割り当てないでください。

キーボードマクロをエクスポートするには、**エクスポート**をクリックして、**保存する場所**フィールドからフォルダを選択し、**ファイル名**ボックスにファイル名を入力して、**OK** をクリックします。

*注:* ショートカットキーの上書きを避けるには、キーボードマクロのインポート後にショートカットキーを再割り当てする必要があります。省略形にはキー割り当てはありません。

### コードセクションの折りたたみの使用

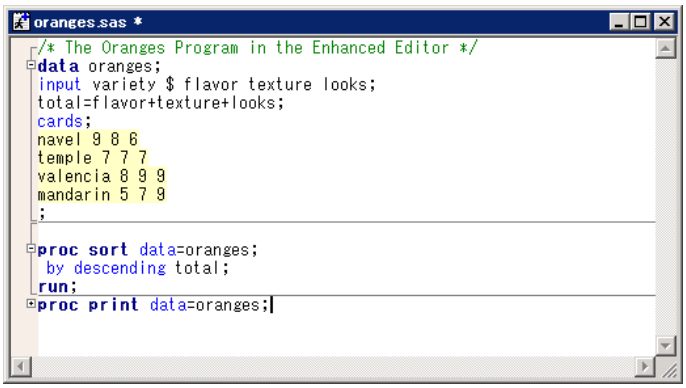
コードセクションの折りたたみでは、コードの1つ以上のセクションを展開または折りたたみできます。セクションは、ステップキーワード、コメント、またはセクションワードやコメント上のスペースから始まります。セクションは、次のステップキーワード、コメント、または次のセクションワードやコメント上のスペースで終了します。ステップキーワードには、DATA ステートメント、PROC ステートメント、%マクロステートメントがあります。シグネチャ行は、ステップキーワードが表示される行です。

展開コードセクションは、シグネチャ行の横の余白にマイナス記号で表示されます。セクションを折りたたむには、マイナス記号をクリックします。

折りたたまれたコードセクションは、余白にプラス記号で表示され、シグネチャ行のみが表示されます。セクションを展開するには、プラス記号をクリックします。

*注:* 展開機能と折りたたみ機能はプログラムでは保存されません。この機能は、現在のエディタウィンドウが開いている間のみ有効です。

***図 3.3*** *折りたたみ可能なコードセグメントが有効なときの拡張エディタ*




余白のかっこおよびエディタウィンドウを横切るセクションラインにより、セクションの始まりと終わりにマークを付けることができます。かっこまたはセクションラインは、**拡張エディタオプション**ダイアログボックスを使用して非表示にできます。

折りたたみ可能なコードセクション、またはかっことセクションラインを無効にするには、**ツール** ⇨ **オプション** ⇨ **拡張エディタ** ⇨ **全般**を選択し、目的とする設定項目を選択します。

- 折りたたみ可能なコードセクションを無効にするには、**コードセクションの表示**チェックボックスを選択しないようにします。
- エディタ内でセクションラインを無効にするには、**ラインを表示する**チェックボックスを選択しないようにします。
- 余白にあるセクションのかっこを非表示にするには、**かっこを表示する**チェックボックスを選択しないようにします。

次の規則は、折りたたまれたセグメントを選択および編集しているときに適用されます。

- 折りたたまれたセグメントを含む余白から行を選択すると、折りたたまれたセグメント内にあるすべてのテキストが選択されます。
- テキスト上で、マウスをドラッグして、またはキーボードを使用してテキストを 1 行選択すると、そのテキスト行のみが選択されます。
- 折りたたまれたセクションから、折りたたまれたシグネチャ行までのテキストを選択すると、選択範囲の先頭から末尾までがコピーされます。選択範囲には、シグネチャ行の上の、非表示の行から選択範囲の末尾までのすべての行が含まれます。

- シグネチャ行から別のセクションまでのテキストを選択すると、シグネチャ行の下の非表示のテキストを含めて、選択範囲の先頭から末尾までがマークされます。
- シグネチャ行で任意のキーを押すと、セクションが展開されます。
- シグネチャ行またはセクションへの貼り付けによって、セクションが展開されます。
- セクションに影響を与えるコメントや引用符などをセクションに入力すると、セクションが展開されます。
- シグネチャ行の先頭で Enter キーを押すと、セクションの先頭にコードが追加されます。
- シグネチャ行の末尾で Enter キーを押すと、セクションの末尾にコードが追加されます。
- **元に戻す**を選択して、コマンドの折りたたみおよび展開を取り消すことはできません。
- テキストを検索したときに、折りたたまれたセグメント内に検索対象のテキストが見つかった場合は、セグメントが展開されます。
- 折りたたまれたセグメント内のテキストを置換する場合、セグメントは展開されます。
- 折りたたまれたセグメント内のテキストを置換する場合、**すべて置換**を選択しても、折りたたまれたセグメントは展開されません。

## *独自のキーワードの作成*

拡張エディタで扱える多くの SAS プログラムのファイル要素に加えて、SAS プロシジャのステートメント、変数、ユーザー定義の形式などプログラミング要素に対応するユーザー定義のキーワードを作成できます。SAS 言語で定義されている同一のキーワードが、ユーザー定義のキーワードとして用いられた場合は上書きされます。

キーワードに名前を付ける場合は、次の規則を使用します。

1. 最初の文字は、アルファベット(A, B, C, ... Z)またはアンダースコア( _ )である必要があります。
2. 以後の文字には、アルファベット、数字(0, 1, ... 9)またはアンダースコアを使用できます。
3. 大文字または小文字を使用できます。キーワードでは大文字と小文字は区別されません。
4. キーワード名にブランクは使用できません。

ユーザー定義のキーワードを作成および書式設定するには、次に示すように**拡張エディタオプション**ダイアログボックスを使用します。

1. **拡張エディタ**ウィンドウをアクティブにします。
2. 次のメニューパスを選択します:**ツール** ⇨ **オプション** ⇨ **拡張エディタ**
3. **全般**タブの**ユーザー定義のキーワード**をクリックします。
4. **追加**をクリックします。
5. **NewKeyword** を独自のキーワードに置き換えます。
6. **OK** をクリックします。
7. **デザイン**タブを選択します。

8. **ファイルの要素**ボックスで、**ユーザー定義のキーワード**を選択します。
9. フォント、前景色、背景色を選択します。
10. **OK** をクリックします。

**ユーザー定義のキーワード**ダイアログボックスでキーワード名を変更するには、次の操作を行います。

1. キーワードを選択します。
2. **名前の変更**をクリックします。
3. キーワード名を変更します。
4. **OK** をクリックします。

**ユーザー定義のキーワード**ダイアログボックスでキーワードを削除するには、次の操作を行います。

1. キーワードを選択します。
2. **削除**をクリックします。
3. **OK** をクリックします。

## *ファイルタイプへのファイル拡張子の割り当て*

次の表に、拡張エディタによって認識されるファイルの種類のデフォルトのファイル拡張子を示します。

***表 3.12*** *拡張エディタによって認識されるデフォルトのファイル拡張子*

| ファイルの種類 | デフォルトのファイル拡張子 |
|---|---|
| SAS プログラムファイル | .sas |
| SCL プログラムファイル | .scl |
| HTML ドキュメント | .htm、.html、.xml |

他のファイル拡張子を SAS プログラム、SCL プログラム、HTML ドキュメント、XML ドキュメントに割り当てるには、次の操作を行います。

1. 次のメニューパスを選択して、**拡張エディタオプション**ダイアログボックスを開きます。**ツール** ⇨ **オプション** ⇨ **拡張エディタ**
2. **全般**タブをクリックします。
3. **ファイルの種類**ボックスからファイルの種類を選択します。
4. **ファイルの拡張子**をクリックしてから**追加**をクリックします。
5. ファイルの拡張子を入力し、Enter キーを押します。

ファイルの拡張子名を変更するには、次の操作を行います。

1. ファイルの拡張子を選択します。

2. **名前の変更**をクリックします。
3. 新しいファイルの拡張子を入力して Enter キーを押します。

ファイルの拡張子を削除するには、拡張子を選択して**削除**をクリックします。

デフォルトのファイル拡張子に戻すには、**標準設定**をクリックします。

## *拡張エディタオプションの設定*

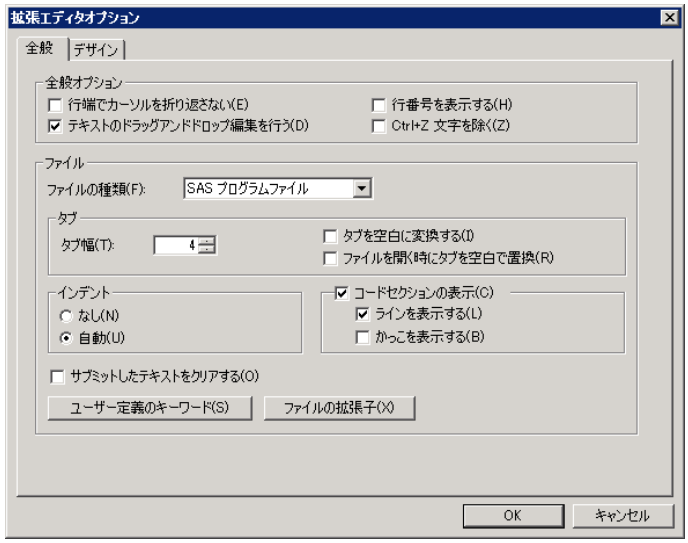
### *拡張エディタオプションウィンドウを開く*

メニューから**拡張エディタオプション**ウィンドウを開くには、**拡張エディタ**ウィンドウがアクティブウィンドウになっていることを確認し、次のメニューパスを選択します:**ツール** ⇨ **オプション** ⇨ **拡張エディタ**

設定を変更するには、ダイアログボックスの上部にあるタブをクリックして必要なオプションを選択します。変更を終了したら **OK** をクリックします。

*注:* 拡張エディタオプションは Microsoft Windows レジストリに格納されます。オプションはマシンや SAS プロファイルカタログには格納されません。変更を行う場合は、オペレーティングシステムのウィンドウ編集権限が必要です。

***図 3.4*** *エディタオプションダイアログボックス*




### *全般タブ*

**全般**タブで、拡張エディタの動作を制御するための次のオプションを指定できます。

**行端でカーソルを折り返さない**
行の最後にあるテキスト文字の直後でマウスポインタをクリックしたときに、挿入位置となる場所を指定します。このチェックボックスが選択されている場合は、マウスポインタをクリックした場所が挿入位置になります。このチェックボックスが選択されていない場合は、行の最後にあるテキスト文字の直後が挿入位置になります。

**テキストのドラッグアンドドロップ編集を行う**
選択したテキストを、ドラッグアンドドロップ編集を使用して移動できるかどうかを指定します。このチェックボックスが選択されている場合、選択したテキストはドラッグアンドドロップを使用して移動できます。チェックボックスが選択されていない場合、選択したテキストはドラッグアンドドロップを使用して移動できません。

**行番号を表示する**
行番号を余白に表示するかどうかを指定します。行番号が表示されるとき、現在の行番号の色は赤になります。

**Ctrl+Z 文字を除く**
DOS エディタで作成されたファイルに含まれることがある Ctrl+Z ファイル終端文字を削除するように指定します。

**ファイルの種類**
タブ、インデント、およびコードセクションの表示を適用するファイルの種類を指定します。ファイルの種類には、HTML ドキュメント、SAS プログラム、SCL プログラム、テキストドキュメントがあります。

**タブ幅**
インデントするスペースの数を指定します。

**タブを空白に変換する**
Tab キーを押したときに、空白文字またはタブ文字を挿入するかどうかを指定します。このチェックボックスを選択すると、通常の空白文字が使用されます。チェックボックスを選択しない場合、タブ文字が使用されます。

**ファイルを開くときにタブを空白で置換**
ファイルを開いたとき、ファイル中のすべてのタブ文字を空白文字に変換するかどうかを指定します。

**インデント**
使用するインデントの種類を選択します。**なし**を選択した場合は、インデントされません。**自動**を選択した場合、次の行は前の行のインデントと同じスペースで自動的にインデントされます。

**コードセクションの表示**
コードセクションを展開および縮小できるかどうかを指定します。このチェックボックスを選択する場合は、折りたたみ可能なコードセクションを折りたたむかまたは展開できます。このチェックボックスを選択しない場合は、すべてのコードがエディタウィンドウに表示されます。**コードセクションの表示**設定が選択されている場合、次の設定が有効になります。

- **ラインを表示する**を選択すると、テキストの各セクションの後に線が表示されます。
- **かっこを表示する**を選択すると、各セクションを囲むかっこが余白に表示されます。

**サブミットしたテキストをクリアする**
プログラムを処理するためにサブミットした後に、**拡張エディタ**ウィンドウの内容をクリアするかどうかを指定します。このチェックボックスが選択されている場合、プログラムをサブミットした後に**拡張エディタ**ウィンドウの内容がクリアされます。このチェックボックスが選択されない場合、プログラムはエディタウィンドウに表示されたままになります。この設定が選択されている場合、F4 キーを使用すると、最後にサブミットしたプログラムをもう一度呼び出せます。

**ユーザー定義のキーワード**
ユーザー定義のキーワードを作成するときに使用する、**ユーザー定義のキーワード**ダイアログボックスが開きます。

**ファイルの拡張子**
**SAS 拡張子**ダイアログボックスを開きます。**SAS 拡張子**ダイアログボックスを使用して、拡張エディタにより認識されるファイル拡張子を定義します。

### *デザインオプション*

次のデザインオプションを使用して、ファイルの要素の前景色、背景色およびフォントスタイルを指定できます。また、配色を作成および保存できます。これらのデザインオプションの使用の詳細については、“デザインオプションの設定” (104 ページ) および “配色の使用” (105 ページ) を参照してください。

**ファイルの種類**
要素の色分けをするファイルの種類を指定します。SAS プログラム、SCL プログラム、HTML および XML ドキュメント、テキストドキュメントのファイル要素について、色分けできます。XML ドキュメントを色分けするには、**HTML ドキュメント**を選択します。**エディタのオプション**ダイアログボックスの起動時に編集しているファイルの種類が、デフォルトに設定されます。

**配色**
指定されたファイルの種類について、保存されたデザインの設定オプションを示す名前です。

**名前**
配色のフォント名を指定します。

**サイズ**
配色のフォントサイズを指定します。

**スクリプト**
指定したフォントに利用できる文字セットを表示します。デフォルトのスクリプトにより使用される文字セットは、Windows の地域別オプションにより決定されています。

**ファイルの要素**
色分けできる、指定されたファイルの種類の要素を一覧表示します。

**前景色**
選択されたファイルの要素に適用される、テキストの色を指定します。

**背景色**
選択されたファイルの要素に適用される、背景色を指定します。

**スタイル**
**標準**、**太字**、**斜体**、**太字 斜体**フォントをファイルの要素に適用するかどうかを指定します。

**下線**
ファイルの要素に下線を付けるかどうかを指定します。

**プレビュー**
選択されたファイルの要素の色とフォントのプレビューを表示します。

### *デザインオプションの設定*

デザインタブの各オプションを設定するとき、**ファイルの種類**リストボックスに指定したファイルの種類の要素に対して設定します。各設定を選択すると、選択した書式が**プレビュー**ボックスに表示されます。指定した書式オプションは、そのファイルの種類の開かれたすべての拡張エディタウィンドウに適用されます。SAS システムを起動したときに拡張エディタファイルに適用される書式オプションは、最後に SAS セッションを終了したときに有効だった書式オプションです。

デザインオプションを指定するには、次の操作を行います。

1. 次のメニューパスを選択して、**エディタのオプション**ウィンドウを開きます:**ツール** ⇨ **オプション** ⇨ **拡張エディタ** ⇨ **デザイン**
2. **ファイルの種類**ボックスからファイルの種類を選択します。

3. 保存されている書式の配色を**配色**ボックスから選択することもできます。配色の使用の詳細については、“配色の使用” (105 ページ) を参照してください。
4. **名前**ボックスから、フォントを選択します。
5. **サイズ**ボックスから、フォントサイズを選択します。
6. **スクリプト**ボックスから、コンピュータにセットアップされている言語に適したスクリプトを選択します。デフォルトのスクリプトは、Windows の地域別オプションにより決定されています。
7. 各ファイル要素について、次の操作を行います。
   a. ファイルの要素を選択します。
   b. **前景色**ボックスをクリックして、ファイルの要素に使用する色を選択します。カスタムカラーを作成するには、**カスタム**を選択し、**色の設定**ダイアログボックスから色を作成します。
   c. **背景色**ボックスをクリックして、ファイルの要素の背景色を選択します。カスタムカラーを作成するには、**カスタム**を選択し、**色の設定**ダイアログボックスから色を作成します。

      *注:* **標準テキスト**の背景色を変更すると、**拡張エディタ**ウィンドウが指定された色に変わります。

   d. **スタイル**ボックスから、**標準**、**太字**、**斜体**、**太字 斜体**のどれかを選択します。
   e. 要素に下線を付けるには、**下線**チェックボックスを選択します。
8. **プレビュー**ボックスで、選択した書式を確認します。プレビュー内のファイルの要素をクリックすると、その要素に割り当てられた色とフォントが表示されます。すべてのファイルの要素の書式設定が終了したら、**OK** をクリックします。

### *配色の使用*

配色は、フォント、フォントサイズ、スクリプトなどの書式オプションの設定を保存したものです。ファイルの種類と配色を選択することにより、拡張エディタの表示オプションを設定できます。個々のファイルの要素を設定する必要はありません。SAS システムでは、いくつかの配色を**配色**ボックスから選択できます。SAS により提供される配色は、デフォルトのスクリプトを使用しています。

配色を作成するには、次の操作を行います。

1. **ファイルの種類**ボックスからファイルの種類を選択します。
2. フォント、フォントサイズ、スクリプトを選択します。
3. 各ファイル要素について、前景色と背景色、フォントスタイル、下線オプションを選択します。
4. **登録**をクリックし、**配色の登録**ダイアログボックスに配色を入力します。
5. **OK** をクリックします。

配色を変更するには、次の操作を行います。

1. **ファイルの種類**ボックスをクリックして、ファイルの種類を選択します。
2. **配色**ボックスをクリックして、配色を選択します。
3. フォントおよびファイル要素に、必要な変更を加えます。

4. **登録**をクリックします。選択された配色名が、**配色名エントリ**ボックスに表示されます。
5. **OK** をクリックします。

配色を削除するには、次の操作を行います。

1. **ファイルの種類**ボックスをクリックして、ファイルの種類を選択します。
2. **配色**ボックスをクリックして、削除する配色名を選択します。
3. **削除**をクリックします。

## *拡張エディタでショートカットキーをカスタマイズする*

### *ショートカットキーの割り当て*

**拡張エディタキー**ダイアログボックスを開くと、ショートカットキーに割り当てたコマンドのみを表示したり、すべてのコマンドを表示したりできます。ショートカットキーに割り当てたコマンドのみを表示するには、**すべてのコマンドを表示する**チェックボックスが選択されていないことを確認してください。ショートカットキーが割り当てられていないコマンドを含めたすべてのコマンドを表示するには、**すべてのコマンドを表示する**チェックボックスを選択します。

ショートカットキーを割り当てるには、次の操作を行います。

1. 次のメニューパスを選択します:**ツール** ⇨ **オプション** ⇨ **拡張エディタキー**
2. **カテゴリ**ボックスから、カテゴリを選択します。マクロは、**ユーザー定義**カテゴリを選択すると表示されます。
3. **コマンド**ボックスからコマンドを選択します。コマンドにすでにショートカットキーが定義されている場合は、**キー**列に表示されます。
4. **キーの割り当て**をクリックします。
5. **新しいショートカットキー**フィールドにカーソルを置きます。
6. 選択されたコマンドに割り当てるショートカットキーを押します。**新しいショートカットキー**フィールドにショートカットキーが表示され、キー割り当てのステータスがダイアログボックスの下部に表示されます。**現在の割り当て**フィールドの値が**なし**の場合、このショートカットキーに割り当てられている他のコマンドはありません。
7. ショートカットキーを割り当てるには、**割り当て**をクリックします。

   *注:* 他のコマンドに割り当てられているショートカットキーを割り当てると、そのコマンドのショートカットキーが削除されます。たとえば、Backspace キーを**新しい省略キーを追加する**コマンドに割り当てた場合は、Backspace キーを押すと**省略キー**ダイアログボックスが表示され、Backspace キーを使用して後退することはできなくなります。

### *ショートカットキーの削除*

ショートカットキーを削除するには、次の操作を行います。

1. 次のメニューパスを選択します:**ツール** ⇨ **オプション** ⇨ **拡張エディタキー**
2. **キーの割り当て**をクリックします。
3. **カテゴリ**ボックスから、カテゴリを選択します。マクロは、**ユーザー定義**カテゴリを選択すると表示されます。

4. **コマンド**ボックスでコマンドを選択します。
5. **現在のキー**リストボックスで、ショートカットキーを選択します。
6. **削除**をクリックします。

#### *ショートカットキーを拡張エディタのデフォルトにリセットする*

ショートカットキーをデフォルトにリセットすると、すべてのマクロのショートカットキーが削除されます。デフォルトのショートカットキーのリストについては、"拡張エディタでのショートカットキー" (631 ページ) を参照してください。

ショートカットキーを、拡張エディタのデフォルトのショートカットキーにリセットするには、次の操作を行います。

1. 次のメニューパスを選択します:**ツール** ⇨ **オプション** ⇨ **拡張エディタキー**
2. **キーの割り当て**をクリックします。
3. **すべてリセット**をクリックします。

### *拡張エディタの有効と無効の切り替え*

デフォルトでは、拡張エディタは SAS システムの起動時に有効になっています。

SAS システムの起動時に拡張エディタを無効にするには、NOENHANCEDEDITOR システムオプションを使用します。詳細については、"ENHANCEDEDITOR システムオプション:Windows" (506 ページ) を参照してください。また、**プリファレンス**ダイアログボックスの編集タブにある**拡張エディタを使用する**設定を使用するか、**wedit** コマンドを使用して、拡張エディタの有効と無効を切り替えることもできます。詳細については、"WEDIT コマンド:Windows" (361 ページ) を参照してください。

拡張エディタを無効にすると、メニューコマンドは利用できなくなります。すべての拡張エディタウィンドウが開いたままの状態で、**表示**メニューまたは **WEDIT** コマンドを使用して新しい拡張エディタウィンドウを開くことができます。SAS の起動時に拡張エディタが無効である場合、**拡張エディタ**ウィンドウは開きません。

拡張エディタが有効になっている場合、**ツール**メニューの**テキストエディタ**コマンドにより**拡張エディタ**ウィンドウが開きます。拡張エディタが無効になっている場合は、**テキストエディタ**コマンドにより SAS NOTEPAD が開きます。

---

## プログラムエディタの使用

### *プログラムエディタの使用の概要*

SAS テキストエディタウィンドウ(プログラムエディタと NOTEPAD)は、他の Windows エディタと同様の方法で操作できます。したがって、SAS コードを編集するために、あらためてテキストエディタの使い方を覚える必要はありません。

### *拡張エディタとプログラムエディタの切り替え*

拡張エディタが SAS の起動時に有効になっている場合、プログラムエディタは無効になっています。SAS の起動時にプログラムエディタを有効にするには、次の方法の 1 つを使用して拡張エディタを無効にします。

- NOENHANCEDEDITOR システムオプションを使用して、SAS を起動します。
- **プリファレンス**ダイアログボックスで、拡張エディタを無効にします。
  1. **ツール** ⇨ **オプション** ⇨ **プリファレンス** ⇨ **表示**を選択します。
  2. **拡張エディタを使用する**チェックボックスを選択しないようにします。
  3. **OK** をクリックします。

NOENHANCEDEDITOR システムオプションの詳細は、“ENHANCEDEDITOR システムオプション:Windows” (506 ページ) を参照してください。

**プログラムエディタ**ウィンドウへは、いつでも**表示**メニューからアクセスできます。

## *ファイルを開く*

プログラムエディタでファイルを開くには、次の操作を行います。

1. プログラムエディタウィンドウをアクティブにした状態で、次の操作の 1 つを実行します。
   - **開く**ツールバーボタン(開いた形のファイルフォルダのアイコン)をクリックします。
   - コマンドバーに `dlgopen` と入力します。
   - 次の順に選択します。**ファイル**、**開く**

   **ファイルを開く**ダイアログボックスが表示されます。
2. **ファイルを開く**ダイアログボックスを使用して、取り込むファイルを選択します。デフォルトでは、SAS ファイル拡張子の付いたファイルが検索されます(このファイルには、SAS コードが記述されています)。**ファイルの種類**フィールドを使用して、デフォルトを変更することもできます。(選択するファイルの種類を変更した場合、SAS によってそのファイルの種類が記憶され、SAS セッション中にそのウィンドウで次にファイルを開いたときにデフォルトとして表示されます。

   *注:* **ファイルを開く**ダイアログボックスのデフォルトのディレクトリを変更するには、SASINITIALFOLDER システムオプションを使用して SAS を起動するか、現在の作業ディレクトリを変更します。詳細については、“SASINITIALFOLDER システムオプション:Windows” (558 ページ) および “SAS の現在のフォルダの変更” (41 ページ) を参照してください。
3. 取り込むファイルにサブミットする SAS コードが含まれている場合、**サブミット**チェックボックスを選択してから **OK** をクリックします。

   *注:* **サブミット**チェックボックスを選択すると、**ファイルを開く**ダイアログボックスを使用してファイルを開くときに、いつもサブミットチェックボックスが選択されている状態になります。開くファイルの内容をサブミットしない場合は、このチェックボックスを選択しないでください。

ファイルを、Windows エクスプローラまたは**お気に入りフォルダ**ウィンドウからプログラムエディタへドラッグアンドドロップすることもできます。ファイルをドラッグアンドドロップするには、次の操作を行います。

1. ソースウィンドウを開きます。
2. ソースウィンドウおよび**プログラムエディタ**ウィンドウの両方が見えるように、ウィンドウを配置します。

3. ソースウィンドウで、開くファイルを探して選択し、マウスの左ボタンでクリックしたままにします。
4. ファイルを**プログラムエディタ**ウィンドウにドラッグし、マウスの左ボタンを離します。

**プログラムエディタ**ウィンドウで 256 文字よりも長い行を含むファイルを開いた場合、256 文字を超える部分が切り捨てられます。これを避けるには、INCLUDE コマンドの LRECL=の値に、最も長い行の文字数と等しい値を指定し、AUTOWRAP コマンドまたは AUTOFLOW コマンドをオンに設定します。**ファイルを開く**ダイアログボックスを使用して 256 文字を超える行を含むファイルを開くには、FILENAME ステートメントを使用して、ファイル参照名を目的のファイルに割り当てます。このファイル参照名を二重引用符で囲み、**ファイルを開く**ダイアログボックスの**ファイル名**フィールドに指定します。

256 文字を超える行を含む SAS プログラムをプログラムエディタにリコールした場合、プログラムエディタではその行が次の行に折り返されます。行が 256 文字を超えて次の行に折り返されている場合、1 行のコードとみなされます。

## *行番号の使用*

z/os などのオペレーティングシステムで **SAS プログラムエディタ**ウィンドウを使用する場合と異なり、Windows 環境では行番号がデフォルトでは表示されません。コマンドバーに **`numbers on`** と入力すると、**プログラムエディタ**ウィンドウに行番号が表示されます。行番号の表示のオンとオフを切り換えるには、**`nums`** と入力します。

プログラムエディタウィンドウまたは NOTEPAD ウィンドウがアクティブになっているときは、**エディタのオプション**ダイアログボックスで行番号の表示を制御することもできます。**エディタのオプション**ダイアログボックスを開くには、次の操作を行います。

1. コマンドバーに **`edop`** と入力するか、次のメニューパスを選択します:**ツール** ⇨ **オプション** ⇨ **プログラムエディタ**
2. **編集**タブを選択します。
3. **行番号を表示する**を選択し、**OK** をクリックします。

## *挿入位置の移動*

SAS テキストウィンドウにおける挿入位置の移動キー(矢印キー、Page Up キー、Page Down キーなど)の機能は、他の Windows アプリケーションで使用する場合と同じです。

Ctrl キーを押しながら左矢印キーまたは右矢印キーを押すと、挿入位置が一度に 1 単語左右に移動します。これらの操作を使用してテキスト上を移動するときに、行の末尾または先頭に到達した場合は、そこで止まります。ファイルの先頭に移動する場合は、Ctrl キーを押しながら Page Up キーを押します。ファイルの末尾に移動する場合は、Ctrl キーを押しながら Page Down キーを押します。

Home キーを押すと、挿入位置は現在の行の先頭に移動します。ただし、アクティブウィンドウのコマンド行(コマンドバーではありません)が有効である場合を除きます。コマンド行がアクティブなときは、Home キーを押すたびに挿入位置が現在位置とコマンド行の間を移動します。F11 キーを押すと、挿入位置はコマンドバーに移動します。コマンド行のオンとオフを切り換えるには、COMMAND コマンドを使用するか、**プリファレンス**ダイアログボックスの**表示**タブで**コマンド行**を選択します。

### タブの使用

多くのテキストエディタでは、タブ文字がそのまま使用されます。その他にタブ文字が空白文字に変換されるエディタもあります。**SAS プログラムエディタ**ウィンドウでは、タブ文字は空白文字に変換されます。Tab キーを押すとスペースが挿入され、すべてのテキストが挿入位置の右側に移動します。

### 改行について

概念的には、改行文字は行頭ではなく行末に作成されます。Enter キーを押すと、改行文字が作成されます。改行文字を削除するには、行の先頭で Backspace キーを押すか、行末で Delete キーを押します。

### テキストの選択

テキストを選択するには、マウスまたは Shift キーを、挿入位置移動キーと組み合わせて使用します。マウスボタンまたは Shift キーを離す位置までを、テキストの範囲として選択できます。アクティブウィンドウのすべてのテキストを選択するには、編集メニューから**すべて選択**を選択します。高度なテキスト選択方法を次に示します。

- 選択する 1 つの単語をダブルクリックします。1 つの行全体を選択するには、Ctrl キーを押したまま対象の行をクリックします。
- Alt キーとマウスボタンを押しながらマウスをドラッグすると、テキストの矩形ブロック(または列)を選択できます(次の図を参照)。

- Shift キーを押したままマウスボタンをクリックすると、現在の挿入位置からクリックした位置までのテキストが選択されます。この方法は、テキストの選択領域を拡張するときにも使用できます。(この機能を使用できるのは、現在のページ内だけです。)

テキストを選択した状態でキー入力を開始すると、選択した範囲では入力テキストに置換されます。選択領域から挿入位置を移動している場合も同じです。マウスを使用してテキストを選択およびコピーする方法については、“クリップボードの使用” (55 ページ) を参照してください。

テキストの選択を解除するには、ウィンドウ内でマウスの左ボタンをクリックします。**編集**メニューの**編集解除**を選択するか、Esc キーを押すことによっても選択を解除できま

す。矢印キーを使用したテキストの選択解除は、コマンドバーに WNAVKEYUNMARK ON と入力して有効にすることもできます。

### *テキストの削除*

テキストを現在選択している場合、Delete キーを押すとそのテキストが削除されます。テキストを選択していない場合は、カーソル位置の右側の文字が削除されます。挿入位置から現在の行の末尾までのテキストを削除するには、Alt キーを押しながら Delete キーを押します。挿入位置から現在の単語の末尾までを削除するには、Ctrl キーを押しながら Delete キーを押します。挿入位置から現在の単語の先頭までを削除するには、Ctrl キーを押しながら Backspace キーを押します。

**編集**メニューを使用してテキストを削除することもできます。ウィンドウ内のすべてのテキストを削除するには、**すべてクリア**をクリックします。選択したテキストのみを削除するには、**クリア**をクリックします。選択したテキストを削除して、そのテキストを Windows クリップボードにコピーするには、**切り取り**をクリックします。

### *テキストの検索と置換*

テキストを検索するには、次の操作を行います。

1. 次のメニューパスを選択して、**検索**ダイアログボックスを開きます。**編集** ⇨ **検索**
2. 次の情報を入力します。

   **検索する文字列**
   検索するテキスト文字列を入力します。このフィールドの初期値は、検索に使用された最後のテキスト文字列です。

   **検索する方向**
   **上へ**または**下へ**チェックボックスを選択します。**上へ**は、カーソル位置からファイルの先頭方向に向けて検索することを指定します。**下へ**は、カーソル位置からファイルの末尾方向に向けて検索することを指定します。

   **完全に一致するワードだけを検索する**
   検索条件とテキストが部分的に一致するワードではなく、完全に一致するワードを検索するときにチェックボックスを選択します。

   **大文字と小文字を区別する**
   大文字と小文字を区別して検索する必要があるときに、チェックボックスを選択します。

3. **次を検索**をクリックします。

テキストを検索および置換するには、次の操作を行います。

1. 次のメニューパスを選択して、**置換**ダイアログボックスを開きます:**編集** ⇨ **置換**
2. 次の情報を入力します。

   **検索する文字列**
   検索および置換するテキスト文字列を入力します。このフィールドの初期値は、検索に使用された最後のテキスト文字列です。

   **置換後の文字列**
   置き換えた後の文字列を入力します。

**検索する方向**
**上へ**または**下へ**チェックボックスを選択します。**上へ**は、カーソル位置からファイルの先頭方向に向けて検索することを指定します。**下へ**は、カーソル位置からファイルの末尾方向に向けて検索することを指定します。

**完全に一致するワードだけを検索する**
検索条件とテキストが部分的に一致するワードではなく、完全に一致するワードを検索するときにこのチェックボックスを選択します。

**大文字と小文字を区別する**
大文字と小文字を区別して検索する必要があるときに、このチェックボックスを選択します。

3. **次を検索**をクリックします。
4. テキストが検索された場合、次の項目の 1 つをクリックします。
   - **置換して次に**をクリックすると、検索された 1 つのテキストが置換文字列と置き換えられます。
   - **すべて置換**をクリックすると、検索されたファイル内のすべてのテキストが、置換文字列と置き換えられます。

## *テキストのドラッグアンドドロップ*

次の表に、ドラッグが可能なテキストの場所と、選択したテキストをドロップできる場所の一覧を示します。

***表 3.13*** *テキストのドラッグアンドドロップの概要*

| ドラッグが可能なテキスト | テキストのドロップ先 |
|---|---|
| 任意の SAS テキストウィンドウ | **プログラムエディタ**ウィンドウなどの、テキストの編集がサポートされている別の SAS ウィンドウ |
| 任意の SAS テキストウィンドウ | テキストのドラッグアンドドロップがサポートされている別の Windows アプリケーション |
| テキストのドラッグアンドドロップがサポートされている Windows アプリケーション | 編集がサポートされている任意の SAS ウィンドウ |
| Windows エクスプローラのテキストファイル項目 | 編集がサポートされている任意の SAS ウィンドウ |

テキストをウィンドウ間でドラッグアンドドロップするには、次の操作を行います。

1. ソースウィンドウとターゲットウィンドウの両方がディスプレイに表示されるように、必要に応じてウィンドウを調整します。

   *注:* ターゲットウィンドウが表示されるようにウィンドウを調整するかわりに、選択したテキストをウィンドウバーのターゲットウィンドウのボタンにドラッグしても、ターゲットウィンドウがアクティブになります。

2. ソースウィンドウから、目的のテキストを選択します。
3. 選択したテキストにポインタを置き、マウスの左ボタンを押したしたままにします。

4. マウスボタンを押したまま、テキストをターゲットウィンドウへドラッグします。
5. テキストを挿入する位置まで挿入位置を移動します。(テキストを SAS コードとして処理するためにサブミットする場合は、挿入位置をウィンドウ内の任意の場所に移動できます。
6. マウスボタンを離します。テキストが挿入位置に挿入されるか、SAS にサブミットされて処理されます。(デフォルトアクションは、ターゲットウィンドウのタイプによって異なります。

ドラッグアンドドロップのデフォルトアクションを無効にするには、マウスの右ボタンを使用してドラッグアンドドロップを開始します。このアクションは、デフォルト以外のドラッグアンドドロップと呼ばれます。選択したテキストをターゲットの SAS ウィンドウにドラッグしてマウスボタンを離すと、ポップアップメニューが表示され、実行するアクションを選択できます。

表 3.14 (113 ページ) に、可能性のある SAS のターゲットウィンドウで使用可能なドラッグアンドドロップアクションの概要を示します。

***表 3.14*** *ドラッグアンドドロップアクションの概要*

| データ | ターゲット | デフォルトアクション | デフォルト以外のアクション |
|---|---|---|---|
| テキスト | SAS テキストエディタ | 移動 | 移動、コピー、キャンセル |
| テキスト | プログラムエディタ | コピー | コピー、サブミット、キャンセル |
| ファイル | SAS テキストエディタ | 無効 | 無効 |
| ファイル | プログラムエディタ | 移動 | コピー、サブミット、キャンセル |
| ファイル | ログ、出力 | サブミット | サブミット、キャンセル |

テキストを SAS ウィンドウから別の Windows アプリケーションにドラッグしたときに実行されるアクションは、アプリケーションによって異なります。ほとんどの場合、SAS と他のアプリケーション間でテキストをドラッグアンドドロップすると、ウィンドウ間でテキストが移動します(つまり、一方のウィンドウで切り取られたテキストが他方のウィンドウに挿入されます)。

ドラッグ補助キーを使用することにより変更できます。ドラッグ補助キーとは、ドラッグアンドドロップしながら押すキーのことです。テキストをウィンドウ間で(移動するのではなく)コピーするには、ドラッグする前に Ctrl キーを押し、キーを押したままドラッグアンドドロップします。マウスボタンを離してテキストをドロップするときに、Ctrl キーも離します。

### *ドラッグ時のスクロール*

SAS テキストエディタウィンドウにテキストをドラッグするときに、ターゲットウィンドウを上下または左右にスクロールできます。このアクションによって、現在表示されていないウィンドウ領域にテキストをドロップできます。

テキストを選択して SAS テキストエディタウィンドウにドラッグするときに、SAS テキストエディタウィンドウの境界付近でマウスポインタの移動を少しの間停止します。ウィンドウが、境界の方向にスクロールします。たとえば、ターゲットウィンドウを下にスクロールするには、マウスポインタをウィンドウ下部の境界の少し上にドラッグし、そこで一時停止します。

ドラッグ時のスクロールは、ドロップ領域の境界付近で一時停止した場合のみに起こります。境界上をすばやく通過したときには起こりません。

### *リッチテキストフォーマットのテキストの使用*

SAS ウィンドウのテキストをクリップボードにコピーして他のアプリケーションのウィンドウに貼り付けるときに、ターゲットウィンドウで RTF ファイル形式がサポートされている場合、SAS で設定されていたすべての書式情報(色情報は除く)が保持されます。たとえば、Windows のメモ帳では書式情報は保持されませんが、Microsoft ワードパッドおよび多くのワードプロセッサでは保持されます。SAS のテキストをドラッグして他のアプリケーションウィンドウにドロップする場合も、同様に保持されます。

表示フォントが Sasfont の場合、SAS からコピーするすべてのテキストは SAS Monospace TrueType フォントで表示されます。下線などの反転表示属性がテキストに設定されている場合は、これらの属性も保持されます(ターゲットウィンドウで RTF がサポートされている場合)。

### *ファイルの保存*

**プログラムエディタ**ウィンドウの内容を保存するには、上書き保存ツールバーボタンをクリックします(ドライブのアイコン)。初めて保存するファイルの場合は、**名前を付けて保存**ダイアログボックスが表示され、ファイル名の入力を求められます。

新しいファイル名を付けてファイルを保存するには、次の操作を行います。

1. 次のメニューパスを選択します:**ファイル** ⇨ **名前を付けて保存**
2. **保存する場所**フィールドでフォルダを選択します。
3. **ファイル名**フィールドにファイル名を入力します。
4. **ファイルの種類**フィールドからファイルの種類を選択します。
5. **OK** をクリックします。

*注:* **名前を付けて保存**ダイアログボックスのデフォルトのディレクトリを変更するには、SASINITIALFOLDER システムオプションを使用して SAS を起動するか、現在の作業ディレクトリを変更します。詳細については、“SASINITIALFOLDER システムオプション:Windows” (558 ページ) および “SAS の現在のフォルダの変更” (41 ページ) を参照してください。

### *自動保存によるプログラムエディタファイルの保存*

SAS には、プログラムエディタでの作業内容が失われないように、ユーザーが指定した間隔でファイルを自動的に保存する機能があります。指定できる間隔は、0(自動保存オフ)～480 分です。デフォルトの間隔は 10 分です。

自動保存ファイルは、カレントフォルダまたは AUTOSAVELOC システムオプションを使用して指定された場所に pgm.asv のファイル名で保存されます。

自動保存のオン/オフの切り替えと間隔を設定するには、次の操作を行います。

1. 次のメニューパスを選択します:**ツール** ⇨ **オプション** ⇨ **プリファレンス** ⇨ **編集タブ**

2. **自動保存するまでの時間**を選択または選択解除します。
3. **分**ボックスに 1 から 480 までの数字を入力して、間隔を指定します。

自動保存のオン/オフの切り替えと間隔の設定には、WAUTOSAVE コマンドを使用することもできます。WAUTOSAVE INTERVAL=*minutes* で *minutes* に間隔を指定すると、自動保存がオンになります。

自動保存機能の詳細は、“編集プリファレンス” (60 ページ) 、“WAUTOSAVE コマンド:Windows” (357 ページ) および “AUTOSAVELOC= System Option” (*SAS System Options: Reference*)を参照してください。

## *SAS テキストエディタ独自の機能について*

次に示す SAS テキストエディタウィンドウの機能は、Windows 環境で一般的に使用される他のエディタの標準機能と異ります。

- SAS テキストエディタウィンドウでは、挿入位置を、行に入力した最後の文字または入力したテキストの最終行より下の位置まで移動できます。
- テキストの一部を選択してから、マウスポインタを選択範囲の外に移動しても、選択範囲はそのまま維持されます。
- Esc キーを押すと、テキストの選択が解除されます。
- Shift キーを押しながら Tab キーを押すと、直前のタブストップに続くブランク文字を削除できます。

*4 章*

# Windows 版での SAS ファイルの使用

**SAS ファイルについて** . . . . . **118**
SAS ファイルについて . . . . . 118
SAS ファイルのファイル拡張子 . . . . . 118
SAS データセット(メンバタイプ:Data または View) . . . . . 120
SAS カタログ(メンバタイプ:Catalog) . . . . . 121
コンパイル済み SAS DATA ステッププログラム(メンバタイプ:Program) . . . . . 122
アクセスディスクリプタファイル(メンバタイプ:Access) . . . . . 122

**マルチエンジンアーキテクチャ** . . . . . **122**
SAS ライブラリ . . . . . 122
SAS エンジン . . . . . 123

**データライブラリの使用** . . . . . **125**
データライブラリ . . . . . 125
ライブラリ参照名の指定 . . . . . 125
グラフィカルユーザーインターフェイスを使用した SAS ライブラリの割り当て . . 126
LIBNAME ステートメントまたは LIBNAME 関数を使用した
SAS ライブラリの割り当て . . . . . 126
環境変数を使用した SAS ライブラリの割り当て . . . . . 129
ライブラリ参照名の割り当てのリストの表示 . . . . . 130
ライブラリ参照名のクリア . . . . . 131
複数のフォルダが格納された SAS ライブラリへのアクセス方法について . . . . . 132
Sasuser データライブラリの更新 . . . . . 133
WORK データライブラリの使用 . . . . . 133
Windows (NTFS)における大きいデータセットの使用 . . . . . 135

**複数の SAS セッションから SAS ファイルにアクセスする** . . . . . **135**

**Windows 版 SAS 9.4 で他のバージョンのファイルを使用する** . . . . . **136**
Windows 版 SAS 9.4 での他のバージョンのファイルの使用について . . . . . 136
リリース 6.08 からリリース 8.2 までのデータセットの使用 . . . . . 137
リリース 6.03 とリリース 6.04 の SAS データセットの使用 . . . . . 138
リリース 6.08 からリリース 6.12 までの SAS データセットの変換 . . . . . 138
SAS 9.4 でバージョン 7 と 8 の SAS カタログを使用する . . . . . 138
SAS 9.4 でバージョン 6 の SAS カタログを変換する . . . . . 138
リリース 6.08 の SAS カタログを SAS 9.4 に変換する . . . . . 139
リリース 6.03 およびリリース 6.04 の SAS カタログを SAS 9.4 に変換する . . . . 140
リリース 6.08 からリリース 6.12 までのデータセットの作成 . . . . . 140

**SAS 9.4 のファイルを以前のリリースで使用する** . . . . . **140**

**SAS 9.4 でリモートホストの SAS ファイルを使用する** . . . . . **141**

**BMDP ファイル、OSIRIS ファイル、SPSS ファイルの読み取り** . . . . . **141**

概要:BMDP ファイル、OSIRIS ファイル、SPSS ファイルの読み取り . . . . . 141
BMDP Engine . . . . . 141
OSIRIS Engine . . . . . 142
SPSS Engine . . . . . 143
**SAS ファイルを動作環境間で転送する** . . . . . **144**
**SAS/ACCESS を使用し、データベースファイルにアクセスする** . . . . . **144**
**SAS ODBC Driver を使用し、他のアプリケーションから SAS データにアクセスする** . . . . . **145**

# SAS ファイルについて

## *SAS ファイルについて*

SAS では、SAS ファイルと呼ばれる、さまざまな特別な構造を持つファイルが作成され、使用されます。Windows では SAS ファイルの格納が管理されますが、オペレーティングシステムから SAS ファイルは処理できません。たとえば、Windows エクスプローラを使用して SAS ファイルをリストすることはできますが、SAS ファイルの処理は SAS でしかできません。SAS ファイルは、外部ファイルとは異なります。外部ファイルを SAS ステートメントおよびコマンドによって処理することはできますが、SAS では外部ファイルは管理できません。

SAS ファイルは、通常は SAS ライブラリに格納されます。Windows 環境では、SAS ライブラリは、SAS からのアクセスが可能な 1 つ以上の Windows フォルダに格納される SAS ファイルのコレクションに名前を付けたものです。SAS ライブラリ内のファイルが初めてアクセスされる時に、そのライブラリに対してエンジンが設定されます。このエンジン名によって、SAS がライブラリ内のファイルを処理するために使用するアクセス方法が指定されます。SAS ライブラリの詳細については、*SAS 言語リファレンス: 解説編* を参照してください。

さまざまなエンジンがあるため、SAS はフォーマットやバージョンの異なる SAS ファイルや、他のベンダのファイルにアクセスすることができます。このため、SAS はマルチエンジンアーキテクチャを持つと言われます。マルチエンジンアーキテクチャと変換ユーティリティを組み合わせることによって、Windows や他の動作環境で作成されたかに関わらず、SAS 9.4 のファイルやバージョン 5 までの SAS によって作成された SAS ファイルにアクセスすることができます。また、マルチエンジンアーキテクチャでは、データベースファイルなど、他社のソフトウェアによって作成されたファイルにアクセスすることもできます。

次のセクションでは、Windows 環境で、さまざまなエンジンを使用して SAS ファイルを作成および使用するために必要な情報を中心に説明します。

## *SAS ファイルのファイル拡張子*

SAS ファイルは SAS ライブラリに格納され、ライブラリのメンバと呼ばれます。各メンバには、メンバタイプが設定されます。SAS では、フォルダ内の SAS ファイルと外部 Windows ファイルを区別する際、重複しないファイル拡張子を使用します。一般的な SAS メンバタイプには、特定のファイル拡張子が割り当てられています。次の表に、Windows ファイル拡張子と、それに対応する V6、V7、V8、V9 Engine の SAS メンバタイプをリストします。詳細については、“マルチエンジンアーキテクチャ” (122 ページ) を参照してください。以前のリリースのファイルの処理の詳細については、“Windows 版 SAS 9.4 で他のバージョンのファイルを使用する” (136 ページ) を参照してください。

*表 4.1 Windows ファイル拡張子と、それに対応する SAS メンバタイプ*

| V6 のファイル拡張子 | V7 以降のファイル拡張子 | 短い拡張子 | SAS メンバタイプ | 説明 |
|---|---|---|---|---|
| .sas | .sas | なし | なし | SAS プログラム |
| .ss2 | .sas7bpgm | ss7 | プログラム | Stored Program (DATA ステップ) |
| | .cfg (バージョン 8 以降) | なし | なし | 構成ファイル |
| .lst | .lst | なし | なし | 出力ファイル |
| .log | .log | なし | なし | ログファイル |
| なし | .sas7baud | st7 | Audit | 監査ファイル |
| .sd2 | .sas7bdat | sd7 | データ | データセット |
| .sv2 | .sas7bvew | sv7 | ビュー | データセットビュー |
| .si2 | .sas7bndx | si7 | インデックス | データセットインデックス<br>インデックスは、SAS データファイルとは別のファイルとして格納されますが、SAS では、SAS データファイルの一部として扱われます。 |
| .sc2 | .sas7bcat | sc7 | カタログ | SAS カタログ |
| .sa2 | .sas7bacs | sa7 | アクセス | アクセスディスクリプタファイル |
| .sf2 | .sas7bfdb | sf7 | FDB | 統合データベースファイル |
| .sm2 | .sas7bmdb | sm7 | MDDB | 多次元データベースファイル |
| なし | .sas7bdmd | s7m | DMDB | データマイニングデータベースファイル |
| なし | .sas7bitm | sr7 | Itemstor | アイテム格納ファイル |
| .su2 | .sas7butl | su7 | ユーティリティ | ユーティリティファイル |
| .sp2 | .sas7bput | sp7 | PUtility | 永久ユーティリティ |
| .stx | なし | なし | なし | 移送ファイル |
| なし | .sas7bbak | なし | なし | バックアップファイル |

***注意:***
**SAS ファイルの拡張子は変更しないでください。変更すると、予測できない結果が発生する可能性があります。** SAS が割り当てた SAS ファイルのファイル拡張子は、ファイル

へのアクセス方法の欠くことのできない要素です。また、オペレーティングシステムのコマンドを使用して SAS ファイルの名前を変更しないでください。SAS ファイルの名前を変更する場合は、DATASETS プロシジャを使用するか、**SAS エクスプローラ**ウィンドウまたは**お気に入りフォルダ**ウィンドウでファイルを選択して、**編集** ⇨ **名前の変更**を選択します。

*注:* SAS セッション中に、WORK および SASUSER ライブラリのファイルを削除しないよう注意してください。SAS では、一時ユーティリティファイルが作成されます。これらのファイルはユーザーが直接アクセスする必要はありませんが、SAS データの処理に必要です。SAS セッションが異常終了した場合、ディスク容量を回復するため、SAS 外部のファイルを手動で削除する必要が生じることがあります。WORK ライブラリ内のファイルを削除するには、SAS 起動時に WORKINIT および WORKTERM システムオプションを使用します。詳細については、"WORKINIT System Option" (*SAS System Options: Reference*)および "WORKTERM System Option" (*SAS System Options: Reference*)を参照してください。

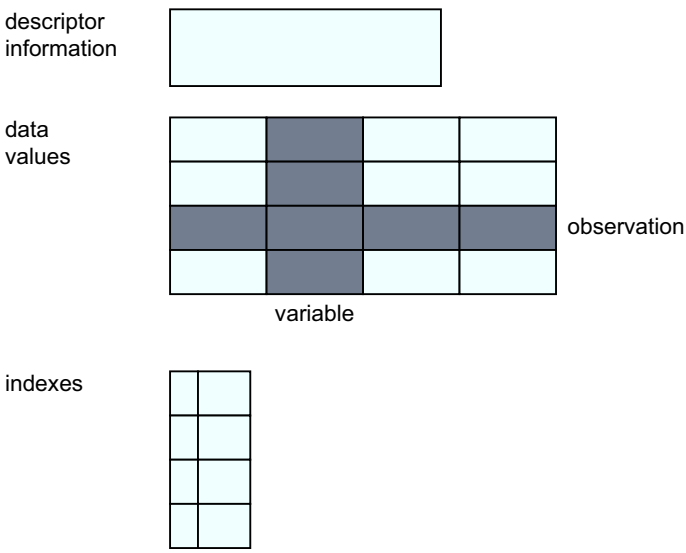
## *SAS データセット(メンバタイプ:Data または View)*

SAS データセットは、ここで説明する SAS データファイルと SAS データビューに対する包括的な用語です。ここでは、SAS データセットの概念の概要を簡単に説明します。詳細については、*SAS 言語リファレンス: 解説編*のデータセットに関するセクションを参照してください。

論理的には、SAS データセットはディスクリプタ情報とデータ値の 2 種類の情報で構成されます。ディスクリプタ部には、データセット名、データセットタイプ、データセットラベル、変数の数、データセット内の変数の名前やラベル、変数の型(文字または数値)、変数の長さ、レコード内での変数の位置、変数のフォーマットなどのディスクリプタ情報が含まれています。ディスクリプタ情報にアクセスするには、"CONTENTS プロシジャ:Windows" (427 ページ) を使用します。

データ値には、変数の値が含まれています。SAS データセットは、オブザベーションの行と変数値の列で構成されるテーブルとして表すことができます。次のテーブルは、SAS データセットのモデルを表しています。

***図 4.1*** *SAS データセットのモデル*




拡張属性は、SAS ファイルのカスタマイズされたメタデータと考えることができます。共通の SAS 属性が事前定義済みの SAS システム属性であるのに対し、拡張属性はユ

ーザーが定義できます。DATA ステップは、拡張属性を入力データセットから出力データセットへ保存します。このデータセットには、ディスクへの保存時に.sas7bxat という新しい拡張子が付きます。DATASETS プロシジャで各種 XATTR ステートメントを使用することで、拡張属性の作成、追加、削除、更新、オプション指定を行うことができます。また、CONTENTS プロシジャを使用してデータセットや各種拡張属性を表示することもできます。詳細については、*SAS 言語リファレンス: 解説編*を参照してください。

SAS データファイル(メンバタイプ:Data)
SAS データファイルは、頻繁に使用されるタイプの SAS ファイルです。SAS データファイルは、Data という SAS メンバタイプであり、DATA ステップ中、および特定の SAS プロシジャ(Base SAS の RANK プロシジャなど)によって作成されます。SAS データファイルのファイル拡張子は、.sas7bdat です。

SAS では、ネイティブとインターフェイスという 2 種類の SAS データファイルが定義されます。ネイティブデータファイルには、SAS が規定するフォーマットのファイル内のデータ値とディスクリプタ情報(前述の説明参照)が格納されます。ネイティブデータファイルは、他の動作環境版のこれまでのバージョンの SAS でも使用されていました。Windows 版 SAS では、ネイティブの SAS データファイルにインデックスを付けられるようになりました。インデックスは、変数またはキーを使用して、SAS データファイル内のレコードにすばやくアクセスするために作成される、補助ファイルです。インデックスは、SAS データファイルとは別のファイルとして格納されますが、SAS では、SAS データファイルの一部として扱われます。インデックスの詳細については、*SAS 言語リファレンス: 解説編*を参照してください。

もう 1 つのタイプのデータファイルは、インターフェイス SAS データファイルです。これらのファイルには、他社のソフトウェア製品が規定するフォーマットのファイル内のデータが格納されます。インターフェイス SAS データファイルには、BMDP、OSIRIS、SPSS などのファイルがあります。これらのファイルには、SAS は読み取り専用でしかアクセスできません。詳細については、“BMDP ファイル、OSIRIS ファイル、SPSS ファイルの読み取り” (141 ページ) を参照してください。

SAS データビュー(メンバタイプ:View)
SAS データビューにはメンバタイプ View が設定されます。SAS データビューにはデータ値が表示され、その値の保存場所が示されますが、実際のデータ値は格納されません。SAS データビューのファイル拡張子は、.sas7bvew です。

ビューには、ネイティブやインターフェイスがあります。ネイティブ SAS データビューは、SQL プロシジャまたは DATA ステップによって作成され、1 つ以上の SAS データファイルあるいは SAS データビュー内のデータのサブセットや組み合わせが表示されます。SQL ビューの詳細については、*Base SAS プロシジャガイド*を参照してください。DATA ステップビューの詳細については、*SAS 言語リファレンス: 解説編*を参照してください。

インターフェイス SAS データビューには、他のソフトウェア製品(データベース管理システムなど)によってフォーマットされたデータのディスクリプタ情報が格納されます。データベースビューにアクセスするには、SAS/ACCESS LIBNAME ステートメントを使用します。詳細については、*SAS/ACCESS Interface to PC Files: Reference* および *SAS/ACCESS for Relational Databases: Reference* を参照してください。

## *SAS カタログ(メンバタイプ:Catalog)*

SAS カタログは、複数のエントリを格納できる、特別なタイプの SAS ファイルです。同じ SAS カタログにさまざまなタイプのエントリを保持できます。たとえば、Sasuser.Profile カタログには、ファンクションキーの定義、グラフィックアプリケーションのフォント、**プリファレンス**ダイアログボックスで選択された内容、その他の対話式ウィンドウプロシジャからの情報が含まれます。SAS カタログのファイル拡張子は、.sas7bcat です。

### コンパイル済み SAS DATA ステッププログラム(メンバタイプ:Program)

コンパイル済み DATA ステッププログラムは、コンパイルされ、SAS ライブラリに格納された DATA ステッププログラムを含む、SAS ファイルです。コンパイル済み DATA ステッププログラムは、必要に応じて、リコンパイルすることなく実行できます。コンパイル済み SAS DATA ステッププログラムの拡張子は、.sas7bpgm です。

コンパイル済みプログラムは、DATA ステップアプリケーションでのみ使用可能です。Stored Program には、グローバルステートメントを除く、すべての SAS 言語要素を含めることができます。ソースプログラムにグローバルステートメントを含めると、コンパイル済みプログラムは格納されますが、グローバルステートメントは格納されません。SAS ログに警告メッセージも表示されません。

このタイプの SAS ファイルの詳細については、*SAS 言語リファレンス: 解説編*を参照してください。

### アクセスディスクリプタファイル(メンバタイプ:Access)

SAS/ACCESS LIBNAME ステートメントによって作成されたディスクリプタファイルは、メンバタイプ ACCESS が設定され、インターフェイス SAS データビューの作成時に使用されます。ディスクリプタファイルには、SAS でサポートされる他のソフトウェア製品によってフォーマットされたデータが示されます。詳細については、*SAS/ACCESS for Relational Databases: Reference*、*SAS/ACCESS Interface to PC Files: Reference*、およびその他の入手可能な SAS/ACCESS ドキュメントを参照してください。

---

## マルチエンジンアーキテクチャ

### SAS ライブラリ

永久 SAS ファイルと一時 SAS ファイルはすべて、SAS ライブラリに格納されます。SAS ライブラリは、オペレーティングシステム内の物理的な場所に格納された SAS ファイルのコレクションです。オペレーティングシステム内の物理的な場所には、SAS では管理しないファイルも格納されますが、SAS ファイルのみが SAS ライブラリの一部と見なされます。どの Windows フォルダでも SAS ライブラリとして定義することができます。

SAS セッションで SAS ライブラリを使用するには、ライブラリにライブラリ参照名(libref)とエンジンを割り当てる必要があります。ライブラリ参照名は、SAS セッションまたは SAS ジョブ中に SAS データライブラリを参照するために使用する名前です。ライブラリ参照名は、**エクスプローラ**ウィンドウから作成することができます。あるいは変数、LIBNAME ステートメント、または LIBNAME 関数を使用したプログラムによって定義することもできます。Windows 環境でライブラリ参照名を使用する方法については、"データライブラリの使用" (125 ページ) を参照してください。ライブラリ参照名の詳細については、*SAS 言語リファレンス: 解説編*を参照してください。

**エクスプローラ**ウィンドウでは、ライブラリ参照名を含むすべての SAS ファイルを、簡単に管理することができます。**エクスプローラ**ウィンドウで SAS ファイルを操作する方法については、SAS ヘルプとドキュメントを参照してください。

# SAS エンジン

## エンジンについて

エンジンは、アクセスメソッドとも呼ばれます。エンジンを使用することで、SAS はさまざまなフォーマットのデータにアクセスでき、マルチエンジンアーキテクチャが実現されます。エンジンは、SAS データセットにのみ適用されます。

SAS がライブラリ内のファイルにアクセスする際に使用するルーチンのセットは、エンジンによって識別されます。このマルチエンジンアーキテクチャにより、SAS ファイルや、他社のソフトウェア製品(データベース管理システムなど)によって作成されたデータなど、さまざまなフォーマットのファイル内にあるデータを処理できます。ファイルの種類に合ったエンジンを使用することで、SAS はファイルへの書き込みやファイルからの読み込みを行うことができます。一部のファイルの種類には、使用するエンジンの指定が必要です。それ以外については、自動的に適切なエンジンが選択されます。エンジンとマルチエンジンアーキテクチャの詳細については、*SAS 言語リファレンス: 解説編*を参照してください。

エンジンには、ライブラリエンジンとビューエンジンの 2 種類があります。ライブラリエンジンは、SAS ライブラリレベルでアクセスを制御します。ライブラリエンジンは、LIBNAME ステートメントまたは LIBNAME 関数で指定します。ビューエンジンは、DATA ステップ、SQL プロシジャ、または SAS/ACCESS によって表示される SAS データビューを読み込むために使用されます。SAS データセットのディスクリプタ部にはビューエンジン名も含まれるため、SAS ビューエンジンは自動的に使用されます。

## ライブラリエンジンの種類

ライブラリエンジンには、ネイティブライブラリエンジンとインターフェイスライブラリエンジンの 2 種類があります。これらのエンジンは、SAS ライブラリモデルをサポートしています。ライブラリエンジンは、基本的な処理特性の決定など、いくつかの重要な機能を実行します。ライブラリエンジンの詳細については、*SAS 言語リファレンス: 解説編*を参照してください。ライブラリエンジンの使用例については、“データライブラリの使用” (125 ページ) を参照してください。

## ネイティブライブラリエンジン

ネイティブライブラリエンジンは、SAS によって作成され、更新される SAS ファイルにアクセスするエンジンです。ネイティブライブラリエンジンには、デフォルトエンジン、互換エンジン、移送エンジンがあります。次の表に、使用可能なこれらのエンジンの名前およびニックネームをまとめます。

***表 4.2*** *ネイティブライブラリエンジン*

| エンジンの種類 | エンジン名 | 説明 |
|---|---|---|
| デフォルト | V9、BASE | SAS System 9、9.1、9.2、9.3、SAS 9.4 のデータファイルにアクセスします。 |
| バージョン 8 互換性 | V8 | バージョン 8 のデータファイルにアクセスします。 |
| バージョン 7 互換性 | V7 | バージョン 7 のデータファイルにアクセスします。 |

| エンジンの種類 | エンジン名 | 説明 |
|---|---|---|
| リリース 6 互換性 | V6 | リリース 6.08 からリリース 6.12 で作成されたデータファイルにアクセスします。64 ビット環境の場合、V6 Engine ではデータの読み取りのみ可能です。 |
| リリース 6.12 互換性 | V612 | リリース 6.12 のデータファイルにアクセスします。 |
| リリース 6.03 およびリリース 6.04 互換性 | V604 | リリース 6.03 およびリリース 6.04 で作成されたデータファイルに、読み取り専用でアクセスします。 |
| 移送 | XPORT | 移送ファイルにアクセスします。 |

デフォルトのエンジンを使用するときは、SAS ジョブで V9 と BASE のどちらのエンジン名を使用するかを、今後のリリースを考慮して選択してください。SAS 9.4 専用のアプリケーションを使用し、最新のリリースに変換する予定がない場合は、エンジン名として V9 を使用します。しかし、新しいリリースの SAS に変換する計画がある場合、最新のデフォルトエンジンを参照するエンジン名として BASE を使用します。BASE を使用すると、プログラムは容易に変換できるようになります。エンジン名 BASE は、Base SAS を参照するのではなく、基本(プライマリ)エンジンを参照します。BASE Engine は、Base SAS 以外でも使用できます。

このドキュメントでは、デフォルトエンジンと言った場合、V9 Engine を指します。ENGINE システムオプションを使用してデフォルトエンジンが変更されていない限り、SAS 9.4 で SAS ファイルにアクセスする場合のデフォルトエンジンは、V9 Engine です。ENGINE システムオプションの値を確認するには、次のいずれかの操作を行います。

- サブミット

```
proc options option=engine;
run;
```

- **ツール** ⇨ **オプション** ⇨ **システム**選択して、**システムオプション**ウィンドウを開きます。次に、**ファイル** ⇨ **SAS ファイル**を選択します。Engine システムオプションに、SAS ライブラリのデフォルトのエンジンが表示されます。

### *インターフェイスライブラリエンジン*

インターフェイスライブラリエンジンは、他社のソフトウェア製品のファイルへのアクセスをサポートします。インターフェイスライブラリエンジンを使用すると、BMDP、OSIRIS、SPSS ファイルへの読み取り専用アクセスを行うことができます。インターフェイスライブラリエンジンの指定は、LIBNAME ステートメントまたは LIBNAME 関数で行います。インターフェイスエンジンの名前については、次の表を参照してください。

***表 4.3*** *インターフェイスライブラリエンジン*

| 名前 | 説明 |
|---|---|
| BMDP | 32 ビット動作環境で BMDP ファイルへの読み取り専用アクセスを可能にします。 |
| OSIRIS | OSIRIS ファイルへの読み取り専用アクセスを可能にします。 |

| 名前 | 説明 |
| --- | --- |
| SPSS | SPSS ファイルへの読み取り専用アクセスを可能にします。 |

これらのエンジンの詳細については、“BMDP ファイル、OSIRIS ファイル、SPSS ファイルの読み取り” (141 ページ) および “ENGINE システムオプション:Windows” (505 ページ) を参照してください。

### *エンジンの決定規則*

LIBNAME ステートメントまたは LIBNAME 関数でエンジン名を指定しなかった場合は、指定されたデータライブラリのライブラリ参照名に割り当てるエンジン(デフォルトエンジンまたは互換エンジン)が自動的に決定されます。Windows 版 SAS では、指定のフォルダにあるファイル拡張子が調べられ、次の規則に従ってライブラリ参照名に割り当てられるエンジンが決定されます。

- フォルダに含まれる SAS データセットのアクセスに、サポートされるネイティブライブラリエンジン(XPORT を除く)の 1 つのエンジンのみが使用される場合は、そのエンジンがライブラリ参照名に割り当てられます。
- 指定のフォルダに SAS データセットがない場合は、デフォルトエンジンがライブラリ参照名に割り当てられます。
- フォルダに含まれる SAS データセットのアクセスに複数のエンジンが使用される場合を混合モードライブラリと呼びます。デフォルトエンジンがライブラリ参照名に割り当てられます。ライブラリ参照名が混合モードライブラリに割り当てられたことを知らせるメッセージが、SAS ログに出力されます。

*注:* 通常は、エンジン名を指定するほうが、自動的に適切なエンジンを決定するよりも効率的です。

ENGINE システムオプションを使用して、混合モードライブラリや SAS ファイルが含まれないライブラリが検出されたときに使用されるデフォルトのエンジンを指定できます。デフォルトでは、ENGINE オプションは V9 に設定されています。詳細については、“ENGINE システムオプション:Windows” (505 ページ) を参照してください。

## データライブラリの使用

### *データライブラリ*

次のセクションでは、データライブラリ、ライブラリ参照名、複数のフォルダが格納されたライブラリ、SAS ユーザーライブラリ、Work データライブラリについて説明します。

### *ライブラリ参照名の指定*

ライブラリ参照名は、フォルダの格納場所(ドライブやフォルダなどのフルパス)を SAS で認識できるように、フォルダに割り当てられるラベル(別名)です。ライブラリ参照名は、ファイルとともに物理的に格納されているものではなく、物理的な場所を示す論理的な概念です。

ライブラリ参照名を SAS プログラムで作成した場合、ライブラリ参照名は、作成したセッション中のみ存在します。ライブラリ参照名が**新規ライブラリ**ダイアログボックスを使用して対話的に作成された場合は、**起動時に有効**を選択して、永久ライブラリ参照名にすることができます。

ライブラリ参照名は、SAS 名の構文規則に従います。SAS 命名規則の詳細については、*SAS 言語リファレンス: 解説編*の SAS 言語規則に関するセクションを参照してください。

次に、ライブラリ参照名を指定する方法を示します。

- SAS ヘルプとドキュメントで説明されている**新規ライブラリ**ダイアログボックスを使用します。
- "LIBNAME ステートメントまたは LIBNAME 関数を使用した SAS ライブラリの割り当て" (126 ページ) を参照してください。
- "環境変数を使用した SAS ライブラリの割り当て" (129 ページ) の説明に従って、変数を定義します。

*注:* LIBNAME ステートメントを使用せずに、直接ドライブ名とデータセット名を引用符で囲んで指定できます。次に、例を示します。

```
data "d:\mydata";
```

## *グラフィカルユーザーインターフェイスを使用した SAS ライブラリの割り当て*

グラフィカルユーザーインターフェイス(GUI)を使用してライブラリ参照名を割り当て、エンジンを指定するには、**新規ライブラリ** ツールバーボタン 、LIBASSIGN コマンド、またはエクスプローラを使用して、**新規ライブラリ**ダイアログボックスを開きます。

- ツールバーでライブラリの作成アイコンをクリックします。
- コマンドバーに **libassign** または **libname** と入力します。

  **LIBNAME** ウィンドウが開いたら、**新規作成**ツールバーボタンをクリックします。
- エクスプローラで、次の操作を行います。

  1. ライブラリフォルダを選択します。
  2. **ファイル** ⇨ **新規作成**を選択するか、ライブラリフォルダを右クリックし、**新規作成**をメニューから選択します。

  *注:* SAS ワークスペースの右側に 2 つ目の**エクスプローラ**ウィンドウが開いている場合、**新規ライブラリ**ダイアログボックスは**ライブラリ** フォルダを右クリックし、**新規作成**を選択して開くこともできます。

**新規ライブラリ**ダイアログボックスの詳細については、SAS ヘルプとドキュメントを参照してください。

## *LIBNAME ステートメントまたは LIBNAME 関数を使用した SAS ライブラリの割り当て*

### *LIBNAME ステートメント構文*

LIBNAME ステートメントや LIBNAME 関数を使用して、ライブラリ参照名やエンジンを作業フォルダなどのフォルダに割り当てることができます。このセクションの例では、LIBNAME ステートメントを使用します。LIBNAME 関数の詳細については、*SAS 関数と CALL ルーチン: リファレンス*を参照してください。

次に、LIBNAME ステートメントの基本的な構文を示します。

**LIBNAME** *libref <engine-name> 'SAS-data-library'*

このステートメントのすべての引数の説明については、*SAS ステートメント: リファレンス* を参照してください。

*注:* AUX、CON、NUL、LPT1 - LPT9、COM1 - COM9、PRN は Windows の予約語です。これらの予約語をライブラリ参照名として使用しないでください。

### *単一フォルダへのライブラリ参照名の割り当て*

SAS 9.4 データセットが C:\MYSASDIR フォルダに格納されている場合、次の LIBNAME ステートメントをサブミットすると、その格納フォルダにライブラリ参照名 TEST を割り当てることができます。

```
libname test V9 'c:\mysasdir';
```

このステートメントは、ライブラリ参照名 TEST からフォルダ C:\MYSASDIR に格納されている SAS 9.4 ファイルにアクセスすることを示します。エンジンの指定は任意です。

### *作業フォルダへのライブラリ参照名の割り当て*

作業フォルダは、メイン SAS ウィンドウのステータス行に表示されます。ライブラリ参照名 MYCURR を現在の SAS 作業フォルダに割り当てるには、次の LIBNAME ステートメントを使用します。

```
libname mycurr '.';
```

### *複数フォルダへのライブラリ参照名の割り当て*

SAS ファイルが複数のフォルダに格納されている場合、次に示すように、1 つのライブラリ参照名を指定し、フォルダの場所を連結すると、複数のフォルダを 1 つの SAS ライブラリとして扱うことができます。

```
libname income ('c:\revenue' 'd:\costs');
```

このステートメントは、2 つのフォルダ **C:\REVENUE** と **D:\COSTS** を 1 つの SAS ライブラリとして扱うことを示します。SAS ライブラリを連結する際、SAS ではライブラリへのアクセスに使用するプロトコルを、読み込み、書き出し、更新のいずれの目的でライブラリにアクセスするかによって使い分けます。

このステートメントは、ライブラリ参照名 INCOME と REVENUE で参照される 2 つのライブラリを、ライブラリ参照名が SALES の 1 つの SAS ライブラリとして扱うことを示します。

```
libname sales (income revenue);
```

このステートメントは、ライブラリ参照名 INCOME と REVENUE で参照される 2 つのライブラリを、ライブラリ参照名が SALES の 1 つの SAS ライブラリとして扱うことを示します。

詳細については、“複数のフォルダが格納された SAS ライブラリへのアクセス方法について” (132 ページ) を参照してください。

*注:* ライブラリ連結の概念は、SASHELP オプションや SASMSG オプションなどのシステムオプションを指定する場合にも適用されます。システムオプションを使用して複数のフォルダを指定する方法の詳細については、“SAS システムオプションでライブラリを連結する構文” (476 ページ) を参照してください。

### *エンジンの割り当て*

V9 Engine 以外のエンジンを使用する別のアクセスメソッドを使用する場合、LIBNAME ステートメントに別のエンジン名を指定できます。たとえば、SAS 9.4 のセッションでバージョン 6.12 の SAS データセットにのみアクセスする場合、次に示すように、LIBNAME ステートメントに V612 Engine を指定できます。

```
libname oldlib V612 'c:\sas612';
```

別の例として、Windows 版 SAS 9.4 と Windows 版バージョン 6 との間で SAS ファイルを共有する場合、SAS ライブラリにライブラリ参照名を割り当てるときに V6 Engine を使用します。次に、LIBNAME ステートメントに V6 Engine を指定する例を示します。

```
libname lib6 V6 'c:\sas6';
```

SAS 9.4 でバージョン 6 の SAS データセットを読み込むことはできますが、バージョン 6 で SAS 9.4 データセットを読み込むことはできません。SAS 9.4 データセットをバージョン 6 のデータセットに変更する方法については、**http:\\support.sas.com\migration\planning\files\regression.html** の移行に関する説明を参照してください。

LIBNAME ステートメントのエンジン名の詳細については、“Windows 版 SAS 9.4 で他のバージョンのファイルを使用する” (136 ページ) および “BMDP ファイル、OSIRIS ファイル、SPSS ファイルの読み取り” (141 ページ) を参照してください。また、LIBNAME ステートメントに関する説明を *SAS ステートメント: リファレンス*でも参照してください。

### *SAS 起動時にライブラリ参照名を使用可能にする*

SAS 起動時ごとに同じライブラリ参照名を割り当てるのではなく、SAS が毎回起動するライブラリ参照名を指定できます。**新規ライブラリ**ダイアログボックスにアクセスするには、**起動時に有効**を選択します。SAS が初期化されるとただちにライブラリ参照名が有効になります。起動時に有効化されるライブラリは、SAS レジストリのエントリ CORE\OPTIONS\LIBNAMES の下に格納されます。

### *フォルダへの複数のライブラリ参照名とエンジンの割り当て*

フォルダに複数のエンジンで作成した SAS ファイルが含まれる場合、指定したライブラリ参照名に割り当てられたエンジンで作成された SAS ファイルのみに、そのライブラリ参照名を使用してアクセスできます。V7、V8、V9 エンジンによって作成された SAS ファイルはすべてのエンジンにアクセスできますが、これらのエンジンは V6 エンジンで作成されたファイルにはアクセスできません。これらのファイルを表示するには、V6 エンジンを指定する必要があります。また、1 つのフォルダに、エンジンの異なる複数のライブラリ参照名を割り当てることができます。たとえば、次のステートメントは有効です。

```
libname one V6 'c:\mydir';
libname two V9 'c:\mydir';
```

ライブラリ参照名 ONE で参照されるデータセットが作成され、アクセスには互換エンジン(V8)が使用されます。一方、ライブラリ参照名 TWO で参照されるデータセットが作成され、アクセスにはデフォルトエンジン(V9)が使用されます。1 つの同じ SAS ライブラリに、(同一のエンジンを使用する)ライブラリ参照名を複数割り当てることもできます。たとえば、次の 2 つの LIBNAME ステートメントは、1 つの同じ SAS ライブラリにライブラリ参照名 MYLIB と INLIB (両方とも V9 Engine を使用)を割り当てます。

```
libname mylib V9 'c:\mydir\datasets';
libname inlib V9 'c:\mydir\datasets';
```

エンジン名と Windows パス名が同一であるため、ライブラリ参照名 MYLIB と INLIB は同一であり、入れ替えて使用することが可能です。

## 環境変数を使用した SAS ライブラリの割り当て

### 環境変数の種類

LIBNAME ステートメントや LIBNAME 関数ではなく、変数を使用してライブラリ参照名を割り当てることもできます。変数によって、ある文字列と別の文字列が Windows 環境内で結び付けられます。SAS では、次の 2 つの変数が認識されます。

- SAS 変数
- Windows 環境変数

SAS ステートメントでライブラリ参照名を使用する場合、SAS は、次の順序でライブラリ参照名の割り当てを確認します。

1. LIBNAME ステートメント、LIBNAME 関数、または**新規ライブラリ**ダイアログボックスを使用して割り当てられたライブラリ参照名(最後の割り当てが優先されます)
2. SAS 変数により割り当てられたライブラリ参照名
3. Windows 環境変数により割り当てられたライブラリ参照名

たとえば、Windows 環境変数 TEMP が C:\Windows\TEMP に割り当てられている場合に、次の LIBNAME ステートメントを使用したとします。

```
libname temp c:\public
```

これにより、LIBNAME が **c:\public** に解決されます。

変数を SAS に定義する方法は次の 2 つです。

- SET システムオプションを使用します。このオプションは SAS の(内部)変数を定義します。
- Windows の SET コマンドを発行します。このコマンドは Windows の(外部)変数を定義します。この他に Windows で、**システムのプロパティ**ダイアログボックスを使用して変数を定義することもできます。このダイアログボックスは、コントロールパネル、または **マイコンピュータ**を右クリックして表示されるメニューから、**プロパティ**を選択して開くことができます。

***注意:***

**エンジンを変数に割り当てることはできません。** 変数をライブラリ参照名として使用する場合は、デフォルトエンジンを使用する必要があります。

変数が使用できる場合は、SAS の起動前にリソースを簡単に割り当てることができます。

### SAS 環境変数をライブラリ参照名として使用する

SET システムオプションを使用して、SAS 変数を定義できます。たとえば、永久 SAS データセットが C:\SAS\MYSASDATA フォルダに格納されている場合に、次の SET オプションを SAS 起動時の SAS コマンドか SAS 構成ファイルに使用すると、変数 TEST をこの SAS ライブラリに割り当てることができます。

```
-set test c:\sas\mysasdata
```

変数を割り当てる際、SAS では変数名を実際に使用するまで参照が有効であるかどうかを確認しません。たとえば、変数 TEST が SAS 構成ファイルで定義されている場合、変数 TEST は SAS から参照されるまで確認されません。したがって、SET オプションの指定にフォルダ名のミススペルなどの間違いがあった場合、SAS ステートメントで変数が使用されるまでは、エラーメッセージは発行されません。

Windows のファイル名は、名前の一部にスペースや一重引用符を使用できるため、SET オプションを指定する際、物理パス名を二重引用符で囲む必要があります。OPTIONS ステートメント内で SET オプションを使用する場合も、ファイル名を引用符で囲む必要があります。SET システムオプションの完全な構文については、“SET システムオプション:Windows” (561 ページ) を参照してください。

変数名を SAS 構成ファイルのシステムオプションの値として使用する場合、使用する前に変数として定義する必要があります。たとえば、変数 TEST を使用する SASUSER システムオプションの前に、次の SET システムオプションを指定する必要があります。

```
-set test "d:\mysasdir"
-sasuser "!test"
```

次に、変数を連結ライブラリで使用する例を示します。

```
-set dir1 "c:\sas\base\sashelp"
-set dir2 "d:\sas\stat\sashelp"
-sashelp (!dir1 !dir2)
```

SAS 構成ファイルまたは SAS プログラムの LIBNAME ステートメントに変数を使用する場合、変数の前に感嘆符(!)を付ける必要があります。

Windows ショートカットを使用して SAS を起動する場合は、SAS 構成ファイルに SET システムオプションを使用することをお勧めします。

#### *Windows 環境変数の使用*

Windows の SET コマンドは、SAS を起動して Windows 環境変数を作成する前に実行できます。SAS を起動する前に変数を定義する必要があります。SAS セッション時に**コマンドプロンプト**ウィンドウから SAS の変数を定義することはできません。

変数は、SAS セッションを起動するコンテキストに割り当てられている場合のみ認識されます。Windows の起動時に実行される Windows の AUTOEXEC.BAT ファイル(これによりグローバル変数が作成されまする)に変数を定義するか、あるいは SAS を起動する**コマンドプロンプト**ウィンドウまたは**システムのプロパティ**ダイアログボックスで変数を定義する必要があります。

**コマンドプロンプト**ウィンドウで変数を定義した後で、**スタート**メニュー(または別のショートカット)から SAS を起動した場合、変数は認識されません。

SET コマンドを使用して定義した変数は、後で SAS 内でライブラリ参照名として使用されます。次の例では、Windows の SET コマンドを使用して、変数 PERM と BUDGET を定義しています。

```
SET PERM=C:\MYSASDIR
SET BUDGET=D:\SAS\BUDGET\DATA
```

### *ライブラリ参照名の割り当てのリストの表示*

#### *エクスプローラウィンドウを使用したライブラリ参照名のリストの表示*

SAS を対話的に実行している場合、**エクスプローラ**ウィンドウを使用してアクティブなライブラリ参照名を表示できます。**エクスプローラ**ウィンドウには、現在の SAS セッションで有効なすべてのライブラリ参照名と、各ライブラリ参照名に割り当てられたエンジンおよび物理パスが表示されます。ライブラリ参照名として定義した変数を SAS セッションで使用したことがある場合は、その変数がすべて表示されます。ライブラリ参照名として定義した変数を SAS プログラム内でまだ使用していない場合、その変数は、SAS の**エクスプローラ**ウィンドウに表示されません。

#### LIBNAME コマンドを使用したライブラリ参照名のリストの表示

SAS セッションで、LIBNAME コマンドを使用して **LIBNAME** ウィンドウを起動できます。**エクスプローラ**ウィンドウには、有効なライブラリ参照名のリストが表示されます。**LIBNAME** ウィンドウを使用して、すべてのライブラリの内容を表示することができます。

#### LIBNAME ステートメントを使用したライブラリ参照名のリストの表示

次の LIBNAME ステートメントは、アクティブなライブラリ参照名を SAS ログに書き込みます。

```
libname _all_ list;
```

### ライブラリ参照名のクリア

#### ライブラリ参照名のクリアの概要

ライブラリ参照名をクリアするには、次のいずれかを使用します。

- “SAS エクスプローラウィンドウ” (131 ページ)
- “LIBNAME ステートメント” (131 ページ)
- “LIBNAME 関数:Windows” (404 ページ)

ライブラリ参照名と参照先のライブラリとの関連付けは、ジョブまたはセッションの終了時に自動的にクリアされます。セッション中にライブラリ参照名を別の SAS ライブラリに割り当てる場合、セッションを終了したり、ライブラリ参照名をクリアしたりする必要はありません。ライブラリ参照名に新しいライブラリを指定すると、自動的にライブラリ参照名の割り当てが変更されます。

#### SAS エクスプローラウィンドウ

**エクスプローラ**ウィンドウを使用してライブラリ参照名をクリアするには

1. クリアするライブラリ参照名のノードを右クリックします。
2. **削除**を選択します。

**エクスプローラ**ウィンドウを使用してライブラリを管理する方法については、The Little SAS Book または SAS ヘルプとドキュメントを参照してください。

#### LIBNAME ウィンドウ

**LIBNAME** ウィンドウを使用してライブラリ参照名をクリアするには

1. コマンドバーで LIBNAME コマンドを発行します。**LIBNAME** ウィンドウが表示されます。
2. クリアするライブラリ参照名のノードを右クリックします。
3. **削除**を選択します。

#### LIBNAME ステートメント

LIBNAME ステートメントを使用してライブラリ参照名をクリアするには、次の構文を使用して LIBNAME ステートメントをサブミットします。

**LIBNAME** *libref*|_all_ <clear>;

ライブラリ参照名を指定すると、指定したライブラリ参照名のみがクリアされます。キーワード_all_を指定すると、現在の SAS セッション中に割り当てたすべてのライブラリ参

照名がクリアされます。MAPS、SASUSER、SASHELP、WORK の割り当ては解除されません。

*注:* 変数によって定義されたライブラリ参照名をクリアすると、変数の定義は残りますが、ライブラリ参照名とはみなされず、**エクスプローラ**ウィンドウに表示されません。この変数を別の LIBNAME ステートメントで使用すると、新たなライブラリ参照名が作成されます。

#### *LIBNAME 関数*

LIBNAME 関数を使用してライブラリ参照名をクリアするには、関数の引数としてそのライブラリ参照名のみを指定します。

```
libname(libref);
```

## *複数のフォルダが格納された SAS ライブラリへのアクセス方法について*

#### *フォルダアクセスのプロトコル*

連結機能を使用して、1 つのライブラリ参照名に複数の物理フォルダを指定する際、アクセスするフォルダを特定するため、次のプロトコルを使用します。

- 入力アクセス権と更新アクセス権
- 出力アクセス権
- 同じ名前のデータセットへのアクセス

次に示すプロトコルは、SAS ファイルにアクセスするすべての SAS ステートメントとプロシジャ(DATA ステップの DATA、UPDATE、MODIFY の各ステートメント、SQL プロシジャ、APPEND プロシジャなど)に適用されます。

#### *入力アクセス権と更新アクセス権*

SAS ファイルにアクセスして入力または更新する場合、名前を基準に検索された最初の SAS ファイルにアクセスします。たとえば、次のステートメントをサブミットし、両方のフォルダにファイル OLD.SPECIES が存在する場合、C:\MYSASDIR フォルダ内のファイルが出力されます。

```
libname old ('c:\mysasdir','d:\saslib');
proc print data=old.species;
run;
```

FSEDIT プロシジャを使用して、OLD.SPECIES を開き更新する場合にもこの方法が適用されます。

#### *出力アクセス権*

データセットにアクセスして出力する場合、データセットが存在する最初のフォルダにデータセットが書き込まれます。そのフォルダが存在しない場合は、エラーメッセージが表示されます。たとえば、次のステートメントをサブミットすると、OLD.SPECIES データセットが最初のフォルダ(C:\MYSASDIR)に書き込まれます。すでに同じ名前のデータセットがある場合は、置換されます。

```
libname old ('c:\mysasdir','d:\saslib');
data old.species;
   x=1;
   y=2;
run;
```

2 番目のフォルダに存在する OLD.SPECIES データセットのコピーは置換されません。

### 名前が同じデータセットへのアクセス

DATA ステートメントと SET ステートメントを使用して同じ名前のデータセットにアクセスする際、SAS ファイルのアクセスプロトコルに関して複雑な問題が発生する可能性があります。たとえば、次のステートメントをサブミットするとき、TEST.SPECIES はもともと 2 番目のフォルダ D:\MYSASDIR 内にのみ存在するとします。

```
libname test ('c:\sas','d:\mysasdir');
data test.species;
   set test.species;
   if value1='y' then
      value2=3;
run;
```

この場合、DATA ステートメントで TEST.SPECIES が開かれ、出力規則に従って出力されます。つまり、連結された最初のライブラリ(C:\SAS)にデータセットが開かれます。SET ステートメントでは、入力規則に基づいて、2 番目のフォルダ(D:\MYSASDIR)内にある既存の TEST.SPECIES データセットが開かれます。つまり、もとの TEST.SPECIES データセットは更新されません。そのうえ、それぞれのフォルダに 1 つずつ、2 つの TEST.SPECIES データセットが存在することになります。

## Sasuser データライブラリの更新

SAS では、ライブラリ参照名が Sasuser の SAS ライブラリが自動的に作成されます。このライブラリには、ユーザーのプロファイルカタログを始めとする SAS ファイルが含まれます。

Windows 版では、デフォルトで、SASUSER ライブラリ参照名は次のフォルダを参照します。C:\Users*user ID*\Documents\My SAS Files\9.4

SASUSER システムオプションを使用して、SASUSER ライブラリ参照名の参照先として別の SAS ライブラリに指定できます。SASUSER フォルダが存在しない場合は、SAS によって作成されます。デフォルト以外のフォルダを使用する場合は、sasv9.cfg 構成ファイルに SASUSER システムオプションを追加します。

SASUSER フォルダには、プロファイルカタログ以外のファイルも格納されます。たとえば、サンプルデータセットもこのフォルダに格納されます。

SASUSER データライブラリは、V9 Engine に割り当てられます。SASUSER データライブラリに割り当てられているエンジンを変更することはできません。このデータライブラリに別のエンジンを割り当てようとすると、エラーメッセージが表示されます。ENGINE システムオプションを別のエンジンに設定している場合でも、SASUSER データライブラリに作成された SAS ファイルは SAS 9.4 ファイルになります。

プロファイルカタログの詳細については、“Profile カタログ ” (22 ページ) を参照してください。SASUSER システムオプションの詳細は、“SASUSER システムオプション:Windows” (559 ページ) を参照してください。

## WORK データライブラリの使用

### 一時ファイルの使用

WORK データライブラリは、一時 SAS ファイルの格納場所です。Windows 版の場合、デフォルトでは WORK データライブラリは!TEMP\SAS Temporary Files フォルダのサブフォルダとして作成されます。このサブフォルダには_TD*nnnnnnnnn_nodename*_という名前が付けられます。詳細については、“Work データライブラリ ” (23 ページ) を参照してください。一時 SAS ファイルは、そのファイルが作成された SAS セッションの間しか使用できません。一時 SAS ファイルは、セッションの終わりに自動的に削除されます。SAS が異常終了した場合、一時ファイルを削除する必要があります。

デフォルトでは、2 レベル名を割り当てられていないファイルは、自動的に一時ファイルとみなされます。作成されたどの一時 SAS データセットにも、自動的に、特殊なライブラリ参照名 WORK が割り当てられます。たとえば、次の DATA ステップを実行してデータセット SPORTS を作成する場合、WORK.SPORTS という名前の一時データセットが作成されます。

```
data sports;
   input @1 sport $10. @12 event $20.;
   datalines;
volleyball co-recreational
swimming   100-meter freestyle
soccer     team
;
```

**エクスプローラ**ウィンドウを表示すると、Work フォルダに Sports データセットを確認できます。

セッション中に次の場所で作成されたすべての一時データセットを表示できます。

- **エクスプローラ**ウィンドウ。ライブラリフォルダのアイコン、Work フォルダのアイコンの順にダブルクリックします。
- **LIBNAME** ウィンドウ。コマンドバーに「**libname**」と入力し、Work フォルダアイコンをダブルクリックします。

WORK データライブラリは、V9 Engine に割り当てられます。WORK データライブラリに割り当てられているエンジンを変更することはできません。このデータライブラリに別のエンジンを割り当てようとすると、エラーメッセージが表示されます。ENGINE システムオプションを別のエンジンに設定している場合でも、WORK データライブラリに作成された SAS ファイルは SAS 9.4 ファイルになります。

### *環境変数の使用*

前述した SASUSER システムオプションでの方法と同様に、Work データライブラリの指定に変数を使用できます。WORK データライブラリのデフォルトの場所を使用しない場合は、この方法を使用します。SAS 構成ファイル内に次のようなオプションを追加して、WORK データライブラリ用の変数を設定します。

```
-set myvar c:\ tempdir
-work !myvar
```

SET オプションは、MYVAR 変数を C:\TEMPDIR フォルダに関連付けます。次に WORK オプションによって、WORK データライブラリにこのフォルダを使用することが指定されます。一時フォルダとその中のファイルは、SAS セッション終了時に削除されます。

### *ユーザーライブラリ参照名の使用*

デフォルトでは、1 レベル名の SAS ファイルは一時ファイルであるため、SAS セッションの終了時に削除されます。しかし、USER ライブラリ参照名を使用すると、1 レベル名の SAS ファイルを永久 SAS ライブラリに格納できます。たとえば、次のステートメントを発行すると、1 レベル名の SAS ファイルがすべて C:\MYSASDIR フォルダに永久保存されます。

```
libname user 'c:\mysasdir';
```

この例のように USER ライブラリ参照名をフォルダに設定し、一時データセットを作成してアクセスできるようにする際、ライブラリ参照名として WORK を使用して、データセットに 2 レベル名を指定する必要があります。

SAS 起動時に USER システムオプションを使用するか、または Windows 環境変数 USER を作成して、USER ライブラリ参照名を割り当てることもできます。USER という

名前の Windows 環境変数がある場合は、SAS 起動時に USER ライブラリ参照名が自動的に割り当てられます。USER システムオプションの詳細については、"USER システムオプション:Windows" (585 ページ) および *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。

*注:* 1 レベル名で保存されたデータセットを他のリリースの SAS で使用可能な形式で格納する場合は、USER ライブラリ参照名に他のエンジンを割り当てることができます。

### *Windows (NTFS)における大きいデータセットの使用*

Windows NT File System (NTFS)を使用して Windows 版 SAS を実行する場合、自動的に 64 ビットファイルの I/O 機能が利用されます。NTFS の使用時の物理ボリュームと論理ボリュームの実際の上限は、2 テラバイトです。

## 複数の SAS セッションから SAS ファイルにアクセスする

単一のマシンあるいはネットワークから複数の SAS セッションを実行している場合、複数のセッションから 1 つの SAS ファイルにアクセスし、読み込むことができます。

SAS/SHARE がインストールされている場合、**VIEWTABLE** ウィンドウ、FSEDIT ウィンドウ、FSVIEW ウィンドウでは、複数のユーザーが 1 つの SAS ファイルを編集できます。**VIEWTABLE** ウィンドウを使用してデータセットを編集する場合は、編集モードをテーブルレベルの編集アクセスまたは行レベルの編集アクセスに設定できます。**テーブルレベルの編集アクセス**を選択すると、そのユーザーだけがデータセットにアクセスできます。行レベルの編集アクセスの場合は、複数のユーザーが同じ SAS ファイルにアクセスできます。ただし、一度に 1 人のユーザーだけが、ある 1 つのレコード(オブザベーション)にアクセスしてそのレコードを変更できます。

**VIEWTABLE** ウィンドウでデータセットを開くには、**エクスプローラ**ウィンドウで次のいずれかの操作を行います。

1. ライブラリアイコンをダブルクリックします。
2. データセットが含まれるライブラリをダブルクリックします。
3. データセットをダブルクリックします。

データセットを編集するには、**編集** ⇨ **編集モード**を選択し、**テーブルレベルの編集**または**行レベルの編集**を選択します。

FSEDIT または FSVIEW を使用してデータセットを編集すると、MEMBER または RECORD に更新モードを設定できます。設定したのが **MEMBER** モードの場合は、そのユーザーだけがデータセットにアクセスできます。設定したのが **RECORD** モードの場合は、複数のユーザーが同じ SAS ファイルに書き出すことができます。ただし、一度に 1 人のユーザーだけが、1 つのレコード(オブザベーション)を更新できます。FSEDIT または FSVIEW を使用してデータセットを開くには、次のいずれかの操作を行います。

FSEDIT または FSVIEW を使用してデータセットを開くには、次のいずれかの操作を行います。

1. コマンドバーに、FSEDIT または FSVIEW と入力します。
2. ライブラリ名を**メンバの選択**ダイアログボックスでダブルクリックします。
3. データセット名をダブルクリックします。

データセットを編集するには、**編集** ⇨ **更新**を選択し、**MEMBER** ラジオボタンまたは **RECORD** ラジオボタンのどちらかを選択します。

RSASUSER システムオプション( “RSASUSER システムオプション:Windows” (553 ページ) を参照)は、SASUSER データライブラリの共有を許可します。複数のユーザーが共通の SAS データセットに更新アクセスを行う必要がある場合は、SAS/SHARE を使用します。

複数のユーザーが 1 つのデータセットとそのメンバにアクセスする際の規則については、SAS ヘルプとドキュメントおよび *SAS/SHARE User's Guide* を参照してください。

# Windows 版 SAS 9.4 で他のバージョンのファイルを使用する

## *Windows 版 SAS 9.4 での他のバージョンのファイルの使用について*

バージョン 8、7、6 で作成された SAS ファイルを処理する場合、一部の例外を除いて、SAS 9.4 形式に変換する必要はありません。

SAS 9.4 ファイル形式は、いくつかの新機能を除き、バージョン 7 と 8 のファイル形式と同じです。表 4.4 (136 ページ) は、前のリリースの SAS ファイルを使用するために必要な操作をまとめたものです。このとき、SAS ライブラリには、同じリリースの SAS ファイルが含まれています。

SAS 9.4 を使用して、Windows 版 SAS のこれまでのリリースで作成されたカタログにアクセスする場合、SAS 9.4 プログラムでカタログを使用する前に、これまでのリリースから SAS 9.4 形式にカタログを変換する必要があります。

次の表に、データセットとカタログ変換の情報を示します。

***表 4.4*** *SAS 9.4 におけるバージョン 6、7、8 のデータセットとカタログの使用の概要*

| バージョンまたはリリース | データセット | カタログ |
|---|---|---|
| バージョン 7 と 8 | 32 ビット SAS の場合、操作の必要はありません。SAS から、バージョン 7 および 8 のデータセットの読み込み、更新、書き込みができます。<br>64 ビット SAS の場合、読み込みと書き込みについての更新はありません。操作の必要はありません。 | 32 ビット SAS の場合、操作の必要はありません。SAS から、バージョン 7 および 8 のカタログの読み込み、更新、書き込みができます。<br>64 ビット SAS の場合は、MIGRATE プロシジャに SLIBREF オプションを指定して、SAS 9 に移行します。 |

| バージョンまたはリリース | データセット | カタログ |
|---|---|---|
| リリース 6.08 - 6.12 | V6 Engine が自動的に検出されます。32 ビット SAS の場合、操作の必要はありません。SAS から、バージョン 6 のデータセットの読み込み、更新、書き込みができます。<br>64 ビット SAS の場合、SAS から、V6 データセットを読み込めますが、書き込めません。 | 32 ビット SAS の場合、SAS から、バージョン 6 のカタログを読み込めますが、書き込めません。<br>64 ビット SAS の場合は、CPORT プロシジャと CIMPORT プロシジャを使用して SAS 9 に変換します。 |
| リリース 6.03 と 6.04 | V604 Engine を使用して、データを読み込みます。リリース 6.03 と 6.04 のデータセットには書き込めません。 | 未サポート |

この表に示されているように、32 ビット SAS の場合は、リリース 6.04 および 6.03 のデータセットを除き、バージョン 6 (32 ビット SAS)、バージョン 7 および 8 のデータセットを SAS 9.4 で読み込み、更新し、書き込めるようにするために SAS 9.4 のデータセットに変換する必要はありません。64 ビット SAS では、Cross-Environment Data Access (CEDA)機能による制限があります。

32 ビット SAS では、バージョン 7 と 8 のカタログも V9 カタログに変換する必要はありません。バージョン 6 の SAS カタログは、読み込み可能ですが、更新できません。バージョン 6 のカタログを更新するには、SAS 9.4 カタログに変換する必要があります。

移行に焦点を当てたページ(**http://support.sas.com/migration**)では、リリース 6.08 からバージョン 8 までで作成された SAS ファイルの使用または変換方法について詳しく説明しています。リモートライブラリサービスを使用してサーバー上の SAS ファイルにアクセスしている場合に、バージョン 6 の SAS ファイルにアクセスする方法については、*SAS/CONNECT User's Guide* を参照してください。

Windows 以外の動作環境で作成された SAS ファイルを使用するには、Windows 環境にファイルを移送する必要があります。動作環境間でのファイルの移送に関する詳細については、*SAS ファイルの移動とアクセス*を参照してください。

### *リリース 6.08 からリリース 8.2 までのデータセットの使用*

リリース 6.12 やバージョン 8 など、同じリリースの SAS で作成された SAS ファイルのみが SAS ライブラリに含まれる場合、これらの SAS データセットに使用するエンジンは自動的に決定されます。SAS ファイルが、複数リリースの SAS データセットが存在する混合モードライブラリに含まれる場合、LIBNAME ステートメントにエンジンパラメータを指定する必要があります。デフォルトエンジンは V9 です。

たとえば、'c:\mydata' SAS ライブラリにバージョン 6 のファイルのみが含まれることが判明している場合、次の SAS ステートメントは、Windows 版で作成された WINDATA.SALEFIGS という名前のバージョン 6 の SAS データセットを印刷します:

```
libname windata 'c:\mydata';
proc print data=windata.salefigs;
   title 'Sales Figures';
run;
```

ライブラリ内の SAS ファイルがすべてバージョン 6 のファイルである場合は、自動的に V6 Engine が検出されるため、エンジンパラメータは省略できます。

同じ例で、ファイルのバージョンが不明である、または SAS ライブラリが混合モードライブラリである場合を想定します。この場合、V6 ファイルにアクセスするには、LIBNAME ステートメントにエンジン名を指定する必要があります。

```
libname windata v6 'c:\mydata';
proc print data=windata.salefigs;
   title 'Sales Figures';
run;
```

リリース 6.03 とリリース 6.04 の SAS ファイルには、特定のエンジンが必要です。詳細については、“リリース 6.03 とリリース 6.04 の SAS データセットの使用” (138 ページ)を参照してください。

### リリース 6.03 とリリース 6.04 の SAS データセットの使用

V604 Engine を使用すると、32 ビット Windows 版 SAS 9.4 のセッションから直接、リリース 6.03 とリリース 6.04 の SAS データセットを読み込むことができます。リリース 6.03 とリリース 6.04 の SAS データセットは、x64 64 ビット SAS と互換性がありません。(リリース 6.04 とリリース 6.03 の SAS データセットには違いがないことに注意してください。)この機能は、PC 版リリース 6.04 と Windows 版 SAS 9.4 の間で SAS データセットを共有する場合に便利です。V604 Engine は、SAS データセット(メンバタイプ DATA)のみサポートします。たとえば、リリース 6.04 の SAS データセット MYLIB.FRUIT を印刷する場合、SAS 9.4 セッションから次のステートメントをサブミットします。

```
libname mylib v604 'c:\sas604';
proc print data=mylib.fruit;
run;
```

### リリース 6.08 からリリース 6.12 までの SAS データセットの変換

バージョン 6 のデータセットに頻繁にアクセスし、バージョン 6 からファイルを読み込む必要がない場合は、この SAS データセットを、SAS 9.4 形式に変換することを検討してください。SAS 9.4 のデータセット形式は、バージョン 6 形式より効率的です。また、データセットを変換しないと、SAS 9.4 の新機能を使用できません。リリース 6.12 のライブラリを移行するには、MIGRATE プロシジャを使用してください。6.12 より前のライブラリの変換方法の詳細については、*SAS ファイルの移動とアクセス*を参照してください。

*注:* 変換の詳細については、移行に焦点を当てたページ(**http://support.sas.com/migration**.

### SAS 9.4 でバージョン 7 と 8 の SAS カタログを使用する

SAS 9.4 ファイル形式は、バージョン 7 と 8 のファイル形式と基本的に同じであるため、32 ビットプラットフォームから、バージョン 7 と 8 のカタログを読み込み、更新、書き込みできます。SAS 9.4 のカタログに変換する必要はありません。

ただし、64 ビットの Windows 動作環境で実行されている SAS では、32 ビットのカタログは読み込めません。

### SAS 9.4 でバージョン 6 の SAS カタログを変換する

動作環境の内部構造が異なるため、CPORT プロシジャと CIMPORT プロシジャを使用して、Windows 版で作成されたバージョン 6 の SAS カタログを SAS 9.4 形式に変換

しないと、Windows 版 SAS 9.4 のセッションでカタログを使用できません。変換手順は次のとおりです。

1. バージョン 6 の SAS セッションで CPORT プロシジャを使用して、変換する SAS カタログが含まれる移送ファイルを作成します。
2. このファイルを、ネットワークやディスケットなどを使用して、SAS 9.4 セッションで読み込み可能な場所に転送します。
3. 9.4 CIMPORT プロシジャを使用して、移送ファイルを読み込み、変換された SAS カタログを作成します。

CPORT プロシジャと CIMPORT プロシジャに関する詳細については、*SAS ファイルの移動とアクセス*および *Base SAS プロシジャガイド*を参照してください。

変換方法は他にもあります。詳細については、*SAS ファイルの移動とアクセス*を参照してください。

### *リリース 6.08 の SAS カタログを SAS 9.4 に変換する*

リリース 6.08 から SAS 9.4 に直接変換する場合、リリース 6.08 の CPORT プロシジャを使用して移送ファイルを作成してから、SAS 9.4 の CIMPORT プロシジャを使用して SAS 9.4 カタログに変換します。ただし、リリース 6.08 のセッションから CPORT プロシジャを使用する場合、HSERVICE および TOOLBOX カタログエントリは移送できません。

リリース 6.08 のカタログの変換方法には、リリース 6.10 からリリース 6.12 まで提供されている C16PORT プロシジャを使用する方法もあります。C16PORT プロシジャは、Windows 版 SAS 6.08 で作成された 16 ビットのカタログを、SAS で使用可能な 32 ビット形式に変換します。SAS のこれまでのリリースのいずれかで C16PORT プロシジャを使用して、後で SAS 9.4 から読み込み可能なカタログを作成できます。(C16PORT プロシジャは SAS 9.4 では使用できません)

Windows 版 SAS 6.08 から SAS 9.4 に SAS カタログを変換するには、次の操作を行います。

1. リリース 6.10、リリース 6.11、リリース 6.12 のセッション中に、C16PORT プロシジャを使用して、リリース 6.08 の SAS カタログを含む移送ファイルを作成します。C16PORT プロシジャの詳細は、各リリースのドキュメントを参照してください。
2. このファイルを、ネットワークやバイナリ FTP などを使用して、SAS 9.4 で読み込み可能な場所に転送します。
3. 9.4 CIMPORT プロシジャを使用して、移送ファイルを読み込み、変換された SAS V9.4 カタログを作成します。

Windows 版 SAS 6.08 を実行している別のマシン上にあるカタログを変換するには、最初にファイルを、ネットワークやバイナリ FTP などを使用して、SAS 9.4 のセッションで読み込み可能な場所に転送する必要があります。

次の例では、リリース 6.12 で C16PORT プロシジャを使用して INLIB.CAT カタログから移送ファイルを作成し、次に CIMPORT プロシジャを使用してリリース 6.12 のカタログ(OUTLIB.CAT)を作成します。

```
   /* Folder where catalog    */
   /* 'cat.sc2' resides       */
libname inlib 'c:\cat608';
   /* Folder where catalog    */
   /* 'cat.sc8' will reside   */
libname outlib 'c:\cat612';
proc c16port file='transprt' c=inlib.cat;
```

```
run;

/* Move the transport file to a location where SAS can read it */
/* Once the file is accessible, run the following procedure.  */

proc cimport infile='transprt' c=outlib.cat;
run;
```

これで、SAS 9.4 からリリース 6.12 の SAS カタログを読み込めるようになります。CPORT プロシジャと CIMPORT プロシジャに関する詳細については、*Base SAS プロシジャガイド*および *SAS ファイルの移動とアクセス*を参照してください。

### *リリース 6.03 およびリリース 6.04 の SAS カタログを SAS 9.4 に変換する*

リリース 6.04 の SAS カタログを SAS 9.4 カタログに変換する場合は、*SAS ファイルの移動とアクセス*を参照してください。

### *リリース 6.08 からリリース 6.12 までのデータセットの作成*

Windows 版 SAS のセッションで、リリース 6.08 からリリース 6.12 のデータセットを作成する場合があります。作成方法は、V6 Engine を使用してバージョン 6 のデータセットを読み込む方法と類似しています。たとえば、次の SAS ステートメントは、V6 Engine を使用して、QTR1 という名前の SAS データセットを作成します。ファイル参照名 MYFILE に割り当てられている外部ファイルの生のデータを読み込みます。

```
libname windata v6 'c:\mydata';
filename myfile 'c:\qtr1data.dat';
data windata.qtr1;
   infile myfile;
   input saledate amount;
run;
```

## SAS 9.4 のファイルを以前のリリースで使用する

SAS ファイルを以前のリリースに戻す場合、CPORT プロシジャと CIMPORT プロシジャは使用しないでください。移送ファイルをインポートしようとすると、エラーが発生します。ファイルの移送の詳細については、*SAS ファイルの移動とアクセス*を参照してください。

32 ビットの Windows 版プラットフォームで作成された SAS 9 ファイルは、バージョン 7 または 8 のセッションでは、バージョン 7 または 8 ではサポートされていない SAS 9 ファイルの新機能を使用しない限り、完全に互換性があります。SAS 9 ファイルは、バージョン 6 ではサポートされませんが、バージョン 6 出力形式に戻してから使用できます。

これらの互換性の問題および SAS 9 ファイルを以前のリリースに戻す最良の方法については、**http://support.sas.com/migration/planning/files/regression.html** の移行に焦点を当てたページで「旧バージョンの SAS での SAS 9 ファイルの使用」を参照してください。

# SAS 9.4 でリモートホストの SAS ファイルを使用する

SAS 9.4 では、以前のバージョンの SAS から、リモートホスト上に作成した SAS 9.4 データセットに直接アクセスできます。たとえば、VSE 上にバージョン 5 のデータセット、または UNIX 上にバージョン 8 のデータセットに存在することがあります。直接アクセスするかわりに、移送データセットを作成し、ホストから Windows にファイルを移送することも可能です。

SAS 9.4 でリモートホストの SAS ファイルを使用する詳しい方法については、*SAS ファイルの移動とアクセス*を参照してください。リリース間の互換性については、`http://support.sas.com/migration` の移行に焦点を当てたページを参照してください。

# BMDP ファイル、OSIRIS ファイル、SPSS ファイルの読み取り

## *概要:BMDP ファイル、OSIRIS ファイル、SPSS ファイルの読み取り*

SAS 9.4 には、SAS プログラムから直接外部データファイルにアクセスできる 3 つのインターフェイスライブラリエンジン(BMDP Engine、OSIRIS Engine、SPSS Engine)が搭載されています。これらのエンジンは、すべて読み込み専用です。これらのエンジンは、データのランダムアクセスをサポートしないシーケンシャルエンジンのため、SET ステートメント内の POINT=システムオプションとともに使用することも、FSBROWSE、FSEDIT、FSVIEW プロシジャとともに使用することもできません。BMDP Engine および OSIRIS Engine を使用する際は、COPY プロシジャまたは DATA ステップを使用してシステムファイルを SAS データセットにコピーし、その SAS データセットに対してこれらの関数を実行します。SPSS Engine を使用する際は、COPY プロシジャや DATA ステップでポータブルファイル形式を使用できます。また、PRINT プロシジャなどのいくつかのプロシジャでは、エンジンがシーケンシャルである、という警告メッセージが表示されます。これらのエンジンでは、ライブラリ参照名に割り当てられている物理ファイル名は、フォルダではなく実際のファイル名です。ライブラリ参照名規則では、このような割り当ては例外的です。

CONVERT プロシジャを使用して、BMDP、OSIRIS、SPSS の各ファイルを SAS データファイルに変換することもできます。詳細については、“CONVERT プロシジャ:Windows” (428 ページ) を参照してください。


- “BMDP Engine” (141 ページ)
- “OSIRIS Engine” (142 ページ)
- “SPSS Engine” (143 ページ)


## *BMDP Engine*

### *BMDP Engine の概要*

BMDP インターフェイスライブラリエンジンを使用すると、SAS プログラムから BMDP 統計ソフトウェアアプリケーションの BMDP DOS ファイルを直接読み込むことができます。次のセクションは、BMDP 保存ファイルの用語の知識があることを前提にしています。

BMDP 保存ファイルを読み込むには、BMDP Engine の使用を明示的に指定する次の LIBNAME ステートメントを発行する必要があります。

LIBNAME *libref* BMDP <'filename'>;

この形式の LIBNAME ステートメントでは、*libref* は SAS ライブラリ参照名、*filename* は BMDP 物理ファイル名です。このライブラリ参照名がすでにファイル参照名として使用されている場合、ファイル参照名に割り当てられている物理ファイル名が使用されるため、*filename* を省略できます。このエンジンは、DOS で作成された BMDP 保存ファイルのみを読み込めます。

1 つの物理ファイルに複数の保存ファイルが格納されるため、CODE=の値を SAS 言語のデータセットのメンバ名として参照します。たとえば、保存ファイルに CODE=ABC および CODE=DEF が含まれ、ライブラリ参照名が MYLIB の場合、これらの CODE= を MYLIB.ABC および MYLIB.DEF として参照できます。CONTENT の場合、その種類に関わらず参照される方法は同一です。したがって、メンバ DEF が CONTENT=CORR でも、CONTENT=DATA として参照されます。

物理ファイルの最初の保存ファイルにアクセスする場合、または保存ファイルが 1 つしか存在しない場合、メンバ名を_FIRST_として参照できます。この方法は、CODE=の値がわからない場合に便利です。

### *BMDP Engine の例*

次の例では、物理ファイル MYBMDP.DAT に保存ファイル ABC が格納されています。この例では、BMDP 物理ファイルにライブラリ参照名 MYLIB を割り当ててから、この保存ファイルに対して CONTENTS プロシジャと PRINT プロシジャを実行しています。

```
libname mylib bmdp 'mybmdp.dat';
proc contents data=mylib.abc;
run;
proc print data=mylib.abc;
run;
```

次の例では、LIBNAME ステートメントを使用して、ライブラリ参照名 MYLIB2 を BMDP 物理ファイルに割り当てます。その後、その物理ファイル内にある最初の保存ファイルのデータを出力しています。

```
libname mylib2 bmdp 'mybmdp.dat';
proc print dat=mylib2._first_;
run;
```

## *OSIRIS Engine*

### *OSIRIS Engine の概要*

ICPSR (Inter-University Consortium on Policy and Social Research)では、データファイルの配信に OSIRIS ファイル形式を使用しています。このため、SAS では OSIRIS インターフェイスライブラリエンジンを提供することで、ICPSR データユーザーをサポートし、CONVERT プロシジャとの互換性を実現しています。この説明については、“CONVERT プロシジャ:Windows” (428 ページ) を参照してください。

読み取り専用の OSIRIS Engine を使用すると、SAS プログラムから OSIRIS データとディクショナリファイルを直接読み込むことができます。これらのファイルは、EBCDIC 形式でなければなりません。つまり、OSIRIS ファイルはホストコンピュータからバイナリ形式でダウンロードする必要があります。次のセクションは、OSIRIS ファイルの用語の知識があることを前提にしています。[1]

OSIRIS ファイルを読み込むには、OSIRIS Engine の使用を明示的に指定する LIBNAME ステートメントを実行する必要があります。この場合の LIBNAME ステートメントの形式は、次のとおりです。

**LIBNAME** *libref* OSIRIS '*data-filename*' DICT='*dictionary-filename*';

この形式の LIBNAME ステートメントでは、*libref* は SAS ライブラリ参照名、*data-filename* は OSIRIS データファイルの物理ファイル名、*dictionary-filename* は OSIRIS ディクショナリファイルの物理ファイル名です。*dictionary-filename* 引数には、変数名またはファイル参照名を指定することもできます変数名またはファイル参照名を指定する場合は、引用符を使用しないでください。このエンジンにはデータファイルとディクショナリファイルの両方が必要なので、DICT=オプションを指定する必要があります。

OSIRIS データファイルには、メンバ名は存在しません。したがって、どのようなメンバ名でも使用できます。異なる OSIRIS データファイルに同じ OSIRIS ディクショナリファイルを使用できます。ファイルごとに LIBNAME ステートメントを記述します。

OSIRIS データディクショナリのレイアウトは、動作環境間で同一です。これは、OSIRIS ソフトウェアは z/OS 環境以外では実行できませんが、エンジンは他の SAS 実行環境で z/OS データディクショナリを受け入れるように設計されているためです。OSIRIS ディクショナリファイルおよびデータファイルは、EBCDIC から ASCII に変換されないことに注意してください。OSIRIS エンジンでは、EBCDIC データが要求されます。OSIRIS データファイルに特定のファイルレイアウトはありません。ファイルレイアウトは、OSIRIS ディクショナリファイルの内容によって指定されます。

#### *OSIRIS Engine の例*

次の例では、データファイルは MYOSIRIS.DAT、ディクショナリファイルは MYOSIRIS.DIC です。この例では、OSIRIS ファイルにライブラリ参照名 MYLIB を割り当ててから、このデータに対して CONTENTS プロシジャと PRINT プロシジャを実行しています。

```
libname mylib osiris 'myosiris.dat'
        dict='myosiris.dic';
proc contents data=mylib._first_;
run;
proc print data=mylib._first_;
run;
```

### *SPSS Engine*

#### *SPSS Engine の概要*

SPSS インターフェイスライブラリエンジンを使用すると、SAS プログラムから SPSS エクスポートファイルを直接読み込めます。SPSS エクスポートファイルは、SPSS EXPORT コマンドを使用して作成する必要があります。[1]SPSS Engine は、読み取り専用エンジンです。

SPSS エクスポートファイルを読み込むには、SPSS Engine の使用を明示的に指定する LIBNAME ステートメントを実行する必要があります。この場合の LIBNAME ステートメントの形式は、次のとおりです。

**LIBNAME** *libref* SPSS <'*filename*'>;

この形式の LIBNAME ステートメントの *libref* 引数は SAS ライブラリ参照名、*filename* はファイル拡張子付きの SPSS 物理ファイル名です。ライブラリ参照名がファイル参照

1 詳細については、Institute for Social Research から提供されているドキュメントを参照してください。

1 詳細については、SPSS Inc.から提供されているドキュメントを参照してください。

名としても指定されている場合、ファイル参照名に割り当てられている物理ファイル名が使用されるため、*filename* を省略できます。SPSS ネイティブファイル形式はサポートされていません。エクスポートファイルは、いずれの動作環境からでも作成できます。

SPSS ファイルには内部名がないため、任意のメンバ名を使用して参照できます。次の例では、_FIRST_を使用しています。

*注:* SPSS には、システム欠損データとユーザー定義の欠損データを含めることができます。SPSS Engine や PROC CONVERT を使用する際、欠損値(ユーザー定義またはシステム)は、システム欠損値に変換されます。ユーザー定義の欠損値を、有効な値として記録する必要があります。データセットの変換には、PROC FORMAT を使用できます(–1 から.A へ、–2 から.B へなど)。

### *SPSS Engine の例*

次の例では、物理ファイル MYSPSS.POR にライブラリ参照名 MYLIB を割り当ててから、CONTENTS プロシジャと PRINT プロシジャをポータブルファイルで実行します。

```
libname mylib spss 'myspss.por';
proc contents data=mylib._first_;
run;
proc print data=mylib._first_;
run;
```

### *SPSS ファイルの再フォーマット*

SAS では、変数全体の幅より小数点以下の桁数が多い数値形式の変数を含む SPSS ファイルを使用できません。たとえば、SPSS ファイルに幅が 17 で、小数点以下の桁数が 35 の変数がある場合、ファイルに対して DATA ステップを実行したり、テーブルビューアでそのファイルを表示したりするとエラーが発生します。SAS で SPSS ファイルを使用するには、変数を再フォーマットする必要があります。

変数の幅に収まる値に小数点以下の桁数を減らすことで、変数を再フォーマットできます。次のコードでは、ステートメント **revision=cat(format,formatl,'.2');** で、小数点以下の桁数を 2 に変換しています。この値は小数点以下の桁数を減らし、変数の幅を超えないようにします。

```
libname abc spss 'FILENAME.POR'; proc contents data=abc._all_ out=new; run; filename sas
```

*注:* OPTIONS NOFMTERR ステートメントでは、DATA ステップやテーブルビューアでデータセットは使用できません。変数の幅より小数点以下の桁数が多い数値変数を再フォーマットしないと、DATA ステップやテーブルビューを使用できません。

## SAS ファイルを動作環境間で転送する

SAS ファイルの動作環境間での転送についての詳細については、*SAS ファイルの移動とアクセス*を参照してください。

## SAS/ACCESS を使用し、データベースファイルにアクセスする

SAS/ACCESS は、SAS と、Windows 環境で実行する複数のデータベース管理システム(DBMS)とのインターフェイスを提供します。このインターフェイスは、3 つのプロシジ

ャと 1 つのインターフェイスビューエンジンで構成されています。これらは、次のタスクを実行できます。

LIBNAME ステートメント
エンジンを特定のデータベースエンジンに割り当て、LIBNAME ステートメントを使用して DATA ステップや SAS プロシジャで直接 DBMS オブジェクトを参照するため、DBMS オブジェクトを SAS データセットのように読み書きできます。

SQL プロシジャのパススルー機能
Oracle サーバーや SQL サーバーなどの複数の異なるリレーショナル DBMS のデータにアクセスします。

インターフェイスビューエンジン
SAS プログラムにディスクリプタファイルを使用できるため、DBMS データへ直接アクセスできます。また、SAS プログラムにディスクリプタファイルを指定できるため、DBMS データを直接更新、挿入、削除できます。

Windows 版 SAS/ACCESS の詳細については、*SAS/ACCESS Interface to PC Files: Reference* およびその他の入手可能な SAS/ACCESS ドキュメントを参照してください。

# SAS ODBC Driver を使用し、他のアプリケーションから SAS データにアクセスする

SAS ODBC Driver には、ODBC (Open Database Connectivity)規格が採用されています。このドライバを使用すると、SAS データソースに対してアクセス、処理、更新を行うことができます。SAS データソースとは、SAS データセット、フラットファイル、VSAM ファイル、SAS/ACCESS がライセンスされている任意のデータベース管理システム(DBMS)のデータなどです。ODBC 規格に準拠する他の Windows アプリケーションのデータへのアクセス方法については、『SAS ヘルプとドキュメント』を参照してください。

SAS ODBC Driver では、データにアクセスするために、TCP/IP プロトコルを使用して、ローカルまたはリモート(SAS/SHARE) SAS Server のセッションとの通信を確立します。TCP/IP プロトコルを使用して、さまざまなホストのプラットフォーム上のリモート SAS Server にアクセスできます。SAS Server とは、独自の SAS セッションで実行される SAS プロシジャ(PROC SERVER または PROC ODBCSERV)です。ODBC 準拠のアプリケーションの代わりに、他の SAS セッションや SAS ODBC ドライバからの入出力要求を受け入れます。SAS データにリモートアクセスを行う場合は、サーバーコンピュータ上に SAS Server がインストールされていなければなりません。ただしクライアントコンピュータ上にインストールされている必要はありません。

SAS ODBC Driver は、Base SAS に同梱されています。SAS ODBC Driver を使用するリモートサーバー構成では、次の SAS プロダクトがインストールされている必要があります。

- Base SAS
- SAS/SHARE

SAS ODBC Driver のインストールと構成方法の詳細については、Windows 版 SAS のインストールドキュメントを参照してください。SAS ODBC Driver の構成方法の詳細については、*SAS Drivers for ODBC: User's Guide* を参照してください。

*5 章*

# Windows 版での外部ファイルの使用

**外部ファイルについて** . . . . . **147**

**外部ファイルの参照** . . . . . **148**
外部ファイルにアクセスする . . . . . 148
ファイル参照名の使用 . . . . . 149
引用符で囲まれた Windows ファイル名の使用 . . . . . 157
作業ディレクトリでのファイルの使用 . . . . . 157

**SAS ステートメントを使用し、外部ファイルにアクセス** . . . . . **157**
SAS ステートメントを使用し、外部ファイルにアクセスする . . . . . 157
FILE ステートメントの使用 . . . . . 158
INFILE ステートメントの使用 . . . . . 159
%INCLUDE ステートメントの使用 . . . . . 159

**SAS コマンドを使用し、外部ファイルにアクセスする** . . . . . **160**
SAS コマンドを使用し、外部ファイルにアクセスする . . . . . 160
FILE コマンドの使用 . . . . . 160
INCLUDE コマンドの使用 . . . . . 161
GSUBMIT コマンドの使用 . . . . . 161

**高度な外部 I/O の操作法** . . . . . **162**
高度な外部 I/O の操作法 . . . . . 162
レコード形式の変更 . . . . . 162
外部ファイルへのデータの追加 . . . . . 163
ドライブマッピングの設定 . . . . . 163
各国語文字を含む外部ファイルの読み込み . . . . . 164

## 外部ファイルについて

外部ファイルは、SAS プログラミングステートメント、生データの記録、プロシジャ出力のような、データやテキストを含むファイルです。SAS では外部ファイルを使用できますが、外部ファイルの管理は行いません。

*SAS 言語リファレンス: 解説編* には、外部ファイルについての基本的な、プラットフォームに依存しない情報を含みます。

転送データライブラリを含む外部ファイルにアクセスする方法について**詳細**は、SAS カスタマーサポートセンター Web ページ(`http://support.sas.com`)にアクセスしてください。

# 外部ファイルの参照

## *外部ファイルにアクセスする*

外部ファイルにアクセスするには、外部ファイルの検索方法を指定する必要があります。外部ファイルにアクセスするには、次のステートメントを使用します。

FILENAME
入出力に使用する外部ファイルにファイル参照名を割り当てます。

FILE
外部ファイルを開いて、データ行を書き込みます。行を書き込むには、PUT ステートメントを使用します。

INFILE
外部ファイルを開いて、データ行を読み込みます。行を読み込むには、INPUT ステートメントを使用します。

%INCLUDE
外部ファイルを開いて、開いたファイルから SAS ステートメントを読み込みます。(その他のステートメントは不要)。

これらのステートメントについては、“Windows 版に固有の SAS ステートメント” (443 ページ) セクションと、*SAS ステートメント: リファレンス*の SAS ステートメントに関するセクションを参照してください。

外部ファイルは、さまざまな SAS ダイアログボックスの入力フィールド(**名前を付けて保存**ダイアログボックスのファイル保存先など)、FILENAME 関数、SAS コマンド(FILE、INCLUDE など)で指定することもできます。

場合によっては、次を使用して外部ファイルを参照できます。

- FILENAME ステートメントや FILENAME 関数に割り当てられたファイル参照名
- SET システムオプションまたは Windows の SET コマンドで定義された環境変数
- 引用符で囲んだ Windows ファイル名
- member-name 構文(または集合構文)
- 引用符で囲んだ単一ファイル名(作業ディレクトリのファイル)

次に、外部ファイルの指定方法について説明します。

SAS には、外部ファイルを指定する方法が複数あるため、外部ファイル参照を解決する規則を次の優先順位で適用します。

1. 引用符で囲んだ、Windows の標準的なファイル指定かどうかを確認します。
2. FILENAME ステートメントまたは FILENAME 関数で定義されたファイル参照名かどうかを確認します。
3. 環境変数のファイル参照名かどうかを確認します。
4. 作業ディレクトリのファイルであるとみなされます。

つまり、外部参照ファイルは、標準の Windows ファイル仕様であるとみなされます。そうでない場合は、ファイル参照が fileref であるか(FILENAME ステートメント、FILENAME 関数、または環境変数で定義されているか)どうかを判断します。ファイル参照がこれらの fileref でない場合は、作業ディレクトリのファイル名であるとみなされ

ます。外部ファイル参照がこれらの選択肢のいずれかで有効でない場合は、外部ファイルにアクセスできないことを示すエラーメッセージが発行されます。

## *ファイル参照名の使用*

### *ファイル参照名の使用の概要*

ファイル参照名を使用するのも、外部ファイルを参照する一つの方法です。ファイル参照名とは、外部ファイルに割り当てられた論理名です。ファイル参照名を割り当てるには、**SAS のエクスプローラ**ウィンドウや**お気に入りフォルダ**ウィンドウでのファイルのショートカット、FILENAME ステートメント、FILENAME 関数を使用します。または、Windows 環境変数を使用してファイルを参照することもできます。このセクションでは、ファイル参照名を割り当てるさまざまな方法と、SAS セッションの実行中にアクティブなファイル参照名のリストを取得したりファイル参照名を消去したりする方法について説明します。

### *ファイルショートカットの割り当て*

対話型 SAS セッションで、**SAS のエクスプローラ**ウィンドウまたは**お気に入りフォルダ**ウィンドウを使用してファイル参照名を作成できます。SAS エクスプローラの[ファイルショートカット]フォルダには、アクティブなファイル参照名のリストが含まれます。SAS エクスプローラで新しいファイル参照名を作成するには、次の操作を行います。

1. ファイルショートカットフォルダを選択し、**ファイル** ⇨ **新規作成**を選択します。
2. **ファイルショートカットの作成**ウィンドウで、ショートカット名(ファイル参照名)とショートカットが示す SAS ファイルへのパスを入力します。
3. **起動時に有効**をオンにしても、後続の SAS セッションすべてのショートカットを再割り当てできます。

**お気に入りフォルダ**ウィンドウを使用してファイルショートカットを割り当てるには、

1. ファイルが格納されているフォルダを開きます。
2. カーソルをファイルに合わせ、マウスを右クリックして、**ファイルショートカットの作成**を選択します。
3. **ファイルショートカットの作成** ダイアログボックスで、ファイルショートカットの名前を入力し、Enter キーを押し、**OK** をクリックします。

SAS プログラムで、これらのファイルショートカットを使用できるようになります。

*注:* ファイルショートカットが有効になるのは、現在の SAS セッションの間のみです。

### *FILENAME ステートメントの使用*

FILENAME ステートメントを使用すると、外部ファイルやディレクトリに論理名を割り当てることができます。

*注:* FILENAME 関数の構文は、FILENAME ステートメントと同様です。FILENAME 関数の詳細については、*SAS 関数と CALL ルーチン: リファレンス*を参照してください。

FILENAME ステートメントの最も簡単な構文は次のとおりです。

FILENAME *fileref* "*external-file*";

たとえば、ファイル C:\MYDATA\SCORES.DAT を読み込む場合、次のステートメントを実行してファイル C:\MYDATA\SCORES.DAT にファイル参照名 MYDATA を割り当てます。

```
filename mydata "c:\mydata\scores.dat";
```

SAS プログラムで、このファイル参照名を使用できるようになります。たとえば、次のステートメントは、ファイル参照名 MYDATA で参照される外部ファイルに格納されているデータを使用して、TEST という名前の SAS データセットを作成します。

```
data test;
   infile mydata;
   input name $ score;
run;
```

*注:* AUX、CON、NUL、PRN、LPT1 - LPT9、COM1 - COM9 は Windows の予約語です。これらの予約語をファイル参照名として使用しないでください。

FILENAME、FILE、INFILE の各ステートメントを使用して、外部ファイルのディレクトリを連結したり、複数の個々の外部ファイルを 1 つの論理外部ファイルに連結することもできます。これらのトピックについては “連結ディレクトリへのファイル参照名の割り当て” (153 ページ) および “連結ファイルへのファイル参照名の割り当て” (154 ページ) を参照してください。

*と?のワイルドカードは、入力ファイル名を照合する際に外部ファイル名やファイル拡張子に使用できます。*は任意の一文字以上と一致することを示すために使用します。?は、任意の 1 文字と一致することを示すために使用します。ワイルドカードは、FILENAME ステートメントと INFILE ステートメント、メンバ名構文(集合構文)での入力以外には使用できません。FILE ステートメントでは、ワイルドカードは無効です。次の filename ステートメントは、文字列 **wild** で始まり、**.dat** で終わる、現在のディレクトリ内の各ファイルから入力を読み込みます。

```
filename wild 'wild*.dat';
data;
   infile wild;
   input;
run;
```

次の例では現在の作業ディレクトリにあるすべてのファイルを読み込みます。

```
filename allfiles '*.*';
data;
   infile allfiles;
   input;
run;
```

FILENAME ステートメントでは、プリンタなどのデバイス名を外部ファイルに割り当てたり、レコード形式やレコード長などのファイル特性を制御するさまざまなオプションを使用できます。これらのオプションの一部については “高度な外部 I/O の操作法” (162 ページ) で説明しています。FILENAME ステートメントの完全な構文については、“FILENAME ステートメント:Windows” (448 ページ) を参照してください。

### *環境変数の使用*

環境変数を定義して、SAS ライブラリに論理名としての機能を果たせるように( “環境変数を使用した SAS ライブラリの割り当て” (129 ページ) を参照)、環境変数を使用して外部ファイルを参照することもできます。SAS 環境変数を定義するには SET システムオプションを使用し、Windows 環境変数を定義するには Windows の SET コマンドを使用します。また、**システム** ダイアログボックスを使用しても、環境変数を定義できます。

*注:* AUX、CON、NUL、PRN、LPT1 - LPT9、COM1 - COM9 は Windows の予約語です。これらの予約語を環境変数として使用しないでください。

環境変数が使用できる場合、SAS の起動前にリソースを簡単に割り当てることができます。ただし、SET システムオプションを使用して特定の SAS セッション用に定義した環境変数は、他のアプリケーションには使用できません。

### *SET システムオプションの使用*

たとえば、外部ファイル **C:\MYDATA\TEST.DAT** を参照する SAS 環境変数を定義するには、SAS 構成ファイルで次の SET オプションを使用します。

```
-set myvar c:\mydata\test.dat
```

次に、SAS プログラムで、環境変数 MYVAR を使用して外部ファイルを参照します。

```
data mytest;
   infile myvar;
   input name $ score;
run;
```

Windows の **スタート** メニューを使用して SAS を起動する場合は、SAS 構成ファイルで SET システムオプションを使用することをお勧めします。

### *SET コマンドの使用*

環境変数の定義に SET システムオプションを使用する代わりに、Windows の SET コマンドを使用します。たとえば、前の例に相当する Windows SET コマンドは、

```
SET MYVAR=C:\MYDATA\TEST.BAT
```

です。また、SET コマンドは、コントロールパネルからアクセスする**システムプロパティ**ダイアログボックスを使用しても定義できます。

環境変数を定義する SET コマンドをすべて発行してから、SAS を起動する必要があります。MS-DOS ウィンドウで環境を定義し、その後 MS-DOS ウィンドウで環境を定義し、その後**スタート** メニューから SAS を起動すると、環境変数は認識されません。

### *ディレクトリへのファイル参照名の割り当て*

ディレクトリにファイル参照名を割り当て、メンバ名構文(または集計構文)を使用して、そのディレクトリの個々のファイルにアクセスできます。

たとえば、1 月の担当地域の販売データすべてをディレクトリ **C:\SAS\MYDATA** に保存する場合は、次の FILENAME ステートメントを発行して、このディレクトリにファイル参照 JAN を割り当てます。

```
filename jan "c:\sas\mydata";
```

SAS プログラムで、メンバ名が付いたこのファイル参照名を使用できるようになります。次の例では、JAN ディレクトリに格納されている 2 つのファイルを参照します。

```
data westsale;
   infile jan(west);
   input name $ 1-16 sales 18-25
         comiss 27-34;
run;
data eastsale;
   infile jan(east);
   input name $ 1-16 sales 18-25
         comiss 27-34;
run;
```

member-name 構文を使用する際、ファイル拡張子が予測される場合は、参照するファイルのファイル拡張子を指定する必要はありません。たとえば、この例では、INFILE

ステートメントでファイル拡張子が.DAT であると予測されます。次の表に、SAS ステートメントや SAS コマンドで予測されるファイル拡張子を示します。

*表 5.1 メンバ名構文を使用して外部ファイルを参照する場合のデフォルトファイル拡張子*

| SAS コマンドまたは SAS ステートメント | SAS ウィンドウ | ファイル拡張子 |
|---|---|---|
| FILE ステートメント | EDITOR | .DAT |
| %INCLUDE ステートメント | EDITOR | .SAS |
| INFILE ステートメント | EDITOR | .DAT |
| FILE コマンド | EDITOR | .SAS |
| FILE コマンド | LOG | .LOG |
| FILE コマンド | OUTPUT | .LST |
| FILE コマンド | NOTEPAD | なし |
| INCLUDE コマンド | EDITOR | .SAS |
| INCLUDE コマンド | NOTEPAD | なし |

たとえば、次のプログラムは、ファイル **C:\PROGRAMS\TESTPGM.SAS** を SAS に送信します。

```
filename test "c:\programs";
%include test(testpgm);
```

C:\PROGRAMS というディレクトリのファイル名 TESTPGM.SAS が検索されます。

デフォルトのファイル拡張子以外のファイル拡張子を持つファイルの場合、次に示すように、ファイル名にファイル拡張子を付けます。

```
filename test "c:\programs";
%include test(testpgm.xyz);
```

ファイル拡張子を持たないファイルの場合、次に示すように、ファイル名を引用符で囲む必要があります。

```
filename test "c:\programs";
%include test("testpgm");
```

次に、SAS が使用するデフォルトのファイル拡張子をさらに解説するために、member-name 構文を使用した例を示します。次の FILENAME ステートメントがサブミットされているとします。

```
filename test "c:\mysasdir";
```

次の例では、ファイル **C:\MYSASDIR\PGM1.DAT** を出力用に開きます。

```
file test(pgm1);
```

次の例では、ファイル **C:\MYSASDIR\PGM1.DAT** を入力用に開きます。

```
infile test(pgm1);
```

次の例では、ファイル **C:\MYSASDIR\PGM1:**を読み込んでサブミットします。

```
%include test("pgm1");
```

これらの例では、SAS ステートメントを使用します。FILE コマンドや INCLUDE コマンドのような SAS コマンドでは、member-name 構文を指定し、表 5.1 (152 ページ) に示す同じデフォルトのファイル拡張子が存在します。

member-name 構文を使用すると、ファイル参照名を使用せずに作業ディレクトリ内のサブディレクトリを参照することもできます。たとえば、作業ディレクトリの直下に PROGRAMS という名前のサブディレクトリがあるとします。このサブディレクトリにあるファイルを参照するときは、サブディレクトリ名 PROGRAMS を使用できます。たとえば、次のステートメントは、*作業ディレクトリ* **\PROGRAMS\PGM1.SAS:**に保存されているプログラムをサブミットします。

```
%include programs(pgm1);
```

次の例は、FILE コマンドを使用して、*作業ディレクトリ* **\PROGRAMS\TESTPGM.DAT:** にアクティブウィンドウの内容を保存します。

```
file programs(testpgm);
```

*注:* ディレクトリ名がすでに定義されているファイル参照名と同じ場合、ファイル参照名はディレクトリ名より優先されます。

### *連結ディレクトリへのファイル参照名の割り当て*

FILENAME ステートメントを使用して外部ファイルのディレクトリを連結する際にも、member-name 構文が役立ちます。たとえば、次の FILENAME ステートメントを発行するとします 。

```
filename progs ("c:\sas\programs",
                "d:\myprogs");
```

このステートメントは、ファイル参照名 PROGS が C:\SAS\PROGRAMS ディレクトリと D:\MYPROGS ディレクトリの両方に格納されているすべてのファイルを参照することを示します。SAS プログラムでファイル参照名 PROGS を使用すると、両方のディレクトリから指定したメンバを検索します。この連結機能を使用するには、SAS で使用するプロトコルを把握している必要があります。使用されるプロトコルは、読み込み、書き込み、更新のいずれの目的でファイルにアクセスするかによって異なります。詳細については、“連結ディレクトリのアクセス法について” (156 ページ) を参照してください。

### *member-name 構文の解決規則の要約*

SAS では、規則を適用し、member-name 構文を使用した外部ファイル参照を解決します。たとえば、SAS ステートメントまたはコマンドの外部ファイルを参照が次のようであるとします。

```
progs(member1)
```

SAS は、次の規則を使用して、この外部ファイル参照を解決します。次の規則を使用して、この外部ファイル参照を解決します。このリストの順位は、優先順位を示します。

1. FILENAME ステートメントで定義されたファイル参照名 PROGS を確認します。
2. SAS または Windows の環境変数 PROGS を確認します。
3. 作業ディレクトリの直下のディレクトリ PROGS を確認します。

メンバ名は、有効な物理ファイル名にする必要があります。前述の例のように拡張子がない場合、SAS は 表 5.1 (152 ページ) に記載されたように適切なデフォルト拡張子を使用します。拡張子が指定されているか、またはメンバ名が引用符で囲まれている場合、拡張子は割り当てられず、指定されたままのファイル名が検索されます。

### 連結ファイルへのファイル参照名の割り当て

SAS の外部ファイルを読み込む際、ファイルの連結を指定できます。連結ファイルは、空白またはカンマで区切った複数のファイル指定(ワイルドカード文字を含む)から構成されます。有効な連結指定の例を次に示します。

- `filename allsas ("one.sas", "two.sas", "three.sas");`
- `filename alldata ("test1.dat" "test2.dat" "test3.dat");`
- `filename allinc "test*.sas";`
- `%include allsas;`
- `infile alldata;`
- `include allinc;`

この連結機能を使用するには、SAS で使用するプロトコルを把握している必要があります。使用されるプロトコルは、読み込み、書き込み、更新のいずれの目的でファイルにアクセスするかによって異なります。詳細については、“連結ファイルのアクセス法について” (156 ページ) を参照してください。

*注:* 連結ファイル指定と連結ディレクトリ指定を混同しないでください。連結ディレクトリ指定も有効であり、“連結ディレクトリへのファイル参照名の割り当て” (153 ページ) で説明されています。

### 長いファイル名で外部ファイルを参照

SAS では、長いファイル名を使用できます。(有効な長いファイル名の詳細については、Windows オペレーティングシステムのドキュメントを参照してください)。ファイル名をダイアログボックス、コマンド、任意の SAS 言語に対する引数として指定する場合は、常に長いファイル名を使用できます。

ステートメントや関数などの SAS 言語を使用して外部ファイル名を指定する際、あいまいさを軽減するためにファイル名を二重引用符で囲む必要があります(一重引用符は、長いファイル名では有効な文字であるため)。複数のファイル名を指定する必要がある場合は、各ファイル名を二重引用符で囲み、ファイル名をスペースで区切ります。

SAS での長いファイル名の有効な使用例を次に示します。

- `libname abc "My data file";`
- `filename myfile "Bernie's file";`
- `filename summer ("June sales" "July sales" "August sales");`
- `include "A really, really big SAS program";`

### UNC パスを使用したファイルの参照

SAS では、UNC (Universal Naming Convention)パスを使用できます。UNC パスを使用すると、ネットワークドライブレターを参照せずに、使用しているコンピュータをネットワークデバイスに接続できます。SAS では、Windows とネットワークソフトウェアがサポートする範囲内で、UNC パスがサポートされています。一般に、SAS でネットワークドライブを参照する場合は、常に UNC パスを参照できます。

UNC パスには、次の構文を使用します。

\\SERVER\SHARE\FOLDER\FILEPATH

ここで

SERVER
ネットワークファイルサーバー名です。

SHARE
　サーバー上の共有ボリュームです。

FOLDER
　共有ボリュームのディレクトリの 1 つです。

FILEPATH
　1 つ以上のサブディレクトリを参照することもある、連続するファイルパスです。

たとえば、次のコマンドには、ネットワークファイルサーバー ZAPHOD のファイルが指定されています。

```
include "\\zaphod\universe\galaxy\stars.sas";
```

### *ファイル参照名の割り当てのリスト*

SAS セッション中に複数のファイル参照名を割り当てた後、ファイル参照名の割り当て先を確認したい場合、**SAS のエクスプローラ**ウィンドウまたは FILENAME ステートメントを使用して、割り当てたファイル参照名をすべて表示できます。

**SAS エクスプローラ**ウィンドウを使用してアクティブなファイル参照を一覧で表示するには、**ファイルショートカット**をダブルクリックします。**エクスプローラ**ウィンドウに、現在の SAS セッションに対してアクティブなファイル参照名がすべて表示されます。ファイル参照名として定義した環境変数を SAS セッションで使用したことがある場合は、その環境変数がすべて表示されます。ファイル参照名として定義した環境変数を SAS プログラムで使用したことがない場合は、**エクスプローラ**ウィンドウにファイル参照名は表示されません。

次の FILENAME ステートメントを使用して、アクティブなファイル参照名を SAS ログに書き込むことができます。

```
filename _all_ list;
```

### *ファイル参照名のクリア*

ファイル参照名をクリアするには、FILENAME ステートメントの次の構文を使用します。

FILENAME *fileref*|_ALL_ <CLEAR>;

ファイル参照名を指定すると、指定したファイル参照名のみがクリアされます。キーワード _ALL_ を指定すると、現在の SAS セッションで割り当てられたファイル参照名がすべてクリアされます。

SAS エクスプローラのファイルショートカットを使用してファイル参照名をクリアするには、

1. 削除するファイルショートカットを選択します。すべてのファイルショートカットは次のように選択できます。**編集** ⇨ **すべて選択**
2. 次に、Delete キーを押すか、次のメニューを選択します。**編集** ⇨ **削除**
3. **OK** をクリックして、ファイルショートカットの削除を確定します。

*注:* 環境変数を使用して定義されているファイル参照名はクリアできません。環境変数を使用して定義されているファイル参照名は、SAS セッション全体に割り当てられます。

ファイル参照名と参照先のファイルとの関連付けは、ジョブまたはセッションの終了時に自動的にクリアされます。現在のセッションでファイル参照名を他のファイルに割り当てる場合は、セッションを終了したり、ファイル参照名を消去したりする必要はありません。新しいファイルの FILENAME ステートメントを発行すると、自動的に割り当てし直されます。

### 連結ディレクトリのアクセス法について

ファイル参照名を複数の物理ディレクトリに割り当てる場合、アクセスするファイルは、入力または出力のどちらの目的でアクセスするかによって異なります。

### 入力

ファイルを開いて入力または更新する場合、メンバ名が一致した最初のファイルにアクセスします。たとえば、ファイル PHONE.DAT が C:\SAMPLES のディレクトリと C:\TESTPGMS のディレクトリの両方に存在する場合に次のステートメントをサブミットすると、C:\SAMPLES にあるファイルが読み込まれます。

```
filename test ("c:\samples","c:\testpgms");
data sampdat;
   infile test(phone.dat);
   input name $ phonenum $ city $ state $;
run;
```

### 出力

ファイルを開いて出力する場合、同じ名前のファイルが後のディレクトリに存在しても、FILENAME ステートメントに最初にリストされたディレクトリのファイルに書き込みます。たとえば、次の FILENAME ステートメントを入力するとします 。

```
filename test ("c:\sas","d:\mysasdir");
```

次の FILE コマンドを発行すると、同じ名前のファイルが D:\MYSASDIR ディレクトリに存在しても、ファイル SOURCE.PGM が C:\SAS ディレクトリに書き込まれます。

```
file test(source.pgm);
```

### 連結ファイルのアクセス法について

ファイル参照名を複数の物理ファイルに割り当てる場合、SAS ステートメントと SAS コマンドの処理は、入力または出力のどちらの目的でファイルにアクセスしているかによって異なります。

### 入力

ファイルを開いて入力する場合、すべてのファイルのデータが入力されます。たとえば、次のステートメントを実行すると、%INCLUDE ステートメントによって 4 つのプログラムの実行がサブミットされます。

```
filename mydata ("qtr1.sas","qtr2.sas",
                 "qtr3.sas","qtr4.sas");
%include mydata;
```

### 出力

ファイルを開いて出力する場合、連結されている最初のファイルにデータが書き込まれます。たとえば、次のステートメントを実行すると、PUT ステートメントによって MYDAT1.DAT への書き込みが行われます。

```
filename indata "dogdat.dat";
filename outdata ("mydat1.dat","mydat2.dat",
                  "mydat3.dat","mydat4.dat");
data _null_;
   infile indata;
   input name breed color;
   file outdata;
   put name= breed= color=;
run;
```

### 引用符で囲まれた Windows ファイル名の使用

### 引用符で囲まれた Windows ファイル名の使用

外部ファイルの参照にファイル参照名を使用する代わりに、引用符で囲まれた Windows ファイル名を使用できます。たとえば、起動する SAS プログラムがファイル C:\MYDIR\ORANGES.SAS に含まれる場合、次のステートメントを実行できます。

```
%include "c:\mydir\oranges.sas";
```

SAS ステートメントに引用符で囲まれた Windows ファイル名を使用する際、参照するファイルが作業ディレクトリにある場合はドライブとディレクトリの指定を省略できます。たとえば、前述の例で、作業ディレクトリが C:\MYDIR の場合、次のステートメントをサブミットできます。

```
%include "oranges.sas";
```

### 作業ディレクトリでのファイルの使用

作業ディレクトリでアクセスする必要のある外部ファイルを保存し、それらの外部ファイルに適切なファイル拡張子がある場合、( 表 5.1 (152 ページ) を参照)、SAS ステートメントで引用符やファイル拡張子がなくとも、ファイル名を参照することができます。たとえば、作業ディレクトリに ORANGES.SAS という名前のファイルがあり、ORANGES がファイル参照名として定義されていない場合、次のステートメントを使用するとファイルをサブミットできます。

```
%include oranges;
```

ただし、このタイプのファイル参照には次の条件が必須です。

- ファイルが作業ディレクトリに格納されています。
- ファイルは適切なファイル拡張子を持ちます。
- ファイル名がファイル参照名として定義されていません。

SAS の作業ディレクトリを決定または変更する方法の詳細については、“SAS 起動時の作業フォルダの設定” (11 ページ) および “SAS の現在のフォルダの変更” (41 ページ) を参照してください。

## SAS ステートメントを使用し、外部ファイルにアクセス

### SAS ステートメントを使用し、外部ファイルにアクセスする

このセクションでは、FILE ステートメント、INFILE ステートメントおよび%INCLUDE ステートメントを使用して外部ファイルにアクセスする簡単な例を示します。Windows 版でのこれらのステートメントの詳細な使用例については、“高度な外部 I/O の操作法” (162 ページ) を参照してください。

- “FILE ステートメントの使用” (158 ページ)
- “INFILE ステートメントの使用” (159 ページ)
- “%INCLUDE ステートメントの使用” (159 ページ)

## FILE ステートメントの使用

FILE ステートメントを使用すると、PUT ステートメントにより書き込まれる行を外部ファイルに送信することができます。[1]

FILE ステートメントを使用した例を示します。この例では、SAS データセット MYLIB.TEST からデータを読み込み、95 点を超えるスコアのみを外部ファイル C:\MYDIR\TEST.DAT に書き出します。

```
filename test "c:\mydir\test.dat";
libname mylib "c:\mydata";
data _null_;
   set mylib.test;
   file test;
   if score ge 95 then
      put score;
run;
```

この例では、1 オブザベーションに付き 1 変数の値のみが外部ファイルに書き出されます。次の例では、PUT ステートメントに_ALL_オプションを指定することで、変数 REGION に値 **west** が含まれる場合に、現在のオブザベーションに含まれるすべての変数が外部ファイルにコピーされます。

```
libname us "c:\mydata";
data west;
   set us.pop;
   file "c:\sas\pop.dat";
   where region="west";
   put _all_;
run;
```

オブザベーション全体を書き出すこの方法は、SAS データセットの変数の値を外部ファイルに書き出す必要のある場合によく使用されます。このため、SAS データセット形式のデータを読み込むことができない他のアプリケーションでデータを使用することができます。

*注:* この例では、PUT ステートメントに_ALL_キーワードを使用します。このコードによって、変数名、等号(=)、および変数の値がすべてファイルに書き出される、名前付き出力が生成されます。名前付き出力の詳細については、*SAS ステートメント: リファレンス*の PUT ステートメントの説明を参照してください。

また、FILE ステートメントには複数のオプションを指定することができます。これらのオプションを指定すると、レコードの形式と長さが制御されます。これらのオプションの一部については “高度な外部 I/O の操作法” (162 ページ) で説明しています。FILE ステートメントの完全な構文については、“FILE ステートメント:Windows” (445 ページ) を参照してください。

FILE ステートメントで使用されるデフォルトのレコード長は、32767 文字です。保存するデータのレコードが、32767 文字を超える場合は、FILENAME ステートメントを使用して、ファイル参照名を定義する必要があります。さらに、FILENAME ステートメントに LRECL=オプションを使用して、正しい論理レコード長を指定するか、FILE ステートメントに LRECL=オプションを指定する必要があります。LRECL=オプションの詳細については、“FILE ステートメント:Windows” (445 ページ) の LRECL=を参照してください。

---

1 FILE ステートメントを使用し、PUT ステートメントの出力を SAS ログやプロシジャ出力と同じ送り先に送信することもできます。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。

デフォルトの 32767 文字の代わりに、OPTIONS ステートメントまたは構成ファイル内の LRECL=に対して、異なる値を指定することもできます。この値はセッション全体の間、有効であり続けます。特定のステップに異なる LRECL=の値を指定する場合、FILENAME ステートメント、FILE ステートメントまたは INFILE ステートメントにその値を指定する必要があります。

### INFILE ステートメントの使用

INFILE ステートメントを使用して、SAS DATA ステップの INPUT ステートメントによって読み込まれるデータのソースを指定します。INFILE ステートメントは、読み込むデータの保存場所と種類を定義する INPUT ステートメントと必ず一緒に使用されます。

INFILE ステートメントの簡単な例を示します。この DATA ステップでは、指定したデータを外部ファイルから読み込み、SURVEY という名前の SAS データセットを作成します。

```
filename mydata "c:\mysasdir\survey.dat";
data survey;
   infile mydata;
   input fruit $ taste looks;
run;
```

ファイル参照名のかわりに、Windows ファイル名を引用符で囲み使用することも可能です。

```
data survey;
   infile "c:\mysasdir\survey.dat";
   input fruit $ taste looks;
run;
```

また、INFILE ステートメントには他のオプションを指定することができます。これらのオプションを指定すると、レコードの形式と長さが制御されます。これらのオプションの一部については “高度な外部 I/O の操作法” (162 ページ) で説明しています。INFILE ステートメントの完全な構文については、“INFILE ステートメント:Windows” (457 ページ) を参照してください。

INFILE ステートメントで使用されるデフォルトのレコード長は、32767 文字です。読み込むデータのレコードが 32767 文字を超える場合は、FILENAME ステートメントを使用して、ファイル参照名を定義する必要があります。さらに、FILENAME ステートメントに LRECL=オプションを使用して正しい論理レコード長を指定するか、INFILE ステートメントで LRECL=オプションを指定する必要があります。LRECL=オプションの詳細については、“INFILE ステートメント:Windows” (457 ページ) の LRECL=を参照してください。

デフォルトの 32767 文字の代わりに、OPTIONS ステートメントまたは構成ファイル内の LRECL=に対して、異なる値を指定することもできます。この値はセッション全体の間、有効であり続けます。特定のステップに異なる LRECL=の値を指定する場合、FILENAME ステートメント、FILE ステートメントまたは INFILE ステートメントにその値を指定する必要があります。

### %INCLUDE ステートメントの使用

%INCLUDE ステートメントをサブミットすると、実行中の SAS プログラムにファイル全体が読み込まれ、このファイルはただちに SAS にサブミットされます。1 つの SAS プログラム内に、個々の%INCLUDE ステートメントを必要な数だけ含めることができます。また、%INCLUDE ステートメントは、10 レベルまでのネスティングが可能です。%

INCLUDE ステートメントを使用すると、モジュール化された SAS プログラムの作成がより簡単になります。%INCLUDE ステートメントと member-name 構文を使用して、C:

\SAS\MYJOBS\PROGRAM1.SAS に格納されているステートメントをサブミットする例を示します。

```
filename job "c:\sas\myjobs";
%include job(program1);
```

また、%INCLUDE ステートメントには複数のオプションを指定することができます。これらのオプションを指定すると、レコードの形式と長さが制御されます。これらのオプションの一部については “高度な外部 I/O の操作法” (162 ページ) で説明しています。%INCLUDE ステートメントの完全な構文については、“%INCLUDE ステートメント:Windows” (455 ページ) を参照してください。

%INCLUDE ステートメントで使用されるデフォルトのレコード長は、32767 文字です。読み込むプログラムのレコードが 32767 文字を超える場合は、FILENAME ステートメントを使用して、ファイル参照名を定義する必要があります。さらに、FILENAME ステートメントに LRECL=オプションを使用して正しい論理レコード長を指定するか、%INCLUDE ステートメントに LRECL=オプションを指定する必要があります。LRECL=オプションの詳細については、“%INCLUDE ステートメント:Windows” (455 ページ) の LRECL=を参照してください。

デフォルトの 32767 文字の代わりに、OPTIONS ステートメントまたは構成ファイル内の LRECL=に対して、異なる値を指定することもできます。この値はセッション全体の間、有効であり続けます。特定のステップに異なる LRECL=の値を指定する場合、FILENAME ステートメント、FILE ステートメントまたは INFILE ステートメントにその値を指定する必要があります。

# SAS コマンドを使用し、外部ファイルにアクセスする

## *SAS コマンドを使用し、外部ファイルにアクセスする*

このセクションでは、FILE コマンドおよび INCLUDE コマンドを使用し、外部ファイルにアクセスする方法について説明します。これらのコマンドでは、**名前を付けて保存**ダイアログボックスおよび**開く**ダイアログボックスと同じ処理が可能です。どの方法で外部ファイルにアクセスするかは、使用する SAS アプリケーションの要件と、好みに応じて選択してください。

## *FILE コマンドの使用*

FILE コマンドの使用法は、FILE ステートメントとは異なります。FILE コマンドは、(たとえば単に DATA ステップで PUT ステートメントの出力先を指定するだけでなく)ウィンドウの現在の内容を外部ファイルに書き出します。

たとえば、**LOG** ウィンドウの内容を外部ファイル名 C:\SASLOGS\TODAY.LOG に保存する場合、**コマンド**ダイアログボックスから次の FILE コマンドを発行することができます。ただし、**LOG** ウィンドウはアクティブである必要があります。

```
file "c:\saslogs\today.log"
```

すでにファイル参照名 LOGS が SASLOGS ディレクトリに割り当てられている場合は、次の FILE コマンドを使用することができます。

```
file logs(today)
```

この場合では、ファイル拡張子は 表 5.1 (152 ページ)に示す.log をデフォルトとして設定されます。

FILE コマンドを使用して既存のファイルに書き込もうとすると、既存のファイルを置換するか、ウィンドウの内容を既存のファイルに追加するか、要求を取り消すかを選択するダイアログボックスが表示されます。

引数を指定せずに FILE コマンドを発行すると、ウィンドウの内容は最新の FILE コマンドで参照されたファイルに書き込まれます。この方法は、プログラムの編集時に頻繁に保存する場合に便利です。ただし、置換か追加かを選択するダイアログボックスは、FILE コマンドを最初に発行するときにしか表示されません。その後は、FILE コマンドでファイル名を指定しない限り、その前に指定したパラメータ(置換または追加)が使用され、選択は要求されません。

SAS メインウィンドウのファイルメニューから**名前を付けて保存** を選択すると、**名前を付けて保存**ダイアログボックスが表示されます。このダイアログボックスの機能は、FILE コマンドと同じです。ただし、FILE コマンドよりも選択肢が多いため柔軟性が増し、より対話的にもなっています。詳細については、次の資料のファイルの保存を参照してください。“ファイルの保存” (82 ページ) および “プログラムエディタの使用” (107 ページ)のファイルの保存に関する説明を参照してください。

FILE コマンドには、複数のオプションを指定することもできます。これらのオプションを指定すると、レコードの形式と長さが制御されます。これらのオプションの一部については “高度な外部 I/O の操作法” (162 ページ)で説明しています。FILE コマンドの完全な構文については、“FILE コマンド:Windows” (344 ページ)を参照してください。

## *INCLUDE コマンドの使用*

INCLUDE コマンドを使用すると、%INCLUDE ステートメントを使用した場合と同じように、外部ファイルを、**エディタ**ウィンドウ、**NOTEPAD** ウィンドウまたはその他のアクティブなウィンドウにコピーすることができます。ただし、INCLUDE コマンドの場合、ファイルはウィンドウにコピーされるだけで、サブミットはされません。

たとえば、ファイル C:\SAS\PROG1.SAS を**エディタ**ウィンドウにコピーするとします。すでにファイル参照名 SAMPLE が正しいディレクトリに割り当てられ、エディタウィンドウがアクティブになっている場合は、**コマンド**ダイアログボックスから次の INCLUDE コマンドを発行してメンバ PROG1 を **エディタ**ウィンドウにコピーすることができます。

```
include sample(prog1);
```

ファイルを SAS セッションにコピーするもう 1 つの方法は、**ファイルを開く** ダイアログボックスを使用することです。ファイルのコピーの他に、**ファイルを開く** ダイアログボックスでは、コピーしているプログラムを起動するなど、他の操作も選択できます。**ファイルを開く**ダイアログボックスでは、柔軟に**エディタ** ウィンドウにファイルをコピーできます。詳細については、“ファイルの保存” (82 ページ) および “プログラムエディタの使用” (107 ページ)のファイルを開く方法に関する説明を参照してください。

INCLUDE コマンドには、複数の引数を指定することもできます。引数を使用して、レコードの形式や長さを制御することができます。これらの引数の一部は次で説明されています。“高度な外部 I/O の操作法” (162 ページ)INCLUDE コマンドの完全な構文については、“INCLUDE コマンド:Windows” (348 ページ)を参照してください。

引数を指定せずに INCLUDE コマンドを発行すると、最新の INCLUDE コマンドで参照されたファイルがコピーされます。以前の INCLUDE コマンドが存在しない場合は、エラーメッセージが表示されます。

## *GSUBMIT コマンドの使用*

GSUBMIT コマンドを使用すると、Windows クリップボードに格納されている SAS ステートメントをサブミットすることができます。クリップボードから SAS ステートメントをサブミットするには、次のコマンドを使用します。

```
gsubmit buffer=default;
```

また GSUBMIT コマンドを使用して、コマンドの中で指定した SAS ステートメントをサブミットすることもできます。GSUBMIT コマンドの詳細については、SAS システムヘルプを参照してください。

*注:* 拡張エディタを使用してサブミットしたプロシジャの実行中は、Windows クリップボードに格納された SAS ステートメントを、GSUBMIT コマンドを使用してサブミットすることはできません。この場合、新たな**拡張エディタ**ウィンドウを開いて SAS ステートメントをコピーし、サブミットすることができます。

# 高度な外部 I/O の操作法

## *高度な外部 I/O の操作法*

このセクションでは、FILENAME ステートメント、FILE ステートメントおよび INFILE ステートメントを使用して、レコードの形式や長さの変更、ファイルへのデータの追加、DRIVEMAP デバイスタイプキーワードによる使用するドライブの指定などの、より高度な入出力タスクを実行する方法を説明します。


- “レコード形式の変更” (162 ページ)
- “外部ファイルへのデータの追加” (163 ページ)
- “ドライブマッピングの設定” (163 ページ)
- “各国語文字を含む外部ファイルの読み込み” (164 ページ)


## *レコード形式の変更*

FILENAME ステートメント、FILE ステートメント、%INCLUDE ステートメント、INFILE ステートメントに RECFM=オプションを使用すると、外部ファイルのレコード形式を指定することができます。次の例は、このオプションの使用方法を示しています。

通常、SAS では、キャリッジリターンとラインフィードの組み合わせ('0D0A'x)またはラインフィード('0A'x)までのデータを一行として読み込みます。ただし、データにこれらのキャリッジコントロール文字が含まれず、固定長レコードが含まれることもあります。この場合は、RECFM=F を指定すると、データを読む込むことができます。

固定長レコードを読み込むには、レコード長を指定する LRECL=オプションと、レコードが固定長レコード形式であることを SAS に認識させる RECFM=オプションを使用する必要があります。次に、必要なステートメントを示します。

```
data test;
   infile "test.dat" lrecl=60 recfm=f;
   input x y z;
run;
```

この例では、レコードが 60 バイトの固定長であるとみなされ、3 つの数値変数 X、Y、Z として読み込まれます。

データにキャリッジリターンやラインフィードが含まれる場合でも、キャリッジコントロール文字としてではなくデータの一部としてこれらのデータを読み込むために、RECFM=F を指定することができます。RECFM=F を指定すると、SAS では、キャリッジコントロールとラインフィードは無視され、指定したレコード長が読み込まれます。

### 外部ファイルへのデータの追加

新しい出力ファイルを作成しないで、既存のファイルの末尾にデータを追加することもできます。この場合は、次の例のように、FILE ステートメントに MOD オプションを使用します。

```
filename myfile "c:\sas\data";
data _null_;
   infile myfile(newdata);
   input sales expenses;
   file myfile(jandata) mod;
   put sales expenses;
run;
```

この例では、外部データファイル C:\SAS\DATA\NEWDATA.DAT から変数 SALES と変数 EXPENSES が読み込まれ、既存のデータファイル C:\SAS\DATA\JANDATA.DAT にレコードが追加されます。

単一ディレクトリの複数のファイルにデータを追加する場合は、FILE ステートメントではなく、FILENAME ステートメントに MOD オプションを使用します。ファイルにデータを追加するには、FAPPEND 関数や PRINTTO プロシジャも使用できます。詳細については、*SAS 関数と CALL ルーチン: リファレンス* の SAS 関数と、*Base SAS プロシジャガイド*の PRINTTO プロシジャを参照してください。

### ドライブマッピングの設定

FILENAME ステートメントに DRIVEMAP デバイスタイプキーワードを使用すると、使用するドライブを指定することができます。

SAS/AF アプリケーションで、この方法を使用すると、ユーザーがハードドライブを選択できるような、選択リストを作成できます。また、DRIVEMAP キーワードを使用すると、使用可能な各種ハードドライブにマクロ変数を割り当てることもできます。

FILENAME ステートメントに DRIVEMAP デバイスタイプキーワードを使用すると、このファイル参照名は読み取り専用として使用されることを意味します。DRIVEMAP デバイスタイプキーワードに関連付けられているファイル参照名を使用して書き出しまたは更新を行うと、ファイルへの書き出しを行う権限がないというエラーメッセージが表示されます。

このキーワードを使用した例を示します。

```
filename myfile drivemap;
data mymap;
   infile myfile;
   input drive $;
   put drive;
run;
```

SAS ログへと書き込まれる情報は、アウトプット 5.1 (164 ページ) の情報と似ています。

**アウトプット 5.1** ドライブのマッピング情報

```
50   filename myfile drivemap;
51
52   data mymap;
53      infile myfile;
54      input drive $;
55      put drive;
56   run;
NOTE: The infile MYFILE is:
      FILENAME=DRIVEMAP,
      RECFM=V,LRECL=32767
A:
C:
D:
J:
K:
L:
M:
N:
R:
S:
T:
U:
NOTE: 12 records were read from the infile MYFILE.
      The minimum record length was 2.
      The maximum record length was 2.
NOTE: The data set WORK.MYMAP has 12 observations
      and 1 variables.
NOTE: The DATA statement used 2.04 seconds.
```

## 各国語文字を含む外部ファイルの読み込み

多くの Windows アプリケーションと同じように、Windows 版 SAS では、ANSI 文字コードを使用して文字データが読み書きされます。現在、SAS では OEM 文字セットを使用してファイルを読み書きするオプションは提供されていません。

Â などの文字は、各国語文字とみなされます。Windows では、16 進数を使用して各文字を表します。外部ファイルが Windows エディタ(Word などのアプリケーション)または SAS で作成された場合、特別な操作は必要ありません。通常どおり、FILENAME ステートメントまたは FILE ステートメントを使用して単純にファイルを読み込みます。

*6 章*

# Windows 版での SAS 出力の管理

**印刷** . . . . . **165**
Windows 環境での SAS の印刷について . . . . . 165
SAS ウィンドウからの印刷 . . . . . 166
印刷前の出力のプレビュー . . . . . 175
SAS 印刷フォームの使用 . . . . . 176
SAS コマンドを使用した印刷 . . . . . 177
プリンタへの DATA ステップの出力の送信 . . . . . 178
プリント出力のファイルへの送信 . . . . . 178
バッチモード印刷 . . . . . 179
デフォルトプリンタの詳細 . . . . . 179
印刷ジョブの取り消し . . . . . 180

**プロシジャ出力の Web ブラウザへの表示** . . . . . **180**
プロシジャ出力の Web ブラウザへの表示について . . . . . 180
HTML 出力のプリファレンスの構成 . . . . . 180
結果ビューアウィンドウの使用 . . . . . 181

**ファイルへのプロシジャの出力と SAS ログの送付について** . . . . . **183**
ファイルへのプロシジャの出力と SAS ログの送付の概要 . . . . . 183
名前を付けて保存ダイアログボックスの使用 . . . . . 183
PRINTTO プロシジャの使用 . . . . . 183
SAS システムオプションの使用 . . . . . 184

**SAS ログ機能を使用したログメッセージの書き込み** . . . . . **185**

**グラフィックの作成** . . . . . **185**
ディスプレイ上でのグラフィックの作成 . . . . . 185
グラフィックの印刷 . . . . . 186
他のアプリケーションからのグラフィックのインポート . . . . . 190
他のアプリケーションで使用するグラフィックのエクスポート . . . . . 192
補足 . . . . . 195

## 印刷

### *Windows 環境での SAS の印刷について*

デフォルトでは、Windows 版 SAS は Microsoft Windows の印刷設定を使用して、他の Windows アプリケーションと同じ方法で出力を管理できます。Windows の印刷設定

を使用すると、Windows TrueType フォントや、プリンタでサポートされているフォントが使用されます。

ODS (Output Delivery System)を使用して、ユニバーサル印刷も使用できます。Windows 環境でユニバーサル印刷を有効化するには、UNIVERSALPRINT オプションと UPRINTMENUSWITCH オプションを指定します。このセクションでは、Windows の印刷機能を使用した場合について解説します。ユニバーサル印刷の詳細については、*SAS 言語リファレンス: 解説編*および "UPRINTMENUSWITCH システムオプション:Windows" (584 ページ) を参照してください。

Windows 印刷の詳細については、Windows ドキュメントを参照してください。"グラフィックの作成" (185 ページ) では、SAS セッションからグラフィックの出力先プリンタを指定する方法について説明しています。

## *SAS ウィンドウからの印刷*

### *概要:SAS ウィンドウからの印刷*

Windows 版 SAS では、他の Windows アプリケーションと同様、ツールバーボタンやダイアログボックスを使用して印刷します。印刷オプションを設定するには**印刷**、**印刷設定**、**ページ設定**の各ダイアログボックスを使用します。印刷プレビューウィンドウで、印刷する内容を確認することもできます。

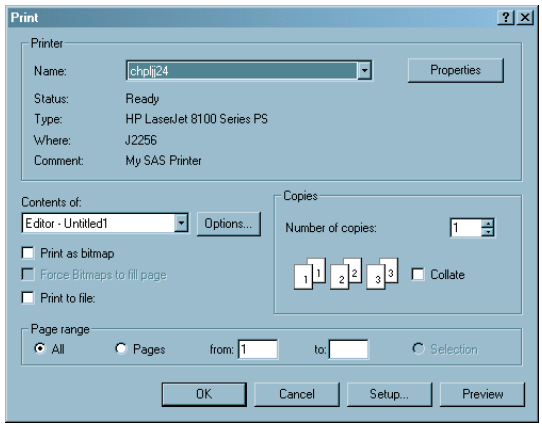
### *印刷オプションの設定*

**印刷**ダイアログボックスでは、次の印刷オプションを設定します。

- 出力先プリンタの変更
- 印刷するウィンドウの指定
- 行番号印刷、ページ番号印刷、カラー印刷
- ビットマップとして印刷
- ファイルへの印刷
- クリップボードの内容の印刷
- 複数部数の印刷
- 印刷範囲または選択したテキストの印刷
- 部単位の印刷

**印刷**ダイアログボックスにアクセスするには、次のメニューパスを選択します: **ファイル** ⇨ **印刷**

***図 6.1*** *印刷ダイアログボックス*

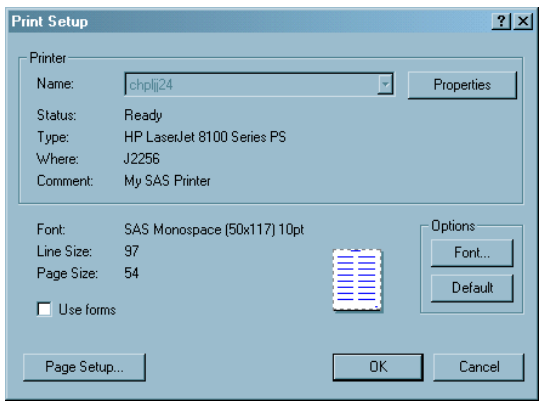



**印刷設定**ダイアログボックスでは、次の印刷オプションを設定します。

- フォントの変更
- フォームの使用

**印刷設定**ダイアログボックスにアクセスするには、次のメニューパスを選択します: **ファイル** ⇨ **印刷設定**

***図 6.2*** *印刷設定ダイアログボックス*

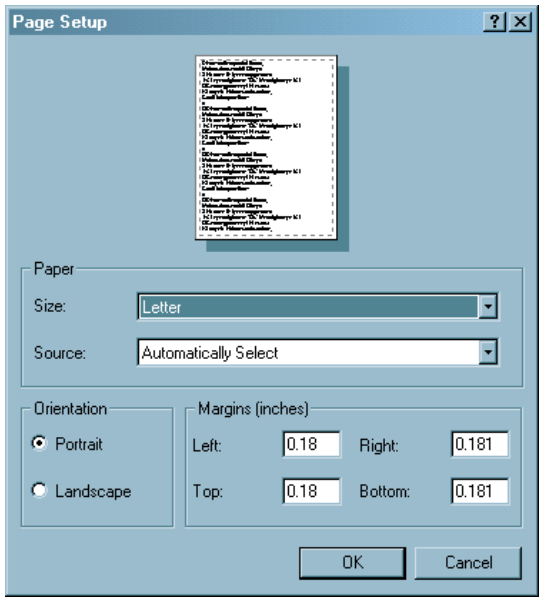



**ページ設定**ダイアログボックスでは、次の印刷オプションを設定します。

- 用紙サイズの変更
- 給紙方法の変更
- 印刷の向き(縦または横)の指定
- 余白サイズの設定

**ページ設定**ダイアログボックスにアクセスするには、次のメニューパスを選択します: **ファイル** ⇨ **ページ設定**

*図 6.3 ページ設定ダイアログボックス*




選択したプリンタのドキュメントプロパティを指定するには、次のメニューパスを選択します。

**ファイル** ⇨ **印刷** ⇨ **プロパティ**

### *印刷可能なウィンドウ*

一部の SAS ウィンドウは、印刷ができません。印刷可能な SAS ウィンドウかどうかを確認するには、ウィンドウをアクティブウィンドウにします。**印刷**ツールバーボタンかファイルメニューの印刷コマンドがアクティブになっていれば、そのウィンドウは印刷可能です。

**印刷**ツールバーボタンがアクティブで、[ファイル]メニューの[印刷]コマンドがアクティブではない場合は、ウィンドウのビットマップが印刷されます。ビットマップとして印刷されるウィンドウには **SAS システムオプション**ウィンドウなどがあります。

### *ウィンドウの印刷*

ウィンドウの内容を印刷するには、ウィンドウをアクティブウィンドウにして、次のいずれかの操作を実行します。

- 現在の印刷設定を使用して印刷するには、**印刷**ツールバーボタンをクリックします。
- 印刷オプションを変更して印刷するには、**ファイル** ⇨ **印刷**を選択して、印刷オプションを選択します。

**印刷**ダイアログボックスに表示される項目は、SAS の実行に使用する Windows 動作環境とアクティブな SAS ウィンドウによって異なります。

### *プリンタの変更*

デフォルトプリンタの設定は、次の順序で検索されます。

1. SYSPRINT システムオプションの値
2. Windows のデフォルトプリンタ

出力先プリンタは SYSPRINT システムオプションの値によって決まり、**印刷**ダイアログボックスの**名前**ボックスに表示されます。

プリンタを変更するには

1. **ファイル** ⇨ **印刷**を選択します。
2. **名前**リストボックスをクリックして、プリンタを選択します。

次の表に、SAS 起動時に SYSPRINT と PRTPERSISTDEFAULT のシステムオプションを使用してプリンタを指定する方法を示します。

***表 6.1*** *SAS 起動時のプリンタの指定*

| プリンタ | 操作 |
|---|---|
| Windows のデフォルトプリンタ | SAS 起動時に SYSPRINT システムオプションを指定しません。 |
| 特定のプリンタ | SAS 起動時に SYSPRINT システムオプションを指定します。 |
| 前の SAS セッションで指定されたプリンタ | SAS 起動時に必ず PRTPERSISTDEFAULT システムオプションを指定します。 |

SAS 起動時に SYSPRINT と PRTPERSISTDEFAULT の両方のシステムオプションが指定された場合、出力先プリンタは SYSPRINT システムオプションの値により決定されます。これらのシステムオプションの詳細については、“SYSPRINT システムオプション:Windows” (578 ページ) および “PRTPERSISTDEFAULT システムオプション]Windows” (549 ページ) を参照してください。

OPTIONS ステートメントまたは **SAS システムオプション**ウィンドウを使用して、出力先プリンタを変更することもできます。**SAS システムオプション**ウィンドウを使用して出力先プリンタを変更するには

1. **ツール** ⇨ **オプション** ⇨ **システム**を選択します。
2. **ログ、プロシジャ出力コントロール**フォルダを選択して、**プロシジャ出力**フォルダを選択します。
3. **Sysprint** をダブルクリックします。
4. Windows のプリンタフォルダに表示されているプリンタ名を**新しい値**ボックスに入力し、**OK** をクリックします。プリンタ名では、大文字と小文字が区別されます。

プリンタ名では、大文字と小文字が区別されます。**印刷**ダイアログボックスのプリンタ情報(**状態**、**種類** , **場所**、**コメント**のフィールド)には、Windows 動作環境から取得された情報が表示されます。

### *印刷フォントの変更*

印刷フォントオプションを使用して、フォント、フォントスタイル、ポイントサイズ、スクリプトを変更できます。フォントのサイズを変更すると、**印刷設定**ダイアログボックスの LINESIZE と PAGESIZE の最大値が再計算されます。

印刷フォントを指定するには

1. 次のメニューパスを指定します。**ファイル** ⇨ **印刷設定** ⇨ **フォント**
2. **フォント**、**フォントスタイル**、**サイズ**を選択します。

3. **OK** をクリックします。

*注:* SAS テーブルレポートと SAS カラムレポートは、モノスペースフォントを想定してフォーマットされます。したがって、プロポーショナルフォントを使用すると、正しくフォーマットされていないレポートが作成される場合があります。

SAS 起動時に SYSPRINTFONT システムオプションを指定する、OPTIONS ステートメントを指定するまたは **SAS システムオプション**ウィンドウを使用して、フォントを変更することもできます。SYSPRINTFONT システムオプションを指定する場合、フォント名やプリンタ名のスペルは完全に同一である必要があります。

**SAS システムオプション**ウィンドウを使用して SYSPRINTFONT を変更するには

1. **ツール** ⇨ **オプション** ⇨ **システム**を選択します。
2. **ログ、プロシジャ出力コントロール**フォルダを選択して、**プロシジャ出力**フォルダを選択します。
3. **Sysprintfont** を右クリックし、**値の変更**を選択します。
4. **新しい値** テキストボックスにフォント値を入力します。値はかっこで囲む必要があります。
5. **OK** をクリックします。

次の SYSPRINTFONT システムオプションは、プリンタ"second-floor"のフォントに Arial、太字、斜体を設定します。

```
sas -sysprintfont="Arial" bold italic named "second-floor";
```

詳細については、“SYSPRINTFONT システムオプション:Windows” (579 ページ) を参照してください。

### *印刷ページの設定*

ページ設定では、用紙、印刷の向き、余白を指定できます。ページを設定するには、**ページ設定** ダイアログボックスまたはシステムオプションを使用します。

**ページ設定**ダイアログボックスを開くには、**ファイル** ⇨ **ページ設定**を選択します。

次の表に、**ページ設定**ダイアログボックスのオプションと関連システムオプションを示します。

***表 6.2*** *印刷ページを設定するためのオプション*

| ページ設定オプション | 説明 | オプションの設定 | 関連するシステムオプション |
|---|---|---|---|
| 印刷の向き | 縦向きに印刷するか、横向きに印刷するかを指定します。 | 縦向きに印刷するには、**縦**を選択します。<br>横向きに印刷するには、**横**を選択します。 | 印刷の向き |
| 余白 | 用紙の上端、下端、左端、右端から空けるスペースの量を指定します。 | インチ数を**左**、**右**、**上**、**下**の各フィールドに入力します。 | LEFTMARGIN、RIGHTMARGIN、TOPMARGIN、BOTTOMMARGIN |

| ページ設定オプション | 説明 | オプションの設定 | 関連するシステムオプション |
|---|---|---|---|
| 用紙サイズ | 印刷する用紙のサイズを指定します。用紙の種類のリストは、次の表を参照してください。 | **サイズ**ボックスをクリックし、用紙サイズを選択します。 | PAPERSIZE |
| 給紙方法 | 用紙トレイ、封筒、手差しなどの給紙方法を指定するか、自動給紙を指定します。 | **給紙方法**ボックスをクリックし、給紙方法を選択します。 | PAPERSOURCE |

OPTIONS ステートメントにシステムオプションを指定するまたは **SAS システムオプション**ウィンドウを使用して、ページ設定オプションを設定することもできます。**SAS システムオプション**ウィンドウでページ設定オプションを設定するには

1. **ツール** ⇨ **オプション** ⇨ **システム** ⇨ **ログ、プロシジャ出力コントロール** ⇨ **ODS プリント**を選択します。
2. システムオプションの上にカーソルを置いて、右マウスボタンをダブルクリックします。
3. 新しい値を入力して、**OK** をクリックします。

***注意:***
**Windows の印刷ダイアログボックスを使用して印刷オプションを変更すると、SAS の印刷システムオプションの値が変更され、これが原因で予想した出力が得られない可能性があります。**

*注:* LINESIZE や PAGESIZE など、SAS システムオプションを使用して印刷オプションを設定した後、Windows の印刷ダイアログボックスを使用して印刷オプションを設定した場合、SAS システムオプションは Windows の印刷ダイアログボックスに指定した値に設定されます。

特定の用紙サイズのサポートは、プリンタ依存です。次に、用紙サイズ名のリストを示します。

LETTER
Letter、8.5X11 インチ

LEGAL
Legal、8.5X14 インチ

A4
A4 シート、210X297 mm

CSHEET
C シート、17X22 インチ

DSHEET
D シート、22X34 インチ

ESHEET
E シート、34X44 インチ

LETTERSMALL
Letter Small、8.5X11 インチ

TABLOID
Tabloid、11X17 インチ

LEDGER
Ledger、17X11 インチ

STATEMENT
Statement、5.5X8.5 インチ

EXECUTIVE
Executive、7.25X10.5 インチ

A3
A3 シート、297X420 mm

A4SMALL
A4 小シート、210X297 mm

A5
A5 シート、148X210 mm

B4
B4 シート、250X354 mm

B5
B5 シート、182X257 mm

FOLIO
Folio、8.5X13 インチ

QUARTO
Quarto、215X275 mm

10X14
10X14 インチ

11X17
11X17 インチ

NOTE
Note、8.5X11 インチ

ENV_9
#9 封筒、3.875X8.875 インチ

ENV_10
#10 封筒、4.125X9.5 インチ

ENV_11
#11 封筒、4.5X10.375 インチ

ENV_12
#12 封筒、4.75X11 インチ

ENV_14
#14 封筒、5X11.5 インチ

ENV_DL
DL 封筒、110X220 mm

ENV_C5
C5 封筒、162X229 mm

ENV_C3
C3 封筒、324X458 mm

ENV_C4
C4 封筒、229X324 mm

ENV_C6
C6 封筒、114X162 mm

ENV_C65
C65 封筒、114X229 mm

ENV_B4
B4 封筒、250X353 mm

ENV_B5
B5 封筒、176X250 mm

ENV_B6
B6 封筒、176X125 mm

ENV_ITALY
Italy 封筒、110X230 mm

ENV_MONARCH
Monarch Envelope、3.875X7.5 インチ

ENV_PERSONAL
6.5 封筒、3.625X6.5 インチ

FANFOLD_US
U.S.Standard Fanfold、14-7/8X11 インチ

FANFOLD_STD- _GERMAN
German Standard Fanfold、8.5X12 インチ

FANFOLD_LGL- _GERMAN
German Legal Fanfold、8.5X13 インチ

### *行サイズとページサイズに影響する印刷オプション*

行サイズとは、1 行に収まる文字数をいいます。ページサイズとは、1 ページの行数をいいます。**ページ設定**ダイアログボックスに表示される行サイズとページサイズは、次の印刷オプションに基づいて自動的に計算されます。

***表 6.3*** *行サイズとページサイズに影響する印刷オプション*

| 印刷設定ダイアログボックス | ページ設定ダイアログボックス | フォントダイアログボックス |
|---|---|---|
| • プリンタ | • 用紙サイズ<br>• 給紙方法<br>• 印刷の向き<br>• 余白設定 | • フォント<br>• フォントスタイル<br>• サイズ |

行サイズとページサイズは、ダイアログボックスから設定できませんが、これらの印刷設定を変更して調整できます。これらの印刷オプションを変更すると、LINESIZE システムオプションおよび PAGESIZE システムオプションも変更されます。

***注意:***

**Windows の印刷ダイアログボックスを使用して印刷オプションを変更すると、SAS の印刷システムオプションの値が変更され、これが原因で予想した出力が得られない可能性があります。** LINESIZE や PAGESIZE など、SAS システムオプションを使用して印刷オプションを設定した後、Windows の印刷ダイアログボックスを使用して印刷オプションを設定した場合、SAS システムオプションは Windows の印刷ダイアログボックスに指定した値に設定されます。

### *行番号、ページ番号、カラー印刷*

行番号印刷、ページ番号印刷、カラー印刷のオプションは、**その他の印刷オプション**ダイアログボックスにあります。このダイアログボックスを開くには、**オプション**を**印刷**ダイアログボックスでクリックします。

**オプション**ボタンは、これらのオプションの印刷を使用できるウィンドウでのみ有効化されます。

行番号とページ番号を印刷するには、ウィンドウの行番号をオンにしたり、NUMBER システムオプションを指定したりする必要はありません。カラー印刷は、出力先がカラープリンタであり、印刷するウィンドウがカラー印刷をサポートしている場合にのみ可能です。

### *ウィンドウのビットマップ形式での印刷*

次の表は、使用可能なビットマップ形式とその印刷方法を示します。

***表 6.4*** *ビットマップ形式の印刷*

| ビットマップ形式 | 印刷ダイアログボックスの選択 |
|---|---|
| アクティブな SAS ウィンドウの印刷 | **ビットマップで印刷** |
| SAS ウィンドウの印刷 | **ビットマップで印刷**<br>**内容**ボックスで、**AWS ウィンドウ (ビットマップ)**を選択します。 |
| スクリーン全体の印刷 | **ビットマップで印刷**<br>**内容**ボックスで、**全画面 (ビットマップ)**を選択します。 |
| ページ全体にビットマップを印刷 | **ページ全体にビットマップを印刷**を選択します。 |

### *印刷部数の設定*

**印刷**ダイアログボックスの**部数**ボックスで、部数を入力するか、上下矢印を使用して部数を選択します。プリンタが部単位の印刷をサポートしている場合は、**部単位で印刷**チェックボックスが使用できるようになります。

OPTIONS ステートメントまたは **SAS システムオプション**ウィンドウに COPIES システムオプションを設定して、部数を設定することもできます。

**SAS システムオプション**ウィンドウを使用して部数を設定するには

1. **ログ、プロシジャ出力コントロール** ⇨ **ODS プリント**を選択します。
2. **部数**をダブルクリックします。
3. **新しい値**ボックスに部数を入力し、**[OK]**をクリックします。

### *印刷するページ範囲の設定*

**印刷**ダイアログボックスの次の設定を使用して、印刷ページを選択します。

- すべてのページを印刷するには、**すべて**をクリックします。
- 印刷範囲を指定して印刷するには
  - **ページ**を選択します。

- **開始**ボックスに開始ページを入力します。
- **終了**ボックスに終了ページを入力します。

- ウィンドウ内で選択した部分のみ印刷するには、**選択した部分**を選択します。**選択した部分**オプションは、選択を行った場合にのみ使用できます。

## 印刷前の出力のプレビュー

### 印刷プレビュー

ウィンドウの内容がどのように印刷されるかを確認するには

1. プレビューするウィンドウを選択します。
2. 次のメニューパスを指定します。**ファイル** ⇨ **印刷プレビュー**

また、**印刷プレビュー**ツールボタンをクリックするまたはコマンドバーに `dlgprtpreview` を入力することもできます。

### 印刷プレビューウィンドウの機能

次の表に、**印刷プレビュー**ウィンドウの機能を示します。

*表 6.5 印刷プレビューウィンドウの機能*

| タスク | 操作 |
|---|---|
| ページ間の移動 | **次へ**または**前へ**の使用 |
| ズームインまたはズームアウト | **ズーム**またはページをクリックします。 |
| 現在のページの確認 | ステータス行に、現在のページとドキュメントのページ総数が表示されます。 |
| ヘルプの表示 | **ヘルプ**をクリックします。 |
| ウィンドウの印刷 | **印刷**をクリックします。 |
| **印刷プレビュー**ウィンドウを閉じる | **閉じる**をクリックします。 |

印刷プレビューウィンドウで使用可能なショートカットキーの一覧については、“印刷プレビュー内でのショートカットキー” (635 ページ) を参照してください。

### 印刷プレビューのショートカットキー

**印刷プレビュー**ウィンドウでは、次のショートカットキーを使用して移動できます。

*表 6.6 印刷プレビューのショートカットキー*

| キー | フルページモードのアクション | ズームモードのアクション |
|---|---|---|
| Page Down | 次のページに進む | 現在のページで下スクロールする |
| Page Up | 前のページに戻る | 現在のページで上スクロールする |

| キー | フルページモードのアクション | ズームモードのアクション |
|---|---|---|
| Ctrl+Page Down | なし | 現在のページで右スクロールする |
| Ctrl+Page Up | なし | 現在のページで左スクロールする |
| Ctrl+Home | 先頭ページへ移動する | 先頭ページへ移動する |
| Ctrl+End | 最終ページへ移動する | 最終ページへ移動する |

SAS ウィンドウによっては、印刷プレビュー機能をサポートしません。

## *SAS 印刷フォームの使用*

### *フォーム使用関連の印刷オプションの設定*

SAS から印刷フォームを使用するには

1. 次のメニューパスを指定します。**ファイル** ⇨ **印刷設定**
2. **フォームを使用する**チェックボックスを選択します。

**フォームを使用する**チェックボックスが**印刷設定ダイアログボックスに表示されない場合は、**PRTSETFORMS システムオプションを使用して有効化します。詳細については、“PRTSETFORMS システムオプション:Windows” (550 ページ) を参照してください。

印刷時、現在の印刷フォームを使用して出力が印刷されます。

バッチ SAS セッションでフォームを使用するには、NOHOSTPRINT システムオプションを使用します。NOHOSTPRINT を指定すると、**フォームを使用する**チェックボックスが選択され、SAS フォームに指定されている文字数、行数、フォント設定が使用されます。

### *現在の印刷フォームの指定*

印刷フォームを現在の印刷フォームに指定するには、次のいずれかの操作を行います。

- デフォルトのフォームを使用するには、コマンドバーに FORMNAME CLEAR を入力します。
- 特定のフォームを使用するには、コマンドバーに FORMNAME*form-name* を入力します。
- SAS 構成ファイルまたは **SAS システムオプション**ウィンドウに FORMS システムオプションを指定します。

現在のフォーム名を確認するには、パラメータを指定せずに FORMNAME コマンドを発行します。フォーム名が、ステータス行のメッセージ領域に表示されます。

FORMNAME コマンドは、ウィンドウによってはサポートされていません。サポートされていない場合は、次のメッセージがステータス行に表示されます。

```
ERROR: Unrecognized command FORMNAME
```

### 印刷フォームの作成

FSFORM コマンドは、SAS 出力の印刷時に使用する印刷フォームを定義する **FORM** ウィンドウを開きます。FORM エントリには、プリンタ、ページフォーマット、マージン、フォント、プリンタ制御言語を指定できます。

SAS 印刷フォームは、対話型 SAS セッションで PRINT コマンドを使用する場合や SAS/AF ウィンドウから印刷したりする場合に特に便利です。

**FORM** ウィンドウを起動するには、次のコマンドを発行します。

FSFORM *catalog-name.form-name;*

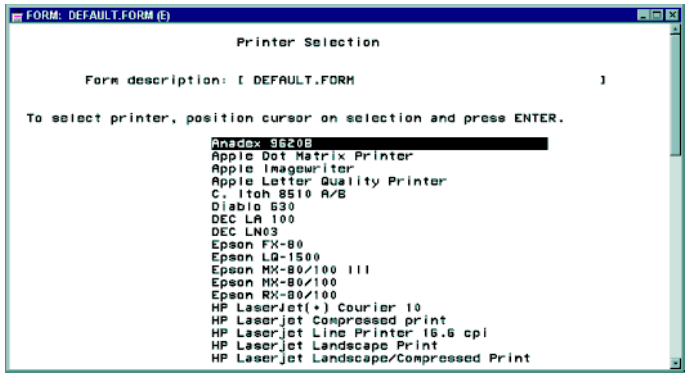
FSFORM コマンドの詳細については、SAS ヘルプとドキュメントを参照してください。

**FORM** ウィンドウのフレームの大半はどの操作環境でも同じですが、FSFORM コマンド発行後に最初に表示されるフレームはプリンタの選択フレームであり、これは Windows で使用できるプリンタのリストです。図 6.4 (177 ページ) には、このフレームのデフォルト情報が示されます。

プリンタの **FORM** ウィンドウのフレーム間で移動するには

1. プリンタ名をクリックするか、プリンタを選択して Enter キーを押します。
2. **ツール**メニューから**次のスクリーン** ⇨ **前の スクリーン**を選択して、フレーム間を移動します。

***図 6.4*** *プリンタの選択フレーム*




プリンタの選択フレームの情報もサイトに依存するため、サイトのプリンタリストは 図 6.4 (177 ページ) に示される情報とは異なります。

プリンタの選択フレームは、印刷フォームを新規作成した場合のみ表示されます。作成したフォームは、ユーザープロファイルカタログか、FSFORM コマンドで指定したカタログ(エントリタイプ FORM)に格納されます。この後、このフォームを修正する時は、プリンタの選択フレームは表示されません。**FORM** ウィンドウの 2 番目のフレームからプリンタの選択フレームに戻ることはできません。

## SAS コマンドを使用した印刷

アクティブウィンドウの内容を印刷する際、メニューを使用せずコマンドを入力する場合、PRINT または SPRINT コマンドを使用します。ウィンドウの中には、SPRINT コマンドが使用できないものもあります。PRINT コマンドまたは SPRINT コマンドの詳細については、SAS ヘルプとドキュメントを参照してください。

次の SAS コマンドは、印刷ダイアログボックスを開きます。

*表 6.7 印刷ダイアログボックスを開く SAS コマンド*

| コマンド | ダイアログボックス |
|---|---|
| DLGPRT | 印刷 |
| DLGPAGESETUP | ページ設定 |
| DLGPRTSETUP | 印刷設定 |
| DLGPRTPREVIEW | 印刷プレビュー |

## プリンタへの DATA ステップの出力の送信

ファイルではなくプリンタに DATA ステップ出力をスプールできます。このアクションを実行するには、次の例のように、FILENAME ステートメントと PRINTER デバイスタイプキーワードを使用します。

```
filename myfile printer;
data _null_;
   set sashelp.shoes;
   file myfile;
   where stores ge 25;
   put _all_;
run;
```

この例では、PRINTER デバイスタイプキーワードは、SYSPRINT システムオプションに指定されたプリンタへの出力を指定します。詳細については、“SYSPRINT システムオプション:Windows” (578 ページ) を参照してください。

## プリント出力のファイルへの送信

### 印刷ダイアログボックスを使用したファイルの印刷

印刷出力をファイルに送信するには、**ファイルへ出力**チェックボックス(**[印刷]**ダイアログボックス)を選択し、出力ファイル名を指定します。このアクションは、保存の操作とは異なります。生成されるプリンタファイルには、**プリンタ設定** ダイアログボックスで選択した、フォントや印刷の向きなどのオプションをサポートするために必要なプリンタ制御言語が、すべて格納されます。ほとんどの場合、このプリンタファイルは、テキストエディタでは判読できません。

### FILENAME ステートメントを使用して、ファイルに出力

FILENAME ステートメントを使用して、プリント出力をファイルに送信することができます。これは、DATA ステップ出力を送る場合に便利です。次に、例を示します。

```
filename myfile printer altdest='c:\results.dat';
data _null_;
   set sashelp.shoes;
   file myfile;
   where stores ge 25;
   put _all_;
run;
```

この例では、DATA ステップの出力はファイルに送信されます。このファイルには、プリンタを使用してフォーマットされた出力を生成するために必要なプリンタ制御言語もすべて格納されています。

### FILE デバイスの使用

ファイルにプリント出力を送信する方法として、**FILE**:デバイスに出力を送信する方法もあります。**プロパティ**ダイアログボックスの**ポート**ページ(Windows プリンタ)で指定します。この **FILE**:デバイスをプリンタに割り当てると、印刷を実行するたびに、ファイル名を入力するよう求められます。出力をファイルに送信するときに、そのファイルがすでに存在する場合は、既存ファイルの内容が上書きされます。プリンタプロパティの変更の詳細については、Windows のマニュアルを参照してください。

## バッチモード印刷

バッチモードで SAS ジョブを実行する場合、**印刷と印刷設定**のダイアログボックスにはアクセスできませんが、Windows プリンタは使用できます。デフォルトのプリンタを指定するには SYSPRINT システムオプションを使用します。プリンタフォントも指定する必要がある場合は、SYSPRINTFONT システムオプションも使用します。SYSPRINT システムオプションの詳細は、“SYSPRINT システムオプション:Windows” (578 ページ) を参照してください。たとえば、SAS 構成ファイルに次のオプションが含まれているとします。

```
-sysprint "f2hp5"
```

さらに、SAS プログラムに次のステートメントが含まれているとします。

```
filename myfile printer;
data _null_;
   set sashelp.shoes;
   file myfile;
   where stores ge 25;
   put _all_;
run;
```

ジョブをサブミットすると、SYSPRINT のプリンタ指定を使用して、DATA ステップの出力が Windows プリンタにスプールされます。

## デフォルトプリンタの詳細

デフォルトプリンタは、次の順序で検索されます。

- SYSPRINT システムオプションの値
- Windows システムのデフォルトプリンタ

SYSPRINT システムの値を表示するには、次のいずれかの方法で**印刷設定**ダイアログボックスを開きます。

- 次を選択します。**ファイル** ⇨ **印刷設定**
- コマンドバーに DLGPRTSETUP を入力

SAS 起動時、PRTPERSISTDEFAULT システムオプションを使用し、SYSPRINT システムオプションを使用しない場合、SYSPRINT システムオプションは前回の SAS セッションの出力先プリンタ名に設定されます。

プリンタ変更の詳細については、“プリンタの変更” (168 ページ) を参照してください。

### 印刷ジョブの取り消し

プリントファイルがフォルダまたは Windows プリンタにスプールされている間は、**キャンセル**を**印刷中止**ダイアログボックスでクリックします。**印刷中止**ダイアログボックスは、プリントファイルが出力先にスプールされている間だけ表示されます。ファイルが小さければ短時間で出力先にスプールされるため、印刷ジョブを取り消す前に**印刷中止**ダイアログボックスが閉じてしまいます。

**印刷中止**ダイアログボックスを表示または非表示にするには、PRTABORTDLGS システムオプションを指定します。有効な PRTABORTDLGS の値を次の表に示します。

*表 6.8 有効な値*

| 印刷の中止ダイアログボックスの表示タイミング | PRTABORTDLGS の値 |
|---|---|
| ファイルまたはプリンタへの出力中 | BOTH |
| 表示しない | NEITHER |
| ファイルへの出力中のみ | FILE |
| プリンタへの出力中のみ | PRINTER |

詳細については、"PRTABORTDLGS システムオプション:Windows" (549 ページ) を参照してください。

## プロシジャ出力の Web ブラウザへの表示

### プロシジャ出力の Web ブラウザへの表示について

Output Delivery System (ODS)を使用すると、プロシジャ出力を HTML 形式で作成して Web ブラウザに表示できます。ODS にはプロシジャ出力を RTF 形式で出力する機能もあります。Microsoft Internet Explorer がインストールされている場合は、HTML 形式および RTF 形式のプロシジャ出力を、結果ビューアを使用してメイン SAS ウィンドウに表示できます。RTF 形式のプロシジャ出力を表示するには、Microsoft Word などの RTF ビューアが必要です。

IE を使用していない場合、HTML 形式のプロシジャ出力は、**プリファレンス**ダイアログボックスの **Web** シートで指定したブラウザに表示されます。IE をインストールすると、HTML 出力をメイン SAS ウィンドウに表示できます。

ODS の使用方法の詳細については、*SAS Output Delivery System: ユーザーガイド*を参照してください。

### HTML 出力のプリファレンスの構成

HTML 出力を作成し、表示するには、**結果**タブと **Web** タブ(**プリファレンス**ダイアログボックス)を構成します。

**プリファレンス**ダイアログボックスにアクセスするには、**ツール** ⇨ **オプション** ⇨ **プリファレンス**を選択します。

**結果**ページ(**プリファレンス**ダイアログボックス)を構成するには:

1. [**結果**]タブをクリックします。
2. **HTML を作成する**チェックボックスを選択します。
3. HTML ファイルの保存場所を選択します。現在の SAS セッションがアクティブである間だけ HTML ファイルを保存するには、**WORK フォルダを使用する folder** チェックボックスを選択します。HTML ファイルを一時フォルダに保存すると、プロシジャのテスト時や多数の HTML ファイルを作成する際に便利です。その一例として、SAS の終了時に一時ファイルがすべて削除されることなどがあります。

   HTML ファイルを WORK フォルダ以外のフォルダに保存するには、**WORK フォルダを使用する** チェックボックスの選択を解除します。次に、パスを**フォルダ**テキストボックスに入力するか、**参照**をクリックしてフォルダを検索します。
4. **スタイル**ボックスをクリックし、スタイルを選択します。スタイルは、出力の色とフォントを定義します。*SAS Output Delivery System: ユーザーガイド*で、10 章: "TEMPLATE Procedure: Overview" (*SAS 9.4 Output Delivery System: Procedures Guide*) を使用して、スタイルを定義します。
5. 結果をすぐに表示するは、**すぐに結果ビューアを表示する**を選択します。
6. 結果ビューアを使用して HTML 出力を表示する場合は、**結果を見る**で、**内部ブラウザ**を選択します。また、**Web** タブで指定した Web ブラウザに HTML 出力を表示する場合は、**標準システムブラウザ**を選択します。Microsoft Internet Explorer がインストールされていない場合は、**標準システムブラウザ**を選択する必要があります。

**Web** ページ(**プリファレンス**ダイアログボックス)を構成するには:

- **Web** タブをクリックします。
- 使用するブラウザを選択します。
  - **標準システムブラウザを使用する**ラジオボタンは、Windows レジストリに登録されたデフォルトブラウザに HTML 出力を表示する場合に選択します。Windows レジストリの表示方法については、Windows のドキュメントを参照してください。
  - **その他ブラウザ**ラジオボタンは、デフォルトブラウザ以外の Web ブラウザに HTML を表示する場合に選択します。次に、ブラウザのパスとファイル名を入力するか、**参照**をクリックしてブラウザのファイル名を指定します。

## *結果ビューアウィンドウの使用*

### *内部ブラウザの構成*

**内部ブラウザ**を、**結果**タブ(**プリファレンス** ダイアログボックス)で選択した場合、Output Delivery System によって作成された HTML 出力は、**結果ビューア**ウィンドウに表示されます。**結果**ウィンドウに表示された HTML 出力やシステム内の HTML ファイルは、自由に開くことができます。

### *ツールバーの使用*

HTML 出力の結果ビューアでは、次のツールバーボタンを使用できます。

戻る
すでに結果ビューアウィンドウで開いたファイルに戻るには、左矢印ボタンを選択します。

進む
すでに結果ビューアで開いたファイルに進むには、右矢印ボタンを選択します。

中止
ファイルのロードを中止するには、信号機のボタンを選択します。

更新
現在のファイルを再ロードするには、輪になった矢印のボタンを選択します。

サイズ
**二重になった A** のボタンを選択して、フォントのサイズを変更します。5 種類のフォントサイズから選択できます。

### *HTML ファイルの表示*

HTML 形式のプロシジャ出力は、ファイルを**結果**ウィンドウから開いて表示できます。プロシジャ出力を生成と同時に表示するには、**結果**タブ(**プリファレンス** ダイアログボックス)の

1. **ツール** ⇨ **オプション** ⇨ **プリファレンス**を選択します。
2. [**結果**]タブを選択します。
3. **すぐに結果ビューアを表示する**を選択します。
4. **OK** をクリックします。

**結果**ウィンドウから別の結果ビューアウィンドウを開くには、次の操作を行います。

1. [**結果**]ウィンドウを選択します。
2. マウスの右ボタンで出力データセット名をクリックします。
3. **新しいウィンドウで開く**を選択します。

使用しているシステム内の HTML ファイルを開くには、次の操作を行います。

1. **結果ビューア**ウィンドウをアクティブウィンドウにします。
2. **開く**ツールバーボタンをクリックします。
3. HTML ファイルを**ファイルを開く** ダイアログボックスで指定します。
4. **OK** をクリックします。

HTML 出力を編集するには、次の操作を行います。

1. **結果**ウィンドウを使用して、マウスの右ボタンで出力データセット名をクリックします。
2. **ソースの編集**を選択します。
3. HTML ファイルを編集します。
4. **保存**ツールバーボタンをクリックします。
5. **更新**ツールバーボタンを選択して、更新した HTML ファイルを結果ビューアに表示します。

# ファイルへのプロシジャの出力と SAS ログの送付について

## ファイルへのプロシジャの出力と SAS ログの送付の概要

このセクションでは、SAS プロシジャ出力と SAS ログをファイルに送る際によく使用される方法を紹介します。一般に、このタスクは動作環境間で違いはありません。このタスクの説明は、SAS ヘルプとドキュメントにあります。ただし、外部ファイル名とデバイスの指定は、システムによって異なります。外部ファイルの参照方法の詳細は、“外部ファイルの参照” (148 ページ) を参照してください。

SAS プロシジャ出力と SAS ログをファイルに送るには、複数の方法があります。選択できる方法は、SAS の稼動方法、出力または SAS ログを送るタイミング、およびユーザーの設定によって異なります。

SAS プロシジャ出力または SAS ログをファイルに送るときは、次のような方法を使用することができます。

- **名前を付けて保存**ダイアログボックスは、**ファイル**メニューまたは保存するウィンドウのポップアップメニューから呼び出すか、コマンドバーに DLGSAVE コマンドを入力して呼び出します。
- FILE コマンド
- PRINTTO プロシジャ
- PRINT や ALTLOG など、さまざまなシステムオプション

## 名前を付けて保存ダイアログボックスの使用

最も簡単にアクティブウィンドウの内容をファイルに保存するには、保存する出力内容がアクティブウィンドウ(ログ、アウトプットウィンドウなど)に含まれていることを確認して、メイン SAS ウィンドウから**ファイル** ⇨ **名前を付けて保存**を選択します。**名前を付けて保存**ダイアログボックスの詳細については、拡張エディタからの “ファイルの保存” (82 ページ) または、プログラムエディタからの “ファイルの保存” (114 ページ) を参照してください。

## PRINTTO プロシジャの使用

SAS のバッチセッションでは、SAS プロシジャ出力と SAS ログは、それぞれデフォルトで *filename*.LST ファイルと *filename*.LOG ファイルに書き込まれます。ここでの *filename* は、SAS ジョブの名前です。たとえば、SYSIN ファイルが MYPROG.SAS の場合は、プロシジャ出力ファイルの名前は MYPROG.LST、ログファイルの名前は MYPROG.LOG になります。これらのデフォルトのファイル名を使用するかわりに、選択した任意のファイルにプロシジャ出力を送ることもできます。ジョブに次のステートメントが使用されている場合、MYOUTPUT というファイル参照名が C:\SAS\FIRST.TXT ファイルに割り当てられます。次に、PROC PRINTTO ステートメントによって、すべての SAS プロシジャ出力が MYOUTPUT に割り当てられたファイルに送られます。

```
filename myoutput 'c:\sas\first.txt';
proc printto print=myoutput;
run;
data uspres;
```

```
   input pres $ party $ number;
   datalines;
Adams F 2
;
run;
proc print;
run;
```

このステートメント以降のプロシジャまたは DATA ステートメントで生成される出力は、すべて C:\SAS\FIRST.TXT ファイルに送られます。デフォルトのプロシジャ出力ファイルには送られません。デフォルトのファイルに戻す場合は、次の例のように、空の PROC PRINTTO ステートメントを実行します。

```
proc printto;
run;
data uspres2;
   input pres $ party $ number;
   datalines;
Lincoln R 16
Grant R 18
;
run;
proc print;
run;
```

このステートメントを発行すると、SAS プロシジャ出力がデフォルトの出力先 (*filename*.LST)にリダイレクトされます。この方法を使用すると、同一 SAS ジョブでもあっても要素ごとに、プロシジャ出力と SAS ログを異なるファイルに送ることができます。

*注:* プロシジャ出力を送ったファイルには、キャリッジコントロール文字が含まれることがあります。このファイルをプログラムエディタに読み込むときに、キャリッジコントロール文字を取り除くには、FILENAME ステートメントの RECFM=オプションに P を指定します。この指定は、プログラムエディタへのファイルの読み込み方法のみに影響し、ファイルそのものには影響しません。

特定のファイルに SAS ログを送る場合、PROC PRINTTO ステートメントで PRINT=オプションのかわりに LOG=オプションを使用します。PRINTTO プロシジャの詳細については、“PRINTTO プロシジャ:Windows” (435 ページ) および *Base SAS プロシジャガイド*を参照してください。

*注:* PRINTTO プロシジャを使用して SAS プロシジャ出力または SAS ログを送ると、**ステータス**ウィンドウにはバッチ出力の送信先の変更は反映されませんが、プロシジャ出力ファイルと SAS ログは *filename*.LST と *filename*.LOG に送られることが示されます。

## *SAS システムオプションの使用*

SAS 出力または SAS ログをファイルに送るには、SAS システムオプションを使用します。たとえば、SAS バッチジョブのプロシジャ出力を、デフォルトファイルのかわりに C:\SASOUTPUT\PROG1.TXT ファイルに送る場合は、次のコマンドを使用して SAS システムを呼び出します。

```
SAS -SYSIN C:\SASPROGS\PROG1
    -PRINT C:\SASOUTPUT\PROG1.TXT
```

この SAS コマンドを実行すると、PROG1.SAS プログラムが実行され、プロシジャ出力が C:\SASOUTPUT\PROG1.TXT ファイルに送られます。SAS ログを同様の方法で送信するには、PRINT システムオプションのかわりに LOG システムオプションを使用します。この他に関連する ALTPRINT システムオプションと ALTLOG システムオプショ

ンについては、“ALTPRINT システムオプション:Windows” (488 ページ) および “ALTLOG システムオプション:Windows” (487 ページ) を参照してください。

*注:* **ステータス**ウィンドウには、プロシジャの出力とログの出力先が記録されるときに、PRINT システムオプションおよび LOG システムオプションの値が反映されます。

## SAS ログ機能を使用したログメッセージの書き込み

SAS ログ機能では、ログイベントを分類、収集、フィルタリングして、さまざまな出力デバイスに書き込むことができます。ログ機能は、問題の診断と解決、パフォーマンスとキャパシティの管理、監査と法規制順守をサポートします。ログ機能には、次の機能があります。

- ログイベントは、概要レベルまたは詳細レベルのログを設定できる階層的な命名体系を使用して分類されます。
- ログイベントは、ファイル、オペレーティングシステム機能、データベース、クライアントアプリケーションなど、複数の出力先に送信できます。出力先ごとに、次の項目を指定できます。
  - レポートするイベントのカテゴリとレベル
  - メッセージのレイアウト(挿入するデータの種類、順序、形式など)
  - 診断レベルやメッセージコンテンツなどの基準に基づくフィルタ
- ログ診断レベルは、プロセスを開始、終了せずに、動的に調整できます。
- Application Response Measurement (ARM) 4.0 サーバーにより処理されるパフォーマンス関連イベントを生成できます。

ログ機能は、ほとんどの SAS Server プロセスで使用されます。SAS プログラム内でログ機能を使用することもできます。

Windows の場合、ログメッセージは Windows イベントビューアに書き込むことができます。

使用動作環境のログ機能の詳細については、*SAS ログ機能: 構成とプログラミングリファレンス*を参照してください。

## グラフィックの作成

### *ディスプレイ上でのグラフィックの作成*

ほとんどの場合、SAS/GRAPH プロシジャを実行すると、出力はモニタに自動的に表示されます。SAS/GRAPH デバイスドライバを指定する必要はありません。グラフィック表示に関する情報は Windows の情報ファイルに格納されており、対話型の SAS セッション中に SAS によって自動的に使用されます。

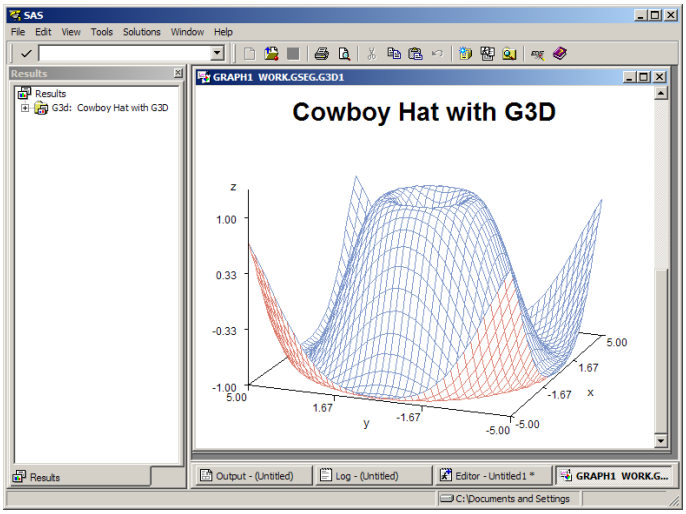
グラフィックを作成する簡単なコードの例を示します。

```
data hat;
   do x=-5 to 5 by .25;
      do y=-5 to 5 by .25;
         z=sin(sqrt(x*x+y*y));
```

```
        output;
      end;
   end;
proc g3d data=hat;
   plot y*x=z;
   title 'Cowboy Hat with G3D';
run;
quit;
```

次の図は、このプログラムの出力を示しています。

***図 6.5*** *Cowboy Hat プログラムの出力*




GOPTIONS ステートメントに DEVICE=オプションを使用してグラフィックをハードコピー装置に送った後で、モニタでの表示に戻す場合は、ドライバを指定する必要があります。グラフィック出力をディスプレイに表示するには、次のステートメントを実行します。

```
goptions device=win;
```

SAS ジョブをバッチモードで実行する場合も、ディスプレイ上にグラフィックを表示するには WIN デバイスドライバを使用する必要があります。

プログラム出力が HTML 形式で表示されるように指定すると、グラフィックが GIF ファイルに変換され、SAS データセットと同じフォルダ内に保存されます。詳細については、*SAS Output Delivery System: ユーザーガイド*を参照してください。

## *グラフィックの印刷*

### *グラフィック印刷の概要*

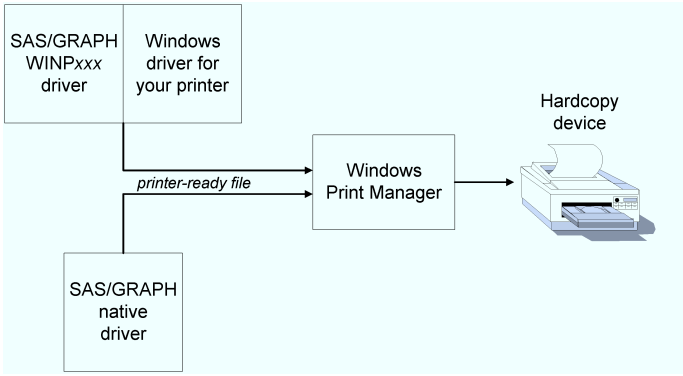
SAS/GRAPH の出力は、次の 2 通りの方法で印刷できます。

- デバイスの “SAS/GRAPH 汎用(WINPxxx)ドライバの使用” (187 ページ) (Windows または出力デバイスで提供されます)による方法。この方法では、SAS/GRAPH から Windows プリンタドライバに汎用グラフィックコマンドが送信され、プリンタドライバによってコマンドがプリンタで使用できる形式に変換されます。
- “SAS/GRAPH ネイティブプリンタドライバの使用” (188 ページ) により、プリンタ固有のグラフィックストリームを作成する方法。SAS/GRAPH プリンタドライバを使用すると、SAS は Windows プリンタドライバを回避し、使用しているデバイスがすぐに処理できるような出力を生成します。SAS/GRAPH プリンタドライバは、ターゲッ

トデバイスに固有のプリンタ言語を使用して出力を生成するため、SAS/GRAPH ネイティブプリンタドライバと呼ばれることもあります。Windows 用の SAS/GRAPH プリンタドライバは、メインフレームおよび UNIX 動作環境で SAS が使用するプリンタドライバと似ています。

Windows プリンタドライバまたは SAS/GRAPH プリンタドライバによって作成された出力は、Windows プリンタに送信され、ユーザーが選択したデバイスのプリントキューに格納されます。図 6.6 (187 ページ) は、これら 2 種類のプリンタドライバを使用して、特定のデバイスに対応した SAS/GRAPH の出力を生成する方法を示しています。

***図 6.6*** *SAS/GRAPH 汎用プリンタドライバと SAS/GRAPH ネイティブプリンタドライバとの比較*




プリンタドライバを選択する方法は、使用している出力デバイスによって異なります。詳細については、“SAS/GRAPH ネイティブドライバと WINPxxx ドライバの選択” (189 ページ) を参照してください。どちらの印刷方法も**印刷と印刷設定** ダイアログボックスまたは SYSPRINT=オプションと GOPTIONS DEVICE=ステートメントを使用して制御できます。

詳細については、次の項目を参照してください。

- “SAS/GRAPH 汎用(WINPxxx)ドライバの使用” (187 ページ)
- “SAS/GRAPH ネイティブプリンタドライバの使用” (188 ページ)
- “GRAPH ウィンドウからの印刷とプレビュー” (189 ページ)
- “SAS/GRAPH ネイティブドライバと WINPxxx ドライバの選択” (189 ページ)

### *SAS/GRAPH 汎用(WINPxxx)ドライバの使用*

SAS/GRAPH 汎用デバイスドライバと Windows プリンタドライバを使用してグラフィックを印刷するには、次の操作を行います。

1. **ファイル**メニューから**印刷設定**を選択します。**印刷設定**ダイアログボックスの**プリンタ**フィールドに適切な Windows プリンタドライバとポートが表示されていることを確認します。**印刷設定**ダイアログボックスでは、インストールしたプリンタドライバとポートを任意に組み合わせて選択できます。新しいドライバとポートの組み合わせをインストールするには、Windows のプリンタの追加ウィザードを使用します。

   または、SYSPRINT システムオプションを使用して、出力先プリンタを割り当てることもできます。たとえば、

   ```
   options sysprint='HP LaserJet III';
   ```

   割り当てることができるのは、Windows で構成済みのプリンタドライバ名のみです。

2. 次のグラフィックオプションで、SAS/GRAPH プログラムを実行します。

   ```
   goptions device=winp xxx;
   ```

WINP*xxx* には、グラフの印刷に使用する出力デバイスのタイプに基づいて、次のいずれかの値を指定します。

WINPRTM
白黒(モノクロ)プリンタの場合

WINPRTG
グレースケールプリンタの場合

WINPRTC
カラープリンタの場合

WINPLOT
プロッタの場合

グラフィック出力の向きは、次の方法で決定されます。

- ROTATE=グラフィックオプションを指定した場合、出力の向きは ROTATE に指定した値に基づいて決定されます。例えば次を指定するとします

```
goptions rotate=landscape
```

。**印刷設定**ダイアログボックスの設定にかかわらず、横向きで出力されます。

- ROTATE=グラフィックオプションを使用しない場合、出力の向きは**印刷設定**ダイアログボックスの設定に基づいて決定されます。

*注:* グラフィックの印刷には、**ページ設定** ダイアログボックスの余白設定が適用されます。グラフィックの印刷時に余白を変更するときに、グラフィックの比率を維持するには、上と下の余白に加える変更量を、左と右の余白に加える変更量に合わせます。

### *SAS/GRAPH ネイティブプリンタドライバの使用*

SAS/GRAPH ネイティブプリンタドライバでは、ターゲットデバイスのネイティブ言語で出力が生成されます。SAS/GRAPH ネイティブドライバの例を次に示します。

PS
ポストスクリプト出力を生成します。

HPLJS3
Hewlett Packard 製品の LaserJet III プリンタで使用される言語 PCL5 で出力を生成します

HP7550
Hewlett Packard 製品の 7550 プロッタで使用される HPGL 出力を生成します

SAS/GRAPH ネイティブプリンタドライバによってターゲットデバイスのネイティブ言語で生成された出力は、SAS によって Windows プリンタ経由でデバイスに送られます。SAS では、ターゲットデバイスに現在割り当てられている Windows ドライバを介さずに、**印刷設定**ダイアログボックスの設定に従って出力先が決定されます。

SAS/GRAPH プリンタドライバを使用してグラフを印刷するには、次のグラフィックオプションを使用して SAS/GRAPH プログラムを実行します。

```
goptions device=driver-name;
```

ここで、*driver-name* には、有効な SAS/GRAPH デバイスドライバの名前を指定します。次の例を考えてみます。

```
goptions device=hplj5p3;
```

このステートメントでは、Hewlett-Packard LaserJet Series V プリンタ用のグラフ形式でフォーマットされます。PROC GDEVICE ステートメントをサブミットすれば、すべての SAS/GRAPH ドライバのリストを出力できます。

グラフを直接プリンタに送らずに、グラフィックストリームファイルまたは GSF とも呼ばれるプリンタファイルに出力するには、GOPTIONS ステートメントの GSFNAME オプションに出力先のファイル名またはファイル参照名を指定します。例を次に示します。

```
filename graphout "graphpic.prn";
goptions gsfname=graphout gsfmode=replace
         device=hpljs2;
```

### *GRAPH ウィンドウからの印刷とプレビュー*

作成したグラフは、印刷前にプレビューし、選択したデバイスに適したフォーマットに変換することができます。グラフを印刷前にプレビューするには、次の GOPTIONS ステートメントを使用して SAS/GRAPH プログラムを実行します。

```
goptions targetdevice=driver-name;
```

ここで、*driver-name* には、WINP*xxx* ドライバまたは SAS/GRAPH ネイティブドライバを指定します。

出力先デバイスを指定すると、ターゲットプリンタに適したカラーと属性で、グラフのフォーマットが変換されます。プレビュー後にグラフを印刷するには、ファイルプルダウンメニューを選択し、**印刷**を選択します。

*注:* グラフの作成前に出力先のデバイスを指定しなかった場合、プレビューしているグラフを印刷するときに、SAS/GRAPH によって**印刷** ダイアログボックスでデバイスドライバ名の指定を要求されます。ほとんどの場合、WINPRTM または WINPRTC ドライバがデフォルトで指定されています。グラフのカラー、向き、およびサイズは、指定した出力デバイスに適していないことがあります。

### *SAS/GRAPH ネイティブドライバと WINPxxx ドライバの選択*

SAS/GRAPH ネイティブドライバまたは WINP*xxx* シリーズのドライバのどちらを使用するかを決めるときは、使用しているデバイスや生成する出力のタイプなどの要素を考慮する必要があります。次の点を考慮してください。

- 使用しているデバイス用の Windows プリンタドライバがない場合は、SAS/GRAPH ネイティブドライバを使用します。
- 使用しているデバイス用の SAS/GRAPH ネイティブドライバがない場合は、WINP*xxx* ドライバを使用します。ただし、そのデバイス用の Windows プリンタドライバが必要です。新しい SAS がリリースされる前に新しいモデルのハードコピーデバイスを入手し、そのデバイスの新機能を使用した Windows ドライバがベンダから提供された場合は、WINPxxx ドライバを使用すればその新機能を利用できます。
- グラフに使用されるサイズ指定を、HSIZE=や VSIZE=などのオプションを使用してカスタマイズする場合は、より信頼性の高い結果を得るため、SAS/GRAPH ネイティブドライバを使用することをお勧めします。
- SAS/GRAPH 出力で TrueType フォントを使用するには、WINPxxx ドライバを使用して、プリンタ用にインストールされたハードウェアフォントを指定する場合と同じ要領でフォントを指定します。TrueType フォントの詳細については、“SAS/GRAPH での TrueType フォントの使用” (625 ページ) を参照してください。

## 他のアプリケーションからのグラフィックのインポート

### 他のアプリケーションからグラフィックをインポートする

SAS/GRAPH では、他社のソフトウェアアプリケーションで作成されたビットマップグラフィックとベクトルグラフィックをインポートできます。この機能には、次の利点があります。

- 別のグラフィックエディタを使用してグラフィックを作成できます。SAS/GRAPH にグラフィックをインポートして専用デバイス上に出力できます。
- 他のアプリケーションのクリップアートとグラフを、SAS/GRAPH で作成したグラフとマージできます。

ビットマップグラフィックを、次の SAS ウィンドウにインポートできます。

- **GRAPH** ウィンドウ(インポートしたグラフィックは、現在のカタログの新しい GRSEG エントリになります)。
- グラフエディタ(インポートしたグラフィックが、現在のグラフに取り込まれます)。
- **イメージエディタ**ウィンドウ。(インポートしたグラフィックは新しいイメージになります)。

SAS では、次の 2 つの方法でビットマップグラフィックを SAS/GRAPH にインポートできます。

- グラフィックの作成に使用したアプリケーションからグラフィックを Windows のクリップボードにコピーします。次に、SAS セッションに切り替えて、**SAS GRAPH** ウィンドウに貼り付けます。“Windows クリップボードからのグラフィックの貼り付け” (190 ページ) を参照してください。
- **SAS GRAPH** ウィンドウ、グラフエディタまたはイメージエディタから、**イメージのインポート**ダイアログボックスを使用してグラフィックファイルをインポートします。“SAS/GRAPH ウィンドウのグラフィックファイルのインポート” (190 ページ) を参照してください。
- “Windows クリップボードからのグラフィックの貼り付け” (190 ページ)
- “SAS/GRAPH ウィンドウのグラフィックファイルのインポート” (190 ページ)

### Windows クリップボードからのグラフィックの貼り付け

Windows アプリケーションで作成したソースグラフィックは、Windows のクリップボードを使用することにより、SAS セッションにコピーできます。

1. グラフィックの作成に使用したアプリケーションでグラフィックを選択し、そのアプリケーションのコピー手順に従ってクリップボードにグラフィックをコピーします。
2. SAS セッションに切り替えます(SAS セッションが実行中でなければ起動します)。
3. **SAS GRAPH** ウィンドウをアクティブにしてから、**貼り付け** を**編集**メニューで選択します。グラフィックが **SAS GRAPH** ウィンドウに貼り付けられます。

### SAS/GRAPH ウィンドウのグラフィックファイルのインポート

SAS/GRAPH のインポートフィルタを利用して、他のアプリケーションで作成したグラフィックファイルを、SAS で利用できる形式に変換できます。

次の表に示す形式でファイルが保存されていれば、他のアプリケーションで作成したグラフィックをインポートできます。

*表 6.9 インポート可能なグラフィックファイル*

| グラフィックファイル形式 | ファイル拡張子 |
|---|---|
| Windows ビットマップ | BMP |
| Windows メタファイル | WMF |
| 拡張メタファイル | EMF |
| DIB (Device independent bitmap) | DIB |
| JPEG | JPG |
| GIF (Graphic Interchange Format) | GIF |
| TIFF (Tag Image File Format) | TIF |
| PC ペイントブラシ | PCX |
| TGA (Truevision Targa) | TGA |
| ESPI (Encapsulated PostScript Interchange) | PS |
| Portable Network Graphics | PNG |
| PCD (Photo CD image) | PCD |
| Portable Pixmap | PBM |
| X Window bitmap | XBM |
| XWD (X Window dump) | XWD |

ビットマップグラフィックを SAS/GRAPH にインポートするには、次の操作を実行します。

1. **GRAPH** ウィンドウをアクティブウィンドウにし、**イメージのインポート**を**ファイル**メニューから選択します。
2. **イメージのインポート**ダイアログボックスを使用して、ソースディレクトリとグラフィックファイルを選択します。**ファイルの種類**フィールドに、適切なファイル形式が表示されていることを確認します。このフィールドは、SAS/GRAPH で使用するインポートフィルタを示しています。ファイルの種類にすべてのファイルを選択すると、インポートするファイルの形式が自動的に検出されます。**OK** をクリックします。

   *注:* AUTO を指定してフェイル形式を自動検索しても、DIB、EMF、WMF ファイル形式は検出されません。

グラフの中に IMAGE カタログエントリを含めることもできます。IMAGE カタログエントリを含める方法の詳細については、SAS ヘルプとドキュメントを参照してください。

## 他のアプリケーションで使用するグラフィックのエクスポート

### 他のアプリケーションで使用するグラフィックのエクスポート

SAS では、SAS/GRAPH で作成したグラフィックを次に説明する方法でエクスポートし、他のワードプロセッサや DTP パッケージで使用したり、インターネットやイントラネット上で表示することができます。

- **GRAPH** ウィンドウ、グラフエディタ、イメージエディタからグラフィックをファイルにエクスポートします( “SAS/GRAPH ウィンドウからファイルへのグラフィックのエクスポート” (192 ページ) を参照)。
- Windows クリップボードの内容を、ビットマップなどの目的のアプリケーションに貼り付けます( “SAS/GRAPH から他の Windows アプリケーションへのグラフィックの貼り付け” (193 ページ) を参照)。
- SAS に組み込まれるドライバを使用して、特定のグラフィックパッケージで使用されるコンピュータグラフィックメタファイル(CGM)を作成します( “他のアプリケーションにエクスポートする CGM ファイルの作成” (193 ページ) を参照)。
- 他の Windows アプリケーションで使用される Windows メタファイルを作成します( “他のアプリケーションにエクスポートする WMF (Windows Metafile)ファイルの作成” (195 ページ) を参照)。

また、SAS/GRAPH を使用して、Web ブラウザで使用される GIF ファイルや VRML ファイル、Adobe Acrobat Reader で使用される PDF ファイルなど、さまざまな形式のグラフィックファイルも作成できます。このような種類のファイルの作成方法の詳細については、*SAS/GRAPH: Reference* および SAS/GRAPH に関する説明を SAS ヘルプとドキュメントで参照してください。

### SAS/GRAPH ウィンドウからファイルへのグラフィックのエクスポート

SAS/GRAPH には、作成したグラフィックを他のアプリケーション(スプレッドシートや DTP プログラムなど)で使用できる形式に変換するエクスポートフィルタがあります。

SAS/GRAPH で作成したグラフィックは、次の表に示す形式でエクスポートできます。

***表 6.10*** *エクスポート可能なグラフィックファイル*

| グラフィックファイル形式 | ファイル拡張子 |
|---|---|
| Windows ビットマップ | BMP |
| Windows メタファイル | WMF |
| 拡張メタファイル | EMF |
| DIB (Device independent bitmap) | DIB |
| JPEG | JPG |
| GIF (Graphic Interchange Format) | GIF |
| TIFF (Tag Image File Format) | TIF |
| Adobe PostScript | PS |

| グラフィックファイル形式 | ファイル拡張子 |
|---|---|
| ESPI (Encapsulated PostScript Interchange) | PS |
| Portable Network Graphics | PNG |
| Portable Pixmap | PBM |

**GRAPH** ウィンドウをアクティブウィンドウにし、ファイルプルダウンメニューから

1. **GRAPH** ウィンドウをアクティブウィンドウにし、**イメージのエクスポート**を**ファイル**メニューから選択します。
2. **イメージのエクスポート**ダイアログボックスで、ファイルの種類を選択します。
3. エクスポートするグラフィックのディレクトリとファイル名を指定します。**OK** をクリックします。

グラフィックを SAS IMAGE カタログエントリに**イメージ**エディタからエクスポートする方法については、SAS/GRAPH の SAS ヘルプとドキュメントを参照してください。

### *SAS/GRAPH から他の Windows アプリケーションへのグラフィックの貼り付け*

Windows のクリップボードを使用すると、SAS のグラフィックを他の Windows アプリケーションに簡単にエクスポートできます。SAS/GRAPH からクリップボードにコピーした情報は、DIB、BMP、WMF の入力に対応した任意のアプリケーションに貼り付けることができます。

SAS/GRAPH の情報をクリップボードにコピーするには、次の操作を行います。

1. **GRAPH** ウィンドウで左マウスボタンを押したまま、グラフのコピーする部分をドラッグします。マウスを移動すると、選択された領域を示す選択ボックスが表示されます。必要な領域の選択後、マウスボタンを離します。

   グラフの一部を選択せずに次のステップを行うと、グラフ全体がクリップボードにコピーされます。

2. **GRAPH** ウィンドウが引き続きアクティブな状態で、Ctrl+C を押します(または**貼り付けバッファにコピー**を**編集**メニューで選択)。

グラフがクリップボードにコピーされます。次に、ターゲットアプリケーションに切り替え、グラフを貼り付けます。**貼り付け**または**形式を選択して貼り付け**オプションを使用します。クリップボードの情報を貼り付ける方法の詳細については、貼り付け先の Windows アプリケーションのドキュメントを参照してください。

### *他のアプリケーションにエクスポートする CGM ファイルの作成*

SAS/GRAPH で作成したグラフは、他のグラフィックパッケージにエクスポートすることができます。この場合、そのグラフィックパッケージ専用のドライバを使用します。パッケージ間でグラフを転送する媒体として、コンピュータグラフィックメタファイル(CGM)ファイルを使用すれば、グラフには構成要素が保持されるため、各要素の編集とサイズ変更を行うことができます。利用できる編集機能は、ターゲットグラフィックパッケージに依存します。

SAS/GRAPH から CGM ファイルを作成するには、GOPTIONS を次のように設定します。

```
filename fileref 'filename.cgm';
goptions device=cgmxxxx gsfname=fileref
```

```
gsfmode=replace;
```

CGM*xxxx* には、ターゲットアプリケーション用の CGM ドライバを指定します。*filename*CGM には、作成するファイルの名前を指定します。CGMOFML と CGMOFMP は、すべての CGM 出力の推奨デバイスドライバです。CGMOFML と CGMOFMP が出力ソフトウェアに適さない場合、表 6.11 (194 ページ) のデバイスを使用できます。この表には、CGM をエクスポートできるグラフィックパッケージと推奨ドライバが示されます。

アスタリスク(*)が付いているドライバは、SAS 9.4 に同梱されています。無効化されているドライバには、表に device カタログでは無効と記されています。デバイスドライバを有効化するには、次の操作を行います。

1. ライブラリ GDEVICE0 に新しい場所を指定します。例を次に示します。

   ```
   LIBNAME GDEVICE0 'directory';
   ```

2. PROC CATALOG を使用して、エントリの一部またはすべてを GDEVICE0 の場所にコピーします。例を次に示します。

   ```
   PROC CATALOG C=SASHELP.DGDEVICE;
   COPY OUT=GDEVICE0.DEVICES;
   RUN;
   ```

CGM デバイスとグラフィックパッケージを使用する方法については、SAS Institute 社のテクニカルサポートに問い合わせてください。

***表 6.11*** *主なグラフィックパッケージ用の CGM ドライバ*

| パッケージ | 推奨ドライバ |
|---|---|
| Aldus PageMaker | CGMAPMA* (dgdevice カタログでは無効) |
| Aldus Persuasion | CGMAPSA* (dgdevice カタログでは無効) |
| BPS 35 MM Express | CGM35 (dgdevice カタログでは無効) |
| Borland Quattro Pro (Windows) | CGMBQWC (dgdevice カタログでは無効) |
| Borland Quattro Pro (DOS) | CGMBQA* (dgdevice カタログでは無効) |
| Frame Tech FrameMaker | CGMFRCA* (dgdevice カタログでは無効)<br>CGMFRGA* (dgdevice カタログでは無効)<br>CGMFRMA* (dgdevice カタログでは無効) |
| DOS 用 Harvard Graphics 2.12 | CGHHG (dgdevice カタログでは無効) |
| DOS 用 Harvard Graphics 3.0 | CGMHG3A* (dgdevice カタログでは無効) |
| Window 用 Harvard Graphics | CGMHGWA* (dgdevice カタログでは無効) |
| ImageBuilder | CGMIMG (dgdevice カタログでは無効) |
| Interleaf 5 | CGMCILFC* (dgdevice カタログでは無効)<br>CGMGILFG* (dgdevice カタログでは無効)<br>CGMMILFM* (dgdevice カタログでは無効) |

| パッケージ | 推奨ドライバ |
|---|---|
| Window 6.0 用 Microsoft Word | CGMMW6C* (dgdevice カタログでは無効) |
| Window 2.0 用 Microsoft Word | CGMMWWC* (dgdevice カタログでは無効) |
| Microsoft PowerPoint | CGMMPPA (dgdevice カタログでは無効) |
| Microsoft Office 97 | CGMOF97L または CGMOFML、<br>CGMOF97P または CGMOFMP |
| Polaroid CI3000 | CI3000 (dgdevice カタログでは無効) |
| DOS 用 WordPerfect 5.1 | CGMWPCA (dgdevice カタログでは無効)<br>CGMWPCAP* (dgdevice カタログでは無効)<br>CGMWPGA (dgdevice カタログでは無効)<br>CGMWPGAP* (dgdevice カタログでは無効)<br>CGMWPMA (dgdevice カタログでは無効)<br>CGMWPMAP* (dgdevice カタログでは無効) |
| Window 用 WordPerfect 5.2 | CGMWPWA* (dgdevice カタログでは無効) |
| DOS 用 WordPerfect Presents | CGMWPCA* (dgdevice カタログでは無効)<br>CGMWPGA* (dgdevice カタログでは無効)<br>CGMWPMA* (dgdevice カタログでは無効) |
| Zenographics Pixie | CGMPIX (dgdevice カタログでは無効) |

### *他のアプリケーションにエクスポートする WMF (Windows Metafile) ファイルの作成*

SAS/GRAPH の WMF ファイルをエクスポートする方法については、SAS Institute 社のテクニカルサポートに問い合わせて、ターゲットソフトウェアの情報を入手してください。

## *補足*

SAS/GRAPH ソフトウェアの使用法の詳細については、*SAS/GRAPH: Reference* を参照してください。Windows 版 SAS で使用するグラフィックとフォントの詳細については、“SAS/GRAPH での TrueType フォントの使用” (625 ページ) を参照してください。

*7 章*

# Windows 版でのパフォーマンスに関する注意点

**ハードウェアに関する注意点** . . . . . **197**
ハードウェアの注意点の概要 . . . . . 197
プロセッサの速度 . . . . . 198
メモリ . . . . . 198
I/O サブシステムの構成 . . . . . 198
**パフォーマンスを最適化する Windows の機能** . . . . . **199**
SAS の応答性の制御 . . . . . 199
複数プロセッサに対応した I/O 拡張 . . . . . 199
メモリベースのライブラリ . . . . . 200
**パフォーマンスを最適化する SAS の機能** . . . . . **203**
**ネットワークのパフォーマンスに関する注意点** . . . . . **204**
**sasiotest ユーティリティ** . . . . . **204**
**高度なパフォーマンス調整メソッド** . . . . . **205**
概要:高度なパフォーマンス調整メソッド . . . . . 205
SORT プロシジャのパフォーマンス向上 . . . . . 205
データセットのサイズの計算 . . . . . 207
対話型処理の効率の向上 . . . . . 209

## ハードウェアに関する注意点

### *ハードウェアの注意点の概要*

SAS のパフォーマンスには、次のハードウェアの要素が影響します。

- プロセッサの速度
- 使用可能な物理メモリ量
- I/O で使用可能な、ディスクへのスループットの容量
- グラフィックアダプタ

これらの要因のすべてが、ユーザーの特定の考慮事項や SAS の実行方法に該当するとは限りません。詳細は、システム管理者に問い合わせてください。

メモリやハードウェアの考慮事項などの詳細については、*Microsoft Windows 向け SAS 9.4 Foundation のシステム要件* および *Microsoft® Windows® for x64 向け SAS 9.4 Foundation のシステム要件* を参照してください。

## プロセッサの速度

一般に、プロセッサが高速であるほど、コンピュータが 1 秒間に実行可能な演算数は多くなります。実行可能な演算数が多いほど、パフォーマンスは向上します。

使用可能なプロセッサキャッシュの量も、パフォーマンスに影響します。キャッシュの量が多いほど、パフォーマンスは向上します。

- 32 ビット SAS には、Intel Pentium 4 クラスのプロセッサまたは Intel 互換のプロセッサを搭載した PC が必要です。
- x64 64 ビット SAS には、Intel64 または AMD64 プロセッサが必要です。

## メモリ

一般に、物理メモリが大きいほど、パフォーマンスは向上します。

## I/O サブシステムの構成

アプリケーションは、データの格納やデータへのアクセスに、I/O を使用します。したがって、I/O が高速なほど、全体のパフォーマンスは向上します。I/O のパフォーマンスに影響する重要な要素は、ディスクコントローラおよびバス、ハードドライブ、ハードドライブの構成です。

ディスクコントローラおよびバス
: 一般に、メモリバッファを使用してデータをキャッシュするハードドライブディスクコントローラでは、従来のコントローラよりスループットが高速化します。このため I/O のパフォーマンスも向上します。

  I/O コントローラの種類は、I/O のパフォーマンスに影響します。ストレージデバイスの構成については、ストレージ管理者に問い合わせるか、次に挙げるドキュメントを参照してください http://support.sas.com/rnd/papers/sgf07/sgf2007-iosubsystem.pdf。

ハードディスクドライブ
: SAS は I/O 指向であるため、システムのパフォーマンスにとって、アクセス時間と転送レートは重要な要素です。Windows サーバーの最大のパフォーマンスを発揮するには、SAS に使用されるファイルシステムあたり、少なくとも 1 秒間で 100MB の I/O スループットレートが推奨されます。

  *注:* IT/SAN 管理者は、ストレージ設定に対して最大の I/O スループットレートを出せるよう I/O サブシステムを構成することができます。

  ディスク容量が小さかったり、ディスクのフラグメント化が進んでいると、I/O のパフォーマンスは低下します。I/O パフォーマンスの低下を防ぐには、ハードディスクドライブを定期的にデフラグする必要があります。

  RAID システムに関する情報については、次のドキュメントを参照してください。
  http://support.sas.com/rnd/papers/sgf07/sgf2007-iosubsystem.pdf

ハードディスクドライブの構成
: 特に大規模なサーバーシステムの場合、ハードディスクドライブの構成は、I/O のパフォーマンスに最も強い影響を与えます。一般に、RAID 構成では、非 RAID 構成より I/O のパフォーマンスが向上します。ご使用のコンピュータに適した構成を決定するには、システム管理者に相談してください。

# パフォーマンスを最適化する Windows の機能

## SAS の応答性の制御

### SAS の応答性の制御の概要

アプリケーションのパフォーマンスレベルを変更して、SAS セッションの相対的応答性を制御できます。Windows のパフォーマンスオプションを使用すると、対話型プログラムとバックグラウンドプログラムのどちらのプログラムに、より多くのプロセッサ時間を割り当てるかを指定できます。アプリケーションのパフォーマンスレベルを決定する際、次のガイドラインを使用します。

- SAS を対話的に実行しているときに、セッションの応答時間を最短にするには、プログラムやアプリケーションのパフォーマンスオプションを設定します。
- SAS をバッチモードで実行しているときに、バッチジョブをより迅速に実行するには、バックグラウンドサービスのパフォーマンスオプションを設定します。

SAS アプリケーションのパフォーマンスを分析するには、Windows のパフォーマンスモニタで SAS パフォーマンスカウンタを指定します。詳細については、“パフォーマンスツール” (217 ページ) を参照してください。

### Windows でのアプリケーションパフォーマンスの最適化

Windows でアプリケーションパフォーマンスを最適化するには、次のステップに従ってください。

1. [コントロール パネル]を開きます。
2. **システムとセキュリティ**をクリックします。
3. **システム**を選択します。
4. **システムの詳細設定**をクリックします。
5. **詳細**タブを選択します。
6. **パフォーマンス**ボックスで**設定**をクリックし、**詳細**タブを選択します。
7. 対話モードの SAS セッションのパフォーマンスを最適化するには、**プログラム**を選択します。
8. バッチモードの SAS セッションのパフォーマンスを最適化するには、**バックグラウンドサービス**を選択します。
9. **OK** をクリックします。

## 複数プロセッサに対応した I/O 拡張

PC にマルチプロセッサが搭載されている場合、SAS では、I/O 拡張が備わった SMP (symmetric multiprocessing)を使用します。より多くの先読み処理が、Windows サーバー上のデータへのシーケンシャルアクセスが大量に発生するプロシジャに対して行われます。先読み処理は、Windows とそのディスクキャッシュ用の処理能力に余裕があるシステムでより多く行われます。

マルチプロセッサ対応の SMP 環境の場合、通常、次の説明が当てはまります。

- 複数のアプリケーションのサーバーとして使用されるマシンとして、SMP ベースのマシンが最もパフォーマンスに優れています。
- SMP ベースの SAS/CONNECT リモートコンピューティング環境が最もパフォーマンスに優れています。

## メモリベースのライブラリ

### メモリベースのライブラリの概要

MEMLIB オプションと MEMCACHE オプションを使用して、メモリベースの SAS ライブラリを作成できます。動作環境によっては、メモリベースの SAS ライブラリのサポートに拡張メモリまたは基本メモリを使用します。

Windows 動作環境において、32 ビット処理では 2 GB の物理メモリが動作環境に使用され、残る 2 GB の物理メモリをアプリケーションで使用できます。PC またはサーバーに 4 GB を上回るメモリがある場合、拡張メモリは 4 GB を超えるメモリとして定義されます。メモリはアプリケーションで使用可能です。拡張メモリはメモリベースのライブラリのサポートに使用できます。

一部の Windows 動作環境では拡張メモリがサポートされません。拡張メモリがサポートされないかインストールされていない動作環境においては、基本メモリによって MEMLIB オプションと MEMCACHE オプションがサポートされます。基本メモリは、32 ビット環境では 4 GB を超えないメモリとして定義され、64 ビット環境では全メモリとして定義されます。

メモリベースのライブラリを使用することで、ディスクに対する I/O が削減され、パフォーマンスが向上します。メモリベースのライブラリは次のような方法で使用できます。

- Work ライブラリのストレージ
- 入出力量が大きい SAS ライブラリの処理
- きわめて大きい SAS ライブラリのキャッシュ

サーバーでは、複数のプロセスがメモリベースのライブラリに同時にアクセスできます。

動作環境のセットアップが完了したら、メモリベースのライブラリにアクセスするために、MEMLIB および MEMCACHE システムオプションと MEMLIB オプションを LIBNAME ステートメントで使用します。

### 64 ビット版の Windows 環境で MEMLIB オプションと MEMCACHE オプションを指定

64 ビット版の Windows 環境では、16 テラバイト(TB)の仮想アドレススペースを使用するため、この環境では拡張メモリは必要ありません。SAS では、基本メモリを使用して、MEMLIB オプションと MEMCACHE オプションをサポートします。

***注意:***

**システムメモリを使い切ると、システムエラーが発生する可能性があります。**
MEMMAXSZ オプションを使用すると、MEMLIB オプションと MEMCACHE オプションに割り当てるシステムメモリの量を制限できます。

MEMLIB オプションと MEMCACHE オプションを使用するには、動作環境が次の必要条件を満たす必要があります。

- インテル製 64 ビットプロセッサまたはインテルと互換性がある 64 ビットプロセッサ。
- 64 ビット版 Windows

- 4 GB 以上の RAM。8 GB 以上のメモリを使用すると、より良いパフォーマンスを得られることがあります。
- Windows 版 SAS 9.4

設定手順は次のとおりです。

- MEMMAXSZ システムオプションを使用して、すべてのメモリをメモリベースのライブラリに使用するように設定します。“MEMMAXSZ システムオプション:Windows” (532 ページ) を参照してください。
- MEMBLKSZ システムオプションを使用して、メモリの一部(ブロックサイズ)をメモリベースのライブラリに使用するように設定します。“MEMBLKSZ システムオプション:Windows” (529 ページ) を参照してください。

### *メモリベースのライブラリとして SAS ライブラリを処理する*

メモリベースの処理に適した SAS ライブラリは、SAS セッション時に何度も参照、更新するデータを格納します。

メモリベースの WORK ライブラリの使用は、巨大な一時ファイルに複数回書き込む PROC SORT などのプロシジャに有益です。WORK ライブラリをメモリベースとして指定するには、SAS 起動時に MEMLIB システムオプションを指定します。

LIBNAME ステートメントで MEMLIB オプションを使用して、ライブラリをメモリベースに指定できます。WORK ディレクトリへのライブラリ参照名などのすべてのライブラリ参照名が、ディスク上の有効なディレクトリを含んでいる必要があります。

ライブラリをメモリベースに指定したら、SAS プログラムでディスクからメモリにライブラリをコピーする必要があります。メモリでのライブラリの処理後、ライブラリのコピーをディスクに戻す必要があります。

***注意:***

**処理の完了後、メモリ内のライブラリをディスクにコピーしてください。** コピーしないと、ライブラリに加えた変更がすべて失われます。SAS セッションが終了するときに、変更は失われます。

次に、LIBNAME ステートメントと PROC COPY ステートメントを使用してライブラリとメモリ間で相互にコピーする方法を示します。

```
/* Set up two librefs, one to the library in memory
and the other to the SAS library on disk.  The library on
disk contains dataset1, dataset2, dataset3 and dataset4. */

libname inmemory "g:\memlib" memlib;
libname ondisk "g:\disk";

/* Copy dataset1, dataset2, dataset3, and dataset4 to memory   */

proc copy in=ondisk out=inmemory;
run;

/* ...Assume dataset1 and dataset4 are updated   */

/* Save the updated datasets back to disk         */

proc copy in=inmemory out=ondisk;
   select dataset1 dataset4;
run;
```

次に示すように、DATA ステートメントを使用して、データセットをメモリにコピーすることもできます。

```
data ondisk.dataset1;
   set inmemory.dataset1;
run;
```

詳細については、“MEMLIB システムオプション:Windows” (532 ページ) および “LIBNAME ステートメント:Windows” (460 ページ) 。

### *SAS ファイルのキャッシュとしてのメモリベースのライブラリの使用*

SAS ファイルのキャッシュは、データを複数回参照する場合に最も役立ちます。たとえば、SAS ファイルのキャッシュを使用すると、データの受け渡しを複数行う SAS プログラムのパフォーマンスが向上します。次の場合に、SAS ファイルをキャッシュすると、パフォーマンスが向上します。

- 他のファイルの書き込み中に、ファイルの読み込み操作を繰り返す場合。ファイルに書き込むと、Windows file system (NTFS)キャッシュは消去されます。
- Scatter Gather I/O がアクティブなときにファイルの読み込み操作を繰り返す場合。Scatter Gather I/O 操作は、NTFS のキャッシュでは行われません。SAS ファイルをキャッシュしないとデータキャッシュは存在しないため、ディスクにアクセスして読み込み操作が行なわれます。

メモリを SAS ファイルのキャッシュとして使用するには、SAS 起動時または OPTIONS ステートメントをサブミットする際に、MEMCACHE システムオプションを指定します。MEMCACHE を 4 に設定すると、メモリにすべてのファイルがキャッシュされます。MEMCACHE を 1 に設定すると、現在メモリにあるファイルのみがメモリにキャッシュされます。次の例のように、OPTIONS ステートメントに MEMCACHE システムオプションを使用すると、SAS ファイルのキャッシュに使用するデータセットを制御できます。

```
/* Example of controlling cached files with the options statement   */

/* Assume cachelib contains 2 data sets, ds1 and ds2.               */
/* Also assume ds1 and ds2 are large enough that they cannot exist */
/* in the cache together. ds1 is read many times, so caching is     */
/* desired. ds2 is accessed only once, so caching is of no         */
/* benefit.  When you use the memcache option, ds1 is cached, and ds2   */
/* is not cached.                                                    */

libname cachelib "e:\tmp";

/* Turn on full caching */

options memcache = 4;

/* Read ds1 and place the data in the cache.  This read operation could be a  */
/* more useful read operation of the file in a real case.              */

data _null_;
  set cachelib.ds1;
run;

/* Change memcache setting to use the cache only for files that   */
/* already exist in the cache.                                    */

options memcache = 1;

/* Data from ds1 will come from the cache and ds2 will not be     */
/* cached.                                                        */
```

```
proc sort data=cachelib.ds1 out=cachelib.ds2;
  by j;
run;

/* Other access of ds1...                                          */

/* All use of the cache can be ended with a memcache system        */
/* option value of 0.                                              */

options memcache = 0;

/* Neither ds1 or ds2 will access the cache.                       */

proc sort data=cachelib.ds1 out=cachelib.ds2;
  by j;
run;
```

MEMCACHE システムオプションの詳細については、次を参照してください。“MEMCACHE システムオプション:Windows” (531 ページ) Scatter Gather I/O の詳細については、次を参照してください。“パフォーマンスを最適化する SAS の機能” (203 ページ) および “SGIO システムオプション:Windows” (562 ページ) 。

## パフォーマンスを最適化する SAS の機能

次に、システムパフォーマンスを向上し、コンピュータのリソースを効率的に使用するために制御可能な SAS の追加機能の一部を示します。SAS のパフォーマンスの最適化についての追加情報については、*SAS 言語リファレンス: 解説編*のシステムパフォーマンスの最適化についての章を参照してください。

- ASCII フラットファイルにアクセスする代わりに、SAS データセットを作成します。フラットファイルを読み込むより効率的に SAS データセットにアクセスできます。

  また、頻繁に使用する既存のデータセットを SAS 9.4 形式に変換する必要があります。

- SAS コードで、多重の IF-THEN 構造の代わりに、IF-THEN-ELSE 条件付き構造を使用します。IF-THEN-ELSE 構造で 1 つの条件が満たされている場合は、上位の構造に制御が戻されます。このとき、ELSE 句は読み飛ばされます。その ELSE 句の中に下位の IF-THEN 構造が含まれていても実行されません。多重の IF-THEN 構造が存在する場合は、各条件が検査されます。
- 配列を使用する際、可能であれば、配列を_TEMPORARY_にします。これにより、メモリの使用量を削減し、メモリを割り当てる時間を短縮します。
- ファイル入出力を減らすプログラミング構造を使用します。ファイル入出力は、SAS 処理で最も時間がかかる処理です。ファイル I/O を削減するには次の方法があります。
  - プロシジャで WHERE ステートメントを使用して、余分なデータ処理を減らします。
  - インデックス付きのデータセットを使用して、必要なオブザベーションへのアクセスを高速化します。
  - SQL プロシジャを使用して、データをサブセット化し、分類します。
- 一度に開ける SAS カタログの数を指定する CATCACHE システムオプションの値を使用します。デフォルトでは、カタログはメモリにキャッシュされず、CATCACHE

は 0 に設定されます。ある SAS アプリケーションで使用するカタログが、後で別の SAS アプリケーションでも必要になる場合は、カタログをキャッシュすると便利です。2 番目の SAS アプリケーションでは、カタログがキャッシュに格納されているため、より効率的にアクセスできます。

*注:* メモリにカタログを格納すると、大量のリソースを消費します。この方法はメモリに問題がない場合のみ、使用してください。

- データセットを圧縮形式で保存します(COMPRESS データセットオプションを使用)。この操作を行うと、SAS で使用する際にオブザベーションの解凍に CPU 時間がかかる場合がありますが、SAS アプリケーションのパフォーマンスは向上します。COMPRESS データセットオプションについては、*SAS データセットオプション: リファレンス*のデータセットオプションの項に記述されています。
- Scatter-read/Gather-write システムオプション(SGIO)を指定すると、データの読み込み/書き込み時に中間バッファ転送が不要になります。BUFNO システムオプションで指定したページ数を読み込み、必要になる前にデータをメモリに配置します。必要になったとき、データはすでにメモリ内にあります。これは、実質的にダイレクトメモリアクセスとなります。SAS ジョブごとに BUFNO システムオプションにさまざまな値を指定し、最大のパフォーマンス効果を得られる値を探してください。詳細は、次を参照してください。Microsoft Windows 環境で、SGIO を使用してより高速な I/O スループットを得るには

  Scatter-read / gather-write 機能は、アクセスパターンがシーケンシャルの場合、INPUT モード、OUTPUT モード、UPDATE モードで開かれた SAS 入出力にのみ有効です。どのような SAS の入出力ファイルも、UPDATE モードまたは RANDOM モードで開かれている場合には、SGIO は無効です。圧縮されたファイルや暗号化されたファイルも、Scatter-read/Gather-write 機能を使用して読み込めます。SGIO システムオプションの詳細については、次を参照してください。“SGIO システムオプション:Windows” (562 ページ) を参照してください。

## ネットワークのパフォーマンスに関する注意点

Windows 環境でネットワークドライブからアプリケーションのダイナミックリンクライブラリ(DLL)ファイルをロードすると、ローカルドライブから DLL ファイルをロードする場合よりもパフォーマンスが低下する可能性があります。

## sasiotest ユーティリティ

sasiotest ユーティリティは、Windows プラットフォームの I/O を測定します。このユーティリティは SAS の外部で動作します。sasiotest はファイルを開き、SAS I/O と同じ方法でデータの読み取りと書き込みを行います。詳細については、“Sasiotest ユーティリティ” (641 ページ)を参照してください。

# 高度なパフォーマンス調整メソッド

## 概要:高度なパフォーマンス調整メソッド

このセクションでは、SORT プロシジャのパフォーマンスの向上や、データセットサイズの計算などの高度なパフォーマンスに関するトピックを説明します。これらの方法は、経験豊富な SAS ユーザーや、使用しているコンピュータ上の SAS の構成に関して詳しい知識があるユーザーが実施してください。


- “SORT プロシジャのパフォーマンス向上” (205 ページ)
- “データセットのサイズの計算” (207 ページ)
- “対話型処理の効率の向上” (209 ページ)


## SORT プロシジャのパフォーマンス向上

### 概要:SORT プロシジャのパフォーマンスの向上

Windows 環境では、PROC SORT ステートメントの 2 つのオプション、SORTSIZE=オプションと TAGSORT オプションを使用できます。これら 2 つのオプションは、SORT プロシジャがソートの実行時に使用するメモリ量を制御します。これらのオプションについては、次の 2 つのセクションで説明します。また、渡されたデータセットのソート処理が行われる場所の確定や、ソート処理に必要なディスク容量の確定についても説明します。SORT プロシジャに関する詳細については、“SORT プロシジャ:Windows” (436 ページ) を参照してください。


- “SORTSIZE オプション” (205 ページ)
- “TAGSORT オプション” (205 ページ)
- “ソートされるファイルの場所の選択” (206 ページ)


### SORTSIZE オプション

PROC SORT ステートメントは、SORTSIZE=オプションをサポートしています。このオプションは、PROC SORT が使用するメモリ量を制限します。

PROC SORT ステートメントに SORTSIZE オプションが使用されない場合、PROC SORT では SORTSIZE システムオプションの値が使用されます。SORTSIZE システムオプションが設定されていないと、SORT プロシジャでは REALMEMSIZE システムオプションで指定したメモリ量が使用されます。SORT プロシジャでは、指定したメモリ量よりも多くのメモリが必要な場合、SASWORK ディレクトリに、ソートを完了するための一時ユーティリティファイルが作成されます。

このオプションのデフォルトの値は、1G です。

### TAGSORT オプション

大きな SAS データセットをソートするディスク容量が足りないシングルスレッド環境では、TAGSORT オプションが役に立ちます。TAGSORT オプションは、マルチスレッド環境でのソートには使用できません。

TAGSORT オプションを指定すると、ソートキー(BY ステートメントに指定された変数)と各オブザベーションのオブザベーション番号だけが、一時ファイルに格納されます。ソートキーとオブザベーション番号を合わせて、タグと呼ばれます。ソート処理が完了

すると、入力データセットのレコードが、タグに基づいてソート順に取り出されます。このため、ソートキーの合計バイト数がレコード長に比べて小さい場合は、一時ディスクの使用量が大幅に減少します。ただし、もう 1 つのデータ(出力データセット)または 2 つのタグセットのうち、いずれか大きい方を格納できるだけのディスク容量を確保する必要があります。TAGSORT オプションを使用すると、一時ディスクの使用量は減少しますが、処理時間は大幅に伸びる可能性があります。

### *ソートされるファイルの場所の選択*

SAS データセットをソートすると、一時ユーティリティファイルが作成されます。マルチスレッドを使用したソートでは、UTILLOC システムオプションを使用して、ユーティリティファイルの場所を指定することができます。ユーティリティファイルのデフォルトの場所は、シングルスレッド式およびマルチスレッド式両方のソートに関する WORK データライブラリです。UTILLOC システムオプションに複数の場所が指定されていると、次に使用可能な場所がユーティリティファイルの場所として使用されます。シングルスレッドを使用したソートでは、ソート中にデータセットの保持に必要なメモリが足りない場合、一時ユーティリティファイルが WORK データライブラリに作成されます。このユーティリティファイルの拡張子は、.sas7butl です。ソートする前に、WORK データライブラリに、この一時ユーティリティファイル用のディスク容量があることを確認してください。

OUT=オプションが IN/DATA=データセットと同じ名前を指定するたび、ソートされたデータセットによって入力データセットが置換されます。OVERWRITE オプションを使用すると、出力データセットが作成される前に入力データセットを削除できるので、ピーク時の空きディスク容量要件が低減されます。

出力データセットはユーティリティファイルのコンテンツから自動的に設定されます。出力データセットが入力データセットを置換している場合、ソートが完了すると、元のデータセットは削除されます。データセットをソートする前に、.sas7butl ファイルのためのディスク容量があることを確認してください。

.sas7butl ファイルとデータセットのソート結果を作成する場所を決める際、次の規則に注意してください

- PROC SORT ステートメントに OUT=オプションを指定しないと、データセットは、データセットがあるドライブのディレクトリまたはサブディレクトリでソートされます。たとえば、2 レベルデータセット名が指定された次のステートメントをサブミットすると、.sas7butl ファイルは指定された WORK ディレクトリ内に SAS セッション用として作成されたサブディレクトリに作成されます。

  ```
  libname mylib 'c:\sas\mydata';
  proc sort data=mylib.report;
     by name;
  run;
  ```

  1 レベルデータセット名を同じように指定した場合、.sas7butl ファイルは WORK データライブラリに作成されます。

- PROC SORT ステートメントに OUT=オプションを使用すると、.sas7butl ファイルは、OUT=オプションで使用されるライブラリ参照名に関連付けられたディレクトリに作成されます。ライブラリ参照名が指定されていない 1 レベル名を使用すると、.sas7butl ファイルは WORK データライブラリに作成されます。たとえば、次の SAS プログラムでは、最初のソートと 2 番目のソートが SAS WORK サブディレクトリで行われます。

  ```
  proc sort data=report out=newrpt;
     by name;
  run;
  libname january 'f:\jandata';
  proc sort data=report out=january.newrpt;
     by name;
  ```

```
run;
```

### *データセットのサイズの計算*

シングルスレッド環境では、データセットサイズの 3 倍から 4 倍の空きディスク容量が常に必要です。たとえば、データセットが 1MB のディスク容量を占める場合、ソートを完了するには、3MB から 4MB のディスク容量が必要です。

マルチスレッド環境で、PROC SORT ステートメントの OVERWRITE オプションを使用する場合は、データセットサイズと同じディスク容量が必要です。OVERWRITE オプションを指定しない場合は、データセットサイズの 2 倍の空きディスク容量が必要です。OVERWRITE オプションに関する詳細については、*Base SAS プロシジャガイド*の「ソートプロシジャ」を参照してください。

SAS データセットに必要な空きディスク容量を推定するには、次の作業を行います。

1. オブザベーションを 1 つと必要な変数を含むダミーの SAS データセットを作成します。
2. ダミーのデータセットを使用して CONTENTS プロシジャを実行します。
3. CONTENTS プロシジャ出力からの情報を使用して簡単な計算を行い、データセットサイズを決めます。

たとえば、1 つの文字変数と 4 つの数値変数を含むデータセットの場合、次のようなステートメントをサブミットします。

```
data oranges;
  input variety $ flavor texture looks;
  total=flavor+texture+looks;
  datalines;
navel 9 8 6
;
proc contents data=oranges;
  title 'Example for Calculating Data Set Size';
run;
```

このステートメントでは、次の出力が生成されます。

*アウトプット 7.1 CONTENTS プロシジャを使用したデータセットサイズの計算例*

```
                    Example for Calculating Data Set Size                    1
                                          19:39 Wednesday, February 12, 2003

                            The CONTENTS Procedure

Data Set Name        WORK.ORANGES                    Observations          1
Member Type          DATA                            Variables             5
Engine               V9                              Indexes               0
Created              Wednesday, October              Observation Length    40
                     3, 2007 07:41:04
Last Modified        Wednesday, October              Deleted Observations  0
                     10, 2007 07:41:04
Protection                                           Compressed            NO
Data Set Type                                        Sorted                NO
Label
Data Representation  WINDOWS_32
Encoding             wlatin1  Western (Windows)


                      Engine/Host Dependent Information

Data Set Page Size          4096
Number of Data Set Pages    1
First Data Page             1
Max Obs per Page            101
Obs in First Data Page      1
Number of Data Set Repairs  0
File Name                   C:\TEMP\SAS Temporary Files\_TD246\oranges.sas7bdat

Release Created             9.0201B0
Host Created                XP_PRO


                  Alphabetic List of Variables and Attributes

                         #    Variable    Type    Len

                         2    flavor      Num       8
                         4    looks       Num       8
                         3    texture     Num       8
                         5    total       Num       8
                         1    variety     Char      8
```

結果データセットのサイズは、データセットのページサイズとオブザベーション数によって異なります。次の公式を使用すると、データセットのサイズを推定できます。

- データページの数 = 1 + (floor(オブジェクトの数 / **ページあたりの最大オブジェクト**))
- サイズ = 1024 + (**データセットページサイズ** * データページの数)

(floor は端数を切り捨てて、最も近い整数にする関数を表します)

アウトプット 7.1 (208 ページ) で示される情報から、例のデータセットのサイズを計算することができます。

- データページの数 = 1 + (floor(1/101))
- サイズ = 1024 + (4096 * 1) = 5120

したがって、例のデータセットでは、5,120 バイトのストレージスペースを使用します。

### *対話型処理の効率の向上*

SAS を対話操作しながら SAS ジョブを実行する際に、ジョブから多数のログメッセージや多数の出力が生成される場合は、AUTOSCROLL コマンドを使用してウィンドウのスクロール機能を抑制することを検討してください。スクロール機能を抑制すると、ジョブ実行中にリソースを使用して[ログ]ウィンドウや[アウトプット]ウィンドウの表示を更新する必要がないため、ジョブの実行速度が速くなります。たとえば、**ログ**ウィンドウから `autoscroll 0` を発行すると、ジョブが終了するまで**ログ**ウィンドウはスクロールしません。**アウトプット**ウィンドウの場合、AUTOSCROLL はデフォルトで 0 に設定されます。

また、**ログ**ウィンドウを最小化することでも、特に多数のログメッセージが生成される場合に、ジョブの実行速度が速くなることがあります。

# 2 部

# 他の Windows アプリケーションでの SAS の使用

*8 章*
***Windows 版 SAS での Windows システムツールの使用*** . . . . . *213*

*9 章*
***Windows 版 SAS/AF ソフトウェアでの OLE の使用*** . . . . . *233*

*10 章*
***OLE を使用した他のアプリケーションからの SAS 制御*** . . . . . *259*

*11 章*
***Windows 版動的データ交換の使用*** . . . . . *267*

*12 章*
***Windows 版の名前の付いていないパイプと名前付きパイプの使用について*** . . . . . *279*

*13 章*
***Windows 版 SAS から外部 DLL にアクセスする*** . . . . . *291*

*14 章*
***Windows 版での SAS/AF プログラマーのための特記注意事項*** . . . . . *313*

*8 章*

# Windows 版 SAS での Windows システムツールの使用

**SAS での Windows システムツールの使用について** . . . . . **213**

**イベントビューアアプリケーションログ** . . . . . **214**
Windows 7 を使用したアプリケーションログへのアクセス . . . . . 214
SAS イベントの表示 . . . . . 214
ユーザー定義関数を使用してメッセージをアプリケーションログに送る . . . . . 214
LOGEVENT.EXE を使用してメッセージをアプリケーションログに送る . . . . . 216

**パフォーマンスツール** . . . . . **217**
パフォーマンスモニタの使用理由について . . . . . 217
パフォーマンスカウンタとオブジェクト . . . . . 217
Windows パフォーマンスモニタの起動 . . . . . 217
パフォーマンスモニタとシステムモニタの SAS カウンタ . . . . . 217
モニタする SAS カウンタの選択 . . . . . 218
DATA ステップ、PROC SORT、PROC SQL のモニタリング例 . . . . . 219
DATA ステップと PROC SORT ステップ間のパフォーマンス検証 . . . . . 219
PROC SQL クエリの検証 . . . . . 220

**SAS を Windows のサービスとして起動する** . . . . . **222**
Windows のサービスとしての SAS の起動について . . . . . 222
SAS サービス構成ユーティリティの起動 . . . . . 223
初期化ファイルの作成 . . . . . 223
SAS サービスのインストール . . . . . 229
SAS サービスの起動 . . . . . 231
SAS サービスの削除 . . . . . 231

## SAS での Windows システムツールの使用について

SAS と Windows に精通したユーザーおよびシステム管理者であれば、Windows のサービスを使用して SAS を起動し、Windows のイベントログツールやパフォーマンスツールを使用して SAS を監視することができます。

SAS では、エラーメッセージを Windows イベントビューアのアプリケーションログに記録することができます。アクセス違反などで SAS タスクが異常終了した場合、SAS ログに加えて、アプリケーションログでもエラー内容を表示することができます。また、SAS/CONNECT から発行された情報メッセージを、アプリケーションログに表示することもできます。詳細については “イベントビューアアプリケーションログ” (214 ページ) を参照してください。

SAS を Windows のサービスとして起動することができます。Windows のサービスを利用すると、SAS を自動的に起動したり、SAS で障害が発生した場合の回復手順を指定することができます。詳細については、“SAS を Windows のサービスとして起動する ” (222 ページ) を参照してください。

# イベントビューアアプリケーションログ

## *Windows 7 を使用したアプリケーションログへのアクセス*

アプリケーションログを開くには、次の操作を行います。

1. [コントロール パネル]を開きます。
2. [**システムとセキュリティ**]を選択します。
3. [**管理 ツール**]を選択します。
4. **イベントビューア**をダブルクリックします。

   *注:* 続行するには許可が必要というメッセージが表示されたら、[**続行**]を選択します。
5. ツリービューで[**Windows ログ**]を選択し、左矢印をクリックして Windows ログを展開します。
6. [**アプリケーションログ**]を選択します。

別の方法でイベントビューアを開始するには、**eventvwr** と[**ブログラムとファイルの検索**]ダイアログボックスに入力し、[**OK**]をクリックします。

## *SAS イベントの表示*

SAS タスクが異常終了した場合、タスクに関する情報がアプリケーションログに記録されます。ソース列には、イベントソースとして SAS が表示されます。SAS/CONNECT からのメッセージに、イベントソースとして SAS Job Spawner が示されます。SAS イベントをダブルクリックすると[**イベントのプロパティ**]ウィンドウが開き、そのイベントに関する情報が表示されます。

## *ユーザー定義関数を使用してメッセージをアプリケーションログに送る*

### *関数の最初のパラメータの指定*

SAS イベントは、SAS コードまたは SAS Component Language (SCL)にユーザー定義関数を使用して、アプリケーションログに送信できます。関数への入力は、特定のテキスト文字列が使用されます。このテキスト文字列は、イベントビューアに表示されるイベントのタイプとテキスト文字列に対応します。

```
yourfunction("type_of_event", "text_string");
```

次に、最初のパラメータに使用できるイベントタイプのリストを示します。

***表 8.1*** *SAS イベントのタイプ*

| イベントのタイプ | 最初のパラメータ値 |
|---|---|

| | |
|---|---|
| エラー | “エラー” |
| 警告 | “警告” |
| 情報 | “INFORMATION” |
| 成功の監査 | “SUCCESSAUDIT” |
| 失敗の監査 | “FAILUREAUDIT” |

最初のパラメータ値は大文字でテーブルに表示されますが、大文字と小文字を混在させることもできます。関数の 2 番目のパラメータは、Windows Event Viewer に表示される文字列です。

### *ユーザー定義関数を使用したイベントログへの書き込み例*

次の例では、SAS によって長い処理が実行される前に、セマフォファイルの存在がチェックされます。

```
%macro pdata(file);
    %let cmdstr = "dir &file";
    options noxwait;
    data _null_;
       call system(&cmdstr);
    run;
    %put &sysrc = sysrc;
    %put &file;
    %if &sysrc=0 %then %do;
       filename indata "&file";
       /* Your data step code for this file.  */
       DATA a;
          infile indata length=linelen;
          length line $ 200;
          input @1 line $ varying200. linelen;
       PROC print;
       run;
    %end;
    %else %do;
       /*  Log an Event of type Error.  */
       %let cmdstr = %str("The file &file did not exist
                           so no data step ran.");
       %put &cmdstr;
       DATA _null_;
          x=ntlog("INFORMATION",&cmdstr);
       run;
    %end;
 %mend;

 %pdata(c:\config.syss)
```

次は、アプリケーションログに書き込む SCL コードの例です。

```
/* Build a frame and add a pushbutton.  Change the Attribute
Name “name” to “object1”.  In the Source window, add the
following code. */
object1:
```

```
x=ntlog("INFORMATION", "This is an INFORMATION event.");
x=ntlog("WARNING", "This is a WARNING event.");
x=ntlog("ERROR", "This is an ERROR event.");
x=ntlog("SUCCESSAUDIT", "This is a SUCCESSAUDIT event.");
x=ntlog("FAILUREAUDIT", "This is a FAILUREAUDIT event.");

return;
```

### *LOGEVENT.EXE を使用してメッセージをアプリケーションログに送る*

Windows Resource Kit に用意されている LOGEVENT.EXE ユーティリティを使用すると、SAS ユーザー独自のメッセージを SAS コード内からアプリケーションログに送信できます。

次の例では、SAS によって長い処理が実行される前に、セマフォファイルの存在がチェックされます。

```
%macro pdata(file);
   %local cmdstr;
   %let cmdstr = "dir &file";
   options noxwait;
   DATA _null_;
      call system(&cmdstr);
   run;
   %if &sysrc=0 %then %do;
      filename indata "&file";
      /* Your data step code for this file.  */
      DATA a;
         infile indata length=linelen;
         length line $ 200;
         input @1 line $ varying200. linelen;
      PROC print;
      run;
   %end;
   %else %do;
      /*  Log an Event of type Error.  */
      %let cmdstr = %bquote(c:\support\sasset2\logevent.exe -s E
      "The file &file did not exist so no data step ran.");
      DATA _null_;
         %sysexec &cmdstr;
      run;
   %end;
%mend;

%pdata(c:\config.syss)
```

アプリケーションログでイベントをダブルクリックすると、[**イベントのプロパティ**]ウィンドウの[**説明**]ボックスにメッセージが表示されます。

LOGEVENT.EXE については、Windows Resource Kit のドキュメントを参照してください。

# パフォーマンスツール

## パフォーマンスモニタの使用理由について

Windows パフォーマンスモニタは、アプリケーションまたはコンピュータシステムの調整および問題の診断に役立ちます。パフォーマンスモニタには、サポートされる Windows オペレーティングシステムのシステムモニタが含まれます。SAS カウンタからの情報と動作環境のカウンタからの情報と関連付けることで、パフォーマンス上の問題により簡単に対応できます。

たとえば、SAS ジョブが動作していないようにみえる場合を考えてみます。この場合、ネットワークドライブ上に大容量のデータセットを生成する、長くて複雑な DATA ステップが実行されている可能性があります。SAS プロセスのプロセスカウンタで I/O Read and Write Bytes/Sec をモニタリングすると、ジョブがまだ動作していることを確認できます。

## パフォーマンスカウンタとオブジェクト

カウンタは、システムでモニタリングされる情報です。パフォーマンスオブジェクトは、個別のプロセス、共用メモリのセクション、およびメモリや論理ディスクなどの物理デバイスを表します。カウンタは、オブジェクトによってグループ化されています。たとえば、メモリオブジェクトには、Available Bytes、Committed Bytes、Page Faults per sec などのカウンタが属しています。Processor オブジェクトには、%Processor Time や%User Time などのカウンタが属しています。

各種システムカウンタおよびアプリケーション定義カウンタを調べることで、パフォーマンス上の問題を確定できます。システム内の問題を追求し、それらの問題をハードウェア、システムソフトウェア、アプリケーションなどの部分に細分化することができます。

## Windows パフォーマンスモニタの起動

Windows で **perfmon** と[**検索**]ダイアログボックス([**スタート**]メニュー)に入力すると、[**パフォーマンスモニター**]ウィンドウが開きます。

*注:* 32 ビットバージョンの SAS のパフォーマンスカウンタを Windows x64 システムで使用するには、32 ビットバージョンの SAS を起動してから 32 ビットバージョンの PerfMon を起動します。32 ビットバージョンの PerfMon を起動するには、Windows エクスプローラを開き、**c:\Windows\SysWOW64** という、32 ビットアプリケーションを含むフォルダに移動します。このフォルダに、32 ビットバージョンの PerfMon が含まれています。32 ビットバージョンの SAS パフォーマンスカウンタは、perfmon.exe または perfmon.msc を起動することでアクセスできます。

## パフォーマンスモニタとシステムモニタの SAS カウンタ

SAS は、SAS オブジェクトに次のアプリケーション定義カウンタを組み込みます。

Virtual Alloc'ed Memory
: SAS が VirtualAlloc() API を用いて割り当てた、コミット済みの仮想メモリ量です。

Disk ReadFile Bytes Read Total
: SAS が ReadFile() API を用いてディスクファイルから読み込んだ、総バイト数です。

Disk ReadFile Bytes Read/Sec
SAS が ReadFile() API を用いてディスクファイルから読み込んだ、毎秒あたりのバイト数です。

Disk WriteFile Bytes Written Total
SAS が WriteFile() API を用いてディスクファイルに書き込んだ、総バイト数です。

Disk WriteFile Bytes Written/Sec
SAS が WriteFile() API を用いてディスクファイルに書き込んだ、毎秒あたりのバイト数です。

Disk SetFilePointer/Sec
SAS が SetFilePointer() API をディスクファイル上で正常に呼び出した、毎秒あたりの回数です。

Memlib/Memcache Current Usage K
現在使用している拡張サーバーメモリの量(バイト単位)です。

Memlib/Memcache Peak Usage K
現在の SAS セッションで使用している拡張サーバーメモリの最大量(バイト単位)です。

## *モニタする SAS カウンタの選択*

SAS カウンタをモニタリングするには、次の手順から使用している動作環境に適した手順を実行します。

***表 8.2*** *SAS カウンタ選択の手順*

| ウィンドウの使い方 |
| --- |
| • **perfmon** と[**ファイル名を指定して実行**]ダイアログボックスに入力します。[**パフォーマンスモニター**]が開きます。<br>• [**モニタツール**]メニューの[**パフォーマンスモニター**]ウィンドウを<br>• +記号のボタンを選択し、モニタリングするカウンタを追加します。<br>• [**次のコンピューターからカウンターを選んでください**]リストで、カウンタに関連付けられているコンピュータを選択します。<br>• [**Process**]を利用可能なカウンタの一覧から選択します。<br>• [**選択したオブジェクトのインスタンス**]リストから[**SAS**]を選択します。<br>• [**追加**>>]を選択して、カウンタを[**追加されたカウンター**]ダイアログボックスに移動します。<br>• [**OK**]を選択します。 |

選択したカウンタに関する情報が、パフォーマンスモニタによってただちに収集されて表示されます。

複数の SAS カウンタをモニタリングすることができます。複数のインスタンスを監視して表示することもできます。各インスタンスは個別の SAS プロセスです。PID number は、SAS セッションのプロセス識別子です。SAS インスタンスは、SAS PID number の形式で表示されます。すべてのプロセスを含むリストは、タスクマネージャを使用して表示できます。

## DATA ステップ、PROC SORT、PROC SQL のモニタリング例

### パフォーマンスモニタの構成

このセクションの例では、パフォーマンスモニタおよびシステムモニタをすべて次のように構成します。

1. SAS とパフォーマンスモニタ、またはシステムモニタを起動します。
2. **グラフの追加**ウィンドウまたは**カウンタの追加**ウィンドウを開き、SAS オブジェクトを選択します。
3. 下向き矢印をクリックして、選択項目を展開します。
4. 以下の SAS カウンタを追加します。まずカウンタを選択し、次に[**追加**>>]を選択します。
   - Disk ReadFile Bytes Read/Sec
   - Disk WriteFile Bytes Written/Sec
   - Disk SetFilePointer/Sec
5. [**OK**]をクリックして、[**カウンタ**]ダイアログボックスを閉じます。
6. Process オブジェクトを選択します。
7. 次の Process カウンタを追加します。
   - %Processor Time
   - %User Time
   - %Privileged Time
8. [**完了**]または[**閉じる**]をクリックします。

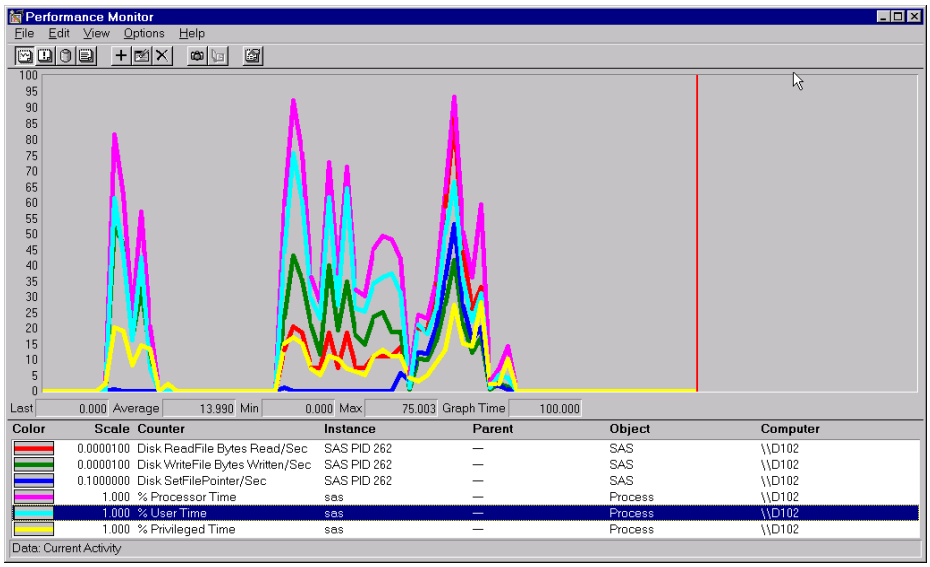
## DATA ステップと PROC SORT ステップ間のパフォーマンス検証

DATA ステップと PROC ステップ間のパフォーマンスの差異を調べるには、次のコードをサブミットします。

```
options fullstimer;
  /* Create a test data set with some random data. */
DATA a (drop=s);
   do i = 1 to 500000;
      x = ranuni(i);
      y = x*2;
      z = exp(x*y);
      output;
   end;
   /* The sleep helps to delineate the subsequent  */
   /* sort in the Performance Monitor graph        */
   s = sleep(15);
run;
PROC sort data = a noduplicates;
   by z y x i;
run;
```

このコードをサブミットすると、パフォーマンスモニタまたはシステムモニタは 図 8.1 (220 ページ) 内の結果と同様な結果を出力します。カウンタ間のスケールの調整が必要になることがあります。

***図 8.1*** *DATA ステップと PROC SORT ステップのパフォーマンス*




DATA ステップでは、Disk ReadFile Bytes Read/Sec カウンタまたは Disk SetFilePointer/Sec カウンタからのアクティビティはかなり少ないことがわかります。それに続く PROC SORT では、これらの 2 つのカウンタからのアクティビティは増加します。これは、ソートのためにデータセットが読み込まれていること(Disk Readfile Bytes Read/Sec)、および一定量のランダム I/O がソートによって実行されていること(Disk SetFilePointer/Sec)を示しています。

アクティビティの一時停止が、DATA ステップの後にある SLEEP 関数によって発生しています。Disk WriteFile Bytes Written/Sec カウンタは、DATA ステップと PROC SORT ステップの両方でアクティブです。

最後に、SAS ログで、Process オブジェクトのカウンタをユーザー時間およびシステム CPU 時間に関連付けられます。

## *PROC SQL クエリの検証*

インデックスを使用した PROC SQL クエリのパフォーマンスを検査するには、次のコードをサブミットします。

1. ステップ 1 およびステップ 2 のコードをサブミットします。ステップ 2 はインデックスを作成します。

```
/* Step 1                                              */
   /* Create a test data set with some random data. */
   /* Do this twice - once with Step 2 and once     */
   /* without Step 2.                                  */

libname sample 'c:\';
DATA sample.a;
   do i = 1 to 500000;
      x = ranuni(i);
      y = x*ranuni(i);
      z = exp(y);
```

```
      output;
   end;
run;

   /* Step 2                                          */
   /* Create a simple index on variable x.            */
   /* Submit this step once.                          */

PROC DATASETS library = sample;
   modify a;
   index create x;
   quit;
```

2. グラフをクリアします。このために[**表示をクリア**]を[**編集**]メニューで選択するか、[**表示をクリア**]ツールバーボタンをクリックします。
3. 図 8.2 (221 ページ) のようなグラフを参照するには、ステップ 3 をもう一度サブミットします。

```
      /* Step 3                                          */
   /* Perform a query on the data.  Do this twice -   */
   /* once with an index and once without an index    */
   /* The query should select about 50% of the        */
   /* observations in the data set.                   */

PROC SQL;
   create table sample.yz as
   select y,z
      from sample.a
      where x > 0.5;
   quit;
```

インデックスを使用しないで SQL プロシジャを実行するには、次の操作を実行します。

1. ステップ 1 をもう一度サブミットします。
2. グラフをクリアします。
3. 図 8.3 (222 ページ) のようなグラフを参照するには、ステップ 3 をもう一度サブミットします。

***図 8.2*** *インデックスを使用した PROC SQL クエリのパフォーマンス*

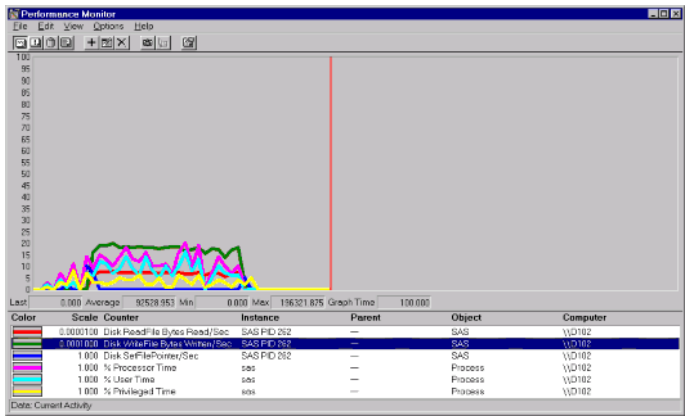
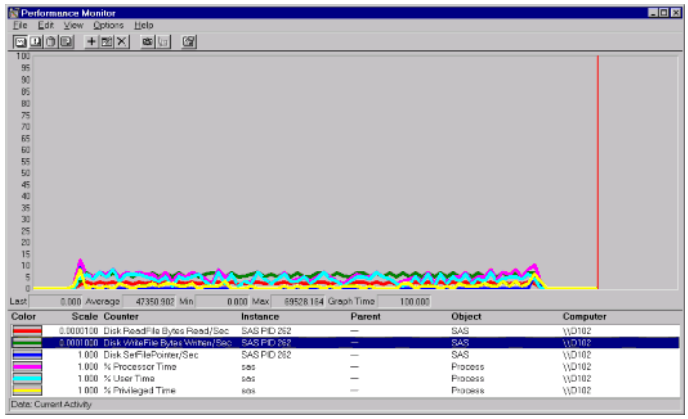


*図 8.3 インデックスを使用しない PROC SQL クエリのパフォーマンス*




次を参照してください。図 8.3 (222 ページ) で、いくつかのカウンターが 10 パーセント以上の平均を示し、図 8.2 (221 ページ) また、Disk WriteFile Bytes Written/Sec カウンターは 25 パーセント以上上昇しているのに対して、前者のカウンターは規模で 10 パーセント以下の平均です。これらの高いカウンタ値は、操作のスループットが全体的に良好であること示しています。

パフォーマンスモニタのグラフまたはシステムモニタのグラフを使用して有効な比較を行うには、各カウンタで同じスケールを使用する必要があることに注意してください。絶対値を調べることによって、カウンタを確認できます。図 8.2 (221 ページ) における Disk WriteFile Bytes Written/Sec の平均値は 92528.953 でした。この値を、平均値が 47350.902 であった 図 8.3 (222 ページ) 内の同じカウンターと対比させます。この操作では、データセットにインデックスが付けられていると、およそ 2 倍の速度でバイトが書き出されたことがわかります。

# SAS を Windows のサービスとして起動する

## *Windows のサービスとしての SAS の起動について*

SAS を Windows のサービスとして起動すると、SAS の自動または手動での起動、SAS で障害が発生した場合の回復手順の指定が可能になります。また、ユーザーは SAS を中断することなく、PC へのログオンとログオフが可能になります。SAS が手動で起動するよう定義されている場合は、SAS をアプリケーションから[`net start` コマンドまたは[**Windows サービス**]ダイアログボックスを使用して起動できます。

Windows のサービスとして SAS を構成するための一般的な手順は、次のとおりです。

1. 初期化(.INI)ファイルを作成します。
2. 作成した INI ファイルをインストールし、SAS を Windows のサービスとして登録します。
3. SAS が自動的に起動するように構成されている場合は、マシンを再起動します。SAS が手動で起動するように構成されている場合は、SAS をアプリケーションから起動することも、すべての Windows 動作環境の**サービス**ウィンドウから起動することもできます。詳細については、“SAS サービスの起動” (231 ページ) を参照してください。

複数の SAS 構成が存在する場合は、構成ごとに初期化ファイルを作成することができます。

SAS には、サービスの構成と INI ファイルのインストールに使用する SAS サービス構成ユーティリティ(SAS SCU)が用意されています。

## SAS サービス構成ユーティリティの起動

SAS SCU は、次のいずれかの方法で起動します。

- [**スタート** ⇨ **すべてのプログラム** ⇨ **SAS** ⇨ **ユーティリティ** ⇨ **SAS サービス構成ユーティリティ**
- SAS SCU ディレクトリから、`sasservicemngr.exe` と入力します。デフォルトの SAS SCU ディレクトリは `c:\Program Files\SASHOME\SASFoundation\9.4\core\sasscu` です。

## 初期化ファイルの作成

### 初期化ファイルの概要

Windows のサービスとして SAS を起動する前に、初期化(.INI)ファイルを構成し、インストールする必要があります。.INI ファイルは Unicode ファイルであり、次の処理をします。

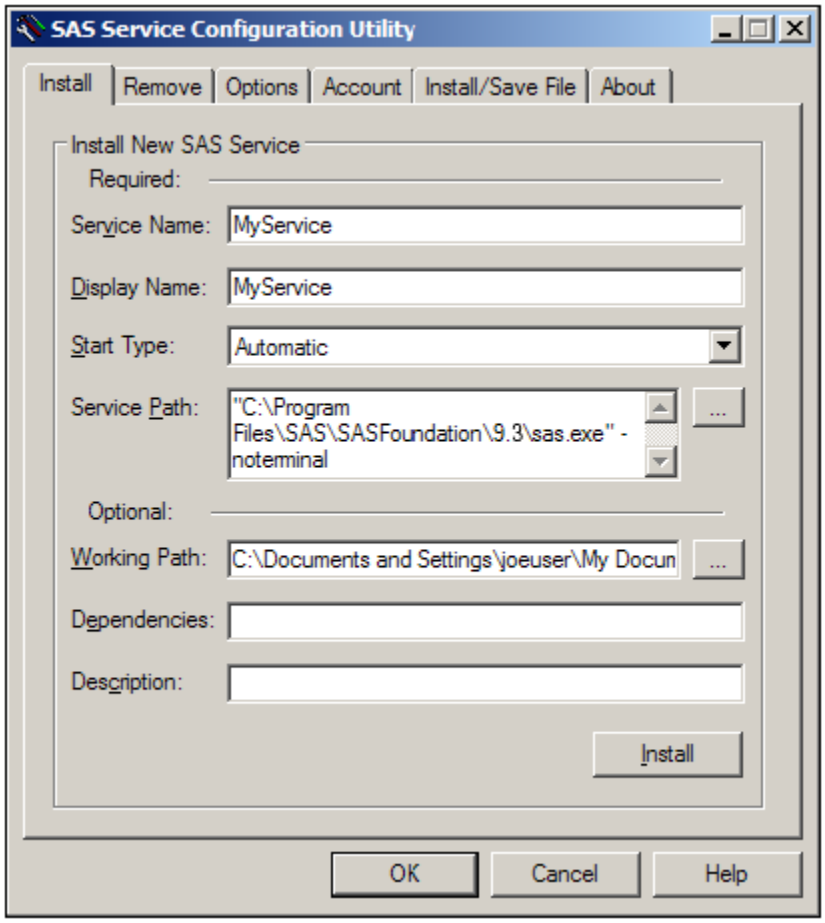
- SAS サービスに名前を付けます
- サービスを自動および手動のいずれで起動するかを指定します
- SAS へのパスと SAS 作業フォルダを定義します
- アプリケーションが SAS サービスに対して持つアクセスのレベルを指定します
- このサービスを起動する前に起動する必要がある Windows の他のサービスを指定します。
- SAS がサービスとして起動しない場合に、Windows で完了すべき処理を定義します。
- SAS サービスをシステムアカウントにするかローカルアカウントにするかを指定します。
- ユーザーに SAS デスクトップからの操作を許可するかどうかを指定します。

初期化ファイルの作成には、SAS SCU グラフィカルユーザーインターフェイス(GUI)を使用することも、メモ帳などの Unicode 対応テキストエディタを使用することもできます。SAS SCU GUI を使用すると、必須の値を指定するだけで SAS SCU によって INI ファイルが作成されます。テキストエディタを使用して INI ファイルを作成する場合は、SAS サービスの設定およびその他の値を指定する必要があります。ファイルは ANSI または ASCII でエンコードされたファイルとしてではなく、Unicode ファイルとして保存してください。表 8.4 (227 ページ) では、SAS サービスの INI ファイルをテキストエディタを使用して作成するために指定する設定を説明しています。

### SAS サービス構成ユーティリティを使用して初期化ファイルを作成

次の図に示される SAS サービス構成ユーティリティ(SAS SCU)を使用して、INI ファイルを作成します。設定を指定したら、[**ファイルのインストール/保存**]をクリックして、INI ファイルを保存し、インストールします。

*図 8.4 SAS サービス構成ユーティリティ*




INI ファイルを構成するには、次のタブを選択し、必要に応じて設定を修正します。

**インストール**タブ

**サービス名**

サービスをインストールしたときに Windows に登録されるサービス名です。このサービス名は、net start または net stop コマンドを発行するときにも使用します。必須フィールドです。デフォルトは **SASService** です。

**表示名**

は、ユーザーインターフェイスアプリケーションに表示されるサービス名です。必須フィールドです。デフォルトは **A SAS Service** です。

**開始タイプ**

SAS サービスを手動で起動するか、自動的に起動するか、または無効にするかを指定します。必須フィールドです。デフォルトは **Manual** です。**Manual** を指定すると、別のプロセスによってサービスが起動されます。**Automatic** を指定すると、システム起動時にサービスが自動的に起動されます。**Disabled** を指定すると、サービスを起動できません。

**サービスパス**

SAS のインストールディレクトリのパスとサービスの開始に使用する SAS コマンドの両方を指定します。必須フィールドです。新規インストールの場合、デフォルトパスは SAS のインストールパスであり、その後にコマンド `sas.exe -noterminal` を続けます。すでに SAS SCU がインストールされている場合は、デフォルトのパスは `c:\Program Files\SASHome\SASFoundation`

**\9.4\で、**後にコマンド **sas.exe -noterminal** を続けます。
NOTERMINAL システムオプションは必須です。SAS プログラムをサービスとして起動するには、SYSIN システムオプションを追加し、その後にサービスパスのプログラムパス名とファイル名を付けます。サービスパスを選択するには、[...]をクリックします。(楕円ボタン)

**作業フォルダ**
SAS サービスを使用してディレクトリの作成、ファイルの保存、およびログの記録を実行するアプリケーションの作業フォルダです。このフィールドは、任意指定です。デフォルトは、ユーザーのプロファイルディレクトリです。Windows では、ユーザープロファイルは **c:\Users***user ID***\Documents\My SAS Files\9.4** にあります。作業フォルダを選択するには、[...]をクリックします。(楕円ボタン)

**依存関係**
このサービスを起動する前に起動する必要がある Windows のサービスを指定します。依存するサービスがインストールされ有効化されている場合は、このサービスを起動する前に依存するサービスが起動されます。依存するサービスがインストールされていても無効化されている場合、このサービスは起動されません。

依存関係を指定するには、サービス名を入力します。複数のサービス名を入力する場合は、パイプ文字(|)で区切ります。例:NetDDE|NetDDEdsdm。

**説明**
サービスの説明を入力します。この説明は、Windows の**サービス**ウィンドウに表示されます。

**削除**タブ
**既存のサービスの削除**
削除するインストール済み SAS サービスの名前を指定します。

**オプション**タブ
**エラーコントロール**
SAS サービスが起動しない場合の、エラー重大度を決定します。次のいずれかを選択できます。

***表 8.3*** *エラーコントロール*

| エラーコントロール | 説明 |
|---|---|
| Ignore | エラーはログに記録されます。起動処理は続行されます。 |
| Normal | エラーがログに記録され、メッセージが表示されます。起動処理は続行されます。 |
| Severe | エラーがログに記録され、起動処理は最後に正常にインストールされた INI ファイルを使用して続行されます。 |
| Critical | エラーのログへの記録が試みられます。最後に成功したことがわかっている INI ファイルが起動処理で使用されていた場合、起動処理は失敗します。最後に成功したことがわかっている INI ファイルが起動処理で使用されていなかった場合は、最後に成功した INI ファイルを使用してサービスの起動が試みられます。 |

**アクセス**
アプリケーションから SAS サービスへのアクセスのレベルです。読み込み、書き込み、実行などのアクセスレベルを選択すると、所定のアクセスタイプ設定が INI ファイルで `1 (TRUE)` に設定されます。すべてのアクセスタイプを構成するには、該当するボックスをクリックします。アクセスタイプ設定の説明は、表 8.4 (227 ページ) を参照してください。アクセスレベルは、次のとおりです。

**読み込み**を選択すると、アプリケーションで**応答**、**クエリの構成**、および**クエリのステータス**の各アクセスタイプ を設定できます。このアクセスレベルを選択すると、INI ファイルで AccessInterrogate=、AccessQryCfg=、および AccessQryStatus=の各設定が `1 (TRUE)` に設定されます。

**書き込み**書き込み ]を選択すると、アプリケーションで[**構成の変更**アクセスタイプが設定できます。このアクセスレベルを選択すると、INI ファイルで AccessChgCfg=設定が `1 (TRUE)` に設定されます。

**実行**を選択すると、アプリケーションで**応答**、**一時停止/再開**、**サービスの開始**, **サービスの停止**、および**コントロールの定義**の各アクセスタイプが設定できます。このアクセスレベルを選択すると、INI ファイルで AccessInterrogate=、AccessPauseCont=、AccessStart=、AccessStop、AccessUserDefCtrl= の各設定が `1 (TRUE)` に設定されます。

**アカウント**タブ

**システムアカウント**
サービスを、このマシンにログオンしているすべてのユーザーで共有するかどうかを指定します。サービスがユーザーとデスクトップでやり取りできるようにするには、**このサービスをデスクトップから操作可能にする**を選択します。**システムアカウント**を選択すると、INI ファイルの ServiceStartName=設定が `LocalSystem` に設定されます。

**このアカウント**
は、サービスが特定のユーザーにのみ許可されていることを指定します。**このアカウント**を選択した場合は、アカウント名をボックスに入力します。次に、パスワードを**パスワード**と**パスワードの確認**ボックスに入力します。

**ファイルのインストール/保存**タブ

**ファイルからインストール**
**ファイルからインストール**ボタンをクリックし、インストールする初期(INI)ファイルを指定します。

**ファイルに設定を保存**
**ファイルに設定を保存**ボタンをクリックすると、SAS サービス構成ユーティリティの GUI で指定した設定がファイルに保存されます。

**ファイル内容を表示**
**ファイル内容を表示**ボックスは、インストールまたは保存したい初期化ファイルを**ファイルの内容**ボックスに表示する場合に選択します。

**著作権情報**タブ
SAS サービス構成ユーティリティの著作権情報が表示されます。

### *Unicode テキストエディタを使用して初期化(INI)ファイルを作成*

メモ帳などの Unicode 対応テキストエディタを使用して SAS サービスの INI ファイルを作成するには、エディタで新規ファイルを作成し、次の表の各設定に有効な値を割り当てます。設定は 1 行に 1 つずつ入力します。

*表 8.4 SAS サービスの INI ファイル設定とデフォルト値*

| 設定名 | 必要事項 | 説明 | 有効な値 | デフォルト | 関連する SAS SCU フィールド |
|---|---|---|---|---|---|
| サービス名= | Yes | Windows に登録される SAS サービス名。 | 最大長は 32 文字。/や \は使用不可。大文字と小文字は区別されない。引用符で囲む必要がある。 | "SAS サービス" | サービス名 |
| 表示名= | Yes | ユーザーインターフェイスアプリケーションに表示されるサービス名。 | 最大長は 256 文字で、大文字と小文字は区別されません。引用符で囲む必要があります。 | "A SAS サービス" | 表示名 |
| バイナリのパス名= | Yes | SAS サービスの INI ファイルがインストールされるディレクトリパスの後にサービスを起動する SAS コマンドを続けて指定します。 | パス名はカッコと引用符の両方で囲む必要があります。 | ["*SAS installation path* \sas.exe -noterminal"] | サービスパス |
| 開始タイプ= | Yes | SAS サービスを手動で起動するか、または自動的に起動するかを指定します。 | SERVICE_AUTO_ START<br>SERVICE_ DEMAND_ START<br>SERVICE_DISABLED | SERVICE_ DEMAND_ START | 開始タイプ |
| 依存関係= | No | このサービスを起動する前に起動する必要がある Windows サービスを指定します。 | 1 つ以上の Windows サービス名を、パイプ文字(\|)で区切って入力する。依存関係は、引用符で囲む。 | なし | 依存関係 |
| 説明 | No | サービスの説明 | 説明には、英数字を使用することができ、引用符で囲む必要があります。 | なし | 説明 |
| WorkDir= | No | SAS サービスによって作成、使用されるファイルを保存するために、アプリケーションが使用するディレクトリ。 | 作業ディレクトリへのパスは、引用符で囲む必要があります。 | "Under Windows c: \Users*user ID*" | 作業フォルダ |
| ErrorControl= | Yes | SAS サービスが起動しない場合の、エラー重大度を決定します。 | SERVICE_ ERROR_ IGNORE<br>SERVICE_ ERROR_ NORMAL<br>SERVICE_ ERROR_ SEVERE<br>SERVICE_ ERROR_ CRITICAL | SERVICE_ ERROR_ NORMAL | エラーコントロール |

| 設定名 | 必要事項 | 説明 | 有効な値 | デフォルト | 関連する SAS SCU フィールド |
|---|---|---|---|---|---|
| Interactive= | Yes | ユーザーが SAS デスクトップからサービスを使用できるかどうかを指定する。 | 1(TRUE)<br>0(FALSE) | 0(FALSE) | 対話型処理 |
| AccessChgCfg= | Yes | SAS サービス構成を修正する。 | 1(TRUE)<br>0(FALSE) | 1(TRUE) | 構成の変更 |
| Access Interrogate= | Yes | ただちに現在のステータスを更新するように SAS サービスに要求します。 | 1(TRUE)<br>0(FALSE) | 1(TRUE) | 応答 |
| Access PauseCont= | Yes | SAS サービスを一時停止したり再開したりする。 | 1(TRUE)<br>0(FALSE) | 1(TRUE) | 一時停止/再開 |
| AccessQryCfg= | Yes | SAS サービス構成に関するクエリを作成します。 | 1(TRUE)<br>0(FALSE) | 1(TRUE) | クエリの構成 |
| AccessQry Status= | Yes | SAS サービスのステータスを Windows に問い合わせます。 | 1(TRUE)<br>0(FALSE) | 1(TRUE) | クエリのステータス |
| AccessStart= | Yes | SAS サービスを起動します。 | 1(TRUE)<br>0(FALSE) | 1(TRUE) | サービスの開始 |
| AccessStop | Yes | SAS サービスを停止します。 | 1(TRUE)<br>0(FALSE) | 1(TRUE) | サービスの停止 |
| AccessUser DefCtrl= | Yes | ユーザー定義コントロールコードを指定します。 | 1(TRUE)<br>0(FALSE) | 1(TRUE) | コントロールの定義 |
| ServiceStart Name= | No | SAS サービスを実行するための適切なユーザー権限を備えた Windows ユーザーアカウント。 | LocalSystem または Windows アカウント名 | LocalSystem | このアカウント |
| Password= | No | Windows アカウントのパスワード。 | 暗号化されたパスワード | なし | パスワード |

テキストエディタを使用して INI ファイルを作成し、ServiceStartName に特定のユーザーを指定する場合は、Windows アカウント名を *domainname\username* 形式で指定し、PASSWORD 設定名に暗号化されたパスワードを含める必要があります。暗号化されたパスワードを作成するには、PWENCODE プロシジャを使用します。例えば、次の PWENCODE プロシジャでは、**mypw** が入力パスワードとして指定されます。

```
proc pwencode in='mypw';
run;
```

SAS ログには暗号化されたパスワード**{sas001}bXlwdw==**が表示されます。次に、**{sas001}bXlwdw==**を Password=設定の値として INI ファイルに指定します。暗号化されたパスワードは、INI ファイルで Password=を指定する場合のみ必須です。それに対して、SAS SCU を使用して INI ファイルを作成する場合は、テキストのパスワードを指定します。SSCU によって、パスワードが暗号化されます。

PWENCODE 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。

## *SAS サービスのインストール*

初期化ファイルの作成が完了したら、初期化ファイルを使用して、SAS をサービスとしてインストールします。SAS サービスは、SSCU、コマンドプロンプト、アプリケーションのいずれかからインストールできます。

SAS SCU を使用して SAS サービスをインストールするには、次の操作を実行します。

1. **ファイルのインストール/保存**タブを選択します。
2. **ファイルからインストール** を選択します。
3. **ァイルを開く**ダイアログボックスで INI ファイル選択します。
4. **開く**をクリックします。

SAS サービスをコマンドプロンプトからインストールするには、SAS サービス構成ユーティリティのディレクトリと INI ファイルが保存されたディレクトリがシステムパスからアクセスできることを確認します。コマンドプロンプトから **sasservicemngr.exe *パス*/*filename*.ini** と入力します。SAS サービスをコマンドプロンプトから入力すると、ユーザーメッセージは無効になります。

*注:* また、SAS SCU GUI のインストールタブからサービスをインストールすることもできます。

SAS サービスをアプリケーションからインストールする場合のコマンドは、**sasservicemngr.exe *パス*/*filename*.ini** です。次の表には、呼び出し元のアプリケーションに返す戻り値が示されています。

***表 8.5*** *SAS サービスのインストールまたは実行による戻り値*

| 数値コード | エラーコード | 説明 |
|---|---|---|
| 0 | SUCCESS | サービスのインストールに成功しました。 |
| 5 | ERROR_ACCESS_DENIED | サービスコントロールマネージャへのアクセスが拒否されました。 |
| 6 | ERROR_INVALID_HANDLE | サービスコントロールマネージャのロード中にエラーが発生しました。 |
| 25 | ERROR_NOT_FULL_PATH_CREATED | フルパスが作成されませんでした。 |
| 26 | USER_CANCELED_INSTALL | ユーザーがインストールをキャンセルしました。 |

| 数値コード | エラーコード | 説明 |
|---|---|---|
| 30 | SUCCESS_NO_REG_DIR | サービスはインストールされましたが、作業ディレクトリの登録に失敗しました。 |
| 35 | ERROR_BINPATH_NOTFOUND | サービスが見つからなかったため、インストールされていません。 |
| 40 | ERROR_USER_CANCEL_NOSRVC | INI ファイルが見つからなかったため、ユーザーによりインストールがキャンセルされました。 |
| 50 | ERROR_MISSING_FILE_ARGUMENT | INI ファイル内に、必要な引数がありません。 |
| 51 | ERROR_INVLAID_FILE_ARGUMENT | INI ファイルの引数に正しくない値が含まれています。 |
| 55 | ERROR_OPENFILE | INI ファイルを開くことができませんでした。 |
| 60 | ERROR_ITEMTOOLARGE | 文字列の値が最大文字数の上限を超えています。 |
| 65 | ERROR_PASSED_DECRYPT_FAILED | パスワードを解読できませんでした。 |
| 87 | ERROR_INVALID_PARAMETER | サービスパラメータが正しくありません。 |
| 123 | ERROR_INVALID_NAME | 指定されたサービス名が無効です。 |
| 1057 | ERROR_INVALID_SERVICE_ACCOUNT | アカウント名が正しくないか存在しません。 |
| 1060 | ERROR_SERVICE_DOES_NOT_EXIST | 指定されたサービスがインストール済みサービスの中に存在しません。 |
| 1065 | ERROR_DATABASE_DOES_NOT_EXIST | 指定されたデータベースが存在しません。 |
| 1072 | ERROR_SERVICE_MARKED_FOR_DELETE | 指定されたサービスに削除マークが付いています。 |
| 1073 | ERROR_SERVICE_EXISTS | ネットワーク上に重複するサービス名が存在します。 |
| 1078 | ERROR_DUPLICATE_SERVICE_NAME | ネットワーク上に重複する表示名が存在します。 |

### *SAS サービスの起動*

SAS サービスは、自動または手動で起動できます。SAS サービスが自動的に起動するように設定されている場合は、システム起動時にサービスが起動します。SAS が手動で起動するよう定義されている場合は、SAS をアプリケーションから[`net start` コマンドまたは[**Windows サービス**]ダイアログボックスを使用して起動できます。

SAS サービスを起動するには、**サービス**ダイアログボックスを使用します。

- Windows 7 および Windows Server 2008 では、**スタート** ⇨ **コントロールパネル** ⇨ **システムとセキュリティ** ⇨ **管理ツール** ⇨ **サービス**を選択します。

  その他すべての Windows 動作環境では、**スタート** ⇨ **コントロールパネル** ⇨ **管理ツール** ⇨ **サービス**を選択します。
- **サービス**リストボックスで、SAS サービスを選択します。
- **スタート**をクリックします。

### *SAS サービスの削除*

SAS サービスを Windows のサービスとして削除するには、SAS SCU またはコマンドプロンプトを使用します。

SAS SCU を使用して SAS サービスを削除するには、次の操作を実行します。

1. SAS SCU を開き、**削除**タブをクリックします。
2. SAS サービスを**既存のサービスの削除**ボックスから選択します。
3. **削除**をクリックします。
4. **Yes** を**サービスの削除**ダイアログボックスで選択して、削除を確認します。

SAS サービスをコマンドプロンプトから削除するには、`sasservicemngr.exe /remove <servicename>`と入力します。

*9 章*

# Windows 版 SAS/AF ソフトウェアでの OLE の使用

**OLE について** . . . . . **234**

**SAS/AF カタログの互換性** . . . . . **234**

**OLE オブジェクトを FRAME エントリに挿入する** . . . . . **234**
FRAME エントリへの OLE オブジェクトの挿入について . . . . . 234
OLE オブジェクトの挿入 . . . . . 235
クリップボードからの OLE オブジェクトの貼り付け . . . . . 236
HSERVICE エントリから OLE オブジェクトの読み込み . . . . . 236
ドラッグによる OLE オブジェクトの挿入 . . . . . 237

**FRAME エントリ内での OLE オブジェクトの編集** . . . . . **238**

**OLE Verb の起動** . . . . . **239**

**リンクされた OLE オブジェクトを使用する** . . . . . **240**
概要:リンクされた OLE オブジェクトを使用する . . . . . 240
リンクダイアログボックスを使用してリンクされたオブジェクトを更新 . . . . . 240
リンクされたオブジェクトの自動更新 . . . . . 241

**OLE オブジェクトの変換** . . . . . **241**

**OLE オブジェクトとアプリケーションの自動化** . . . . . **243**
概要:OLE オブジェクトとアプリケーションの自動化 . . . . . 243
OLE オートメーションサーバーが返した配列値へのアクセス . . . . . 244
値のプロパティの使用 . . . . . 245
OLE サーバーメソッドのオプションパラメータの指定 . . . . . 246
外部 OLE オートメーションインスタンスの作成 . . . . . 246
例:Microsoft Excel スプレッドシートへの SAS データの入力 . . . . . 248

**SAS/AF アプリケーションで OLE カスタムコントロール(OCX)を使用する** . . . . . **250**
概要:SAS/AF アプリケーションで OLE カスタムコントロール(OCX)を使用する 250
OLE コントロールを FRAME エントリに挿入 . . . . . 251
OLE コントロールの登録 . . . . . 251
OLE コントロールのプロパティへのアクセス . . . . . 251
SCL メソッドを使用した OLE コントロールの操作 . . . . . 253
OLE コントロールイベントへの応答 . . . . . 253

# OLE について

OLE は、異なるアプリケーションの複数の情報ソースを、1 つのドキュメントに統合する機能です。このようなオブジェクトには、テキスト、グラフィック、グラフ、サウンド、ビデオクリップなどがあります。

リリース 6.08 以降の SAS では OLE 1.0 がサポートされ、SAS/AF FRAME エントリや SAS/EIS アプリケーションへの OLE オブジェクトのリンクや埋め込みが可能です。SAS 9.4 では OLE 2.0 がサポートされ、SAS/AF フレームや SAS/EIS アプリケーションを拡張できるように多数の機能が提供されています。

*注:* Windows 版 SAS(および OLE 2.0)では、OLE 1.0 からのすべての機能がサポートされます。

SAS は、オブジェクトコンテナまたはクライアントとして機能します。FRAME エントリ内に配置するオブジェクトを作成、更新するアプリケーションは、サーバーと呼ばれます。OLE オートメーション機能を使用して、他のアプリケーションから SAS をサーバーとして使用することもできます。

OLE の一般的な情報については、Windows 動作環境のドキュメントを参照してください。SAS/AF ソフトウェアの OLE 機能を使用する際に表示されるエラーメッセージの詳細については、“OLE の使用” (620 ページ) を参照してください。

# SAS/AF カタログの互換性

Windows NT 版のリリース 6.09 や Windows 版のリリース 6.10 以降の OLE HSERVICE エントリを含む SAS/AF カタログは、SAS 9.4 のセッションでライブラリ参照名を、この SAS/AF カタログに割り当てるだけで、移植できます。

*注:* SAS 9.4 で作成された HSERVICE エントリを含む SAS/AF カタログは、CPORT プロシジャの V6.08 オプションを使用してリリース 6.08 への移植が可能ですが、移植される機能はリリース 6.08 で利用できる機能に限られます。

HSERVICE エントリは、そのエントリが作成されたプラットフォームでしか利用できません。Windows 版 SAS を使用する SAS/AF アプリケーションに含まれる OLE 機能は、他の動作環境には移植できません。移植性を考え、各種の Microsoft Windows は同一プラットフォームと見なします。

32 ビットアーキテクチャのカタログは、64 ビットのアーキテクチャへ移動なしでは使用できません。逆の場合も同じです。CPORT オプションと CIMPORT オプションを使って移動を行います。

# OLE オブジェクトを FRAME エントリに挿入する

### *FRAME エントリへの OLE オブジェクトの挿入について*

SAS では、OLE を容易にするため、オブジェクトの選択リストに次の 3 つの項目が用意されています。

**OLE - オブジェクトの挿入**
OLE オブジェクトを、登録されたサーバーアプリケーションに関連する種類の新規オブジェクト、既存のファイルから作成されたオブジェクトまたは OLE コントロールとして挿入します。

**OLE - 形式を選択して貼り付け**
OLE オブジェクトを、Windows クリップボードから FRAME エントリに貼り付けます。

**OLE - オブジェクトの読み込み**
SAS カタログ内の既存の HSERVICE エントリを参照するオブジェクトを作成します。

これらの 3 つの項目は、SAS/AF ソフトウェアの 3 つの OLE クラス(INSERT、PASTE、READOLE)に対応しています。

選択リストを使用してオブジェクトを挿入するほかに、他の Windows アプリケーションでオブジェクトを選択してドラッグし、開いている FRAME エントリにドロップすることもできます。この操作は、BUILD モードや、フレームオブジェクトまたは作業領域オブジェクトが SAS_DND_OLEOBJ 表現のドロップ領域として登録されていれば実行時に行えます。

## *OLE オブジェクトの挿入*

OLE オブジェクトを FRAME エントリに挿入するには、次の操作を実行します。

1. **COMPONENTS** ウィンドウから、**V6 オブジェクト**項目を選択してオブジェクトツリーを展開します。
2. 選択リストのオブジェクトリストをスクロールし、**OLE - オブジェクトの挿入**を選択して、その上でマウスの左ボタンを押したままにします。
3. **OLE - オブジェクトの挿入**を、FRAME エントリでオブジェクトを挿入する位置にドラッグします。オブジェクトを配置するには、マウスボタンを離します。**オブジェクトの挿入**ダイアログボックスが表示されます。
4. 挿入するオブジェクトの種類を選択します。選択可能なオブジェクトのリストは、システムに登録された OLE 対応アプリケーションの種類によって異なります。オブジェクトの種類を選択し、[OK]をクリックすると、その種類のオブジェクトが FRAME エントリに挿入されます。

   また、**ファイルから作成**をクリックして、ファイルからオブジェクトを作成することもできます。指定するファイルは、OLE オブジェクトを作成できるアプリケーションで作成されている必要があります。たとえば、Microsoft Excel がインストールされている場合は、Excel のスプレッドシートファイルからオブジェクトを作成できます。また、そのオブジェクトを、埋め込みオブジェクトではなく、リンクオブジェクトにすることもできます。リンクオブジェクトの詳細については、“リンクされた OLE オブジェクトを使用する” (240 ページ) を参照してください。

   挿入するオブジェクトの種類またはファイル名の選択が終了したら、**OK** をクリックします。オブジェクトが FRAME エントリに挿入されます。
5. **BUILD** ウィンドウをアクティブにしてから、次を選択します。**ビュー** ⇨ **プロパティウィンドウ**

   プロパティウィンドウで、オブジェクトを選択し、**オブジェクト属性**を選択します。

   オブジェクトエントリの名前を **Entry** エントリフィールドに入力します。デフォルトでは、現在のカタログのエントリが指定されます。オブジェクトの名前を**名前**で変更することもできます。HSERVICE エントリは、FRAME 編集セッションで**保存**または**終了**を選択するまでは作成されません。

**OK** をクリックします。

### *クリップボードからの OLE オブジェクトの貼り付け*

OLE オブジェクトを Windows クリップボードから貼り付けるには、次の操作を実行します。

1. 他の Windows アプリケーションから、FRAME エントリに挿入するオブジェクトまたはデータを Windows クリップボードにコピーするか切り取ります。
2. **COMPONENTS** ウィンドウから、**V6 オブジェクト**項目を選択してオブジェクトツリーを展開します。
3. 選択リストのオブジェクトリストをスクロールし、**OLE - 形式を選択して貼り付け**を選択し、その上でマウスの左ボタンを押したままにします。**OLE - 形式を選択して貼り付け**をフレームにドラッグします。**形式を選択して貼り付け**ダイアログボックスが表示されます。
4. クリップボードの内容に基づいて、挿入する OLE オブジェクトの種類を選択します。この種類は、コピーしたデータを作成したアプリケーションによって決定されます。たとえば、通常は、Microsoft Word データを Microsoft Word オブジェクトとして貼り付けます。
5. FRAME エントリに実際のデータを埋め込むかわりに OLE オブジェクトをデータにリンクする場合は、**リンクの貼り付け**を**形式を選択して貼り付け**ダイアログボックスで選択します。リンクオブジェクトの詳細については、“リンクされた OLE オブジェクトを使用する” (240 ページ) を参照してください。

   *注:* 保存していないドキュメントなど一時的なソースからデータを貼り付けると、データソースへのリンクを後で試行する際にこのデータソースが存在せず、検索できません。Windows クリップボードにコピーする前に、データファイルを保存してください。
6. 貼り付けるオブジェクトの種類を選択したら、**OK** をクリックします。オブジェクトが FRAME エントリに貼り付けられます。
7. **ビュー** ⇨ **プロパティウィンドウ**の順に選択します。オブジェクトを**プロパティ**ボックスから選択し、**オブジェクト属性**をクリックします。
8. オブジェクトエントリの名前を **Entry** エントリフィールドに入力します。デフォルトでは、現在のカタログのエントリが指定されます。オブジェクトの名前を**名前**で変更することもできます。HSERVICE エントリは、FRAME 編集セッションで**保存**または**終了**を選択するまでは作成されません。

   **OK** をクリックします。

### *HSERVICE エントリから OLE オブジェクトの読み込み*

HSERVICE エントリとして SAS カタログに格納されている既存の OLE オブジェクトを読み込むには、次の操作を実行します。

1. **COMPONENTS** ウィンドウから、**V6 オブジェクト**項目を選択してオブジェクトツリーを展開します。
2. 選択リストのオブジェクトリストをスクロールし、**OLE - オブジェクトの読み込み** を **BUILD** ウィンドウにドラッグします。
3. カーソルをブランクオブジェクトに合わせ、右クリックし、**オブジェクト属性**を選択します。

4. In the **OLE-オブジェクト属性の読み込み**ウィンドウで、HSERVICE エントリの名前を**エントリ**フィールドに入力します。デフォルトでは、現在のカタログのエントリが指定されます。エントリの検索に**選択**ウィンドウを使用する場合は、**エントリ**フィールドの隣の矢印をクリックします。

   **OK** をクリックします。オブジェクトが FRAME エントリに挿入され、選択した位置に表示されます。

   *注:* 読み込んだ HSERVICE エントリの名前は変更できません。HSERVICE エントリに別の名前を割り当てる場合は、オブジェクトを読み込む前に、HSERVICE エントリをコピーして別の名前にしておきます。

## *ドラッグによる OLE オブジェクトの挿入*

### *OLE オブジェクトを FRAME エントリに挿入する*

OLE オブジェクトをドラッグアンドドロップで FRAME エントリに挿入するには、次の操作を実行します。

1. サーバーアプリケーションを使用して、オブジェクトを作成します。たとえば、Microsoft Excel チャートオブジェクトを FRAME エントリに埋め込む場合は、Microsoft Excel を使用してオブジェクトを作成します。または、他のアプリケーションに埋め込まれている OLE オブジェクトを選択することもできます。
2. SAS とサーバーアプリケーションの両方を実行して、サーバーアプリケーション(およびそのオブジェクト)と、**SAS BUILD:DISPLAY** ウィンドウ(および FRAME エントリ)の両方がスクリーンに表示されるようにアプリケーションウィンドウを配列します。
3. サーバーアプリケーションでオブジェクトを選択します。マウスボタンを押したまま、オブジェクトをサーバーアプリケーションから FRAME エントリでのオブジェクトを配置する位置までドラッグします。カーソルが矢印付きのボックスに変わり、FRAME エントリがオブジェクトの有効なドロップ場所であることが示されます。ドラッグアンドドロップで OLE オブジェクトを挿入する場合、FRAME エントリ内にあらかじめ領域を作成する必要はありません。また、修飾キーを使用してドラッグすることもできます。それについては次で説明されています。“ドラッグ操作の変更” (237 ページ)ドラッグアンドドロップの動きを制御するには、次の操作を行います。

   マウスボタンを離す(つまり、オブジェクトを"ドロップする")と、オブジェクトが FRAME に挿入され、自動的に OLE オブジェクトの名前と HSERVICE エントリが作成されます。選択した位置にオブジェクトが表示されます。

### *実行時に OLE オブジェクトをドラッグする*

SAS/AF アプリケーションの実行時に、OLE オブジェクトをドラッグアンドドロップできるようにします。そのためには、OLE オブジェクトの種類を、有効なドラッグアンドドロップ表現で登録する必要があります。

OLE オブジェクトは SAS_DND_OLEOBJ 表現で登録されている必要があります。ドラッグアンドドロップするオブジェクトの登録の詳細については、SAS/AF オンラインドキュメントから、FRAME アプリケーション開発環境の操作方法と Widget クラスに関する情報を参照してください。

### *ドラッグ操作の変更*

デフォルトでは、他のアプリケーションから SAS に OLE オブジェクトをドラッグすると、(読み込み専用で削除できないオブジェクトでない限り)オブジェクトは移動されます。このデフォルト操作を無効にするには、ドラッグ変更キーを使用します。次のキーを押すことによって、別のドロップ操作の実行が指示されます。

- サーバーアプリケーションからオブジェクトをコピーするには、Ctrl キーを押しながら、オブジェクトをターゲットウィンドウにドロップします。Ctrl キーを押すと、カーソルが、ボックスとプラス記号(＋)が付いた矢印へと変わります。
- SAS/AF FRAME エントリ内にデータへのリンクを作成するには、Ctrl キーと Shift キーを押しながら、オブジェクトを **BUILD** ウィンドウにドロップします。Ctrl キーと Shift キーを押すと、カーソルが、ボックスとプラス記号(＋)が付いた矢印へと変わります。(この機能は、アプリケーションによって異なる可能性があります)。一時的なソースからリンクオブジェクトを貼り付けないでください。データソースが存在しないと、データソースが検索されません。

非デフォルトのドラッグアンドドロップ操作がサーバーアプリケーションでサポートされていれば、その操作を実行することもできます。マウスの右ボタンを使用してオブジェクトを選択し、FRAME エントリにドラッグアンドドロップします。マウスボタンを離すと、移動、コピー、オブジェクトへのリンクのいずれかを選択するポップアップメニューが表示されます。ポップアップメニューの項目は、OLE オブジェクトの種類によって異なる場合があります。

## FRAME エントリ内での OLE オブジェクトの編集

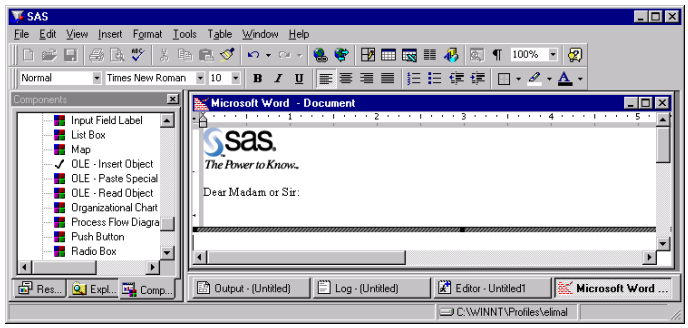
OLE 2.0 の最も優れた機能の 1 つに、ビジュアル編集があります。これは、埋め込まれているオブジェクトを、他のアプリケーションへと切り替えることなく、埋め込み先で編集できる機能です。

作成時に FRAME エントリの OLE オブジェクトをビジュアル編集するには、右クリックし、**編集**を選択します。実行時にビジュアル編集を行うには、単純にオブジェクトをダブルクリックします。オブジェクトのアプリケーションがサーバーアプリケーションとしてビジュアル編集をサポートしている場合は、次のようになります。

- FRAME エントリのオブジェクトの表示が、その実際のオブジェクトの編集セッションに変更されます。オブジェクトの境界に、サーバーアプリケーションから提供されるツールが挿入されます。
- SAS メニューバーは、サーバーアプリケーションのメニューバーに適応して変化します。ファイルメニューとウィンドウメニューは同じままですが、その他のメニューバーはサーバーアプリケーションのメニューに変更されます。
- サーバーアプリケーションに、ツールバーアイコンや移動可能なツールボックスなどのツールが用意されている場合は、それらのアイテムも使用できるようになります。

例えば、図 9.1 (238 ページ) では Microsoft Word のオブジェクトが再生された状態での SAS/AF FRAME エントリを表示します。

***図 9.1*** *Word オブジェクトが再生された SAS/AF FRAME エントリ*



このように変換すると、サーバーアプリケーションに用意されているツールやメニューをすべて使用してオブジェクトを編集できます。

ビジュアル編集セッションを終了するには、FRAME エントリ内でオブジェクト以外の部分をクリックします。SAS はセッションの制御を再開し、SAS のデフォルトのメニューおよびツールに切り替わります。

*注:*

1. オブジェクトが以前に保存されている(HSERVICE エントリが作成されている)場合のみ、HSERVICE エントリが、ビジュアル編集セッション終了時に自動的に更新されます。保存されていない場合は、SAS/AF ソフトウェアの**ファイル**メニューから*上書き保存*または*終了*を選択して、HSERVICE エントリを作成する必要があります。TESTAF モード時にオブジェクトを変更して、変更を HSERVICE エントリに保存する場合、BUILD モードに戻る前にローカルメニューから*更新*を選択して、オブジェクトの内容を更新する必要があります。
2. BUILD モードでビジュアル編集中に FRAME エントリ内で OLE オブジェクトを移動しても、ビジュアル編集セッションを終了するためにオブジェクト以外の場所をクリックすると、オブジェクトが元の場所に戻ります。オブジェクトを FRAME エントリ内の別の位置に移動する場合は、ビジュアル編集セッションを終了し、オブジェクト領域をドラッグして移動します。
3. 大部分の OLE オブジェクトは、ダブルクリックして再生する必要があります。ただし、シングルクリックするだけで再生できる種類のオブジェクトものもあります。
4. リンクオブジェクトまたは OLE オブジェクトのサーバーアプリケーションでビジュアル編集がサポートされない場合にオブジェクトを編集しようとすると、オブジェクトを編集するためにサーバーアプリケーションが別個のインスタンスとして起動されます。この操作は、オープン編集と呼ばれ、リンクオブジェクトや OLE 1.0 のすべてのオブジェクトで使用されます。

## OLE Verb の起動

OLE コントロールを除く各 OLE オブジェクトには、ダブルクリックしたときに実行されるデフォルト操作があります。多くのオブジェクトのデフォルト操作は、**編集**です (OLE 2.0 オブジェクトには、ビジュアル編集セッションを起動します。リンクされたオブジェクトや、すべての OLE 1.0 オブジェクトには、オープン編集セッションを起動します)。ただし、**編集**が 2 次操作のオブジェクトもあります(たとえば、メディアクリップオブジェクトでは、**再生**が 1 次操作です。また、実行できる操作を 2 つ以上持つオブジェクトが多く存在します。そのため、そのようなオブジェクトは 2 つ以上の OLE Verb を認識します (OLE オブジェクトを BUILD:DISPLAY モードでダブルクリックしてもデフォルトの操作は実行されませんが、TESTAF モードでダブルクリックすると実行されることに注意してください)。

BUILD モードで OLE オブジェクトの OLE Verb のメニューを表示するには、オブジェクトを右クリックしてください。OLE オブジェクトの名前が、ポップアップメニューの一番下に表示されます。カスケードメニューに、そのオブジェクトで有効な OLE Verb のリストが表示されます。このメニューから動詞を選択して、その操作を実行します。デフォルトの動詞は、動詞リストの先頭に表示されます。

たとえば、Microsoft Excel オブジェクトは、**編集**(ビジュアル編集)と**開く**(オープン編集)を認識します。メディアクリップオブジェクトは、**再生**と**編集**を認識します。

有効な動詞のリストは、**関連する Verb** の項目を**オブジェクト属性**ダイアログボックスでクリックすることでも表示できます。リストには、動詞の名前が表示されるだけで、こ

のリストから動詞を起動することはできません。前述したように、リストの先頭にある動詞は、デフォルトの動詞です。

SCL を使用すると、特定の OLE オブジェクトで認識される動詞であれば、_EXECUTE_メソッドに、この動詞を引数として使用して起動できます。たとえば、次のコードは動詞である**再生**を(OLE オブジェクト **mediaobj** で)起動します。

```
call notify('mediaobj', '_EXECUTE_', 'Play');
```

_EXECUTE_メソッドの 1 つの呼び出しに、複数の動詞を指定できます。_EXECUTE_メソッドの詳細については、“_EXECUTE_” (610 ページ) を参照してください。

# リンクされた OLE オブジェクトを使用する

## *概要:リンクされた OLE オブジェクトを使用する*

OLE リンクオブジェクトは、オブジェクトのサーバーアプリケーションに関する情報を保持し、ディスク上のデータファイルを参照します。ただし、オブジェクト自体のデータは含みません。リンクオブジェクトが保持しているのは、リンク元の内容を表す静的なイメージです。

**リンク**ダイアログボックスから、リンクオブジェクトの更新を次のように指定できます。

- オブジェクトが参照するソースファイルを更新する場合は、**リンク**ダイアログボックスを使用すると、リンクされたオブジェクトを自動的に更新できます(変更を反映する前に、FRAME エントリを再ロードしておく必要があります)。
- 手動で、**今すぐ更新**を**リンク**ダイアログボックスで選択するか、SCL の_UPDATE_メソッドを使用します。
- 手動により、**リンク**ダイアログボックスまたは SCL の_UPDATE_メソッドを使用して、リンクオブジェクトが別のソースファイルを参照するように設定します。

FRAME エントリに挿入する、OLE リンクオブジェクトについて、次に説明します。

- ビジュアル編集ではなく、オープン編集のみをサポートします(詳細については、“FRAME エントリ内での OLE オブジェクトの編集” (238 ページ) を参照)。FRAME 内に表示されているオブジェクトをダブルクリックすると、サーバーアプリケーションが別ウィンドウで起動し、オブジェクトのデータファイルを開きます。

  サーバーアプリケーションを使用して、リンクオブジェクトが参照するデータファイルを開くと、リンクオブジェクトのデータを更新することもできます。
- 既存のデータファイルを参照する必要があります。オブジェクトのリンク元のデータファイルが保存されている場所を変更する場合は、オブジェクトのリンク情報を更新する必要があります。

**OLE - 形式を選択して貼り付け**を使用してリンクオブジェクトを作成する場合は、永続的なデータソースを貼り付ける必要があります。つまり、データソースをディスクに保存してあることが必要です。リンクオブジェクトをサーバーアプリケーションの一時的なデータソースから作成し、そのままデータソースを保存しないでサーバーアプリケーションを終了すると、データを見つけられないためオブジェクトを更新できません。

## *リンクダイアログボックスを使用してリンクされたオブジェクトを更新*

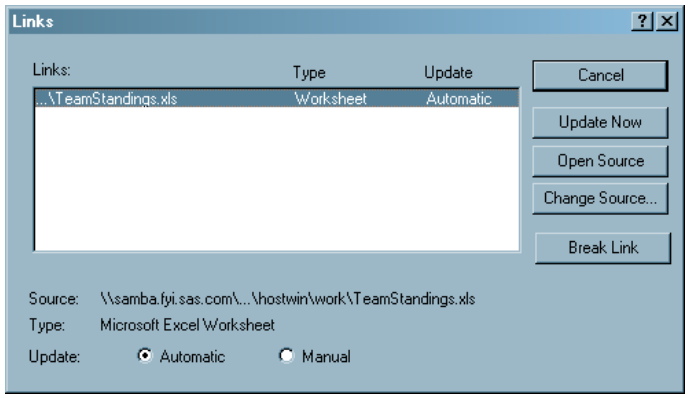
**リンク**ダイアログボックスを使用してリンク情報を更新するには、次の操作を行います。図 9.2 (241 ページ) に示されています。

1. オブジェクトを右クリックします。ポップアップメニューが表示されます。オブジェクトの種類が一番下のメニュー項目として表示されます。
2. 一番下のメニュー項目をクリックします。オブジェクトの有効な OLE Verb を含むカスケードメニューが表示されます。
3. **リンク**をクリックします。**リンク**ダイアログボックスが表示され、FRAME エントリ内のすべてのリンクオブジェクトに関するリンク情報が示されます(FRAME エントリ内にリンクオブジェクトが存在しない場合、**リンク**の項目は無効になります)。
4. 必要に応じて、**リンク**ダイアログボックスを使用して、リンクオブジェクトの情報を変更します。たとえば、データファイルが別の場所に存在する場合は、リンクオブジェクトのリンク元を変更できます。

また、OLE リンクオブジェクトの**リンク**ダイアログボックスを開くには、コマンド行から DLGLINKS コマンドを使用することもできます。SCL の _EXECUTE_ メソッドを使用して、DLGLINKS コマンドを発行することもできます。例を次に示します。

```
call notify('linkobj','_execute_','dlglinks');
```

***図 9.2** [リンク]ダイアログボックス*




### *リンクされたオブジェクトの自動更新*

リンクオブジェクトのソースを SCL プログラムを使用して変更するには、_UPDATE_ メソッドで、オブジェクトに割り当てる新しい HSERVICE エントリを指定します。OLE オブジェクトの _UPDATE_ メソッドは、3 つ目の引数として HSERVICE エントリ名を受け取ります(このメソッドは、Widget クラスの _UPDATE_ メソッドより優先されます)。OLE _UPDATE_ メソッドの構文については、“_UPDATE_” (615 ページ)を参照してください。

## OLE オブジェクトの変換

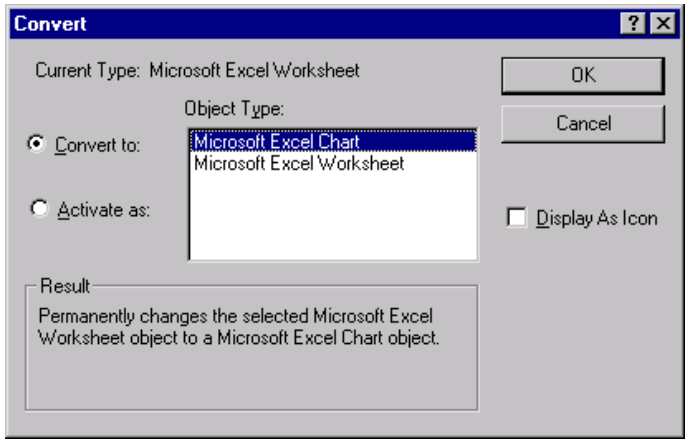
OLE オブジェクトは、1 つのサーバーアプリケーションにのみ関連付けられます。しかし、OLE オブジェクトには、作成に使用したサーバーアプリケーションとは別のサーバーアプリケーションで使用できるように変換できるものもあります。

オブジェクトを変換するには、**変換**ダイアログボックスを使用します。図 9.3 (242 ページ) で示されています。このダイアログボックスでは、次のことができます。

- オブジェクトの表示方法をアイコンからオブジェクトの内容に変更したり、オブジェクトの内容からアイコンに変更したりすることができます。

- オブジェクトのサーバーアプリケーションの種類を別のサーバーアプリケーションに変更することができます。
- オブジェクトの種類を変更せずに、オブジェクトを作成したアプリケーションとは異なるサーバーアプリケーションを使用して、オブジェクトを再生することができます。

*図 9.3 [変換]ダイアログボックス*




SAS/AF FRAME エントリ内の OLE オブジェクトを変換するには、次の操作を実行します。

1. オブジェクトを右クリックします。
2. ポップアップメニューの一番下で、オブジェクトの名前を選択します。カスケードメニューが表示されます。
3. カスケードメニューから**変換**を選択します。**変換**ダイアログボックスが表示され、その中に、選択したオブジェクトの変換先として有効なオブジェクトの種類が表示されます。
4. オブジェクトを他の種類に実際に変換する場合は、変換後の種類を選択し、**OK** をクリックします。

   オブジェクトの表示をアイコンまたは内容のいずれかに切り替えるには、**アイコンで表示**をクリックします。

   オブジェクトを他のサーバーを使用して有効化するには、**有効化の方法**をクリックし、使用するサーバーアプリケーションを選択します。
5. **OK** をクリックします。

この方法以外に、OLE オブジェクトの**変換**ダイアログボックスを開くには、オブジェクトを選択し、コマンド行で DLGCONVERT コマンドを発行する方法もあります。DLGCONVERT コマンドを発行するのに、SCL の_EXECUTE_メソッドを使用することもできます。例を次に示します。

```
call notify('sheetobj','_execute_',
            'dlgconvert');
```

# OLE オブジェクトとアプリケーションの自動化

## 概要:OLE オブジェクトとアプリケーションの自動化

一部の Windows アプリケーションでは、スクリプト言語を使用することで、オブジェクトや外部アプリケーションの制御および更新を自動化することができます。SAS/AF では、OLE オートメーションに、SAS コンポーネント言語(SCL:SAS Component Language)を使用できます。SCL コードを使用して OLE オブジェクトに命令を送信することにより、SAS/AF アプリケーションでのユーザーの操作に基づいて、オブジェクトのデータを更新することができます。

SAS/AF ソフトウェアでは、次の OLE オブジェクトやアプリケーションを自動化できます。

- OLE クラスを使用する、FRAME エントリに埋め込まれた OLE オブジェクト
- OLE クラスを使用する、FRAME エントリにリンクされた OLE オブジェクト
- OLE オートメーションクラスを使用する、FRAME エントリに割り当てられていない OLE アプリケーション

SCL を使用すると、OLE オートメーションをサーバーとしてサポートする OLE オブジェクトまたはアプリケーションと通信できます。この通信では、SAS はクライアントとして機能し、オートメーションアプリケーションはサーバーとして機能します。サーバーは、OLE オートメーションオブジェクトを提供します。OLE オートメーションオブジェクトを制御するには、SCL コードを使用します。SCL メソッドを使用すると、OLE メソッドをサーバーに送信して実行することができます。また、制御するオブジェクトのプロパティを取得および設定することもできます。OLE オートメーションサーバーでは、複数の種類のオブジェクトをサポートできます。各種オブジェクトには、それぞれ独自のメソッドおよびプロパティを定義できます。使用できる SCL メソッドについては 表 9.1 (243 ページ) に一覧があり、詳細については “OLE クラスメソッドの要約” (607 ページ) で説明しています。

*注:* 表の SCL OLE オートメーションメソッドと、OLE オートメーションサーバー提供のメソッドを混同しないようにしてください。SAS/AF では、_COMPUTE_および_DO_の SCL メソッドを使用すると、OLE オートメーションサーバーでサポートされるメソッドにアクセスできます。OLE オートメーションサーバーごとに異なるメソッドがサポートされますが、メソッドを呼び出すには、必ず_COMPUTE_メソッドまたは_DO_メソッドを使用する必要があります。サブクラス化することで、この表にあるメソッドなどをカプセル化する新しいメソッドの作成が可能です。

***表 9.1*** *OLE オートメーションクラスメソッド*

| OLE オートメーションメソッド | 説明 |
|---|---|
| _COMPUTE_ | OLE オートメーションサーバーでサポートされるメソッドを呼び出して値を返します |
| _DO_ | OLE オートメーションサーバーでサポートされるメソッドを呼び出します。(値は返しません)。 |
| _GET_PROPERTY_ | OLE オートメーションサーバーが公開するプロパティの値を取得します |

| OLE オートメーションメソッド | 説明 |
|---|---|
| _GET_REFERENCE_ID_ | OLE オートメーションサーバーが提供するオブジェクトの参照識別子を返します |
| _IN_ERROR_ | オブジェクトのエラーステータスを返します |
| _NEW_ | OLE オートメーションサーバーの外部インスタンスに SCL 識別子を割り当てます。 |
| _SET_PROPERTY_ | OLE オートメーションサーバーが公開するプロパティの値を設定します |

*注:* オートメーションサーバーと、OLE オートメーションメソッドを使用する SAS 間で受け渡される戻り値と引数は、サーバーによって pass-by-reference として定義される引数を含む参照ではなく、値で受け渡されます。引数には、変更可能な値を参照するポインタではなく、実際の静的な値が含まれます。

### *OLE オートメーションサーバーが返した配列値へのアクセス*

SCL メソッドと OLE オートメーションサーバーを使用すると、次のことが可能になります。

- SCL リストとして SAS に渡される 1 次元配列を受信できます。
- 多次元の SCL 配列を送受信できます。

最初の例では、SCL コードで Microsoft Excel ワークシートにリストボックスを作成、データを入力し、リストボックスの内容を SCL リストに保存します。

```
list=makelist();  /* create the SCL list */
   /* Add a Listbox in a worksheet */
call send(worksht, '_COMPUTE_', 'Listboxes',
          listbox);
call send(listbox, '_DO_', 'Add', 20, 50,
          40, 100);
call send(worksht, '_COMPUTE_', 'Listboxes',
          1, listone);
   /* Fill the Listbox with a range of */
   /* values from the worksheet        */
call send(listone, '_SET_PROPERTY_',
          'ListFillRange', 'A1:A3');
   /* Get the contents of the Listbox */
call send(listone, '_GET_PROPERTY_',
          'List', list);
```

次の SCL コードでは、複数の SCL 配列を使用して、新しい Microsoft Excel ワークシートを作成します。

```
Init:
/* Initialization */
   HostClass = loadclass('sashelp.fsp.hauto');

/* Instantiate the Excel object and make it visible */
   call send (Hostclass, '_NEW_', ExcelObj, 0, 'Excel.Application');
   call send (ExcelObj, '_SET_PROPERTY_', 'Visible', -1);
```

```
/* Get the Workbook Object, add a new Sheet and get the Sheet object */
   call send (ExcelObj, '_GET_PROPERTY_', 'Workbooks', WkBkObj);
   call send (WkBkObj, '_DO_', 'Add');
   call send (ExcelObj, '_GET_PROPERTY_', 'ActiveSheet', WkShtObj);

   dcl char names{3,2) =  ('Lucy', 'Ricky',
                          'Julliette', 'Romeo',
                          'Elizabeth', 'Richard');

/* Set the range to be A1:A4 and fill that range with names */
   call send(WkShtObj, '_COMPUTE_', 'Range', 'A1', 'B3', RangeObj);
   call send(RangeObj, '_SET_PROPERTY_', 'Value', names);

   dcl num primes{2,4} = ( 1, 3, 5, 7,
                           11, 13, 17, 23);

/* Set the range to be A5:D6 and fill that range with ints values */
   call send(WkShtObj, '_COMPUTE_', 'Range', 'A5', 'D6', RangeObj);
   call send(RangeObj, '_SET_PROPERTY_', 'Value', primes);

   dcl char totals{1,4} = ('=SUM(A5,A6)',
                           '=SUM(B5,B6)',
                           '=SUM(C5,C6)',
                           '=SUM(D5,D6)');

/* Set the range to be A1:A4 and fill that range with totals */
   call send(WkShtObj, '_COMPUTE_', 'Range', 'A7', 'D7', RangeObj);
   call send(RangeObj, '_SET_PROPERTY_', 'Value', totals);

   dcl char vals{7,4};

   call send(WkShtObj, '_COMPUTE_', 'Range', 'A5', 'D7', RangeObj);
   call send(RangeObj, '_GET_PROPERTY_', 'Value', vals);
return;
```

## *値のプロパティの使用*

OLE オートメーションサーバー(OLE カスタムコントロールを含む)では、プロパティまたはメソッドの 1 つを値のプロパティとして指定できます。オートメーションコードが、プロパティ名またはメソッド名が明確に指定されていないサーバーが提供するオブジェクトにアクセスする際に、値のプロパティはデフォルトのプロパティまたはメソッドとして使用されます。

SCL コードでは、_GET_PROPERTY_または_SET_PROPERTY_を呼び出す際にプロパティ名が空白、あるいは、_DO_または_COMPUTE_を使用する際にメソッド名が空白であると、サーバーの値のプロパティにアクセスされます。たとえば、テキストのプロパティが値プロパティの場合は、コード:

```
call notify('sascombo', '_set_property_', '',
            'An excellent choice');
```

を次のように記述します。

```
call notify ('sascombo', '_set_property_',
             'Text', 'An excellent choice');
```

SAS 提供の SAS コンボボックスと SAS エディット両方の制御により、テキストが値プロパティとして指定されています。

### *OLE サーバーメソッドのオプションパラメータの指定*

一部の OLE サーバーアプリケーションでは、任意指定のパラメータを含むメソッドが公開されています。メソッドがサポートするパラメータに値を指定しないと、OLE サーバーでは、それらのパラメータにデフォルトの値が使用されます。どのパラメータが任意指定であるかについては、使用する OLE サーバーアプリケーションのマニュアルを参照してください。

SAS では、任意指定のパラメータを削除し、その位置に SAS 欠損値を指定することもできます。デフォルトの欠損値文字はピリオドです。ただし、この文字は、MISSING システムオプションを使用すると変更できます。

たとえば、Microsoft Excel では、大部分が任意指定の 11 個の引数が渡される ChartWizard メソッドをサポートします。次の SCL コードは、すべての引数を使用してこのメソッドを呼び出します。

```
call send(chart, '_DO_', 'ChartWizard', hcell,
                 -4098, 6, 1, 0, 0, 1,
                 "Automation at work!",
                 'Column', 'Value', 'Row');
```

次に、任意指定のパラメータを削除し、欠損値文字に入れ替えた同等の SCL コードを示します。

```
call send(chart, '_DO_', 'ChartWizard', hcell,
                .,.,.,.,.,.,
                "Automation at work!",
                 .,.,.);
```

*注:* SCL コードは、メソッド呼び出し時に、任意指定のパラメータの位置を順守する必要があります。欠損値文字を引数として指定する場合は、OLE サーバーのメソッドで任意指定のパラメータの位置に欠損文字を指定する必要があります。

### *外部 OLE オートメーションインスタンスの作成*

外部 OLE オートメーションインスタンスは、ローカルマシン上またはリモートマシン上のアプリケーションのインスタンスです。外部 OLE アプリケーションを自動化する前に、OLE オートメーションクラスのインスタンスを作成する必要があります。この作業は、FRAME エントリに埋め込むか、またはリンクするオブジェクトを自動化する場合には必要ありません。これは、FRAME エントリにオブジェクトを配置すると、インスタンスが自動的に作成されるためです。OLE クラスと異なり、OLE オートメーションクラスは Widget クラスから派生したものではありません。そのため、FRAME エントリに含めるための視覚的なコンポーネントを持ちません。かわりに、自動化を実行する SCL コード内で、LOADCLASS 関数を使用して、HAUTO クラスのインスタンスをロードする必要があります。例を次に示します。

```
hostcl=loadclass('sashelp.fsp.hauto');
```

OLE オートメーションクラスのインスタンスを作成した後に、新しいインスタンスを、CALL SEND を使用してメソッドを呼び出す際に必要な SCL オブジェクト識別子と OLE サーバーアプリケーションに割り当てる必要があります。識別子を取得するには、新しく作成した OLE オートメーションクラスのインスタンスに対して_NEW_メソッドを使用します。次の例では、オブジェクト識別子を **oleauto** に保存し、オブジェクトをローカルマシンの Microsoft Excel (Windows レジストリにあり、**Excel.Application** という識別子を持つ)に割り当てます。

```
call send(hostcl, '_NEW_', oleauto, 0,
          'Excel.Application');
```

リモートマシンに OLE オートメーションクラスのインスタンスを作成するには、分散 COM の構成プロパティ(DCOMCNFG.EXE)を使用したリモートインスタンスの開始を許可するようにリモートマシンを構成する必要があります。DCOMCNFG.EXE は、\Windows\SYSTEM32 フォルダに収められています。分散 COM の構成プロパティの詳細は、Windows のマニュアルを参照してください。次の例は、リモートマシン上に Microsoft Excel のインスタンスを作成します。作成後、そのインスタンスへのメソッド呼び出しおよびプロパティ呼び出しは、インスタンスがローカルマシンにある場合と同様に動作します。

```
Init:
  HostClass = loadclass('sashelp.fsp.hauto');
  ExcelObj = 0;

  /* Define the machine name and put it in a list  */

  machineName = '\\Aladdin';
  inslist = makelist();
  attrlist = makelist ();

  rc = insertc (attrlist, machineName, -1, 'remoteServer');
  rc = insertl (inslist, attrlist, -1, '_ATTRS_');

  /* Instantiate the Excel object and make it visible */

  call send (HostClass, '_NEW_',ExcelObj, inslist,
  'Excel.Application');
  call send (ExcelObj, '_SET_PROPERTY_', 'Visible', -1);
return;
```

_NEW_ メソッドの詳細については、“_NEW_” (614 ページ) を参照してください。

OLE オートメーションオブジェクトのインスタンスを作成すると、フレームに埋め込まれているか、またはリンクされているオブジェクトを自動化する方法とまったく同じ方法で、オブジェクトを自動化できます。次の表に、オブジェクトの種類間の主な違いを示します。

***表 9.2*** *OLE 自動化オブジェクト*

| SAS OLE オブジェクト | SAS OLE 自動化オブジェクト |
|---|---|
| Widget クラスから派生 | Object クラスから派生 |
| 視覚的なコンポーネント(FRAME エントリに配置するオブジェクト)を持ちます。 | FRAME エントリ内に視覚的なコンポーネントを持たない。 |
| FRAME エントリ内の領域にドラッグアンドドロップでオブジェクトを配置することにより作成。 | LOADCLASS ステートメントと SCL の _NEW_メソッドを使用することにより作成。 |
| OLE サーバーによってサポートされる、特定の種類のデータオブジェクトを表します。 | OLE サーバーによってサポートされる最上位のアプリケーションを表す。このアプリケーションを使用すると、特定の種類のデータオブジェクトを開くことができます。 |

| SAS OLE オブジェクト | SAS OLE 自動化オブジェクト |
| --- | --- |
| FRAME エントリのオブジェクト名を渡して、CALL NOTIFY でメソッドを呼び出すことができます。 | _NEW_メソッド、_GET_PROPERTY_メソッド、または_COMPUTE_メソッドが返すオブジェクト識別子を渡して、CALL SEND でメソッドを呼び出す必要があります。 |

## *例:Microsoft Excel スプレッドシートへの SAS データの入力*

次の表に、SAS データセットのデータを Microsoft Excel スプレッドシートに入力する SCL コードを示します。

***表 9.3*** *Microsoft Excel スプレッドシートに入力する SCL コード*

| 操作 | SCL コード |
| --- | --- |
| OLE オートメーションクラスのインスタンスをロードし、Excel を呼び出します。自動化の進行状況を確認できるように、オブジェクトを Visible に設定する。 | `LAUNCHXL:`<br><br>`hostcl = loadclass('sashelp.fsp.hauto');`<br>`call send(hostcl, '_NEW_', excelobj, 0,`<br>`'Excel.Application');`<br>`call send(excelobj, '_SET_PROPERTY_', 'Visible', 'True');`<br>`return;` |
| 現在の Workbooks プロパティの識別子を取得し、ワークシートを追加します。次に、新しいワークシートの識別子を取得する。 | `CREATEWS:`<br>`call send(excelobj, '_GET_PROPERTY_', 'Workbooks',`<br>`wbsobj);`<br>`call send(wbsobj, '_DO_', 'Add' );`<br>`call send(excelobj, '_GET_PROPERTY_', 'ActiveSheet',`<br>`wsobj);` |
| SAS データセットを開きます。 | `dsid=open('sasuser.class','i');`<br><br>`call set(dsid);`<br>`rc=fetch(dsid);`<br>`nvar=attrn(dsid, 'NVARS');`<br>`nobs=attrn(dsid, 'NOBS');` |

| 操作 | SCL コード |
|---|---|
| データセットのデータ値を全検索し、Excel ワークシートのセルにデータを 1 行ずつ入力します。 | `do`<br>`col=1 to nvar;`<br>`  call send(wsobj, '_COMPUTE_', 'Cells',1,col,retcell);`<br>`  var=varname(dsid,col);`<br>`  call send(retcell,'_SET_PROPERTY_', 'Value',var);`<br>`end;`<br>`do while (rc ne -1);`<br>`   do row = 1 to nobs;`<br>`      do col = 1 to nvar;`<br>`         r=row+1;`<br>`         call send (wsobj, '_COMPUTE_', 'Cells', r ,col,retcell);`<br>`         if vartype(dsid,col) eq 'N' then do;`<br>`            varn=getvarn(dsid,col);`<br>`            call send(retcell, '_SET_PROPERTY_', 'Value' ,varn);`<br>`         end;`<br>`         else do;`<br>`            varc=getvarc(dsid,col);`<br>`            call send(retcell, '_SET_PROPERTY_', 'Value' ,varc);`<br>`         end;`<br>`      end;`<br>`      rc=fetch(dsid);`<br>`   end;`<br>`end;`<br>`dsid=close(dsid);`<br>`return;` |
| ワークシートを閉じ、Excel セッションを終了します。_TERM_メソッドにより、OLE オートメーションインスタンスが削除される。 | `QUITXL:`<br>`call send(excelobj,'_GET_PROPERTY_', 'ActiveWorkbook',`<br>` awbobj);`<br>`call send(awbobj,  '_DO_', 'Close', 'False');`<br>`call send(excelobj, '_DO_', 'Quit');`<br>`call send(excelobj, '_TERM_');`<br>`return;` |

この例から、アプリケーションオブジェクトの自動化には、オブジェクトのプロパティとメソッドの知識が必要であることがわかります。Excel オートメーションオブジェクトに使用する SCL コマンドを決定するには、Excel 内のマクロの記録機能を使用して、自動化するタスクを実行します。Visual Basic のコードが生成されます。Visual Basic コードは、同等の SCL ステートメントや SCL 関数よりも簡単です。

次の表には、Visual Basic コードおよび同等の SCL の一部を示します。

***表 9.4*** *Visual Basic コードの例および同等の SCL*

| Visual Basic のコード | SCL の OLE オートメーション |
|---|---|
| Excel を起動して表示します<br>`Set excelobj = CreateObject("Excel.Application")`<br>`excelobj.Visible = True` | `hostcl = loadclass('sashelp.fsp.hauto');`<br><br>`call send ( hostcl,  '_NEW_', excelobj, 0,`<br>`   'Excel.Application');`<br>`call send (excelobj,'_SET_PROPERTY_',`<br>`   'Visible','True');` |

| Visual Basic のコード | SCL の OLE オートメーション |
|---|---|
| 新しいワークシートを作成します<br>`Dim wbsobj, wsobj As Object`<br>`Set wbsobj = excelobj.Workbooks`<br>`wbsobj.Add`<br>`Set wsobj = excelobj.ActiveSheet` | `call send(excelobj,'_GET_PROPERTY_',`<br>`'Workbooks', wbsobj);`<br>`call send(wbsobj, '_DO_', 'Add');`<br>`call send(excelobj,'_GET_PROPERTY_',`<br>`'ActiveSheet', wsobj );` |
| セルの値を設定します<br>`wsobj.Cells(row + 1, col).Value`<br>`=var` | `r=row+1;`<br>`call send(wsobj,'_COMPUTE_', 'Cells', r, col,`<br>`retcell);`<br>`call send(retcell,'_SET_PROPERTY_',`<br>`'Value' ,var);` |
| Excel アプリケーションオブジェクトを閉じます<br>`excelobj.ActiveWorkbook.Close`<br>`("False")`<br>`excelobj.Quit` | `call send(excelobj,'_GET_PROPERTY_',`<br>`'ActiveWorkbook', awbobj);`<br>`call send(awbobj, '_DO_', 'Close', 'False');`<br>`call send(excelobj,'_DO_', 'Quit');`<br>`call send(excelobj,'_TERM_');` |

# SAS/AF アプリケーションで OLE カスタムコントロール (OCX)を使用する

## *概要:SAS/AF アプリケーションで OLE カスタムコントロール(OCX)を使用する*

OLE カスタムコントロールとは、OLE オブジェクト自体のプロパティとメソッドを公開するインターフェイスを持つ、特別な種類の OLE オブジェクトまたは OLE オブジェクトのコレクションです。これらのオブジェクトを制御するには、オブジェクトのグラフィカルインターフェイスと SCL コードを使用します。

OLE カスタムコントロールは、次の点で他の OLE オブジェクトと異なります。

- ユーザー操作に基づいて、FRAME エントリで応答できるイベントを生成します。OLE コントロールをアクティブにしても、オブジェクトの SCL ラベルはデフォルトでは実行されません。
- 使用する環境に基づき、色、フォントなどの環境に関するプロパティを推定します。

OLE コントロールは、OCX ファイル拡張子が付けられ、独自の動的リンクライブラリにまとめられます。SCL コードを使用すると、FRAME エントリによる、OLE コントロール(マウスクリックやキー入力など)が生成するイベントへの応答が可能になります。OLE コントロールが生成するイベントは、OLE コントロールによって異なります。詳細は、使用している OLE コントロールのマニュアルを参照してください。FRAME エントリにコントロールを挿入後、OLE コントロールオブジェクトの**オブジェクト属性**を選択し、次に**イベントマップ**を選択すると、イベントマップを表示できます。

*注:* SAS が提供する OLE コントロールには 32 ビットコンテナが必要で、そのため 16 ビットコンテナのみのサポートしか提供しない Windows アプリケーションでは使用できません。また、SAS は 32 ビットコンテナなので、16 ビットコントロールを使用することができません。

### OLE コントロールを FRAME エントリに挿入

FRAME エントリに OLE コントロールを挿入するには、次の操作を行います。

1. **COMPONENTS** ウィンドウから、**V6 オブジェクト**項目を選択してオブジェクトツリーを展開します。
2. **OLE - オブジェクトの挿入**を選択リストでダブルクリックします。**オブジェクトの 挿入**ダイアログボックスが表示されます。また、**OLE - オブジェクトの挿入** を選択リストから **BUILD** ウィンドウにドラッグすることもできます。マウスボタンを離すと、**オブジェクトの挿入**ダイアログボックスが表示されます。
3. **コントロールの作成**ラジオボタンを選択して、登録済み OLE カスタムコントロールのリストを表示します。使用する OLE コントロールがリストにはないがシステム上にある場合、OLE コントロールを登録する必要があります。詳細については、“OLE コントロールの登録” (251 ページ) を参照してください。
4. 挿入する OCX コントロール名を選択します。

### OLE コントロールの登録

Windows で OLE コントロールを使用するには、Windows にコントロールを登録する必要があります。SAS に用意されている 2 つの OLE コントロール(SAS コンボボックス、SAS エディット)は、SAS インストール時に自動的に登録されます。

SAS アプリケーションや他のアプリケーションで使用する他のコントロールをインストールするには、登録実行プロセスによってコントロールがインストールされている場合を除き、Windows にコントロールを登録する必要があります。登録されるまでは、OLE コントロールを**オブジェクトの挿入**ダイアログボックスで使用することはできません。

OLE コントロールを登録するには、次の操作を行います。

1. “OLE コントロールを FRAME エントリに挿入” (251 ページ) に記述されているステップ 1～4 を完了して、登録済みコントロールが表示された**オブジェクトの挿入**ダイアログボックスを開きます。
2. **コントロールの追加**をクリックして、参照ファイル選択ダイアログボックスを開きます。
3. ダイアログボックスで、登録するコントロール(通常、ファイル拡張子は OCX)を選択します。

   **OK** をクリックすると、コントロールが**オブジェクトの挿入**ダイアログボックスの登録済みコントロールのリストに追加されます。

### OLE コントロールのプロパティへのアクセス

#### 概要:OLE コントロールプロパティへのアクセス

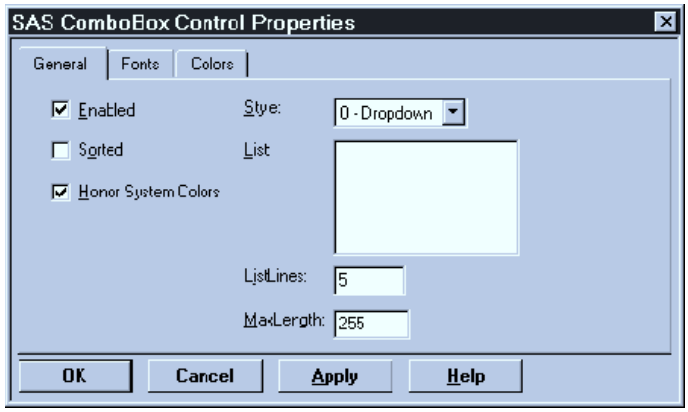
OLE コントロールのプロパティは、SCL メソッドを使用して設定または取得できます。OLE コントロールの中には、プロパティページからプロパティの一部にアクセスできるものがあります。プロパティページを使用すると、データを対話的に設定または取得できます。

### OLE コントロールのプロパティページへのアクセス

コントロールのプロパティページを開くには、**BUILD:DISPLAY** ウィンドウのコントロールの領域内を右クリックし、ポップアップメニューから**プロパティ**を選択します。コントロールのプロパティページが表示されます。図 9.4 (252 ページ) では、OLE コントロールのプロパティページの例を表示します。

OLE コントロールには、SCL の _EXECUTE メソッドを使用して、コントロールの**プロパティ**ページを表示できる Properties Verb が用意されています。または、OLE コントロールのポップアップメニューから、OLE コントロール名のカスケードメニューを選択することもできます。そのカスケードメニューから Properties Verb を利用できます。

***図 9.4*** *プロパティページの例*




プロパティページを使用して、公開されたプロパティの一部の設定を表示または変更できます。

DISPLAY モードでは、コントロールはアクティブではないため、コントロールのインターフェイスを操作することはできません。コントロールは TESTAF モードでアクティブになります。

### SCL コードを使用したプロパティへのアクセス

SAS/AF アプリケーションで OLE コントロールを使用すると、コントロールのプロパティへのアクセスをプログラムできます。しかし、OLE コントロールのプロパティには、プロパティページに公開されていない属性があります。このような場合、コントロールのプロパティにアクセスするには、_SET_PROPERTY_ メソッドと _GET_PROPERTY_ メソッドを使用します。

プロパティにアクセスするには、次の情報が必要です。

- SAS/AF FRAME エントリの OLE コントロールのオブジェクトラベル
- アクセスするプロパティの名前
- プロパティに保持されるデータの種類

たとえば、FRAME エントリにコンボボックスコントロール **sascombo** がある場合、次のように、リストのスタイルを **simple**(整数の 1)に設定できます。

```
call notify ('sascombo', '_set_property_',
             'Style', 1);
```

プロパティからデータを取得するには、読み込むデータと同じ種類の変数を使用する必要があります。たとえば、コンボボックスの編集フィールドにユーザーが指定したテキストを取得するには、次のように記述します。

```
length text $ 200;
call notify ('sascombo', '_get_property_',
```

```
'Text', text);
```

## SCL メソッドを使用した OLE コントロールの操作

OLE コントロールでは、その内容と動作を制御するメソッドを使用できます。OLE コントロールに対してメソッドを実行するように通知するメッセージを送信するには、_DO_ または_COMPUTE_の SCL メソッドを使用します。

- OLE コントロールメソッドが操作を実行して値を返さない場合は、SCL に_DO_メソッドを使用します。たとえば、SAS コンボボックスの OLE コントロールには、次のような、リスト内の項目をすべて消去するメソッドがあります。

  ```
  call notify('sascombo', '_DO_', 'Clear');
  ```

- OLE コントロールメソッドが値を返す場合は、SCL に_COMPUTE_メソッドを使用します。SCL コード内に、メソッドの終了時に戻り値を保存する変数を指定します。たとえば、SAS コンボボックスの OLE コントロールには、次のリストの特定の位置にあるアイテムを返すメソッドがあります。

  ```
  length item $ 80;
  call notify('sascombo', '_COMPUTE_',
              'GetItem', 2, item);
  ```

  この呼出が戻る場合、アイテムには位置 2(リスト上の 3 番目のアイテム)にあるアイテムのテキストを含みます。

## OLE コントロールイベントへの応答

### 概要:OLE コントロールイベントへのレスポンス

OLE コントロールは、SCL コードを使用して応答できるイベントを生成します。SAS/AF イベントの場合と同様に、OLE イベントに対して SCL コード内にラベルを作成できます。


- “OLE コントロールイベントへの SCL コードの割り当て” (253 ページ)
- “イベントから引数値を取り込む” (254 ページ)
- “例:SCL コードへの OLE コントロールイベントのマップ” (255 ページ)
- “例:OLE カスタムコントロールのサブクラス化” (255 ページ)
- “コンボボックスリストへのアイテムの追加” (256 ページ)
- “コンボボックスのアイテムの検索” (256 ページ)
- “コントロールのテキスト値の取り込み” (256 ページ)


### OLE コントロールイベントへの SCL コードの割り当て

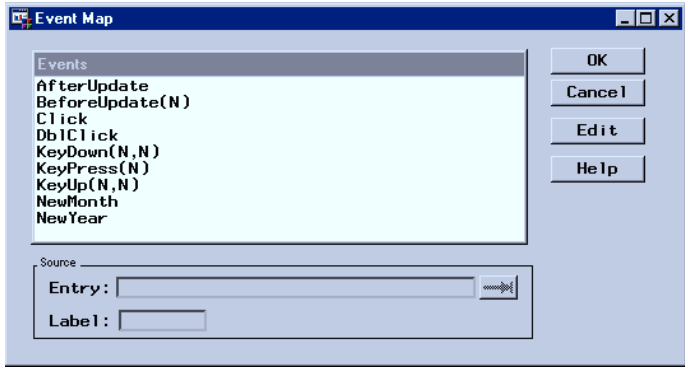
OLE コントロールイベントの発生時に実行する SCL コードを割り当てるには、次の操作を行います。

1. **BUILD** ウィンドウで OLE コントロールオブジェクトを選択します。
2. **オブジェクト属性**をオブジェクトのポップアップメニューから選択します。
3. **イベントマップ**を**オブジェクト属性**ダイアログボックスから選択します。次に示されるように、**イベント マップ** ダイアログボックスが表示されます。図 9.5 (254 ページ)を参照。

4. **イベント マップ**ダイアログボックスで、SCL コードを使用して応答するイベントを選択します。
5. イベント処理コードが格納された SCL、FRAME または PROGRAM のソースエントリと、必要に応じて SCL ラベルを指定します。

   *注:* FRAME エントリと共に保存されるものと同じ SCL ソースエントリを指定できます。ただし、FRAME エントリを使用してコードをコンパイルする以外に、イベントハンドラが SCL ラベルを認識するように、FRAME コンテキストの枠外 (BUILD:SOURCE ウィンドウと BUILD:DISPLAY ウィンドウの枠外)の SCL エントリもコンパイルする必要があります。したがって、FRAME エントリに割り当てられていない SCL ソースエントリに OLE コントロールのイベント処理コードを保存すると、効率が上がります。

***図 9.5*** *[イベントマップ]ダイアログボックス*




*注:* 大部分の OLE コントロールには LostFocus イベントが含まれます。このイベントは、コントロールがウィンドウのフォーカスを喪失する際に生成されます。SAS/AF では OLE コントロールと通信する方法であるため、LostFocus イベントをマップすると、フォーカス喪失直後のコントロールにフォーカスが戻る場合があります。FRAME エントリで LostFocus イベントに応答できますが、フォーカスの動作に異常が発生することがあります。

### *イベントから引数値を取り込む*

便利なパラメータが含まれる OLE コントロールのイベントもあります。たとえば、SAS コンボボックスコントロールは、押されたキーの ASCII 値を通知する KeyPress イベントを生成します。特定のイベントが引数を FRAME エントリに返す場合は、戻り値の種類が**イベントマップ**ダイアログボックスに表示されます。数値は **N** で表され、文字値は **C** で表されます。

OLE コントロールのイベントの戻り値を取り込むには、SCL で METHOD ステートメントを使用して、イベント処理コードにメソッドを定義する必要があります。METHOD ステートメントの引数リストで、OLE コントロールが返すと予想される種類の変数を指定します。この変数には、イベントの戻り値が含まれます。これで、必要なときにその変数をイベントハンドラ内で使用できます。

たとえば、SAS コンボボックスコントロールで KeyPress イベントをトリガとしたキーの値を取り込み、キーを ASCII 文字として通知するとします。KeyPress イベントは、押されたキーの ASCII 値を表す整数を返します。イベント処理コードは次のようになります。

```
   /* Label specified in Event Map dialog box */
KEYPRESS:
   /* Define a method with an
      integer argument */
method keyval 8;
```

```
      /* Convert the integer to an
         ASCII character */
   keychar=byte(keyval);
  put keychar=; /* Output the character */
endmethod;
```

### 例:SCL コードへの OLE コントロールイベントのマップ

次のいずれかの方法で、OLE コントロールイベントをマップできます。

- **イベント マップ**ウィンドウのイベントごとに SCL コードの各ラベルセクションにマップします。各コードはそれぞれ異なる操作を実行します。
- **イベント マップ**ウィンドウのすべてのイベントを SCL コードの 1 つのラベルセクションにマップします。_GET_EVENT メソッドを使用して、トリガされたイベントを検出し、検出されたイベントを処理します。
- これらの方法を組み合わせて使用します。1 つのイベント(クリックイベントなど)を、_OBJECT_LABEL_メソッドを使用してオブジェクトのラベルを実行する SCL コードに割り当てます。残りのイベントを、_GET_EVENT_メソッドを使用してイベントと適切な操作を確定する 1 つのラベルにマップします。OLE コントロールの場合、オブジェクトのラベルは、デフォルトでは実行されません。

次に、OLE コントロールのすべてのイベントが 1 つのラベルにマップされる場合に、SCL コードがどのように構成されるかを示します。この SCL コードは、順番にオブジェクトのラベルを実行し、イベントを特定し、処理します。

```
length event $ 80;
  /* All OLE control events are mapped to
     this label */
RUNLABEL:
     /* Call the object's label */
  call send(_self_, '_OBJECT_LABEL_');
return;
  /* This is the label of the OLE control */
OBJ1:
     /* Determine the last event */
  call notify('obj1', '_GET_EVENT_', event);
  select (event);
     when('Click') put 'Click received';
     when('DblClick') put 'DblClick received';
     otherwise put event=;
  end;
return;
```

### 例:OLE カスタムコントロールのサブクラス化

1 つ以上の OLE カスタムコントロールを頻繁に使用する SAS/AF アプリケーションを作成する場合、独自のメソッドを記述することで、SCL コード中に_DO_メソッドおよび_COMPUTE_メソッドを指定することなくコントロールで認識されるメソッドを抽出できます。

そのためには、OLE クラスのサブクラスを作成し、派生クラスにメソッドを追加します。OLE コントロールを FRAME エントリに挿入するには、定義する新しいクラスのインスタンスとして OLE コントロールを挿入します(**OLE - オブジェクトの挿入**)次の例には、コントロールのメソッドを抽出できるサンプルコードが含まれています。サブクラスの作成方法の詳細は、含まれていません。SAS/AF クラスのサブクラスの作成については、SAS/AF のオンラインドキュメントを参照してください。

### コンボボックスリストへのアイテムの追加

このメソッドを使用すると、SAS コンボボックスコントロールのリストに新しいアイテムを追加できます。SAS コンボボックスコントロールでは、0 から採番される番号を使用して、リスト項目の位置を示します。つまり、最初の項目が位置 0、2 番目の項目が位置 1、となります。 次のメソッドを使用すると、位置 1 が最初の項目であることを示すように位置番号を指定できます。

```
/* Add a new item to a ComboBox list.  */
ADDITEM:
method text $200 row 8 rc 8;
     /* adjust for zero-based index */
  ocxrow = row-1;
  call send(_self_, '_COMPUTE_', 'AddItem',
            text, ocxrow, rc);
  if ( rc = 0 ) then
    _MSG_="ERROR: Could not add item to list.";
endmethod;
```

このコードを、ADD_ITEM と呼ばれる新しいメソッドにマップすると、この構文を使用して、コントロールに新しいアイテムを追加できます。

```
   /* Adds 'Item 1' at the first position */
   /* in the control                      */
length success 8;
call notify('sascombo', 'ADD_ITEM',
            'Item 1', 1, success);
```

### コンボボックスのアイテムの検索

次のメソッドは、指定したアイテムを検索して、リスト内の位置を返します。前の例と同様に、このメソッドは位置番号を 0 からではなく 1 から始まるように調整します。

```
FINDITEM:
method text $200 row 8;
  call send(_self_, '_COMPUTE_', 'FindItem',
            text, row);
  row = row + 1; /* adjust for zero-based */
endmethod;       /* index                */
```

このコードを FIND_ITEM メソッドにマップすると、次の例のように使用できます。

```
length position 8;
call notify('sascombo','FIND_ITEM',
            'Lost Item', position);
```

### コントロールのテキスト値の取り込み

SAS コンボボックスコントロールおよび SAS エディットコントロールの両方には、テキストプロパティがあります。テキストプロパティにアクセスするには、_GET_PROPERTY_メソッドとプロパティ名を使用します。OLE サブクラスからより簡単に、より直感的にアクセスするには、_GET_TEXT_メソッドを無効にして、このコードにマップします。

```
GETTEXT:
method text $200;
   call send(_self_, '_GET_PROPERTY_',
             'Text', text);
endmethod;
```

SAS/AF の他の Widget オブジェクトのテキストへのアクセス時と同じ方法を使用して、コントロールのテキストプロパティにアクセスします。

```
length text $ 200;
call notify('sasedit', '_GET_TEXT_', text);
```

*10 章*

# OLE を使用した他のアプリケーションからの SAS 制御

**SAS の自動化について** . . . . . **259**

**SAS のインスタンスの作成** . . . . . **260**

**SAS セッションからのフィードバックの取得** . . . . . **260**

**OLE を用いた SAS の自動化の例** . . . . . **261**
Visual Basic の使用例 . . . . . 261
SAS オートメーションオブジェクトの作成 . . . . . 261
SAS セッションがビジーかどうかを特定 . . . . . 261
SAS セッションの表示/非表示の切り替え . . . . . 262
SAS セッションのメイン SAS ウィンドウタイトルの設定 . . . . . 262
SAS ライブラリの割り当てと SAS プロシジャの実行 . . . . . 262
SAS セッションの終了 . . . . . 262

**SAS OLE オートメーションオブジェクトで使用するメソッドとプロパティ** . . . . . **262**

**ディクショナリ** . . . . . **263**
コマンドメソッド . . . . . 263
QueryWindow メソッド . . . . . 263
Quit メソッド . . . . . 264
Submit メソッド . . . . . 264
Top メソッド . . . . . 264
SAS オートメーションオブジェクトを制御するためのプロパティ . . . . . 265

## SAS の自動化について

SAS を、OLE オートメーションサーバーとして実行することができます。つまり、Visual Basic などのように OLE オートメーションコントローラとして機能するアプリケーションを使用して、SAS セッションを作成することができます。さらに、このセッションの制御に、SAS 提供のメソッドやプロパティを使用することができます。

Windows アプリケーションの多くでは、オートメーションのスクリプト言語として、Visual Basic または Visual Basic for Applications が使用されます。このドキュメントの例ではすべて Visual Basic を使用していますが、OLE オートメーションコントローラとして機能するアプリケーションであればどのアプリケーションを使用しても、同じ結果を得ることができます。

## SAS のインスタンスの作成

SAS のインスタンスを作成する(SAS セッションを起動する)には、Windows レジストリに登録されているプログラム識別子のような SAS プログラム識別子を使用して OLE オブジェクトを作成する必要があります。SAS プログラム識別子は、**SAS.Application** です。SAS セッションをインスタンス化(インスタンスの作成)する Visual Basic の一例です。

```
Dim OleSAS as Object
Set OleSAS = CreateObject("SAS.Application")
```

この例では識別子 **OleSAS** を新たな SAS セッションに置きます。この識別子を使用すると、SAS で利用可能なメソッドやプロパティにアクセスできます。

OLE オートメーションを使用して、既存の SAS オートメーションオブジェクトを制御するには、オートメーション制御言語を使用します。Visual Basic では、次のものが使用できます。

```
Dim OleSAS as Object
Set OleSAS = GetObject(,"SAS.Automation")
```

このコードで SAS のインスタンスが作成されるのは、インスタンスがすでに存在している場合であることに注意してください。また、既存の SAS セッションが、OLE オートメーションオブジェクトとして作成されている必要があります。たとえば、Visual Basic で **CreateObject** を使用すると作成できます。[**スタート**]メニューを使用するなどの他の方法を使用して起動した SAS セッションを制御する場合、OLE オートメーションは使用できません。

## SAS セッションからのフィードバックの取得

SAS には、RC と ResultString の 2 つのプロパティが用意されています。このため、自動化している SAS セッションからの情報を、SAS セッションを制御しているアプリケーションに渡すことができます。RC プロパティには数字が含まれます。ResultString プロパティには文字列が含まれます。

SAS セッション内からの、これらのプロパティの値を設定するには、次の構文を用いて SETRC 機能を使用します。

```
error=SETRC("result-string", rc-number);
```

where *result-string* は ResultString プロパティに、*rc-number* は RC プロパティにそれぞれ割り当てられる値です。

たとえば、Submit メソッドを使用して、DATA ステップコードをサブミットします。このとき、処理の一部としてエラーコードが返されます。ここで、RC プロパティまたは ResultString プロパティを使用すると、そのエラーの値を確認できます。次に、Visual Basic の例を示します。

```
Private Delcare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Using Visual Studio 2010 vb.net, I had to change this to:
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim OleSAS As Object
        Dim x = 0
```

```
    Dim Response
    OleSAS = CreateObject("SAS.Application")
    OleSAS.Submit("data _null_;error=setrc('Error string', 2.0);put error; run;")
    Sleep(500)
    Do While x = 0
        If (OleSAS.Busy) Then
            Sleep(500)
        Else
            If (OleSAS.RC <> 0) Then Response = MsgBox(OleSAS.ResultString,
            vbOKOnly, "Error Message is")
            x = 1
        End If
    Loop
End Sub
```

# OLE を用いた SAS の自動化の例

## *Visual Basic の使用例*

次の例では、OLE オートメーションを使用して SAS を制御するために、スクリプト言語として Visual Basic を使用しています。OLE オートメーションコントローラとして機能する Windows アプリケーションであれば、任意のスクリプト言語を使用できます。


- “SAS オートメーションオブジェクトの作成” (261 ページ)
- “SAS セッションがビジーかどうかを特定 ” (261 ページ)
- “SAS セッションの表示/非表示の切り替え” (262 ページ)
- “SAS セッションのメイン SAS ウィンドウタイトルの設定” (262 ページ)
- “SAS ライブラリの割り当てと SAS プロシジャの実行” (262 ページ)
- “SAS セッションの終了” (262 ページ)


## *SAS オートメーションオブジェクトの作成*

次の Visual Basic コードは、オブジェクトを定義し、そのオブジェクトに関連付ける SAS のインスタンスを作成します。

```
Dim OleSAS As Object
Set OleSAS = CreateObject("SAS.Application")
```

## *SAS セッションがビジーかどうかを特定*

次の Visual Basic コードは、Busy プロパティを使用して SAS セッションに照会し、セッションがビジー処理になっているかどうかをテストします。

```
If (OleSAS.Busy) Then
   Response = MsgBox("SAS Session is Busy",
              vbOKOnly, "SAS Session")
Else
   Response = MsgBox("SAS Session is Idle",
              vbOKOnly, "SAS Session")
```

```
End If
```

### SAS セッションの表示/非表示の切り替え

次の Visual Basic コードは、SAS セッションを、現在の状態に基づいて非表示にしたり表示したりします。

```
If OleSAS.Visible = False then
    OleSAS.Visible=True
Else
    OleSAS.Visible=False
End If
```

### SAS セッションのメイン SAS ウィンドウタイトルの設定

次の Visual Basic コードは、SAS セッションのメイン SAS ウィンドウにタイトルを割り当て、タイトルをメッセージボックスに表示します。

```
OleSAS.Title = "Automation Server"
Response = MsgBox(OleSAS.Title, vbOKOnly,
           "Title Is", 0, 0)
```

### SAS ライブラリの割り当てと SAS プロシジャの実行

次の Visual Basic コードは、SAS コードを SAS セッションにサブミットし、SAS ライブラリを割り当て、サンプルデータに対して INSIGHT プロシジャを実行します。

```
OleSAS.Submit("libname insamp
                'c:\sas\insight\sample';
               proc insight data=insamp.drug;
               run;")
```

### SAS セッションの終了

次の Visual Basic コードは、SAS セッションを使用する OLE オートメーションコントローラが他にない場合に、その SAS セッションを終了します。

```
OleSAS.Quit
Set OleSAS = Nothing
```

## SAS OLE オートメーションオブジェクトで使用するメソッドとプロパティ

インスタンスを作成すると、SAS OLE オートメーションオブジェクトにより、次のメソッドとプロパティがサポートされます。

# ディクショナリ

## コマンドメソッド

SAS コマンドラインから入力されたときと同じようにコマンドを呼び出します。

### 構文

Command("sas-command")

### 詳細

デフォルトでは、アクティブウィンドウでコマンドを受け取ります。コマンドを受け取るウィンドウを変更するには、CommandWindow プロパティを変更します。

### 例

次の Visual Basic コードは、SAS セッションを起動し、**BUILD** ウィンドウを開きます。

```
Set OleSAS = CreateObject("SAS.Application") OleSAS.Command("build")
```

## QueryWindow メソッド

指定されたウィンドウが SAS セッション内に存在するかどうかを照会します。

### 構文

QueryWindow("window-name")

### 詳細

QueryWindow は **True** または **False** を、指定されたウィンドウが自動化された SAS セッションで開かれているかどうかに基づいて返します。存在するウィンドウが非表示の場合でも、QueryWindow は **True** を返します。

指定するウィンドウ名は、SAS のウィンドウ名と一致している必要があります。window-name 引数では、大文字と小文字は区別されません。

### 例

次の Visual Basic コードは、既存の SAS セッションを取得し、**BUILD** ウィンドウが開かれているかどうかをチェックします。ウィンドウが開いていない場合は、このコードによって BUILD が呼び出されます。

```
Dim OleSAS as Object Set OleSAS = GetObject(,"SAS.Application") If (Not OleSAS.QueryWind
```

## Quit メソッド

SAS セッションを終了する

### 構文

Quit

### 詳細

Quit メソッドを発行するオートメーションコントローラが、特定の SAS セッションを使用している唯一のコントローラである場合、その SAS セッションは終了します。SAS セッションを使用しているオートメーションプロセスが他に少なくとも 1 つ存在する場合、セッションは動作を続けます。

## Submit メソッド

DATA ステップコードまたは PROC ステップコードをサブミットし、処理します。

### 構文

Submit("SAS-program-code")

### 詳細

*SAS-program-code* として指定する文字列には、セミコロンで区切って複数の SAS ステートメントを指定できます。文字列の内容は SAS にサブミットされ、処理されます。

### 例

次の例では、データライブラリを参照し、SAS/AF アプリケーションを起動します。

```
Dim OleSAS as Object Set OleSAS = CreateObject("SAS.Application") OleSAS.Visible = True
```

## Top メソッド

SAS セッションを手前に表示します。

### 構文

Top

### 詳細

Top メソッドは、Visible プロパティが True に設定されている場合にのみ影響します。

### 例

次の例では、SAS セッションを起動し、起動した SAS セッションのウィンドウを表示し、このウィンドウを手前に配置します。

```
Dim OleSAS as Object Set OleSAS = CreateObject("SAS.Application") OleSAS.Visible = True
```

## SAS オートメーションオブジェクトを制御するためのプロパティ

SAS オートメーションオブジェクトのさまざまなプロパティを指定します。

### 詳細

#### *プロパティと説明*

次のプロパティは SAS オートメーションオブジェクトに適用されます。

Busy
SAS が、アイドル中または作動中(プロシジャ、DATA ステップなどの実行中)のどちらであるかを示します。このプロパティは読み取り専用です。

CommandWindow
ウィンドウのタイトルに基づきウィンドウを設定し、Command メソッドを使用して指定したコマンドを受け取ります。指定する名前は、ウィンドウ名のスペルと正確に一致している必要があります。ただし、このプロパティでは、大文字と小文字は区別されません。名前が設定されると、設定したウィンドウは、それ以降に Command メソッドで指定したコマンドを受け取ります。この状態は、CommandWindow を変更するか、Null に設定(""と指定)するまで変わりません。名前が Null (デフォルト)の場合は、現在のアクティブウィンドウがコマンドを受け取ります。このプロパティは、読み書きできます。

CommandWindowVisible
CommandWindow プロパティで指定したウィンドウを表示するかどうかを制御します。プロパティを **`False`** に設定すると、CommandWindow プロパティで指定したウィンドウが非表示に設定されます。CommandWindow プロパティが Null の場合、このプロパティは無効です。このプロパティは、読み書きできます。

ConfirmExit
SAS 終了時の動作を制御します。値が `0` の場合は、SAS の終了前に確認ボックスは表示されません。値が `1` の場合は、SAS の終了前に確認ボックスが表示されます。値が `2` の場合は、デフォルトの操作が適用されます。デフォルトの操作では、SAS の終了方法(**プリファレンス**ダイアログボックスなど)を定義する別のメソッドにより制御されます。

Height
SAS アプリケーションウィンドウの高さを、ピクセル単位で指定します。このプロパティは、読み書きできます。

Parent
SAS アプリケーションウィンドウを表示する親ウィンドウの名前を指定します。この値を別のウィンドウのものに変更すると、SAS アプリケーションウィンドウは、新しいフレームに適合するようにサイズ変更されます。このプロパティは、読み書きできます。

RC
ユーザー関数により渡されるリターンコードを返します。このプロパティを SAS セッション内から設定するには、SETRC 関数を使用します。このプロパティは、オートメーションコントローラからの読み取り専用です。

ResultString
ユーザー関数により渡される文字列を返します。このプロパティを SAS セッション内から設定するには、SETRC 関数を使用します。このプロパティは、オートメーションコントローラからの読み取り専用です。

Title
メイン SAS ウィンドウのタイトルを指定します。このプロパティは、読み書きできます。

Visible
SAS を表示するかどうかを制御します。このプロパティは、読み書きできます。

Width
SAS アプリケーションウィンドウの幅を、ピクセル単位で指定します。このプロパティは、読み書きできます。

X
SAS アプリケーションウィンドウの左上の横の位置をピクセル単位で指定します。このプロパティは、読み書きできます。

Y
SAS アプリケーションウィンドウの左上の縦の位置をピクセル単位で指定します。このプロパティは、読み書きできます。

*11 章*

# Windows 版動的データ交換の使用

**動的データ交換(DDE)の概要** . . . . . **267**

**SAS での DDE 構文** . . . . . **268**

**DDE 外部ファイルの参照** . . . . . **269**
DDE 外部ファイルの参照の概要 . . . . . 269
DDE トリプレットの使用 . . . . . 269
DDE を使用して他のアプリケーションを制御する . . . . . 270

**DDE の例** . . . . . **270**
DDE 例の概要 . . . . . 270
X コマンドを使用して DDE サーバーを開く . . . . . 270
DDE を使用して、Microsoft Excel にデータを書き込む . . . . . 271
DDE を使用して、Microsoft Word にデータを書き込む . . . . . 271
DDE を使用して、Microsoft Excel からデータを読み込む . . . . . 271
DDE を使用して、Microsoft Word からデータを読み込む . . . . . 272
DDE と SYSTEM トピックを用いて、Excel を使用するアプリケーションのコマンドを起動する . . . . . 272
DDE で NOTAB オプションを使用する . . . . . 273
DDE HOTLINK の使用 . . . . . 274
!DDE_FLUSH 文字列を使用してデータを動的に転送する . . . . . 275
マクロ変数を使用した DDE コマンドの発行 . . . . . 275
欠損データの読み取り . . . . . 276

## 動的データ交換(DDE)の概要

動的データ交換(DDE)は、Windows アプリケーション間で情報を動的に交換する方法です。DDE は、クライアント/サーバーの関係を利用して、クライアントアプリケーションがサーバーアプリケーションに情報を要求できるようにします。SAS は常にクライアントです。この役割において、SAS はサーバーアプリケーションに情報を要求し、サーバーアプリケーションにデータやコマンドを送信します。

DDE は、DATA ステップ、SAS マクロ機能、SAS/AF アプリケーションなど、データを要求、生成する SAS のあらゆる機能で使用できます。DDE にはさまざまな用途があります。たとえば、Windows のスプレッドシートアプリケーションやデータベースアプリケーションからのデータ取得に使用できます。

次の URL にアクセスすると、動的データ交換(DDE)によってサポートされるアプリケーションにアクセスできます。Word および Excel アプリケーションがサポートされています。PowerPoint アプリケーションはサポートされていません。

http://support.microsoft.com/kb/142821

*注:* 現在は SAS を含む多くの Windows プログラムで、アプリケーション間の通信を容易にするため OLE がサポートされています。OLE をサポートするアプリケーションとデータを共有する場合は、SAS に組み込まれた OLE サポートを使用してください。詳細については、“OLE について” (234 ページ) を参照してください。

## SAS での DDE 構文

SAS で DDE を使用するには、次の構文の FILENAME ステートメントを発行します。

**FILENAME** *fileref* DDE '*DDE-triplet*' <DDE-options>;

説明

*fileref*
有効な fileref です (“外部ファイルの参照” (148 ページ) に詳細があります)。

DDE
動的データ交換の使用を指定するデバイスタイプキーワードです。

'*DDE-triplet*'
DDE 外部ファイルの名前です。

*DDE-options*
次のいずれかを指定できます。

COMMAND
DDE サーバーアプリケーションに、リモートコマンドを発行することができます。詳細については、“DDE を使用して他のアプリケーションを制御する” (270 ページ) を参照してください。

HOTLINK
SAS に DDE HOTLINK の使用を指示します。このオプションの使用例については、“DDE HOTLINK の使用” (274 ページ) を参照してください。

LRECL=*record-length*
レコード長をバイト単位で指定します。Windows ではデフォルト値は 32767 です。*record-length* の値は、1 から 1,073,741,823 (1 GB)の範囲から指定できます。

NOTAB
SAS に、変数間のタブ文字を無視するよう指示します。このオプションの使用例については、“DDE で NOTAB オプションを使用する” (273 ページ) を参照してください。

RECFM=*record-format*
レコードの形式を制御します。Windows 版では、次の値が有効です。

F 固定長形式を指定します。

N バイナリ形式を指定し、ファイルはバイトストリームとして処理されます。LRECL を指定しない場合、デフォルト設定では、一度に 32767 バイトがファイルから読み込まれます。

P 印刷形式を指定します。

V|D 可変長形式を指定します。これはデフォルトの設定です。

***注意:***

**DDE で扱うデータ値がブランクになっている場合や欠損している場合は注意してください。** サンプルコードについては “欠損データの読み取り” (276 ページ) を参照してください。

# DDE 外部ファイルの参照

## *DDE 外部ファイルの参照の概要*

DDE で使用するファイル参照名を定義すると、*DDE トリプレット*引数はその DDE 外部ファイルを参照します。

## *DDE トリプレットの使用*

DDE トリプレットはアプリケーションに依存するため、実行するアプリケーションによって異なります。アプリケーションの DDE トリプレットの詳細は、アプリケーションのドキュメントを参照してください。

トリプレットの形式は、次のとおりです。

'*application-name*|*topic*!*item*'

説明

*application-name*
サーバーアプリケーションの実行可能なファイル名。たとえば、Microsoft Word の *アプリケーション名*は、**winword** です。Microsoft Excel の場合は、**excel** になります。

*topic*
対話のトピック(SAS と DDE サーバーアプリケーション間)。通常は、データを共有するドキュメントまたはスプレッドシートのフルパスのファイル名になります。

*item*
クライアントとサーバーアプリケーション間で指定した対話範囲。スプレッドシートアプリケーションでは、通常、セルの範囲になります。Microsoft Word などのドキュメントベースのアプリケーションでは、item はブックマークなど、ドキュメント内の場所を定義するものになります。

これらのすべての引数において、有効な値はサーバーアプリケーションによって異なります。DDE をサーバーとしてサポートするソフトウェアパッケージでは、アプリケーションの付属マニュアルのトリプレット情報に関するトピックに利用可能な値が記載されています。

*注:* DDE を使用して通信する前に、サーバーアプリケーションを起動する必要があります。また、DDE トリプレットの形式は、アプリケーションや、同じアプリケーションでもバージョンによって異なる場合があります。

たとえば、C:\TEMP にある Microsoft Word ドキュメント TESTDDE.DOC (ブックマーク名 NUMBER)にテキストを追加するには、次のコードを使用します。

```
filename test dde 'winword|"c:\temp\testdde.doc"
                  !NUMBER' notab;
```

アプリケーション名 は **winword** で、トピックは**"c:\temp\testdde.doc"**、範囲は**!NUMBER** です。

たとえば、SAS を使用して **C:\EXCEL\SALES.XLS** に保存されている Sales Data という名前の Microsoft Excel スプレッドシートに、最初の 4 行と 2 列を作成するとします。次のコードを使用します。

```
filename test dde 'Excel|c:\excel\
                   [Sales.xls]Sales Data!R1C1:R4C2'
```

アプリケーション名は **Excel**、トピックは **c:\excel\[Sales.xls] Sales Data**、範囲は **R1C1:R4C2** です。

サーバーアプリケーションで、DDE トリプレットを Windows クリップボードにコピーできる場合、SAS に DDE トリプレットを表示できます。そのためには、サーバーアプリケーションで情報を選択して、Windows クリップボードにコピーします。SAS に戻り、**ソリューション** ⇨ **アクセサリー** ⇨ **DDE トリプレット**を選択します。

### *DDE を使用して他のアプリケーションを制御する*

DDE サーバーアプリケーションでは、DDE リンクを使用してアプリケーションを制御し、実行する特定のコマンドをサポートします。このようなコマンドを使用するには、DDE トリプレット内で特別な topic 名 SYSTEM を使用し、item 名をブランクにします。次に、アプリケーションから入力を行う場合は INPUT ステートメントを、サーバーアプリケーションにコマンドを発行する場合は PUT ステートメントを使用します。

topic 名 SYSTEM を認識しない DDE サーバーアプリケーションの場合、DDE リンクを定義する FILENAME ステートメントに COMMAND オプションを指定できます。COMMAND オプションを指定する場合は、DDE トリプレット内で item 名を指定しないでください。

*注:* SAS/AF と OLE オートメーションを使用すると、サーバーとして OLE 2.0 をサポートする Windows アプリケーションを自動化できます。SAS と OLE の使用に関する詳細については、"OLE オブジェクトとアプリケーションの自動化" (243 ページ)

## DDE の例

### *DDE 例の概要*

このセクションでは、Windows 版での動的データ交換(DDE)の使用例を示します。これらの例では、DDE サーバーとして Microsoft Excel と Microsoft Word を使用していますが、サーバーとして DDE が使用できるアプリケーションであれば、どのアプリケーションでも SAS と通信できます。

これらの例を実行するには、最初に Microsoft Excel および Microsoft Word を起動してから、例で使用されているスプレッドシートまたはドキュメントを開く必要があります。

*注:* DDE の例は、ホスト固有のサンプルプログラムに含まれています。これらのプログラムには、**ヘルプ**メニューからアクセスできます。

### *X コマンドを使用して DDE サーバーを開く*

SAS コード内で X コマンドを使用して、DDE サーバーアプリケーションを起動できます。XWAIT オプションと XSYNC オプションを無効にする必要があります。

```
options noxwait noxsync;
x '"C:\program files\microsoft office\office14\excel.exe"';
```

パスにスペースが含まれる場合は、パスを二重引用符で囲む必要があります。一重引用符は、X コマンド用です。

### DDE を使用して、Microsoft Excel にデータを書き込む

最初の例では、SAS セッションから Excel スプレッドシートにデータが送信されます。送信先は、行 1 から 100 と列 1 から 3 のセルです。データを送信するには、次のプログラムをサブミットします。

```
/* The DDE link is established using   */
/* Microsoft Excel SHEET1, rows 1      */
/* through 100 and columns 1 through 3 */
filename random dde
   'excel|sheet1!r1c1:r100c3';
data random;
   file random;
   do i=1 to 100;
      x=ranuni(i);
      y=10+x;
      z=x-10;
      put x y z;
   end;
run;
```

### DDE を使用して、Microsoft Word にデータを書き込む

この例では、Microsoft Word ドキュメント内の指定したブックマークに文字列が送信されます。DDE を Microsoft Word に使用する場合と、Microsoft Excel に使用する場合との違いに注意してください。

```
filename testit dde 'winword|"c:\temp\testing.doc"
                     !MARK' notab;

data _null_;
   file testit;
   put 'This is a test.';
run;
```

*注:* Microsoft Word に書き出す場合は、FileOpen.Name、FileSave、FileClose、Insert などの Visual Basic コマンドを使用してください。PUT ステートメントの中に、Word が認識しないマクロが含まれている場合は、次のメッセージが表示されます。

```
Ambiguous name detected:TmpDDE
```

### DDE を使用して、Microsoft Excel からデータを読み込む

次の例のように、DDE を使用すると、Excel アプリケーションから SAS にデータが取り込まれます。

```
/* The DDE link is established using   */
/* Microsoft Excel SHEET1, rows 1      */
/* through 10 and columns 1 through 3  */
filename monthly
  dde 'excel|sheet1!r1c1:r10c3';
data monthly;
   infile monthly;
```

```
   input var1 var2 var3;
run;
proc print;
run;
```

### DDE を使用して、Microsoft Word からデータを読み込む

この例では、Microsoft Word ドキュメント内の指定したブックマークからデータが取り込まれます。

```
filename testit dde 'winword|"c:\temp\testing.doc"
                     !MARK' notab;

libname workdir 'c:\temp';

/* Get ready to read the first bookmark.  */

data workdir.worddata;
   length wordnum $5;
   infile testit;
   input wordnum $;
run;

proc print;
run;
```

### DDE と SYSTEM トピックを用いて、Excel を使用するアプリケーションのコマンドを起動する

DDE を使用して、SAS から Excel または他の DDE 互換プログラムに直接コマンドを発行することができます。次の例では、X コマンドを使用して Excel アプリケーションが起動され、SHEET1 という名前のスプレッドシートがロードされます。データが、SAS から Excel の行 1 列 1 から行 20 列 3 のセルに送信されます。データ範囲の選択とデータの並べ替えに必要なコマンドが発行されます。最後に、スプレッドシートが保存され、Excel アプリケーションが終了します。

```
/* This code assumes that Excel       */
/* is installed on the current        */
/* drive in a directory called EXCEL. */

options noxwait noxsync;
x '"c:\program files\microsoft office\office14\excel.exe"';

/* Sleep for 60 seconds to give */
/* Excel time to start up.       */

data _null_;
   x=sleep(60);
run;

/* The DDE link is established using   */
/* Microsoft Excel SHEET1, rows 1      */
/* through 20 and columns 1 through 3  */

filename data
```

```
   dde 'excel|sheet1!r1c1:r20c3';
data one;
   file data;
   do i=1 to 20;
      x=ranuni(i);
      y=x+10;
      z=x/2;
      put x y z;
   end;
run;

/* Microsoft defines the DDE topic */
/* SYSTEM to enable commands to be */
/* invoked within Excel.           */

filename cmds dde 'excel|system';

/* These PUT statements are       */
/* executing Excel macro commands */

data _null_;
   file cmds;
   put '[SELECT("R1C1:R20C3")]';
   put '[SORT(1,"R1C1",1)]';
   put '[SAVE()]';
   put '[QUIT()]';
run;
```

### DDE で NOTAB オプションを使用する

SAS では、DDE リンクを介して受信する変数はタブ文字で区切られていると想定されます。同様に、SAS からリンクを介してデータを送信する際、タブ文字を使用し変数を区切ります。DDE デバイスタイプキーワードを使用する FILENAME ステートメントに NOTAB オプションを指定すると、SAS では、変数の区切り文字としてタブ文字以外も受け入れられます。

NOTAB オプションは、ブランクを含む文字列全体を 1 つのセルに格納する場合にも使用されます。たとえば、SAS と Excel アプリケーションの間にリンクが確立され、SAS 変数にブランクを含む文字列が存在する場合、通常は文字列内の各単語ごとに 1 つのセルに格納されます。ブランクを含む文字列全体を 1 つのセルに格納するには、NOTAB オプションを次の例のように使用します。

```
/* Without the NOTAB option, column1  */
/* contains 'test' and column2        */
/* contains 'one'.                    */

filename test
  dde 'excel|sheet1!r1c1:r1c2';
data string;
   file test;
   a='test one';
   b='test two';
   put a $15. b $15.;
run;

/* You can use the NOTAB option to store   */
```

```
/* each variable in a separate cell. To    */
/* do this, you must force a tab           */
/* ('09'x) between each variable, as in    */
/* the PUT statement.                      */
/* After performing this DATA step, column1*/
/* contains 'test one' and column2         */
/* contains 'test two'.                    */

filename test
  dde 'excel|sheet1!r2c1:r2c2' notab;
data string;
   file test;
   a='test one';
   b='test two';
   put a $15. '09'x b $15.;
run;
```

### DDE HOTLINK の使用

HOTLINK オプションを指定すると、スプレッドシートの指定範囲にあるデータが更新されるたびに、DDE リンクがアクティブになります。また、DDE では、HOTLINK オプションを指定すると、データをポーリングして、指定範囲にあるデータが変更されたかどうかを確認することができます。データが変更されていない場合、HOTLINK オプションは 0 バイトのレコードを返します。次の例では、スプレッドシート SHEET1 の行 1 列 1 のセルに、日別の総生産量が含まれます。このセルの値が変更されるたびに、新しい値が読み込まれ、オブザベーションがデータセットに出力されます。この例では、2 番目のセル(行 5 列 1 のセル)がステータスフィールドとして定義されます。ユーザーがデータ入力を完了し、このフィールドに任意の文字を入力すると、DDE リンクが終了します。

```
/* Enter data into Excel SHEET1 in   */
/* row 1 column 1. When you          */
/* are through entering data, place  */
/* any character in row 5            */
/* column 1, and the DDE link is     */
/* terminated.                       */

filename daily
  dde 'excel|sheet1!r1c1' hotlink;
filename status
  dde 'excel|sheet1!r5c1' hotlink;
data daily;
   infile status length=flag;
   input @;
   if flag ne 0 then stop;
   infile daily length=b;
   input @;

   /* If data have changed, then the */
   /* incoming record length         */
   /* is not equal to 0.             */

   if b ne 0 then
      do;
         input total $;
```

```
        put total=;
        output;
      end;
run;
```

複数の DDE セッションを作成することもできます。前の例では、2 つの独立した DDE リンクを使用しています。HOTLINK オプションが使用され、item で複数のセルが参照されている場合、そのいずれかのセルが変更されると、すべてのセルが送信されます。

HOTLINK オプションが指定されていない場合、DDE は単一のワンタイムデータ転送として実行されます。DDE が処理されたときにスプレッドシートのセルに格納されている値が転送されます。

### !DDE_FLUSH 文字列を使用してデータを動的に転送する

DDE を使用すると、DDE リンクの動作中に DDE バッファをダンプするタイミングをプログラムすることもできます。通常、DDE バッファ内のデータが転送されるのは、DATA ステップの最後で DDE リンクが終了したときです。ただし、PUT ステートメントで特別な文字列**'!DDE_FLUSH'**が発行されると、SAS では DDE バッファの内容がダンプされます。この関数を使用すると、DATA ステップからデータを動的に転送できるなど、非常に柔軟性のある方法で DDE を使用することができます。次の例では、Microsoft Excel のマクロシートを作成します。次に、Sheet1 から NewSheet に名前を変更するなど、マクロシートへのコマンドを書き込みます。これらのコマンドを書き込んだ後、!DDE_FLUSH を使用すると、書き込まれた DATA ステップで Excel マクロが実行されます。

```
filename cmds dde 'excel|system';
data _null_;
file cmds;
/* Insert an Excel Macro Sheet */
put '[workbook.insert(3)]';
run;

/* Direct the Output to the Newly created Macro Sheet */
filename xlmacro dde 'excel|macro1!r1c1:r5c1' notab;

data _null_;
file xlmacro;
put '=workbook.name("sheet1","NewSheet")';
put '=halt(true)';
/* Dump the contents of the buffer, allowing us to both write and */
/* execute the macro in the same DATA Step                        */
put '!dde_flush';
file cmds;
/* Run Macro1 */
put '[run("macro1!r1c1")]';
put '[error(false)]';
/* delete the Macro Sheet */
put '[workbook.delete("macro1")]';
run;
```

### マクロ変数を使用した DDE コマンドの発行

次の例は、マクロ変数を使用して Microsoft Excel にコマンドを発行します。この例のマクロ変数 excelOne は、Excel ワークブックの場所 **C:\test.xls** の代わりに使用さ

れています。アンパサンドやパーセントなどのマクロトリガは、一重引用符内ではテキストとして処理されるため、マクロ引用関数を使用する必要があります。%STR を使用して、個々のアポストロフィをマスクします。対でない単独のアポストロフィやかっこの前にはパーセント記号を付け、各アポストロフィが個々に処理されるようにします。%STR によりアポストロフィがマスクされた後に、マクロ変数&excelOne が解決されます。%UNQUOTE を使用して%STR によるマスクを削除し、解決された値の前後のアポストロフィを復元することで、次の結果が得られます。

```
'[open("C:\test.xls")]'
```

```
options mprint symbolgen;
filename cmds dde 'excel|system';

%let excelOne=C:\test.xls;

data _null_;
file cmds;
put %unquote(%str(%'[open("&excelOne")]%'));
run;
```

## 欠損データの読み取り

次の例は、SHEET1 という名前の Excel スプレッドシートから、欠損データを読み込む方法を示しています。この例では、列 1 から 3 と行 10 から 20 のデータを読み込みます。これらのデータセルのうちのいくつかは空白です。データの例を次に示します。

```
...
10   John    Raleigh      Cardinals
11   Jose    North Bend   Orioles
12   Kurt    Yelm         Red Sox
13   Brent                Dodgers
...
```

次のコードを使用すると、これらのデータを SAS データセットに正しく読み込むことができます。

```
filename mydata
  dde 'excel|sheet1!r10c1:r20c3';
data in;
   infile mydata dlm='09'x notab
          dsd missover;
   informat name $10. town $char20.
            team $char20.;
   input name town team;
run;
proc print data=in;
run;
```

この例では、NOTAB オプションが指定されているため、Excel アプリケーションから送信されたタブはブランクに変換されません。したがって、タブ文字をデータ値間の区切り文字として使用できます。DLM=オプションは、区切り文字を指定します。**09'x** は、タブ文字の 16 進表記です。DSD オプションによって、2 つの連続した区切り文字が欠損値と見なされます。デフォルトの区切り文字はカンマです。DSD 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。MISSOVER オプションは、現在の行にすべての INPUT ステートメント変数の値がない場合に、新しい入力行の検索を防げます。MISSOVER オプションを指定した場合、INPUT ステートメントが現在のレコードの最後に到達すると、見つからなかった値が欠損データとして設定されます。

INFORMAT ステートメントによって、DATA ステップで強制的に修正リスト入力が使用されます。これはこの例において非常に重要です。修正リスト入力を使用しないと、間違った結果が導かれます。修正リスト入力を使用する必要性は、DDE 固有のものではありません。CARDS ステートメントでデータを使用する場合でも、そのデータがブランクまたはカンマで区切られているかに関係なく、修正リスト入力が必要になります。

*12 章*

# Windows 版の名前の付いていないパイプと名前付きパイプの使用について

**パイプの概要** . . . . . **279**

**名前の付いていないパイプの使用** . . . . . **280**
名前の付いていないパイプについて . . . . . 280
名前の付いていないパイプの構文 . . . . . 280
リダイレクトの順序の使用 . . . . . 281
名前の付いていないパイプの例 . . . . . 281

**名前付きパイプの使用** . . . . . **282**
名前付きパイプについて . . . . . 282
名前付きパイプの構文 . . . . . 282
CALL RECONNECT ルーチンの使用 . . . . . 284
SCL の名前付きパイプの使用 . . . . . 284
名前付きパイプの例 . . . . . 284

## パイプの概要

パイプとは、2 つのプロセス間の通信チャネルです。パイプを使用すると、ある端点のハンドルを持つプロセスが、他の端点のハンドルを持つ別のプロセスと通信することができます。つまり、Windows の専用アプリケーションと SAS セッションの間で情報を受け渡すことができます。

パイプは、一方向または双方向でテストできます。一方向通信を使用すると、一方のアプリケーションはパイプにデータを書き出すことだけが可能となり、他方のアプリケーションがパイプからデータを読み込みます。双方向通信を使用すると、双方のアプリケーションがデータを読み書きできます。パイプには次の 2 種類があります。

名前の付いていないパイプ
: 一方向通信を処理します。匿名パイプ(または単純にパイプ)とも呼ばれ、一般に、親プロセスと子プロセスとの間の通信に使用されます。SAS 内では、SAS が親プロセスとして子プロセスを起動し、そこからデータを読み込みます。

名前付きパイプ
: 関連のない 2 つのプロセス間の一方向通信または双方向通信を処理します。つまり、一方のプロセスが他方のプロセスによって開始されることはありません。実際、2 つのアプリケーションがネットワーク上のパイプを通じて通信します。SAS 内で名前付きパイプを使用すると、別のアプリケーションまたは別の SAS セッションと通信できます。

# 名前の付いていないパイプの使用

## 名前の付いていないパイプについて

名前の付いていないパイプを使用すると、SAS 以外のプログラムを呼び出してプログラムの入力、出力、エラーメッセージを SAS にリダイレクトできます。この機能によって、外部プログラムのデータを、中間データファイルを作成せずに取り込むことができます。

名前の付いていないパイプで外部 Windows アプリケーションを処理するには、このアプリケーションプログラムでは、データの読み込み元が標準入力ファイル(STDIN)、出力の書き込み先が標準出力ファイル(STDOUT)、エラーの書き込み先が標準エラーファイル(STDERR)である必要があります。次に、これらのファイルに割り当てられている数値ファイルハンドルを示します。

*表 12.1 ファイルハンドル*

| ファイル | ファイルハンドル |
|---|---|
| STDIN | 0 |
| STDOUT | 1 |
| STDERR | 2 |

他のアプリケーションから STDERR を取り込む場合、エラーメッセージの出力先は、デフォルトで SAS ログに設定されます。他のアプリケーションの STDIN に書き出しを行う場合は、DATA ステップで PUT ステートメントを使用します。SAS では同じアプリケーションの STDIN への書き出しと STDOUT からの読み込みができるため、名前の付いていないパイプを使用することで、外部プログラムにデータを送信し、また同じ外部プログラムから出力とエラーメッセージを取り込むことができます。リダイレクトを使用すると、STDIN、STDOUT、STDERR をリダイレクトできます。

Windows デスクトップから SAS を起動した場合は、STDIN と STDOUT をプログラムで使用できません。

詳細については、“リダイレクトの順序の使用” (281 ページ) を参照するか、Windows の資料を参照してください。

## 名前の付いていないパイプの構文

名前の付いていないパイプを使用するには、次の構文の FILENAME ステートメントを実行します。

**FILENAME** *fileref* PIPE '*program-name*' *option-list*;

この FILENAME ステートメントの構文には、次の引数を使用できます。

*fileref*
任意の有効な fileref です。詳細は、“外部ファイルの参照” (148 ページ) を参照してください。

PIPE
名前の付いていないパイプを使用することを示すデバイスタイプキーワード。

*program-name*
外部の Windows アプリケーションプログラムを指定します。この引数にはプログラムの完全パス名を指定する必要があります。また、プログラムを含むディレクトリのパスは、Windows の PATH 環境変数に含まれる必要があります。この引数には、プログラムオプションを指定することもできます。たとえば、次の引数を指定すると、すべての株式データに対して STOCKMKT プログラムが呼び出されます。

```
'stockmkt.exe -all'
```

*option-list*
LRECL=オプションや RECFM=オプションなど、FILENAME ステートメントで有効なオプション。Windows 版での FILENAME ステートメントに使用可能なオプションの完全なリストは、“FILENAME ステートメント:Windows” (448 ページ) を参照してください。

## リダイレクトの順序の使用

標準の入力、出力、エラーの各コマンドを使用する Windows アプリケーションでは、名前の付いていないパイプ機能を使用できます。多くの Windows システムコマンドでは標準の入出力コマンドとエラーコマンドが使用されるため、SAS 内で名前の付いていないパイプを使用して Windows システムコマンドを使用することができます。特に指定しない限り、名前の付いていないパイプは、STDOUT と STDERR の出力先を 2 つの異なるファイルに設定します。STDOUT と STDERR を同一のファイルに格納するには、リダイレクトを使用します。たとえば、Windows DIR コマンドで STDERR を STDOUT にリダイレクトするには、次のように指定します。

```
filename listing pipe 'dir *.sas 2>&1';
```

この例では、このコマンドの実行中にエラーが発生した場合、STDERR (2)が同一ファイルに STDOUT (1)としてリダイレクトされます。このように、SAS では動作環境の機能を活用することができます。ファイルハンドルをリダイレクトする機能は、SAS ではなく Windows オペレーティングシステムの機能です。

## 名前の付いていないパイプの例

次の例では、Windows 版 SAS で名前の付いていないパイプの機能を使用して、金融レポートを作成します。この例では、最新の株式市場の情報を提供するプログラム (stockmkt.exe)があると仮定しています。SAS で株式市場レポートを呼び出し、株式市場プログラムで作成された最新のデータを取り込む必要があります。次に、SAS セッション内でパイプを作成して使用する方法を示します。

```
filename stocks pipe 'stockmkt.exe -all' console=min;
data report;
   infile stocks;
   input stock $ open close change;
run;
proc print;
   var stock open close change;
   sum change;
   title 'Stock Market Report';
run;
```

この例では、FILENAME ステートメントの PIPE デバイスタイプキーワードによって、ファイル参照名 STOCKS が名前の付いていないパイプとして定義されます。STOCKMKT.EXE は、株式市場データを生成するプログラム名です。ホストオプション

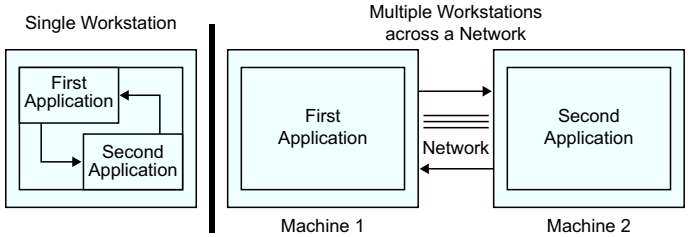
CONSOLE=MIN が指定されているため、STOCKMKT.EXE プログラムを実行するため開かれるコマンドプロンプトウィンドウが、最小化されて開かれます。INFILE ステートメントによって、SAS から STOCKMKT.EXE プログラムが呼び出され、このプログラムからパイプを介してデータが読み込まれます。Windows タスクバーにコマンドプロンプトウィンドウボタンが表示されますが、ユーザーが STOCKMKT.EXE プログラムの実行に気付くことなく、このプログラムは終了します。ファイル参照名 STOCKS はすでに名前の付いていないパイプとして定義されているため、STOCKMKT.EXE からの標準出力は SAS にリダイレクトされ、INFILE ステートメントによって取り込まれます。SAS プログラムは変数の値を読み込み、PRINT プロシジャを使用してレポートを表示します。STOCKMKT.EXE によって生成されたエラーメッセージがあれば、すべて SAS ログに出力されます。

# 名前付きパイプの使用

## *名前付きパイプについて*

名前付きパイプは、Windows 版 SAS において、他のアプリケーションとの通信に使用できる最も強力なツールの 1 つです。名前付きパイプを使用すると、同一コンピュータ上のアプリケーション間、またはネットワークで接続された複数のコンピュータ上のアプリケーション間で、データやメッセージを双方向に交換できます。次の図は、この 2 種類の通信方法を表しています。

***図 12.1*** *名前付きパイプを使用した通信*




アプリケーションとは、SAS セッションまたはその他の Windows アプリケーションです。たとえば、名前付きパイプと PRINTTO プロシジャを組み合わせて使用することにより、SAS プロシジャの結果を別の Windows アプリケーションに出力することができます。つまり、複数の SAS セッション間の通信に使用することも、または他の Windows アプリケーションと SAS セッションの通信に使用することもできます。

複数の SAS セッション間で通信するか、または、SAS セッションと名前付きパイプをサポートする他の Windows アプリケーションとの間で通信するかということに関係なく、パイプはクライアント/サーバー関係で定義します。1 つのプロセスがサーバーとして定義されます。1 つ以上のプロセスがクライアントとして定義されます。この構成では、複数のクライアントからサーバーにデータを送信すること、またはサーバーからさまざまなクライアントにデータを送信することができます。名前付きパイプは、さまざまなオプションを使用することにより、サーバーとクライアントとの間の処理を調整することができます。

## *名前付きパイプの構文*

ファイル参照名を使用する場所であれば、どのような箇所でも名前付きパイプが利用できます。名前付きパイプを使用するには、次の構文の FILENAME ステートメントを実行します。

**FILENAME** *fileref* NAMEPIPE '*pipe-specification*' <*named-pipe-options*>;

この FILENAME ステートメントの構文には、次の引数を使用できます。

*fileref*
任意の有効な fileref です。詳細は、“外部ファイルの参照” (148 ページ) を参照してください。

NAMEPIPE
名前付きパイプの使用を指定するデバイスタイプキーワードです。

*pipe-specification*
パイプの名前です。

この引数には、次の 2 つの相互排他的な構文があります。

\\.\PIPE*パイプ名*
スタンドアロンのコンピュータ上でパイプを確立する、またはネットワーク上にサーバーパイプを定義することを指定します。*パイプ名*引数には、パイプの名前を指定します。

*server-name*\PIPE*pipe-name*
ネットワークで接続された名前付きパイプサーバーに対するクライアントパイプを確立することを指定します。この場合は、必ず二重のバックスラッシュ(\\)を入れてください。*パイプ名*引数には、クライアントパイプの名前を指定します。*サーバー名*引数には、名前付きパイプサーバーの名前を指定します。

*named-pipe-options*
次のいずれかのオプションを使用することができます。最初に記載されている値がデフォルト値です。

SERVER | CLIENT
パイプのモードを指定します。デフォルト値は SERVER です。

BLOCK | NOBLOCK
現在使用できるデータがない場合に、クライアントまたはサーバーがデータが読み込まれるまで待つかどうかを指定します。BLOCK は待つことを示し、これがデフォルト値です。NOBLOCK は待たないことを示します。NOBLOCK を指定することにより、利用可能なデータがパイプにない場合は、ただちにプログラムに制御を戻すことができます。パイプへの書き出しは、常に BLOCK が指定されたとみなされます。

BYTE | MESSAGE
パイプのタイプを指定します。デフォルト値は BYTE です。BYTE パイプと MESSAGE パイプの違いは、MESSAGE パイプがエンコードされたレコード長を含むのに対して、BYTE パイプは含まないことです。

RETRY=*seconds*
クライアントまたはサーバーが、パイプが確立されるまで待つ時間の長さを指定します。*seconds* の最小値は 10 です。このオプションを使用することによって、クライアントとサーバーの同期のための時間が設定されます。デフォルトの待ち時間は 10 秒です。

*seconds* 引数には、特別な場合を示す 2 つの値があります。

-2 は、クライアントがサーバーの RETRY=オプションで定義された時間だけ待つことを示します。このオプションを使用する場合は、必ず SERVER を指定する必要があります。指定しないと、パイプ接続が失敗します。

-1 は、クライアントまたはサーバーがパイプの接続が確立されるまで無期限に待つことを示します。

EOFCONNECT
サーバーを定義するときにのみ有効です。ファイル終端文字(EOF)をクライアントから受け取ったときに、サーバーが次のクライアントに接続することを示します。

これらのオプションはすべて、Microsoft Win32 SDK などの Windows プログラマ用リファレンスで使用される用語と整合性がとれています。

## CALL RECONNECT ルーチンの使用

特殊な SAS CALL ルーチンである CALL RECONNECT ルーチンを使用すると、サーバーは現在のクライアントとの接続を切断し、利用可能な次のクライアントに接続することができます。通常、パイプのクライアント側がサーバーにファイル終端文字を送信すると、パイプが終了します。パイプ接続を任意の時点で切断するには、サーバー SAS セッションで CALL RECONNECT ルーチンを使用します。このルーチンの説明図は、“CALL RECONNECT ルーチン” (289 ページ) を参照してください。

## SCL の名前付きパイプの使用

SCL コードを使用して名前付きパイプを確立するには、FOPEN 関数を使用してファイル(またはパイプ)を開いてからアクセスする必要があります。その場合、クライアントアプリケーションとサーバーアプリケーションの両方に適切なオープンモードを指定して、両者がパイプ経由で通信できるようにする必要があります。使用可能な異なるノードの要約を示します。

*表 12.2 SCL コード使用時のノード*

| サーバーのアクセスモード | クライアントのアクセスモード |
|---|---|
| I (入力) | O (出力) |
| O (出力) | S (順次) |
| U (更新) | O (出力)または S (順次) |

## 名前付きパイプの例

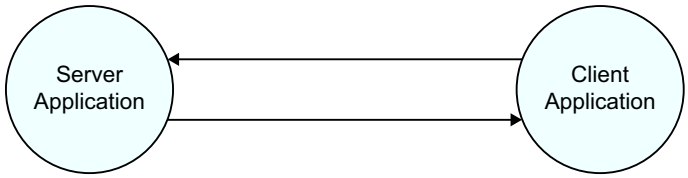
### 名前付きパイプ例の概要

名前付きパイプを理解するには、その使用法を説明するさまざまなパターンの例を確認することをお勧めします。ここで紹介するほとんどの例では、名前付きパイプは、2 つの SAS セッション間に確立されます。しかし、名前付きパイプは、SAS と、他の名前付きパイプをサポートするアプリケーションの間にも確立することができます。

### 簡単な名前付きパイプ: サーバーとクライアントの 1 対 1 接続

名前付きパイプのもっとも単純な構成は、図 12.2 (285 ページ) に示すように、サーバーとクライアントの 1 対 1 接続です。

**図 12.2** サーバーとクライアントの 1 対 1 接続




次の例では、WOMEN という名前のパイプが、2 つの SAS セッション間に確立されます。サーバー SAS セッションは、選択したデータをクライアント SAS セッションに送信します。サーバーとクライアントのどちらを先に開始することができ、一方の接続を 30 秒間待機します。

最初の SAS セッションで、サーバーとして名前付きパイプを作成します。

```
   /* Creates a pipe called WOMEN, acting */
   /* as a server. The server waits 30    */
   /* seconds for a client to connect.    */
filename women namepipe '\\.\pipe\women'
         server retry=30;
   /* This code writes three records into */
   /* the named pipe called WOMEN.        */
data class;
   input name $ sex $ age;
   file women;
   if upcase(sex)='F' then
      put name age;
   datalines;
MOORE M 15
JOHNSON F 16
DALY F 14
ROBERTS M 14
PARKER F 13
;
```

2 番目の SAS セッションで、SAS ステートメントを使用して 2 つの SAS セッション間でデータを交換します。たとえば、次のプログラムをクライアントセッションからサブミットできます。

```
   /* Creates a pipe called WOMEN, acting */
   /* as a client.  The client waits 30   */
   /* seconds for a server to connect.    */
filename in namepipe '\\.\pipe\women' client
         retry=30;
data female;
   infile in;
   input name $ age;
proc print;
run;
```

次に、サーバーとクライアントの 1 対 1 接続の例をもう 1 つ示します。この例は、RESULTS という名前のパイプと PRINTTO プロシジャを組み合わせて使用することにより、SUMMARY プロシジャの結果を別の Windows アプリケーションに出力する方法を示しています。

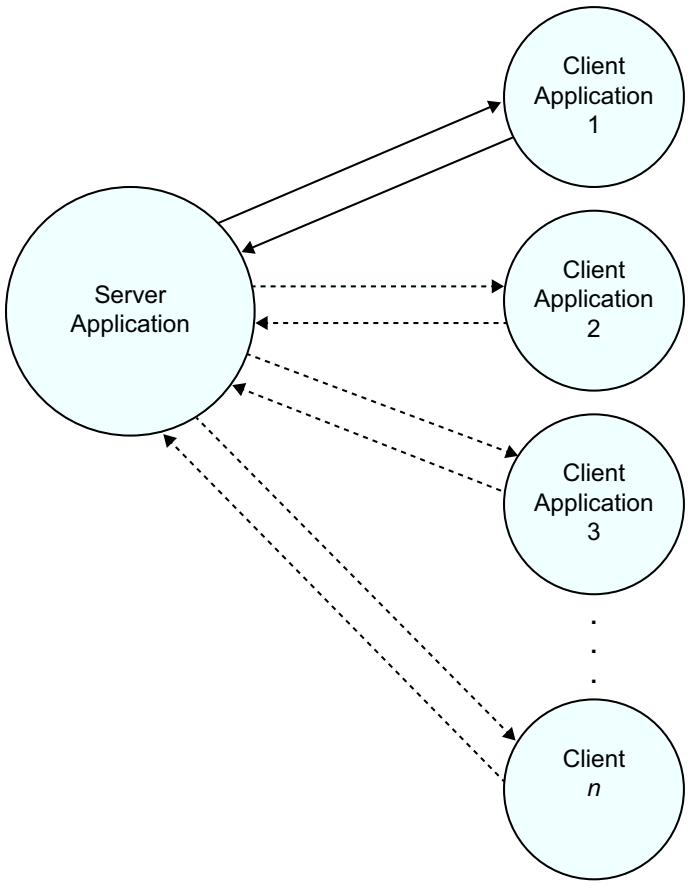
```
filename results namepipe '\\.\pipe\results'
     server retry=60;
proc printto print=results new;
run;
```

```
proc summary data=monthly;
run;
```

### 複数クライアントに接続する 1 つのサーバー

1 つのサーバーを複数のクライアントに接続するようにできます。この場合、名前付きパイプの構成は 図 12.3 (286 ページ) に示すようなものになります。

**図 12.3** 複数クライアントに接続する 1 つのサーバー




この構成では、初めにサーバーと最初のクライアントがデータ接続されます。この接続が終了すると、サーバーは 2 番目のクライアントを接続します。以降、この動作が繰り返されます。最後のクライアントの接続が終了すると、最初のクライアントに接続が戻ります。ただし、プログラムがそのように記述されている必要があります。

クライアントから次のクライアントに正しく接続が移動するようにするには、EOFCONNECT オプションを使用する必要があります。次に、1 つの SAS サーバーセッションと 2 つのクライアントに対して EOFCONNECT オプションを使用する例を示します。ここでは、クライアントは、スタンドアロンの PC、またはネットワーク上で接続された PC 上に存在することを前提としています。

最初の SAS セッションで、次のステートメントをサブミットします。

```
/* Creates a pipe called SALES, acting  */
/* as a server. The server waits 30     */
/* seconds for a client to connect.     */
/* After the client has disconnected,   */
/* this server SAS session tries to     */
/* connect to the next available client */
```

```
filename daily namepipe '\\.\pipe\sales'
         server eofconnect retry=30;
   /* This program reads in the daily     */
   /* sales figures sent from each client.*/
data totsales;
   infile daily;
   input dept $ item $ total;
run;
```

2 番目の SAS セッションで、次のステートメントをサブミットします。

```
  /* Creates a pipe called SALES, acting   */
  /* as a client. The client waits forever */
  /* for a server to connect. After the    */
  /* first client has disconnected, the    */
  /* second client connects with the server.*/
  /* The first client is the TOYS dept.    */
filename dept1 namepipe '\\.\pipe\sales'
         client retry=-1;
data toys;
   input item $ total;
   dept='TOYS';
   file dept1;
   put dept item total;
   datalines;
DOLLS 100
MARBLES 10
BLOCKS 50
GAMES 60
CARS 40
;
  /* The second client is the SPORTS dept.*/
  /* These data could come from a separate */
  /* SAS session.                          */
filename dept2 namepipe '\\.\pipe\sales'
         client retry=-1;
data sports;
   input item $ total;
   dept='SPORTS';
   file dept2;
   put dept item total;
   datalines;
BALLS 30
BATS 65
GLOVES 15
RACKETS 75
FISHING 20
TENTS 115
HELMETS 45
;
```

### *NOBLOCK オプション*

次の例では、NOBLOCK オプションを使用して、パイプを読み込んだ際に利用可能なデータがない場合に、プログラムの処理を続行することを指定します。デフォルト値の BLOCK を使用した場合、パイプは、パイプ内にデータが見つかるまで無期限に待ちます。EOFCONNECT オプションを指定すると、クライアントが送信したファイル終端文

字(EOF)を受信したサーバーに対して、他のクライアントに接続するように指示します。RETRY=オプションを指定すると、クライアントは無期限にサーバーを待ちますが、サーバーは接続可能な他のクライアントセッションが存在するかどうかを 20 秒間検索します。ここでは、クライアントは、スタンドアロンの PC、またはネットワーク上で接続された PC 上に存在することを前提としています。サーバーは一度に 1 つのクライアントに接続します。また、その間他のクライアントは、サーバーへの接続を順番に待つキューに格納されます。

最初の SAS セッションで、次のステートメントをサブミットします。

```
   /* Defines a named pipe called LINE.   */
   /* Use the NOBLOCK option to specify   */
   /* that if no data are available when  */
   /* the read is performed, then continue.*/
   /* Use the EOFCONNECT option to tell   */
   /* the server to try to connect with a */
   /* new client if an end-of-file is     */
   /* encountered. Use the RETRY= option  */
   /* to tell the server to look for any  */
   /* new clients for 20 seconds.         */
filename data namepipe '\\.\pipe\line' server
         noblock eofconnect retry=20;
   /* This DATA step reads in all data  */
   /* from any clients connected to the */
   /* named pipe called LINE.           */
data all;
   infile data length=len;
   input @;
      /* If the length of the incoming  */
      /* record is 0, then no data were */
      /* found in the pipe; otherwise,  */
      /* read the incoming data.        */
   if len ne 0 then
      do;
         input machine $ width weight;
         output;
      end;
run;
proc print;
run;
```

次の各 DATA ステップは、単一のネットワークで接続された複数の PC 上で実行することができます。

```
   /* Defines a named pipe called LINE.  */
   /* The RETRY= option is set such that */
   /* the clients wait forever until a   */
   /* server is available                */
   /* (that is, RETRY=-1).               */
   filename data namepipe '\\.\pipe\line'
            client retry=-1;
   /* This is information from the */
   /* first machine/client.        */
data machine1;
   file data;
   input width weight;
   machine='LINE_1';
   put machine width weight;
```

```
   datalines;
5.3 18.2
3.2 14.3
4.8 16.9
6.4 20.8
4.3 15.4
6.1 19.5
5.6 18.9
;
   /* This is information from the */
   /* second machine/client.       */
filename data namepipe '\\.\pipe\line'
         client retry=-1;
data machine2;
   file data;
   input width weight;
   machine='LINE_2';
   put machine width weight;
   datalines;
4.3 17.2
5.2 18.4
6.8 19.9
3.4 14.5
5.3 18.6
4.1 17.1
6.6 19.5
;
```

### *CALL RECONNECT ルーチン*

次の例は、2 つのクライアントに接続するように名前付きパイプサーバーを設定する方法を示しています。(この例では、3 つのアクティブ SAS セッションが必要です)。前のクライアントがデータを送信してから少なくとも 30 秒経過した場合に、CALL RECONNECT ルーチンを使用し、名前付きパイプ経由で次のクライアントに再び接続します。各クライアントは、SAS サーバーセッションにデータを送信します。

サーバー SAS セッションで、次のステートメントをサブミットします。

```
filename data namepipe '\\.\pipe\orders'
         server noblock eofconnect retry=30;
data all;
   infile data length=len missover;
   input @;
      /* If the length of the incoming  */
      /* record is 0, then no data were */
      /* found in the pipe; otherwise,  */
      /* read the incoming data         */
   if len ne 0 then
      do;
         input operator $ item $ quantity $;
         if item='' or quantity='' then
            delete;
         else
            output;
         put operator= item= quantity=;
      end;
      /* If no data are being transmitted,*/
```

```
      /* try reconnecting to the next     */
      /* available client.                */
   else
      do;
            /* Use the named pipe fileref */
            /* as the argument of */
            /* CALL RECONNECT. */
         call reconnect('data');
      end;
   run;
```

2 番目の SAS セッション(最初のデータ入力オペレータ)で、次のステートメントをサブミットします。

```
filename data namepipe '\\.\pipe\orders'
         client retry=-1;
data entry1;
   if _n_=1 then
     do;
        window entry_1
         #1 @2 'ENTER STOP WHEN YOU ARE FINISHED'
         #3 @5 'ITEM NUMBER - ' item $3.
         #5 @5 'QUANTITY - ' quantity $3.;
     end;
     do while (upcase(_cmd_) ne 'STOP');
        display entry_1;
        file data;
        put 'ENTRY_1' +1 item quantity;
        item='';
        quantity='';
     end;
   stop;
run;
```

3 番目の SAS セッション(最初のデータ入力オペレータ)で、次のステートメントをサブミットします。

```
filename data namepipe '\\.\pipe\orders'
         client retry=-1;
data entry2;
   if _n_=1 then
     do;
       window entry_2
        #1 @2 'ENTER STOP WHEN YOU ARE FINISHED'
        #3 @5 'ITEM NUMBER - ' item $3.
        #5 @5 'QUANTITY - ' quantity $3.;
     end;
     do while (upcase(_cmd_) ne 'STOP');
        display entry_2;
        file data;
        put 'ENTRY_2' +1 item quantity;
        item='';
        quantity='';
     end;
   stop;
run;
```

*13 章*

# Windows 版 SAS から外部 DLL にアクセスする

**SAS のダイナミックリンクライブラリ(DLL)の概要** . . . . . **291**

**SASCBTBL 属性テーブル** . . . . . **292**
SASCBTBL 属性テーブルの概要 . . . . . 292
属性テーブルの構文 . . . . . 293
属性テーブルの重要性 . . . . . 296

**外部 DLL を使用する場合に特に注意すべき点** . . . . . **297**
PEEKLONG 関数を使用して文字列引数にアクセスする . . . . . 297
効率的に外部 DLL にアクセス . . . . . 298
SAS 変数を構造引数としてグループ化 . . . . . 299
MODULE の引数として定数と式を使用 . . . . . 300
MODULE 引数で使用する出力形式および入力形式の指定概要: . . . . . 301
MODULE ログメッセージについて . . . . . 305

**例** . . . . . **307**
文字列引数の更新 . . . . . 307
引数を値として渡す . . . . . 308
PEEKCLONG を使用して返されたポインタにアクセスする . . . . . 308
構造の使用 . . . . . 309
DLL ルーチンを PROC IML から起動 . . . . . 310

## SAS のダイナミックリンクライブラリ(DLL)の概要

ダイナミックリンクライブラリ(DLL)とは、任意のプログラミング言語で作成された 1 つ以上のルーチンを含む実行可能ファイルです。DLL は、多くのアプリケーションで使用される、便利なルーチンを格納する手法です。アプリケーションが DLL 内のルーチンを呼び出す必要がある場合、DLL をロードし、ルーチンを起動して、処理完了後に DLL をアンロードします。SAS には、外部ルーチンを SAS から呼び出すためのルーチンと関数が用意されています。DLL ルーチンへは、DATA ステップ、IML プロシジャ、SCL プログラムからアクセスできます。外部 DLL 内のルーチンを起動するには、MODULE 群の SAS CALL ルーチンおよび関数(MODULE、MODULEN、MODULEC、MODULEI、MODULEIN、MODULEIC を含む)を使用します。このドキュメントでは、MODULE 群の SAS CALL ルーチンおよび関数を総称して、MODULE 関数と呼びます。

外部 DLL ルーチンにアクセスするための一般的な方法を次に示します。

1. アクセスする DLL ルーチンを記述するテキストファイルを作成します。期待される引数と、戻り値がある場合は、このテキストファイルに含めます。この属性ファイル

は次に示すような特殊な形式である必要があります。“SASCBTBL 属性テーブル” (292 ページ)

2. FILENAME ステートメントを使用して、作成した属性ファイルに SASCBTBL ファイル参照名を割り当てます。
3. DATA ステップか SCL コードで、CALL ルーチンや関数(MODULE、MODULEN、または MODULEC)を使用して DLL ルーチンを呼び出します。使用する特定の関数は、期待される戻り値の型(なし、数値、文字)によって異なります。(また、PROC IML ステップ内で MODULEI、MODULEIN 、MODULEIC を使用することもできます)MODULE 関数については “MODULE 関数:Windows” (407 ページ) で説明します。

***注意:***

**外部 DLL へは、経験のあるプログラマのみアクセスするようにしてください。** 外部 DLL の関数にアクセスすると、処理制御が外部関数に移ります。誤って実行した場合や、外部関数の信頼性が低い場合、データが失われたり、コンピュータのリセットが必要になる可能性があります。

# SASCBTBL 属性テーブル

## *SASCBTBL 属性テーブルの概要*

MODULE ルーチンは、SAS に情報がない外部関数を起動します。このため、この関数の引数情報を指定することで、MODULE ルーチンで引数を検証し、必要に応じて変換することが可能になります。たとえば、整数の引数が必要なルーチンを呼び出すとします。SAS では、すべてのルーチンの数値引数に浮動小数点値を使用しているため、外部ルーチンを呼び出す前に、浮動小数点値を整数に変換する必要があります。MODULE ルーチンでは、SASCBTBL ファイル参照名により参照される属性テーブルから、この属性情報を検索します。

この属性テーブルは、MODULE 関数で呼び出せるルーチンの説明を含むシーケンシャルテキストファイルです。このテーブルによって、MODULE 関数で DLL ルーチンを呼び出す際に渡すパラメータリストを作成するために、指定された引数をどのように解釈するかが定義されます。

MODULE ルーチンは、SASCBTBL ファイル参照名で参照するファイルを開いて、テーブルを検索します。このファイル参照名を定義しないと、MODULE ルーチンは引数を変更せずに、要求された DLL ルーチンを呼び出します。

***注意:***

**属性テーブルを定義せずに MODULE 関数を使用すると、SAS がクラッシュしたり、コンピュータが強制的にリセットされる原因となります。** 外部関数を呼び出すには、属性テーブルを使用する必要があります。

属性テーブルには、ROUTINE ステートメントを使用して呼び出す DLL ルーチンの説明と、ARG ステートメントを使用してルーチンに関連付ける引数の説明を含める必要があります。

## 属性テーブルの構文

### 概要:属性テーブルの構文

属性テーブルファイル内では任意の場所にコメントを挿入できます。行の最初の非空白文字であるアスタリスク(*)、またはステートメントの終端のセミコロンに続いて、コメントを挿入します。コメントの終端には、セミコロンを使用します。

### ROUTINE ステートメント

次に、ROUTINE ステートメントの構文を示します。

**ROUTINE** *name* MINARG=*minarg* MAXARG=*maxarg*
<CALLSEQ=BYVALUE|BYADDR>
<STACKORDER=R2L|L2R>
<STACKPOP=CALLER|CALLED>
<TRANSPOSE=YES|NO> <MODULE=*DLL-name*>
<RETURNS=SHORT|USHORT|LONG|INT64|ULONG |DOUBLE|DBLPTR|CHAR<*n*>>
<RETURNREGS=DXAX>;

次に、ROUTINE ステートメント属性について説明します。

ROUTINE *name*
ROUTINE ステートメントを開始します。MODULE 関数を使用して呼び出す DLL 関数ごとに、ROUTINE ステートメントが必要です。*name* の値は、MODULE 関数の'*module*'引数の一部として指定したルーチン名や序数と一致する必要があります。*module* には、DLL 名(MODULE 属性で指定されない場合。詳細は、次の説明を参照)、ルーチン名や序数を指定します。たとえば、MODULE 関数の呼び出しに **`KERNEL32,GetPath`** を指定できるようにするには、ROUTINE *name* を **`GetPath`** にする必要があります。

*name* 引数では、大文字と小文字が区別されます。この引数は、ROUTINE ステートメントに必須です。

MINARG=*minarg*
DLL ルーチンに必要な引数の最小数を指定します。多くの場合、この値は MAXARG と同じです。しかし、一部のルーチンでは異なる値を指定できます。これは、必須属性です。

MAXARG=*maxarg*
DLL ルーチンに必要な引数の最大数を指定します。これは、必須属性です。

CALLSEQ=BYVALUE | BYADDR
DLL ルーチンで使用する呼び出しシーケンスメソッドを指定します。BYVALUE では値での呼び出しを、BYADDR ではアドレスでの呼び出しを指定します。デフォルト値は、BYADDR です。

FORTRAN と COBOL は、アドレスでの呼び出し言語です。通常、C は値での呼び出し言語ですが、特定のルーチンではアドレスでの呼び出しとして実行されることがあります。

MODULE ルーチンでは、すべての引数で同一の呼び出しメソッドを使用する必要はありません。ARG ステートメントに BYVALUE オプションと BYADDR オプションを使用することで、例外を設定できます。詳細については、このセクションの後半を参照してください。

STACKORDER=R2L | L2R
DLL ルーチンで要求されるスタック内の引数の順序を指定します。R2L を指定した場合、スタック内の引数を C 言語規約に従って配置します。最後の引数(呼び出し構文ではもっとも右にある引数)が最初に格納され、次に一つ隣の引数が格納されます。したがって、外部ルーチンへスタックが引き渡される際には、最初の引数が、最初に扱われる項目となります。デフォルト値は R2L です。

L2R を指定した場合、スタック内の引数を逆順に配置します。最初の引数が最初に格納され、2 番目の引数が次に格納されます。したがって、外部ルーチンへスタックが引き渡される際には、最後の引数が、最初に扱われる項目となります。Pascal ではこの呼び出し規則が使用されます。また一部の C ルーチンでも使用されることがあります。

STACKPOP=CALLER | CALLED
呼び出す側のルーチンまたは呼び出される側のルーチンのどちらを使用するかを指定します。指定したルーチンによって、ルーチンからスタックが戻された際にスタックがポップ(スタックポインタが更新)されます。デフォルト値は、CALLER (ルーチンを呼び出すコード)です。Microsoft の _stdcall 属性を 32 ビットコンパイラで使用する場合など、ルーチンによっては、呼び出される側のルーチンがスタックをポップするように指定する必要があります。

TRANSPOSE=YES | NO
DLL ルーチンを呼び出す前に、複数の行と複数の列で行列を転置するかどうかを指定します。この属性は、IML プロシジャ内から MODULEI、MODULEIC、MODULEIN で呼び出されたルーチンにのみ適用されます。

行順優位で行列を格納しない言語で作成されたルーチンを呼び出す場合は、TRANSPOSE=YES に設定する必要があります。たとえば、FORTRAN は列順優位です。

3 列と 2 行の行列の場合、次のようになります。**列 1 2 3 ---------- 行 1 | 10 11 12 2 | 13 14 15**

PROC IML では、この行列を 10、11、12、13、14、15 のように順番にメモリに格納します。しかし、FORTRAN ルーチンでは、この行列を 10、13、11、14、12、15 のように認識します。

デフォルト値は NO です。

MODULE=*DLL-name*
ルーチンの常駐場所にある実行可能なモジュール(DLL)名を指定します。MODULE 関数は、PATH 環境変数によって指定されたディレクトリを検索します。ROUTINE ステートメントに MODULE 属性を指定する場合、MODULE 関数の *モジュール*引数にモジュール名を指定する必要はありません(呼び出す DLL ルーチン名が属性テーブルで重複している場合は除く)。MODULE 関数については “MODULE 関数:Windows” (407 ページ) で説明します。

複数の ROUTINE ステートメントに、同一の MODULE 名を使用できます。また、異なる DLL に常駐すれば、重複する ROUTINE 名を使用することも可能です。

RETURNS=SHORT | USHORT | LONG | INT64 | ULONG | DOUBLE | DBLPTR | CHAR<*n*>
DLL ルーチンが返す値の種類を指定します。戻り値は、MODULEC(文字を返す)または MODULEN(数値を返す)のどちらを使用するかに基づき、適切に変換されます。次のような戻り値があります。

SHORT
短整数

USHORT
符号なし短整数

LONG
長整数

ULONG
符号なし長整数

INT64
長整数

DOUBLE
倍精度浮動小数点数

DBLPTR
浮動小数点レジスタを使用せずに、倍精度浮動小数点のポインタを使用します。DLL ルーチンの倍精度浮動小数点数値の扱いについては、DLL ルーチンのドキュメントを参照してください。

CHAR*n*
*n* バイト長までの文字列のポインタ。文字列は、ヌル終端文字列である必要があります。必要に応じて、空白が埋め込まれたり、切り捨てられてたりします。*n* を指定しないと、MODULE 関数では渡された SAS 文字変数の最大長が使用されます。

RETURNS 属性を指定しない場合、MODULE および MODULEI 呼び出しルーチンのみを使用してルーチンを呼び出す必要があります。RETURNS 属性を省略して、MODULEN/MODULEIN 関数または MODULEC/MODULEIC 関数を使用してルーチンを呼び出す場合、予期しない値が返されます。

ROUTINE ステートメントの後ろに、MAXARG=オプションで指定した数の ARG ステートメントを指定する必要があります。ARG ステートメントは、引数が MODULE ルーチン内で指定される順序で指定する必要があります。

### *ARG ステートメント*

ARG ステートメントの構文を次に示します。

**ARG** *argnum* NUM|CHAR <INPUT|OUTPUT|UPDATE> <NOTREQD|REQUIRED> <BYADDR|BYVALUE> <FDSTART> <FORMAT=*format*>;

次に、ARG ステートメント属性について説明します。

ARG *argnum*
引数の番号を定義します。この属性は必須です。引数を、最初のルーチン引数 (ARG 1)から昇順に定義します。

NUM | CHAR
引数が、数値または文字のどちらであるかを定義します。この属性は必須です。

ここで NUM を指定したにもかかわらず文字引数をルーチンに渡した場合、引数は標準的な数値入力形式を使用して変換されます。ここで CHAR を指定したにもかかわらず数値引数をルーチンに渡した場合、引数は BEST12 入力形式を使用して変換されます。

INPUT | OUTPUT | UPDATE
引数がルーチンへの入力、または出力引数、あるいは両方であるかを示します。INPUT を指定した場合、引数は変換されて DLL ルーチンへ渡されます。OUTPUT を指定した場合、引数は変換されずに、DLL ルーチンからの出力値で更新されます。UPDATE を指定した場合、引数は変換されて DLL ルーチンへ渡され、さらに DLL ルーチンからの出力値で更新されます。

定数でも式でもない可変引数には、OUTPUT と UPDATE のみを指定できます。

NOTREQD | REQUIRED
引数が必須であるかどうかを示します。NOTREQD を指定すると、MODULE ルーチンで引数を省略できます。省略した引数の後に引数が存在する場合、省略された引数のプレースホルダとしてカンマを挿入します。たとえば、ルーチン XYZ への 2 番目の引数を省略するには、**call module('XYZ',1,,3);を指定します。**

***注意:***
**NOTREQD を指定する場合には注意が必要です。MODULE の呼び出しに引数が指定されていない場合、DLL ルーチンはその引数にアクセスする必要はありません。このルーチンから引数へアクセスしようとすると、システムがクラッシュします。**

REQUIRED 属性は、引数が必須であり、省略できないことを示します。デフォルト値は REQUIRED です。

BYADDR | BYVALUE
引数を、参照または値のどちらで受け渡すかを示します。

ROUTINE ステートメントに CALLSEQ=BYVALUE が指定されなかった場合、BYADDR がデフォルト値です。その場合、BYVALUE がデフォルトです。値での呼び出しルーチンを使用する場合、このルーチンにアドレスで渡す引数も含まれるときは、BYADDR を指定します。

FDSTART
この後に続く引数は、1 つの引数としてポインタが渡される構造に組み込まれる値の集合であることを指定します。MODULE 関数に別の FDSTART 引数が現れるまで、後続のすべての引数がその構造の一部として扱われることに注意してください。

FORMAT=*format*
DLL ルーチンの引数を表す出力形式を指定します。SAS で用意されているすべての出力形式、PROC FORMAT スタイルの出力形式、または SAS/TOOLKIT 出力形式が有効です。引数に UPDATE 属性または OUTPUT 属性を指定した場合、この出力形式には必ず、対応する有効な入力形式が必要なことに注意してください。

FORMAT=属性は必須ではありません。ただし、属性テーブルの ARG ステートメントの主な用途が出力形式の指定である場合は、この属性の指定をお勧めします。

***注意:***
**誤った出力形式を使用すると、無効な値を生成またはシステムがクラッシュすることがあります。**

## *属性テーブルの重要性*

MODULE ルーチンは、属性テーブルの情報の精度に大きく依存します。この情報に誤りがある場合、システムのクラッシュを含めた予期しない結果を招くことがあります。

2 つの引数(整数とポインタ)が必要なルーチン **xyz** の例を次に示します。整数は、操作を指定するコードです。たとえば、操作 1 によって、20 バイトの文字列が引数 2(ポインタ)で指定された領域に書き出されます。

ここで、MODULE ルーチンを使用しますが、属性テーブルに渡される文字引数が 10 文字の長さしかないことを示す **xyz** を呼び出すとします。

```
routine xyz minarg=2 maxarg=2;
arg 1 input num byvalue format=ib4.;
arg 2 output char format=$char10.;
```

MODULE への 2 番目の引数に対する LENGTH ステートメントが指定した値に関わらず、MODULE は **xyz** ルーチンへの 10 バイト領域へとポインタを渡します。**xyz** に

よってその 10 バイト領域に 20 バイトが書き込まれる場合、MODULE が提供した文字列の後の 10 バイトのメモリも上書きされ、予期せぬ結果を招くことがあります。

```
data _null_;
    length x $20;
    call module('xyz',1,x);
    run;
```

どの 10 バイトが上書きされたかによっては、呼び出しが成功する場合もあります。ただし、データが失われたり、システムがクラッシュする可能性があります。

また、PEEKLONG 関数と PEEKCLONG 関数は、指定したポインタの有効性に依存します。ポインタが無効な場合、SAS がクラッシュする可能性があります。たとえば、次のコードはクラッシュを招きます。

```
data _null_;
   length c $10;
     /* trying to copy from address 0!!!*/
   c = peekclong(0,10);
run;
```

PEEKLONG や PEEKCLONG を使用する際は、ポインタが有効であることを確認してください。

# 外部 DLL を使用する場合に特に注意すべき点

### *PEEKLONG 関数を使用して文字列引数にアクセスする*

SAS 言語ではポインタをデータ型として扱わないため、ポインタを表すには、PIB4.入出力形式を使用する必要があります。ポインタの表現後、SAS PEEKLONG 関数を使用して、アドレス値に格納されたデータにアクセスすることができます。

たとえば、SERVICES.DLL というライブラリに GetPath というルーチンがあるとします。このルーチンでは、整数関数コードと、ポインタを示すポイント位置の、2 つの引数が指定されます。関数コードは、GetPath のアクションを指定します。2 番目の引数は、システム文字列を参照するために、GetPath によって更新されるポインタへのポイント位置を示します。この場合、C 言語での呼び出しコードは次のようになります。

```
GetPath(1,&stgptr);
printf("GetPath indicates string is
   '%s'.\n",stgptr);
```

MODULE を使用することで、対応する属性テーブルエントリーは

```
ROUTINE GetPath MINARG=2 MAXARG=2
  MODULE=SERVICES;
ARG 1 NUM INPUT BYVALUE FORMAT=PIB4.;
ARG 2 NUM OUTPUT BYADDR FORMAT=PIB4.;
```

となり、次のように呼び出されます。

```
call module('GetPath',1,stgptr);
put stgptr= stgptr=hex8.;
```

STGPTR のポインタ値が 0035F780 の場合、STGPTR は実際には 10 進法の値 3536768 に設定されます。3536768 は 0035F780 の 10 進法での同値です。そこで、上記の PUT ステートメントがを発行します。

```
STGPTR=3536768 STGPTR=0035F780
```

しかし、ここで必要なのはアドレス 0035F780 にあるデータであって、ポインタそのものの値ではありません。このデータにアクセスするには、PEEKCLONG 関数を使用します。

PEEKCLONG 関数に、数値変数(たとえば前述の STGPTR など)を使用したポインタと、バイト数(文字数)の長さの、2 つの引数を指定します。PEEKCLONG 関数から、ポインタが示す場所にある文字を含む、指定した長さの文字列が返されます。

この例では、GetPath によって、2 番目の引数のポインタ値がヌル終端文字列のアドレス C:\XYZ に設定されると仮定します。文字データへのアクセス方法は、次のとおりです。

```
call module('SERVICES,GetPath',1,stgptr);
length path $64;
path = peekclong(stgptr,64);
i = index(path,'00'x);
if i then substr(path,i)=' ';
/* path now contains the string */
```

PEEKCLONG 関数では、STGPTR のポインタによって参照される場所から 64 バイト分がコピーされます。必要なデータはヌル終端まで(ヌルは含まず)のため、INDEX 関数でヌル終端を検索します。次に、その場所から後の文字を全て削除して空白にする必要があります。

また、この筋書きでは、$ CSTR 出力形式を使ってコードを少し単純にできます。

```
call module('SERVICES,GetPath',1,stgptr);
length path $64;
path = put(peekclong(stgptr,64),$cstr64.);
```

$ CSTR 出力形式は指定された幅の文字列を入力として受け付けます。ヌル終端を検索し、そこから出力文字列をブランクで埋め込みます。

詳細については、“PEEKLONG Function” (*SAS Functions and CALL Routines: Reference*) を参照してください。

## *効率的に外部 DLL にアクセス*

MODULE ルーチンでは、ステップ(DATA ステップ、PROC IML ステップ、SCL ステップ)ごとに、SASCBTBL ファイル参照名によって参照される属性テーブルが読み込まれます。MODULE によってテーブルが解析され、後のステップで使用できるように属性情報が格納されます。MODULE 関数を使用すると、ルーチン名およびモジュール名が一致する場合、その格納された属性情報が SAS によって検索されます。ステップ内で最初に DLL にアクセスしたときに、SAS によって DLL がロードされ、要求されたルーチンのアドレスが決定されます。呼び出す各 DLL はステップ中、ロードされた状態となり、後続の呼び出しでリロードされることはありません。すべてのモジュールおよびルーチンは、ステップ完了時にアンロードされます。たとえば、属性テーブルがのような基本的な形式を取ったとします。

```
* routines XYZ and BBB in FIRST.DLL;
ROUTINE XYZ MINARG=1 MAXARG=1 MODULE=FIRST;
ARG 1 NUM INPUT;
ROUTINE BBB MINARG=1 MAXARG=1 MODULE=FIRST;
ARG 1 NUM INPUT;
* routines ABC and DDD in SECOND.DLL;
ROUTINE ABC MINARG=1 MAXARG=1 MODULE=SECOND;
ARG 1 NUM INPUT;
ROUTINE DDD MINARG=1 MAXARG=1 MODULE=SECOND;
ARG 1 NUM INPUT;
```

それから、DATA ステップがのようなものだとします。

```
filename sascbtbl 'myattr.tbl';
data _null_;
   do i=1 to 50;
      /* FIRST.DLL is loaded only once */
      value = modulen('XYZ',i);
      /* SECOND.DLL is loaded only once */
      value2 = modulen('ABC',value);
      put i= value= value2=;
   end;
run;
```

この例では、DATA ステップ編集の間、属性テーブルは MODULEN によって構文解析されます。最初の反復ループ(i=1)では、MODULEN による呼び出し時に FIRST.DLL をロードし、XYZ ルーチンにアクセスします。次に、SECOND.DLL をロードし、ABC ルーチンにアクセスします。次の反復ループ(i=2 で開始)では、FIRST.DLL および SECOND.DLL はロードされた状態であるため、MODULEN 関数は XYZ ルーチンおよび ABC ルーチンにのみアクセスします。DATA ステップの終了時には、どちらの DLL もアンロードされます。

属性テーブルには、特定のステップではアクセスしないルーチンの記述を格納することができます。格納するルーチンの数に制限はありません。これにより追加のオーバーヘッド(属性記述を格納するための内部メモリから数バイトの空白を除く)を生じることはありません。前述の例では、BBB および DDD は属性テーブルにありますが、DATA ステップからはアクセスされていません。

## *SAS 変数を構造引数としてグループ化*

通常、外部ルーチンを呼び出す際、ポインタを構造に渡す必要があります。構造の一部は 、ルーチンへの入力として使用できる場合があります。他の部分は、ルーチンによって置換されたりルーチンに挿入されたりする場合があります。SAS の言語が構造化されていない場合でも、MODULE に、特定の引数の組み合わせを 1 つの構造としてグループ化するように指定することは可能です。ARG ステートメントの FDSTART オプションを使用し、属性テーブルでの構造を開始する引数にフラグを設定します。指定された引数と、そこから次の FDSTART オプションが指定されるまでのすべての引数が、SAS によって 1 つのブロックとしてまとめられます。このブロックを示すポインタが引数として DLL ルーチンに渡されます。

たとえば、USER32.DLL の Win32 API の一部である GetClientRect ルーチンについて考えてみます。このルーチンは、ウィンドウのクライアント領域の座標を取得します。この処理によって、座標が必要なウィンドウのウィンドウハンドルを取得するには、GetForegroundWindow という別のルーチンも使用する必要があります。

これらのルーチンの C 言語プロトタイプはです。

```
HWND GetForegroundWindow(VOID);
BOOL GetClientRect(HWND hWnd, LPRECT lprc);
```

C 言語では、ルーチンを呼び出すコードはになります。

```
typedef struct tagRECT {
    int left;
    int top;
    int right;
    int bottom;
    } RECT;
  /* RECT is a structure variable */
....                  /* other code */
```

```
/* Need the window handle first */
hWnd=GetForegroundWindow();
/* Function call, passing the address */
/* of RECT                            */
GetClientRect(hWnd, &RECT);
```

MODULE を使用してこれらのルーチンを呼び出すには、次の属性テーブルエントリ

```
routine GetForegroundWindow
  minarg=0
  maxarg=0
  stackpop=called
  module=USER32
  returns=long;
routine GetClientRect
  minarg=5
  maxarg=5
  stackpop=called
  module=USER32;
arg 1 num input byvalue format=pib4.;
arg 2 num update fdstart format=ib4.;
arg 3 num update         format=ib4.;
arg 4 num update         format=ib4.;
arg 5 num update         format=ib4.;
```

次の DATA ステップが使用されます。

```
filename sascbtbl 'sascbtbl.dat';
data _null_;
  hwnd=modulen('GetForegroundWindow');
  call module('GetClientRect',hwnd,left,
       top,right,bottom);
  put left= top= right= bottom=;
run;
```

ARG ステートメントで引数 2 に FDSTART オプションを使用することによって、引数 2 とそれ以降のすべての引数を、1 つのパラメータブロックとしてグループ化することが指示されます。

PUT ステートメントからのログには、次のように出力されます。

```
LEFT=2 TOP=2 RIGHT=400 BOTTOM=587
```

### *MODULE の引数として定数と式を使用*

MODULE 関数の引数として、任意の種類の式を渡すことができます。属性テーブルで、引数が入力引数、出力引数、更新引数のどれであるかを示します。

入力引数を定数や数式として指定できます。しかし、出力引数および更新引数は修正されたり戻されたりするため、これらのパラメータには変数しか渡すことができません。更新可能な値が必要な引数に定数または数式を指定すると、エラーを指摘する警告メッセージが発行されます。処理は継続されますが、MODULE ルーチンでは定数または数式の引数を更新できません。したがって、更新しようとした引数の値は失われます。

次の例を参照してください。属性テーブルを示します。

```
* attribute table entry for ABC;
routine abc minarg=2 maxarg=2;
arg 1 input format=ib4.;
```

```
arg 2 output format=ib4.;
```

MODULE 呼び出しと DATA ステップ:

```
data _null_;
  x=5;
  /* passing a variable as the    */
  /*   second argument - OK       */
  call module('abc',1,x);
  /* passing a constant as the    */
  /*   second argument - INVALID  */
  call module('abc',1,2);
  /* passing an expression as the */
  /*   second argument - INVALID  */
  call module('abc',1,x+1);
run;
```

この例において、最初の MODULE 呼び出しは正しい呼び出しです。変数 **x** は、**abc** ルーチンから 2 番目の引数に対して戻される値によって更新されます。2 番目の MODULE 呼び出しは、定数が渡されているため、正しくありません。MODULE は定数が渡されたことを示す警告メッセージを発行し、定数を渡すかわりに一時的な領域を渡します。3 番目のの MODULE 呼び出しは、数式が渡されるため、正しくありません。DATA ステップからの一時的な場所の使用の原因となります。その戻り値は失われます。

## *MODULE 引数で使用する出力形式および入力形式の指定概要:*

### *MODULE の引数で使用する出力形式と入力形式を指定する*

各 DLL ルーチン引数に SAS 入出力形式を指定するには、ARG ステートメントの FORMAT 属性を属性テーブルに指定します。入出力形式は、数値および文字値の DLL ルーチンへの渡し方、ルーチン完了時の読み取り方を示します。

通常、使用する出力形式は、プログラミング言語で決められた変数の型に対応します。次のセクションで、さまざまなプログラミング言語の変数の型に対応する、適切な入出力形式について説明します。

*注:* 文字データをポインタとしてではなく、文字列へと渡す場合の詳細については、次を参照してください。“$BYVALw.ファイルの種類” (304 ページ)

### *C 言語形式*

***表 13.1*** *C 言語形式*

| C 言語の型 | SAS 出力形式/入力形式 |
|---|---|
| double | RB8. |
| float | FLOAT4. |
| signed int | IB4. |
| signed short | IB2. |
| signed long | IB4. |

| C 言語の型 | SAS 出力形式/入力形式 |
|---|---|
| char * | IB4.(32 bit SAS) |
| char * | IB8 (x64) |
| unsigned int | PIB4. |
| unsigned short | PIB2. |
| unsigned long | PIB4. |
| char[*w*] | $CHAR*w*.または$CSTR*w*( “$CSTRw.ファイルの種類” (304 ページ) を参照) |

### *FORTRAN 言語形式*

***表 13.2*** *FORTRAN 言語形式*

| FORTRAN 言語の型 | SAS 出力形式/入力形式 |
|---|---|
| integer*2 | IB2. |
| integer*4 | IB4. |
| real*4 | FLOAT4. |
| real*8 | RB8. |
| character**w* | $CHAR*w*. |

MODULE ルーチンで FORTRAN 文字引数をサポートする場合は、引数がディスクリプタによって渡されないときに限られます。

### *PL/I 言語形式*

***表 13.3*** *PL/I 言語形式*

| PL/I 言語の型 | SAS 出力形式/入力形式 |
|---|---|
| FIXED BIN(15) | IB2. |
| FIXED BIN(31) | IB4. |
| FLOAT BIN(21) | RB4. |
| FLOAT BIN(31) | RB8. |
| CHARACTER(*w*) | $CHAR*w*. |

PL/I の説明は、ここのセクションの説明を完全なものにするために追加されています。PL/I ルーチンが呼び出し可能なことを保証するものではありません。

## COBOL 言語形式

**表 13.4** COBOL 言語形式

| COBOL 言語の型 | Windows 版 SAS 出力形式/入力形式 | 説明 |
|---|---|---|
| PIC S*xxxx* BINARY | IB*w*. | バイナリ整数 |
| COMP-2 | RB8. | 倍精度浮動小数点 |
| COMP-1 | RB4. | 単精度浮動小数点 |
| PIC *xxxx* または S*xxxx* | F*w*. | 出力可能数値 |
| PIC *yyyy* | $CHAR*w*. | 文字 |

次に示す COBOL 言語の型は、SAS 入出力形式と正確に一致しない場合があります。これはゾーン 10 進とパック 10 進が、Intel アーキテクチャに基づくシステム向けに本来定義されていないためです。

**表 13.5** COBOL 言語形式(ゾーン 10 進とパック 10 進)

| COBOL 言語の型 | Windows 版 SAS 出力形式/入力形式 | 説明 |
|---|---|---|
| PIC S*xxxx* DISPLAY | ZD*w*. | ゾーン 10 進 |
| PIC S*xxxx* PACKED-DECIMAL | PD*w*. | パック 10 進 |

次に示す COBOL 言語の型は、真にネイティブな同等物がなく、対応する S370F*xxx* 入出力形式と連結して使用可能です。S370F エレメントにより、IBM とメインフレーム型表現が PC 環境で読み書き可能になります。

**表 13.6** COBOL 言語形式(PC 環境)

| COBOL 言語の型 | SAS 出力形式/入力形式 | 説明 |
|---|---|---|
| PIC *xxxx* DISPLAY | S370FZDU*w*. | 符号なしゾーン 10 進 |
| PIC S*xxxx* DISPLAY SIGN LEADING | S370FZDL*w*. | 前符号付きゾーン 10 進 |
| PIC S*xxxx* DISPLAY SIGN LEADING SEPARATE | S370FZDS*w*. | 分離した前符号付きゾーン 10 進 |
| PIC S*xxxx* DISPLAY SIGN TRAILING SEPARATE | S370FZDT*w*. | 分離した後符号付きゾーン 10 進 |

| COBOL 言語の型 | SAS 出力形式/入力形式 | 説明 |
|---|---|---|
| PIC *xxxx* BINARY | S370FIBU*w*. | 符号なしバイナリ整数 |
| PIC *xxxx* PACKED-DECIMAL | S370FPDU*w*. | 符号なしパック 10 進 |

### *$CSTRw.ファイルの種類*

文字引数をヌル終端文字列として渡す場合 、$CSTR*w*.出力形式を使用します。この出力形式では、文字引数の最後にある非ブランク文字が検索され、この最後の非ブランク文字の後にヌル終端を追加した文字列のコピーが渡されます。たとえば、次のコードフラグメントには 属性テーブルエントリが表示されます。

```
* attribute table entry;
routine abc minarg=1 maxarg=1;
arg 1 input char format=$cstr10.;
```

次の DATA ステップを使用できます。

```
data _null_;
     rc = module('abc','my string');
     run;
```

$CSTR 出力形式では、**abc** ルーチンに渡す前に、ヌル終端が文字列 **my string** に追加されます。これは、次の属性エントリと同等です。

```
* attribute table entry;
routine abc minarg=1 maxarg=1;
arg 1 input char format=$char10.;
```

次の DATA ステップが使用されます。

```
data _null_;
     rc = module('abc','my string'||'00'x);
     run;
```

最初の例のほうがよりわかりやすく、変数または式の引数を使用する場合にはより簡単に使用できます。

$CSTR 入力形式では、ヌル終端文字列が、指定された長さのブランク埋め込み文字列に変換されます。DLL ルーチンが文字引数を更新する場合は、引数属性に$CSTR 入力形式を使用します。

### *$BYVALw.ファイルの種類*

MODULE 関数を使用し、値によって単一の文字を渡すと、引数は自動的に整数にプロモートされます。MODULE 呼び出しで文字式を使用するには、$BYVAL*w*.と呼ばれる特別な出力形式/入力形式を使用する必要があります。$BYVAL*w*.出力形式/入力形式では、単一の文字が受け入れられ、数値が作成されます。数値のサイズは、*w* によって決定されます。$BYVAL2.では短整数値が作成され、$BYVAL4.では長整数値が作成され、$BYVAL8.では倍精度浮動小数点数値が作成されます。この例を、C 言語で表現すると、次のとおりになります。

```
long xyz(a,b)
  long a; double b;
  {
  static char c = 'Y';
  if (a == 'X')
     return(1);
  else if (b == c)
```

```
    return(2);
  else return(3);
  }
```

この例では、**xyz** ルーチンに long と double の 2 つの引数が必要です。long が **x** の場合、long の実際の値は 10 進では 88 になります。ASCII **x** は 16 進の 58 として格納され、0x00000058 (つまり 10 進の 88 )として表示される long 値にプロモートされます。**a** の値が **x** (または 88)の場合、1 が返されます。2 番目の引数 double の値が 89 と解釈される **y** の場合は、2 が返されます。

**xyz** に引数として文字を渡す場合を考えます。C 言語では、これらの引数を次のように呼び出します。

```
x = xyz('X',(double)'Z');
y = xyz('Q',(double)'Y');
```

**x** の値と **q** の値が自動的に、(この例のために long と同じである)整数にプロモートされます。**z** と **y** に対応する整数値は double に割り当てられます。

MODULEN 関数を使って **xyz** を呼び出すには、文字を渡すということを属性テーブルに示す必要があります。

```
routine xyz minarg=2 maxarg=2 returns=long;
arg 1 input char byvalue format=$byval4.;
arg 2 input char byvalue format=$byval8.;
```

注記:ARG ステートメントに、BYVALUE オプションも表示されることは重要です。このオプションを使用しない場合、MODULEN 関数では、値ではなくポインタをルーチンに渡すものと仮定します。

次に、MODULEN を呼び出し、文字を渡す DATA ステップを示します。

```
data _null_;
     x = modulen('xyz','X','Z');
     put x= ' (should be 1)';
     y = modulen('xyz','Q','Y');
     put y= ' (should be 2)';
     run;
```

### *MODULE ログメッセージについて*

MODULE の制御文字列パラメータで **i** を指定すると、情報メッセージがログに出力されます。このメッセージを使用して、誤った引数を渡していないか、または属性テーブルのコードが誤っていないかを確認できます。

IML プロシジャ内から MODULEIN を使用する場合を想定します。MODULEIN 関数を使用して、論理 TRYMOD.DLL に格納された **changi** ルーチンが呼び出されます。この例では、MODULEIN によって定数 6 および行列 x2(4x5 行列で整数行列に変換される)が渡されます。**changi** の属性テーブルは次の通りです。

```
routine changi module=trymod returns=long;
arg 1 input num format=ib4. byvalue;
arg 2 update num format=ib4.;
```

MODULEIN は次の IML ステップに呼び出されます。

```
proc iml;
     x1 = J(4,5,0);
     do i=1 to 4;
        do j=1 to 5;
           x1[i,j] = i*10+j+3;
```

```
      end;
   end;
y1= x1;
        x2 = x1;
                y2 = y1;
rc = modulein('*i','changi',6,x2);
....
```

**'*i'**制御文字列により、アウトプット 13.1 (306 ページ) に示される行がログに出力されます。

***アウトプット 13.1*** *MODULEIN 出力*

```
---PARM LIST FOR MODULEIN ROUTINE---  CHR PARM 1 885E0AA8 2A69 (*i)
CHR PARM 2 885E0AD0 6368616E6769 (changi)
NUM PARM 3 885E0AE0 0000000000001840
NUM PARM 4 885E07F0
0000000000002C400000000000002E40000000000000304000000000000031400000000000003240
0000000000003840000000000000394000000000000003A400000000000003B400000000000003C40
0000000000041400000000000804140000000000
---ROUTINE changi LOADED AT ADDRESS 886119B8 (PARMLIST AT 886033A0)--- PARM 1
06000000     <CALL-BY-VALUE>
PARM 2 88604720
0E0000000F000000100000001100000012000000180000001900000001A0000001B0000001C000000
220000002300000024000000250000002600000002C0000002D0000002E0000002F00000030000000
---VALUES UPON RETURN FROM changi ROUTINE---   PARM 1 06000000
<CALL-BY-VALUE>
PARM 2 88604720
140000001F0000002A000000350000004000000082000008D000000980000000A3000000AE000000
F0000000FB000000060100001101000001C0100005E01000069010000740100007F0100008A010000
---VALUES UPON RETURN FROM MODULEIN ROUTINE---  NUM PARM 3 885E0AE0
0000000000001840
NUM PARM 4 885E07F0
000000000003440000000000003F400000000000004540000000000804A400000000000005040
0000000004060400000000000A0614000000000000063400000000006064400000000000C06540
0000000000006E400000000000606F4000000000
```

この出力は、4 つのセクションに分けられます。

1. 最初のセクションには、MODULEIN に渡される引数が記述されています。

   'CHR PARM *n*'の部分は、文字パラメータ *n* が渡されたことを示します。この例では、885E0AA8 が MODULEIN へ渡された最初の文字パラメータの実際のアドレスです。アドレスの値は 16 進数の 2A69 で、その値を ASCII で表現したもの(「*i」)が 16 進数値の後にかっこ付きで表示されます。2 番目のパラメータも、同様に出力されます。最初の 2 つの引数は、対応する ASCII を出力します。これらが出力される理由は、他の引数に、読み取り不可能なバイナリデータが含まれている可能性があるためです。

   この残りのパラメータは、16 進値(この例では NUM PARM 3 および NUM PARM 4)のみが表示されています。

   MODULEIN の 3 番目のパラメータは数値で、アドレス 885E0AE0 にあります。浮動小数点数 6 を 16 進数で表現した値が表示されています。4 番目のパラメータは、アドレス 885E07F0 にあり、4x5 行列のすべてのデータがある領域へのポインタを示します。**i** オプションがあるため、引数全体が出力されます。行列が大きい場合にこのオプションを使用するとログが膨大になるため、注意が必要です。

2. ログの 2 番目のセクションでは、要求されたルーチンに渡される引数がリストされます。この例では変更された引数がリストされています。このセクションは、引数が正しくルーチンに渡されているかどうかを確認するために重要です。このセクション

の 1 行目は、ルーチン名およびメモリのアドレスです。また、MODULEIN が作成したパラメータブロックがある場所のアドレスも含まれます。

ログには、各引数が渡されたときのステータスが格納されます。たとえば、この例の最初のパラメータは、ログに示すとおり callbyvalue です。2 番目のパラメータは、行列のアドレスです。ログには、アドレスとそのアドレスが指すデータが出力されます。

最初のパラメータと行列に含まれる値がすべて長整数であることに注意してください。これは属性テーブルで出力形式 IB4 が指定されているためです。

3. 3 番目のセクションには、ログには **changi** からの戻り値に基づく引数値が含まれています。callbyvalue 引数には変更はありませんが、他の引数(行列)に異なる値が含まれています。
4. ログ出力の最後のセクションには、引数が MODULEIN 呼び出しルーチンに戻されたときの値が含まれています。

# 例

## *文字列引数の更新*

この例では、Win32 ルーチンの GetTempPathA を使用します。このルーチンは、引数としてバッファへのポインタおよびバッファ長が必要です。GetTempPathA は、一時パスを表すヌル終端文字列をバッファに書き込みます。GetTempPathA ルーチンの C 言語プロトタイプを示します。

```
DWORD WINAPI GetTempPathA
    (DWORD nBufferLength, LPSTR lpBuffer);
```

属性テーブルを示します。

```
routine GetTempPathA
  minarg=2
  maxarg=2
  stackpop=called
  returns=long;
arg 1 input  byvalue format=pib4.;
arg 2 update         format=$cstr200.;
```

STACKPOP=CALLED オプションが使われていることに注意してください。すべての Win32 サービスルーチンでこの属性が要求されます。最初の引数は、入力引数であるため、値が渡されます。2 番目の引数は、バッファ内容が更新される更新引数です。$CSTR200.出力形式により、ヌル終端のある 200 バイトの文字列を使用できます。

次に、関数を呼び出す SAS コードを示します。この例では、DLL 名(KERNEL32)が明示的に指定されます(MODULE 属性が属性ファイルで使用されなかったため)。

```
filename sascbtbl "sascbtbl.dat";
data _null_;
  length path $200;
  n = modulen( '*i',
      "KERNEL32,GetTempPathA", 199, path );
  put n= path=;
run;
```

*注:* KERNEL32.DLL は、Windows によって提供される内部 DLL です。そのルーチンは Microsoft Win32 SDK に記述されています。

このコードによって、次のログメッセージが作成されます。

```
NOTE: Variable PATH is uninitialized.
N=7 PATH=C:\TEMP
```

この例ではバッファ長に 199 を使用しています。これは PATH が保持できる最大 200 文字のうち、1 文字はヌル終端用に予約されているためです。$CSTR200.入力形式では、制御が DATA ステップに戻る際に、ヌル終端および後続の文字が後置ブランクに置換されます。

### *引数を値として渡す*

この例では、KERNEL32.DLL の Win32 API の一部である Beep ルーチンを呼び出します。Beep の C プロトタイプは、次のとおりです。

```
BOOL Beep(DWORD dwFreq, DWORD dwDuration)
```

使用する属性テーブルは、次のとおりです。

```
routine Beep
  minarg=2
  maxarg=2
  stackpop=called
  callseq=byvalue
  module=kernel32;
arg 1 num format=pib4.;
arg 2 num format=pib4.;
```

両方の引数が値によって渡されるため、この例では、ROUTINE ステートメントに CALLSEQ=BYVALUE 属性が含まれます。各 ARG ステートメントに BYVALUE オプションを指定する必要はありません。

次に、Beep 関数の呼び出しに使用される SAS コードのサンプルを示します。

```
filename sascbtbl 'sascbtbl.dat';
data _null_;
  rc = modulen("*e","Beep",1380,1000);
run;
```

コンピュータのスピーカーはビープ音を出します。

### *PEEKCLONG を使用して返されたポインタにアクセスする*

次の例では、KERNEL32.DLL の Win32 API の一部である **lstrcat** ルーチンを使用します。**lstrcat** は 2 つの文字列を引数として受け入れ、連結し、連結した文字列へのポインタを返します。C プロトタイプは、次のとおりです。

```
LPTSTR lstrcat (LPTSTR lpszString1,
                LPCTSTR lpszString2);
```

次に、適切な属性テーブルを示します。

```
routine lstrcat
  minarg=2
  maxarg=2
  stackpop=called
  module=KERNEL32
  returns=ptr;
```

```
  arg 1 char format=$cstr200.;
  arg 2 char format=$cstr200.;
```

**lstrcat** を使用するには、返されたポインタによって参照されるデータに、SAS PEEKCLONG 関数を使用してアクセスする必要があります。**lstrcat** にアクセスするサンプル SAS プログラムを示します。

```
filename sascbtbl 'sascbtbl.dat';
data _null_;
  length string1 string2 conctstr $200;
  length charptr $20;
  string1 = 'This is';
  string2 = ' a test!';
  charptr=modulec('lstrcat',string1,string2);
  concatstr = peekclong(charptr,200);
  put concatstr=;
run;
```

次の出力はログに表示されます。

```
conctstr=This is a test!
```

MODULEN からの戻り値にもとづいて、ポインタ値は RC に格納されます。この例では、PEEKCLONG 関数を使用して指定の場所に 200 バイトが返され、$CSTR200.出力形式を使用してヌル終端を置換するブランク埋め込み文字列が作成されます。

PEEKLONG 関数に関する詳細は次を参照してください。“PEEKCLONG Function” (*SAS Functions and CALL Routines: Reference*) および “PEEKLONG Function” (*SAS Functions and CALL Routines: Reference*)

## 構造の使用

“SAS 変数を構造引数としてグループ化” (299 ページ) には FDSTART 属性を使用して複数の引数を 1 つの構造引数として DLL ルーチンに渡す方法が記述されています。GetClientRect 属性テーブルおよび対応する C 言語についても説明されています。この例では、属性テーブルを定義した後の GetClientRect 関数の起動方法を示します。

最も簡単な方法を使用すると、変数が初期化されていないという警告メッセージが生成されます。

```
filename sascbtbl 'sascbtbl.dat';
data _null_;
     hwnd=modulen('GetForegroundWindow');
     call module('GetClientRect',hwnd,
       left,top,right,bottom);
     put _all_;
     run;
```

この警告が表示されないようにするには、RETAIN ステートメントを使用して変数を 0 に初期化します。また MODULEN ステートメントでは、略記表現を使用して変数リストを指定することもできます。

```
data _null_;
     retain left top right bottom 0;
     hwnd=modulen('GetForegroundWindow');
     call module('GetClientRect',hwnd,
           of left--bottom);
     put _all_;
     run;
```

この例では、OF キーワードの後に、2 つのダッシュによって区切られた変数リストが指定されています。ログ出力は、アクティブウィンドウにより異なりますが、ほぼ次のとおりです。

```
HWND=3536768 LEFT=2 TOP=2 RIGHT=400
BOTTOM=587
```

### *DLL ルーチンを PROC IML から起動*

この例では、IML プロシジャ内で行列を引数として渡す方法を示します。ここでは、4x5 の行列を作成します。各セルは、10*x*+*y*+3 に設定されます。*x* は行番号、*y* は列番号です。たとえば、行 1 列 2 は(10*1)+2+3、すなわち 15 と設定します。

この例では、論理 TRYMOD DLL から複数のルーチンが呼び出されます。**changd** ルーチンを使用して、100*x*+10*y* を各要素に追加します。*x* は C の行番号(0 から 3)、*y* は C の列番号(0 から 4)です。**changd** の最初の引数は、合計に追加する量を示します。**changdx** ルーチンは **changd** と同じ処理を行いますが、changdx では転置行列が期待される点が異なります。**changi** ルーチンは **changd** と同じ処理を行いますが、changi では整数の行列が期待される点が異なります。**changix** ルーチンは **changdx** と同じ処理を行いますが、changix では整数が期待される点が異なります。

*注:* 3 つの引数の最大値は IML プロシジャから DLL ルーチンを呼び出すときに送ることができます。

この例では、4 つの行列 x1、x2、y1、y2 が、それぞれ MODULEIN で呼び出された後、すべて同じ値に設定されます。属性テーブルエントリは、次のとおりです。

```
routine changd module=trymod returns=long;
arg 1 input num format=rb8. byvalue;
arg 2 update num format=rb8.;
routine changdx module=trymod returns=long
  transpose=yes;
arg 1 input num format=rb8. byvalue;
arg 2 update num format=rb8.;
routine changi module=trymod returns=long;
arg 1 input num format=ib4. byvalue;
arg 2 update num format=ib4.;
routine changix module=trymod returns=long
  transpose=yes;
arg 1 input num format=ib4. byvalue;
arg 2 update num format=ib4.;
```

PROC IML のステップは、次のとおりです。

```
proc iml;
     x1 = J(4,5,0);
     do i=1 to 4;
        do j=1 to 5;
           x1[i,j] = i*10+j+3;
           end;
        end;
     y1= x1; x2 = x1; y2 = y1;
     rc = modulein('changd',6,x1);
     rc = modulein('changdx',6,x2);
     rc = modulein('changi',6,y1);
     rc = modulein('changix',6,y2);
     print x1 x2 y1 y2;
     run;
```

PRINT ステートメントの結果は、次のようになります。

```
 X1
 20        31        42        53        64
130       141       152       163       174
240       251       262       273       284
350       361       372       383       394
 X2
 20        31        42        53        64
130       141       152       163       174
240       251       262       273       284
350       361       372       383       394
 Y1
 20        31        42        53        64
130       141       152       163       174
240       251       262       273       284
350       361       372       383       394
 Y2
 20        31        42        53        64
130       141       152       163       174
240       251       262       273       284
350       361       372       383       394
```

*14 章*

# Windows 版での SAS/AF プログラマーのための特記注意事項

**SAS の表示画面と機能の制御** . . . . . **313**

**メイン SAS ウィンドウの制御** . . . . . **314**
概要:メイン SAS ウィンドウの制御 . . . . . 314
メイン SAS ウィンドウを制御する SAS システムオプション . . . . . 314
メイン SAS ウィンドウを制御する SAS コマンド . . . . . 315

**SAS から外部 DLL にアクセスする** . . . . . **318**

**カスタムツールバーコントロールの設計、保存、ロード** . . . . . **318**

**SAS/AF アプリケーションの自動起動** . . . . . **319**

**独自のロゴとアイコンを SAS/AF アプリケーションに関連付ける** . . . . . **319**

## SAS の表示画面と機能の制御

Windows 版 SAS では、SAS/AF のプログラミングの際に、メイン SAS ウィンドウの表示画面と機能を詳細に制御することができます。次のことが可能です。

- SAS システムオプションとウィンドウ環境コマンドを使用して、メイン SAS ウィンドウの表示を制御することができます。
- DATA ステップや SAS Component Language(SCL)コマンドを使用して、ダイナミックリンクライブラリ(DLL)を呼び出すことができます。
- カスタムツールバーコントロールを設計、保存、ロードすることができます。
- SAS の起動後ただちに SAS/AF プログラムを呼び出すことができます。
- 特定の SAS/AF アプリケーションの実行に必要な最低限の SAS ファイルのサブセットを配布することができます。
- 独自のロゴとアイコンを SAS/AF アプリケーションに関連付けることができます。
- SCL コードを使用して、他のユーザーに電子メールを送信することができます。
- Web ブラウザを呼び出して、ドキュメントをオンライン表示することができます。

# メイン SAS ウィンドウの制御

## *概要:メイン SAS ウィンドウの制御*

SAS システムオプションとウィンドウ環境コマンドを使用して、メイン SAS ウィンドウの表示画面と機能を変更することができます。エンドユーザーにはそれが SAS だとわからないほど大幅に変更することもできます。

## *メイン SAS ウィンドウを制御する SAS システムオプション*

メイン SAS ウィンドウを制御するシステムオプションについては、次の表を参照してください。

***表 14.1*** *メイン SAS ウィンドウに対する SAS システムオプション*

| オプション | 説明 |
|---|---|
| AWSCONTROL | タイトルバー、システムメニュー、最小化および最大化ボタンなどのシステムコントロールを、メイン SAS ウィンドウから削除します。 |
| AWSDEF | SAS が初期化された際の、メイン SAS ウィンドウの位置と大きさを指定します。 |
| AWSMENU | メイン SAS ウィンドウのメニューバーを表示するかどうかを指定します。 |
| AWSMENUMERGE | メインメニューに、Windows 固有のメニュー項目を組み込むかどうかを指定します。 |
| AWSTITLE | メイン SAS ウィンドウのタイトルバーに表示されるテキストを指定します。 |
| ICON | SAS の初期化時に、メイン SAS ウィンドウを最小化します。 |
| REGISTER | **AWS のファイル**メニューにメニュー項目として含める Windows プログラムを指定します。 |
| SASCONTROL | システムコントロールと最小化および最大化ボタンを、SAS アプリケーションウィンドウから削除します。 |
| SOLUTIONS | メインメニューの**ソリューション**メニューを表示したり、非表示したりにします。 |
| SPLASH | SAS の起動時にロゴ(スプラッシュ画面)を表示するかどうかを指定します。 |
| SPLASHLOC | SAS の起動時に表示するスプラッシュ画面のビットマップの場所を指定します。 |

| オプション | 説明 |
|---|---|
| TOOLSMENU | メインメニューの**ツール**メニューを表示したり、非表示したりにします。 |
| VIEWMENU | メインメニューの**表示**メニューを表示したり、非表示にしたりします。 |
| WEBUI | Web 拡張(マウスポインタをオブジェクトに合わせるとオブジェクトが選択され、シングルクリックによりデフォルトの操作が呼び出される)を有効にするかどうかを指定します。 |
| WINDOWSMENU | メインメニューに **Window** メニューを表示したり、非表示にしたりします。WINDOWSMENU は、NOAWSMENUMERGE を指定した場合にのみ有効です。 |

## メイン SAS ウィンドウを制御する SAS コマンド

表 14.2 (315 ページ) SAS メインウィンドウの表示と動作を制御するために使用できる SAS コマンドを表示します。

***表 14.2*** *メイン SAS ウィンドウに対する SAS ウィンドウ環境コマンド*

| オプション | 説明 |
|---|---|
| AWSMAXIMIZE | メイン SAS ウィンドウを最大化します。 |
| AWSMINIMIZE | メイン SAS ウィンドウを最小化します。 |
| AWSRESTORE | メイン SAS ウィンドウを元のサイズに戻します。 |
| COLOR | アプリケーションウィンドウの構成要素の色を設定します。 |
| COMMAND | コマンドバーまたはダイアログボックスの外観を制御します。 |
| DLGABOUT | **著作権情報** ダイアログボックスを呼び出します。 |
| DLGCONVERT | 選択された OLE オブジェクトの**変換** ダイアログボックスを呼び出します。 |
| DLGCDIR | **フォルダの変更** ダイアログボックスを呼び出します。 |
| DLGENDR | **SAS の終了確認** ダイアログボックスを呼び出します。 |
| DLGFIND | **検索**ダイアログボックスを呼び出します。 |
| DLGFONT | **フォントの選択**ダイアログボックスを呼び出します。 |

| オプション | 説明 |
|---|---|
| DLGLIB | **ライブラリ** ダイアログボックスを呼び出します。 |
| DLGLINKS | **OLE リンクの設定** ダイアログボックスを呼び出します。 |
| DLGOPEN | プログラムエディタの**ファイルを開く** ダイアログボックスを呼び出します。 |
| DLGPAGESETUP | **ページ設定**ダイアログボックスを呼び出します。 |
| DLGPREF | **プリファレンス**ダイアログボックスを呼び出します。 |
| DLGPRT | **印刷** ダイアログボックスを呼び出します。 |
| DLGPRTPREVIEW | **印刷プレビュー**ウィンドウを呼び出します。 |
| DLGPRTSETUP | **印刷設定**ダイアログボックスを呼び出します。 |
| DLGREPLACE | **置換**ダイアログボックスを呼び出します。 |
| DLGRUN | **実行** ダイアログボックスを呼び出します。 |
| DLGSAVE | **名前を付けて保存** ダイアログボックスを呼び出します。 |
| DLGSMAIL | **電子メール** ダイアログボックスを呼び出します。 |
| FILEOPEN | 拡張エディタの**ファイルを開く** ダイアログボックスを呼び出します。 |
| NEXTWIND | 開いている次の SAS ウィンドウを表示します。 |
| PMENU | ウィンドウ環境でコマンド行の表示/非表示を切り替えます。 |
| PREVWIND | 開いている前の SAS ウィンドウを表示します。 |
| RESHOW | 現在開いている SAS ウィンドウを再表示します。 |
| TOOLCLOSE | ツールボックスまたはツールバーを閉じます。 |
| TOOLEDIT | ツールバーまたはツールボックスの**カスタマイズ**ダイアログボックスを呼び出します。 |
| TOOLLARGE | ツールバーまたはツールボックスのボタンのサイズを切り替えます。 |
| TOOLLOAD | 指定された構成のツールボックスまたはツールバーを開きます。 |
| TOOLSWITCH | アクティブウィンドウに割り当てられているツールバーまたはツールボックスを自動的にロードするかどうかを切り替えます。 |

| オプション | 説明 |
|---|---|
| TOOLTIPS | ツールチップの表示/非表示を切り替えます。 |
| WATTACH | SAS を使用して送信する電子メールにアクティブウィンドウの内容を添付するかどうかを切り替えます。 |
| WDOCKVIEW | ドッキング可能ウィンドウを使用可能にします。 |
| WEDIT | 新しい**拡張エディタ**ウィンドウを呼び出します。 |
| WEMAILFMT | 電子メールメッセージに添付するテキストウィンドウの形式(.RTF または.TEXT)を指定します。 |
| WHIDECURSOR | カーソルの表示を抑制します。 |
| WHSBAR | 水平スクロールバーの表示/非表示を切り替えます。 |
| WINSERT | 挿入モードの 有効化/無効化を切り替えます。 |
| WMENUPOP | SAS アプリケーションウィンドウのポップアップメニューを有効または無効にします。 |
| WMRU | **ファイル**メニューに表示されるリストに保持するファイル名の数を指定します。 |
| WNEWTITLE | アクティブウィンドウの内容とタイトルを消去します。 |
| WNEXTEDIT | **拡張エディタ**ウィンドウから別の **拡張エディタ**ウィンドウをアクティブにします。 |
| WPOPUP | アクティブウィンドウのポップアップメニューを表示します。 |
| WSCREENTIPS | スクリーンチップの表示/非表示を切り替えます。 |
| WSTATUSLN | ステータスバーの表示/非表示を切り替え、各表示領域の割合を制御します。 |
| WUNDO | 直前の編集操作を取り消します。 |
| WVSBAR | 垂直スクロールバーの表示/非表示を切り替えます。 |
| WWINDOWBAR | メイン SAS ウィンドウの下部に、ウィンドウバーを表示します。 |
| ZOOM | アクティブな SAS アプリケーションウィンドウを最大化します。 |

## SAS から外部 DLL にアクセスする

DATA ステップまたは SCL 内で SAS MODULE 群の関数を使用することで、外部ダイナミックリンクライブラリ(DLL)に格納されたルーチンにアクセスすることができます。作成した DLL や購入した DLL にアクセスする際には、この方法を使用します。また、OS の DLL にもアクセスすることができます。

外部 DLL にアクセスするには、次の情報が必要です。

- DLL の名前
- 関数名または序数
- 関数の引数の 記述
- リターンコードの記述

***注意:***

**外部 DLL へは、経験のあるプログラマのみアクセスするようにしてください。**外部 DLL にアクセスすると、コンピュータの制御が SAS から DLL 関数に渡されます。誤って実行した場合や、DLL 関数の信頼性が低い場合、データが失われる可能性や、コンピュータの再起動が必要になることがあります。(同時に起こることもあります)。

外部 DLL ルーチンにアクセスする一般的な方法は、次のとおりです。

1. アクセスする DLL ルーチンを記述するテキストファイルを作成します。期待される引数と、戻り値がある場合は、このテキストファイルに含めます。この属性ファイルは、特別なフォーマットである必要があります。
2. FILENAME ステートメントを使用して、作成した属性ファイルに SASCBTBL ファイル参照名を割り当てます。
3. DATA ステップか SCL コードで、MODULE、MODULEN または MODULEC を使用して DLL ルーチンを呼び出します。使用する特定の関数は、期待される戻り値の型(なし、数値、文字)によって異なります。(また、PROC IML ステップ内で MODULEI、MODULEIN 、MODULEIC を使用することもできます)

*注:* MODULE ルーチンは、特に SASCBTBL ファイル、SAS 出力形式と入力形式、その他の SAS ルーチンとともに使用した場合に柔軟で強力なツールになります。このため、外部ルーチンを呼び出すときには、十分な注意が必要です。誤って実行した場合、データが失われる可能性や、コンピュータの再起動が必要になります。

SAS 内から DLL にアクセスするための詳細な情報については、13 章, “Windows 版 SAS から外部 DLL にアクセスする” (291 ページ) を参照してください。

## カスタムツールバーコントロールの設計、保存、ロード

SAS/AF アプリケーションでは、カスタムツールバーを構成することで、ユーザーにとって使いやすいツールを提供することができます。これらのツールは、ウィンドウ環境コマンドに割り当てることが可能です。カスタムツールバーの作成と保存についての詳細な情報については、“ツールバーのカスタマイズ” (67 ページ) を参照してください。

SAS/AF アプリケーションを他のマシンに配布する場合は、必ずカスタムツールの構成が格納されたカタログエントリを含めてください。

デフォルトでは、ツール切り替え機能が有効化され、SAS/AF アプリケーションでカスタムツールバーを使用できます。ツール切り替え機能は、TOOLSWITCH OFF コマンドを発行することで無効にすることができます。

## SAS/AF アプリケーションの自動起動

SAS 提供のシステムオプション INITCMD を使用して、SAS/AF プログラムを自動的に起動することができます。このオプションを使用すると、[プログラムエディタ]、[ログ]、[アウトプット]ウィンドウは作成されませんが、指定した SAS/AF アプリケーションとウィンドウ環境コマンドが実行されます。

INITCMD オプションの一般的な構文は次のとおりです。

-INITCMD "*af-command*" <*DM-command-1…DM-command-n*>

*af-command* は AF アプリケーションを起動するコマンド、*DM-command-1* から *DM-command-n* まではウィンドウ環境コマンドを表します。

たとえば、次のようにオプションを指定すると、SAS/AF アプリケーションを起動し、カスタムツールバーをロードすることができます。

```
-initcmd "AF c=mylib.myapp.myfirst.frame;
  toolload bar mylib.myapp.profile.toolbox"
```

INITCMD システム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。

## 独自のロゴとアイコンを SAS/AF アプリケーションに関連付ける

SAS 提供のアイコンのかわりに、独自のロゴ画面とアイコンを使用することができます。

*注:* このための手順には、リソースの作成と、独自のダイナミックリンクライブラリ(DLL)の作成が含まれます。DLL の作成の詳細については、Microsoft Win32 Software Development Kit を参照してください。

SAS の起動時に独自のロゴを表示するには、次の操作を実行します。

1. 表示するロゴを作成し、Windows ビットマップ(BMP ファイル拡張子)として保存するかまたはリソースとしてコンパイルし、DLL を作成します。
2. SAS の起動時に、SPLASHLOC オプションにビットマップを含むファイルのフルパス名を指定します。ビットマップが DLL の場合は、リソース番号も指定する必要があります。デフォルトのリソース番号は 1 です。詳細については、次を参照してください。"SPLASHLOC システムオプション:Windows" (571 ページ)

SAS を起動すると、指定したロゴが表示されます。

SAS/AF アプリケーションで独自のアイコンを使用するには、次の操作を実行します。

1. SAS の起動時に、USERICON システムオプションを使用して、目的のアイコンが格納されたリソースファイルを指定します。リソースファイルをコンパイルするには、Windows ソフトウェア開発ツールを使用します。USERICON オプションの詳細につ

いては、次を参照してください。“USERICON システムオプション:Windows” (585 ページ) .

2. SAS/AF ソフトウェアを使用して FRAME エントリを作成します。
3. アイコンを表示するプッシュボタンの buttonStyle 属性を選択します。プッシュボタンの iconStyle 属性に[大きいアイコン]を選択すると、アイコンを拡大できます。
4. アイコンを表示するプッシュボタンのアイコン属性値の省略記号をクリックします。**アイコンの選択**ウィンドウが表示されます。アイコンカテゴリが、ウィンドウの上部に表示されます。下矢印をクリックして、[ユーザーアイコン]カテゴリを選択します。リソースファイルに含まれるユーザー定義アイコンが表示されます。プッシュボタンのアイコンを選択します。

# 3 部

# Windows 版の SAS 言語機能

15 章
**Windows 版に固有のデータセットオプション** . . . . . 323

16 章
**Windows 版に固有の SAS コマンド** . . . . . 325

17 章
**Windows 版に固有の SAS 出力形式** . . . . . 377

18 章
**Windows 版に固有の SAS 関数と CALL ルーチン** . . . . . 389

19 章
**Windows 版に固有の SAS 入力形式** . . . . . 415

20 章
**Windows 版に固有の SAS プロシジャ** . . . . . 425

21 章
**Windows 版に固有の SAS ステートメント** . . . . . 443

22 章
**Windows 版に固有の SAS システムオプション** . . . . . 471

23 章
**Windows 版での変数の長さと精度** . . . . . 595

24 章
**Windows 版に固有の SAS マクロ機能** . . . . . 599

*15 章*

# Windows 版に固有のデータセットオプション

**Windows 版に固有の SAS データセットオプション** . . . . . **323**

**ディクショナリ** . . . . . **323**
SGIO データセットオプション:Windows . . . . . 323

## Windows 版に固有の SAS データセットオプション

データセットオプションには、SAS データセットにのみ適用される操作を指定します。データセットオプションを使用して、次の操作を実行できます。

- 変数の名前を変更する
- 最初または最後の *n* オブザベーションを選択して処理します。
- 処理または出力データセットから変数を削除する
- データセットのパスワードを指定する

## ディクショナリ

### SGIO データセットオプション:Windows

データセットへの分散/集合入出力機能を有効化します。

**該当要素:** DATA ステップと PROC ステップ

#### 構文

**SGIO**= YES| NO

#### *必須引数*

**YES**
SAS データセットの scatter-read / gather-write 機能を有効化します。scatter-read / gather-write 機能は、SAS セッションが終了するまで有効です。

**NO**
SAS データセットの scatter-read / gather-write 機能を有効化しません。

## 詳細

SAS コード内で参照される SAS I/O ファイルであれば、SGIO データセットオプションを指定できます。SGIO を使用して処理する SAS I/O が多い場合は、SGIO システムオプションを指定し、SGIO を有効化しないデータセットに対して SGIO を無効化します (SGIO=no)。

## 比較

SGIO データセットオプションを指定すると、そのデータセットは SGIO を使用して処理されます。SGIO システムオプションで、すべての SAS ファイルが SGIO を用いて処理するのを指定します。

## 例

### *例 1: SGIO オプション*

SAS ジョブで参照される SAS I/O ファイルであれば、SGIO データセットオプションを指定できます。DATA ステップでの単純な指定例を次に示します。

```
data mike(sgio=yes);
    input x y z;
    datalines;
1 2 3
run;
```

### *例 2: SGIO オプション*

複雑な指定例を次に示します。

```
data master(sgio=yes)
   merge daily1(sgio=yes) daily2(sgio=no) daily3(sgio=yes)
... more SAS statements ...
run;
```

*注:* 詳細については以下の情報を参照してください。SGIO の使用

*16 章*

# Windows 版に固有の SAS コマンド

**Windows 版に固有の SAS コマンド** . . . . . **326**
Windows 版に固有の SAS コマンドの概要 . . . . . 326
Windows 動作環境でサポートされないコマンド . . . . . 327

**ディクショナリ** . . . . . **327**
AUTOSCROLL コマンド:Windows . . . . . 327
AWSMAXIMIZE コマンド:Windows . . . . . 328
AWSMINIMIZE コマンド:Windows . . . . . 328
AWSRESTORE コマンド:Windows . . . . . 329
CAPS コマンド:Windows . . . . . 329
COLOR コマンド:Windows . . . . . 330
COMMAND コマンド:Windows . . . . . 330
CUT コマンド:Windows . . . . . 332
DLGABOUT コマンド:Windows . . . . . 332
DLGCDIR コマンド:Windows . . . . . 333
DLGCOLUMNSIZE コマンド:Windows . . . . . 333
DLGCOLUMNSORT コマンド:Windows . . . . . 333
DLGCONVERT コマンド:Windows . . . . . 334
DLGENDR コマンド:Windows . . . . . 334
DLGFIND コマンド:Windows . . . . . 334
DLGFONT コマンド:Windows . . . . . 335
DLGLIB コマンド:Windows . . . . . 335
DLGLINKS コマンド:Windows . . . . . 336
DLGOPEN コマンド:Windows . . . . . 336
DLGPAGESETUP コマンド:Windows . . . . . 338
DLGPREF コマンド:Windows . . . . . 338
DLGPRT コマンド:Windows . . . . . 339
DLGPRTPREVIEW コマンド:Windows . . . . . 340
DLGPRTSETUP コマンド:Windows . . . . . 340
DLGREPLACE コマンド:Windows . . . . . 341
DLGRUN コマンド:Windows . . . . . 341
DLGSAVE コマンド:Windows . . . . . 342
DLGSMAIL コマンド:Windows . . . . . 344
FILE コマンド:Windows . . . . . 344
FILEOPEN コマンド:Windows . . . . . 346
FILL コマンド:Windows . . . . . 346
GSUBMIT コマンド:Windows . . . . . 347
HOME コマンド:Windows . . . . . 347
ICON コマンド:Windows . . . . . 348
INCLUDE コマンド:Windows . . . . . 348
PMENU コマンド:Windows . . . . . 350
SAVE コマンド:Windows . . . . . 351

STORE コマンド:Windows . . . . . 351
SUBTOP コマンド:Windows . . . . . 352
TOOLCLOSE コマンド:Windows . . . . . 353
TOOLEDIT コマンド:Windows . . . . . 353
TOOLLARGE コマンド:Windows . . . . . 353
TOOLLOAD コマンド:Windows . . . . . 354
TOOLSWITCH コマンド:Windows . . . . . 355
TOOLTIPS コマンド:Windows . . . . . 355
WATTACH コマンド:Windows . . . . . 356
WATTENTION コマンド:Windows . . . . . 357
WAUTOSAVE コマンド:Windows . . . . . 357
WBROWSE コマンド:Windows . . . . . 358
WCOPY コマンド:Windows . . . . . 358
WCUT コマンド:Windows . . . . . 359
WDOCKVIEW コマンド:Windows . . . . . 359
WDOCKVIEWMINIMIZE コマンド:Windows . . . . . 360
WDOCKVIEWRESIZE コマンド:Windows . . . . . 360
WDOCKVIEWRESTORE コマンド:Windows . . . . . 361
WEDIT コマンド:Windows . . . . . 361
WEMAILFMT コマンド:Windows . . . . . 362
WEXITSAVE コマンド:Windows . . . . . 362
WFILE コマンド:Windows . . . . . 363
WHIDECURSOR コマンド:Windows . . . . . 363
WHSBAR コマンド:Windows . . . . . 364
WINSERT コマンド:Windows . . . . . 365
WMENUPOP コマンド:Windows . . . . . 365
WMRU コマンド:Windows . . . . . 366
WNAVKEYUNMARK コマンド:Windows . . . . . 367
WNEWTITLE コマンド:Windows . . . . . 367
WNEXTEDIT コマンド:Windows . . . . . 368
WPASTE コマンド:Windows . . . . . 368
WPGM コマンド:Windows . . . . . 368
WPOPUP コマンド:Windows . . . . . 369
WRTFSAVE コマンド:Windows . . . . . 369
WSCREENTIPS コマンド:Windows . . . . . 370
WSTATUSLN コマンド:Windows . . . . . 371
WUNDO コマンド:Windows . . . . . 372
WVSBAR コマンド:Windows . . . . . 372
WWINDOWBAR コマンド:Windows . . . . . 373
X コマンド:Windows . . . . . 374
ZOOM コマンド:Windows . . . . . 374

# Windows 版に固有の SAS コマンド

## *Windows 版に固有の SAS コマンドの概要*

対話型 SAS セッション中、コマンドバー、SAS ウィンドウのコマンド行、キーボード、ツールバーからコマンドを発行できます。SAS では、セッションの操作や特定のタスクの実行に利用できるさまざまなコマンドがサポートされています。多くのコマンドは、SAS メニューや SAS ウィンドウからでも実行できる機能を呼び出すもう 1 つの手段に過ぎません。操作に慣れてくると、コマンドを使用する方が作業効率が高くなる場合があります。また、タスクに必要なパラメータが SAS のインターフェイスでサポートされるものと異なる場合、コマンドを使用することでタスクをより柔軟に実行できます。

SAS ウィンドウ環境のほとんどのコマンドについて、SAS ヘルプとドキュメントで説明しています。ここで解説するコマンドには、Windows 版固有の構文または動作があります。

コマンドの発行の詳細については、“SAS コマンドの発行” (42 ページ) を参照してください。

### *Windows 動作環境でサポートされないコマンド*

次の SAS コマンドは、Windows 環境でサポートされていません。

| | | |
|---|---|---|
| PCLEAR | SMARK | WMOVE |
| PLIST | WDRAG | WSHRINK |
| SCROLLBAR | WGROW | |

Windows の機能を使用したほうが効率的なため、これらのコマンドは Windows 環境でサポートされていません。たとえば、SCROLLBAR コマンドとウィンドウサイズ変更のコマンドは、スクロールバーとウィンドウサイズを変更するバーがグラフィカルユーザーインターフェイスに統合されている Windows 環境では必要ありません。

# ディクショナリ

## AUTOSCROLL コマンド:Windows

ログウィンドウとアウトプットウィンドウで出力を表するときのスクロールの頻度を指定します。

**Windows 固有:** デフォルト値

### 構文

**AUTOSCROLL** <*number-of-lines* | PAGE | MAX>

### 詳細

Windows 版の場合、**アウトプット**ウィンドウでの AUTOSCROLL コマンドのデフォルト値は 0 です。つまり、ステートメントの実行中は、アウトプットウィンドウに出力は表示されません。これにより、最大のパフォーマンスが得られます。**ログ**ウィンドウでのデフォルト値は、SAS 起動時の**ログ**ウィンドウの行数の半分の値です。

スクロールにより、プログラムの実行時間が長くなる可能性があります。[ログ]ウィンドウと[アウトプット]ウィンドウのスクロール頻度を少なくすると、プログラムの実行速度が上がります。

**プリファレンス** ダイアログボックスの**詳細** ページでもスクロールオプションを設定できます。

### 関連項目:

- “セッションプリファレンスについて ” (58 ページ)
- SAS のヘルプとドキュメント内の“AUTOSCROLL Command”

## AWSMAXIMIZE コマンド:Windows

メイン SAS ウィンドウを最大化します。

**Windows 固有:** すべて

### 構文

**AWSMAXIMIZE** <ON | OFF>

#### *引数なし*

メイン SAS ウィンドウを最大サイズまたは元のサイズに切り替えます。

#### *オプション引数*

**ON**
メイン SAS ウィンドウを最大化します。このオプションには、[最大化]ボタンをクリックするのと同じ効果があります。

**OFF**
メイン SAS ウィンドウを元のサイズに戻します。

### 詳細

AWSMAXIMIZE コマンドを使用すると、メイン SAS ウィンドウを Windows デスクトップ全体に拡大できます。

## AWSMINIMIZE コマンド:Windows

メイン SAS ウィンドウを最小化します。

**Windows 固有:** すべて

### 構文

**AWSMINIMIZE** <ON | OFF>

#### *引数なし*

メイン SAS ウィンドウを最小サイズまたは元のサイズに切り替えます。

#### *オプション引数*

**ON**
メイン SAS ウィンドウを最小化します。このオプションには、[最小化]ボタンをクリックするのと同じ効果があります。

**OFF**
メイン SAS ウィンドウを元のサイズに戻します。

## AWSRESTORE コマンド:Windows

メイン SAS ウィンドウを元のサイズに戻します。

**Windows 固有:** すべて

### 構文

**AWSRESTORE** <ON | OFF>

***引数なし***

メイン SAS ウィンドウを最大サイズまたは元のサイズに切り替えます。

***必須引数***

**ON**
メイン SAS ウィンドウを元のサイズに戻します。このオプションには、メイン SAS ウィンドウのタイトルバーメニューで**元のサイズに戻す**を選択するのと同じ効果があります。

**OFF**
メイン SAS ウィンドウをデフォルトのサイズに戻します。

### 詳細

AWSRESTORE コマンドまたは AWSMAXIMIZE コマンドを使用して、メイン SAS ウィンドウを最大サイズまたは元のサイズに切り替えることができます。

## CAPS コマンド:Windows

大文字で書き込むかどうかを指定します。

**Windows 固有:** すべて

### 構文

**CAPS**

### 詳細

CAPS コマンドを使用すると、これから入力するテキストやウィンドウで変更したテキストが、すべて大文字に変換されます。

Windows 版では、カーソルを行の外に移動したり*または* Enter を押したりすると、文字が大文字に変換されます。

### 関連項目:

SAS ヘルプとドキュメント内の“CAPS Command”

## COLOR コマンド:Windows

選択したウィンドウ部分の色と強調表示を指定します。

**Windows 固有:** 影響を受けるウィンドウの構成要素

### 構文

**COLOR** *field-type* <*色* | NEXT <*highlight*>>

#### *必須引数*

***field-type***
色を変更するウィンドウの領域またはテキストの種類を指定します。

#### *オプション引数*

***色***
ウィンドウまたは選択したウィンドウ部分の色を指定します。

**NEXT**
色を、次に使用できる色に変更します。NEXT の値は、最後に入力した色に基づいて決まります。色の順序は、使用しているモニタによって異なります。

***highlight***
強調表示属性を指定します。

### 詳細

Windows 版では、COLOR コマンドを使用して、枠、メニューバー、ポップアップメニューの背景、タイトルバーの各画面構成要素の色を変更できません。これらの画面構成要素の色を変更するには、Windows のコントロールパネルを使用します。

さらに、Windows のウィンドウ構成要素では、HIGHLIGHT 強調属性と BLINK 強調属性はサポートされていません。

### 関連項目:

SAS ヘルプとドキュメント内の Other COLOR コマンド

## COMMAND コマンド:Windows

コマンドバーのオプションを指定します。

**Windows 固有:** 有効なオプション

### 構文

**COMMAND** <<WINDOW <"*title*"> | BAR <SORT=MCU | MRU> <FOCUS> <MAX=*max-commands*><AUTOCOMPLETE | NOAUTOCOMPLETE>> | CLOSE>

#### *引数なし*

アクティブウィンドウのコマンド行のオン／オフを切り替えます。

### *オプション引数*

**WINDOW <"*title*">**
デスクトップ上の任意の場所に移動できる独立したウィンドウとして、コマンドバーを表示します。"*title*"引数の指定は任意です。また、二重引用符で囲んで指定する必要があります。*title* を指定すると、*title* というタイトルを持つコマンドウィンドウが表示されます。

**BAR**
コマンドバーをメニューバーの下の固定位置に表示します。

**SORT=MCU|MRU**
コマンドバーのドロップダウンリストでコマンドを表示する順序を指定します。使用頻度の高い順序(MCU)または最後に使用した順序(MRU)でコマンドを並べることができます。

コマンドに WINDOW 引数または BAR 引数を指定してから、SORT 引数を指定する必要があります。

**FOCUS**
コマンドバーにウィンドウのフォーカスを置きます。

**MAX=*max-commands***
コマンドバーのドロップダウンリストに"登録する"コマンドの最大数を指定します。有効な値は 0 から 50 です。

コマンドに WINDOW 引数または BAR 引数を指定してから、MAX 引数を指定する必要があります。

**AUTOCOMPLETE | NOAUTOCOMPLETE**
コマンドバーにコマンドを入力する際に、以前入力したコマンドに一致する場合に残りが自動的に入力されるかどうかを指定します。

コマンドに WINDOW 引数または BAR 引数を指定してから、AUTOCOMPLETE 引数を指定する必要があります。

**CLOSE**
コマンドバーを閉じることを指定します。

## 詳細

これらのオプションの一部は、**ツールのユーザー設定**ダイアログボックスを使用して設定できます。ただし、**コマンド**ウィンドウのタイトルは、このコマンドでしか指定できません。

コマンドバーが閉じているときに COMMAND FOCUS を発行すると、次のように動作します。

- コマンドバーは、前回閉じたときの状態(メイン SAS ウィンドウにドッキングした状態、または独立したウィンドウとしてドッキングが解除された状態)で、開かれます。
- ウィンドウフォーカスがコマンドバーに置かれます。

## 関連項目:

- SAS ヘルプとドキュメント内の“COMMAND コマンド”
- “コマンドバーを使用してコマンドの発行” (43 ページ)
- “表示プリファレンス” (60 ページ)
- “DLGPREF コマンド:Windows” (338 ページ)

# CUT コマンド:Windows

選択したテキストをウィンドウから切り取ります。

**Windows 固有:** サポートされているオプション

## 構文

**CUT** <LAST | ALL>

### *オプション引数*

**LAST**
テキスト範囲が複数選択されている場合、最後に選択したテキストを切り取り、それ以外の選択をすべて解除します。複数のテキスト範囲が選択されている場合に 1 つの範囲を切り取るには、LAST と ALL のどちらかの引数を指定する必要があります。

**ALL**
複数のテキスト範囲が選択されている場合、現在選択されているすべてのテキストを切り取ります。

## 詳細

CUT コマンドは、選択されているテキストを現在のウィンドウから削除し、Windows クリップボードに格納します。

Windows 版の CUT コマンドでは、APPEND オプションと BUFFER=オプションはサポートされません。

## 関連項目:

- SAS ヘルプとドキュメントの“CUT コマンド”
- “クリップボードの使用” (55 ページ)
- “WCUT コマンド:Windows” (359 ページ)

# DLGABOUT コマンド:Windows

**SAS システムについて**ダイアログボックスを開きます。

**Windows 固有:** すべて

## 構文

**DLGABOUT**

## 詳細

メニューから **SAS システムについて** ダイアログボックスにアクセスするには、[ヘルプ] メニューを選択し、**SAS 9 について**を選択します。

## DLGCDIR コマンド:Windows

**フォルダの変更**ダイアログボックスを開きます。

**Windows 固有:** すべて

### 構文

**DLGCDIR**

### 詳細

**フォルダの変更**ダイアログボックスから、新しい作業フォルダを選択できます。

### 関連項目:

- “SAS の現在のフォルダの変更” (41 ページ)
- “SASINITIALFOLDER システムオプション:Windows” (558 ページ)

## DLGCOLUMNSIZE コマンド:Windows

**列の設定**ダイアログボックスを開きます。

**Windows 固有:** すべて

### 構文

**DLGCOLUMNSIZE**

### 詳細

SAS ウィンドウに、詳細を表示する一覧ビューが含まれる場合、**列の設定** ダイアログボックスを使用して、列のサイズをピクセル単位で指定できます。**SAS のエクスプローラ**ウィンドウは、一覧ビューが含まれるウィンドウの一例です。

## DLGCOLUMNSORT コマンド:Windows

**列の並べ替え**ダイアログボックスを開きます。

**Windows 固有:** すべて

### 構文

**DLGCOLUMNSORT**

### 詳細

SAS ウィンドウに一覧ビューが含まれる場合、**列の並べ替え** ダイアログボックスを使用して列を並べ替えられます。**SAS のエクスプローラ**ウィンドウは、一覧ビューが含まれるウィンドウの一例です。

## DLGCONVERT コマンド:Windows

**変換**ダイアログボックスを開きます。

**Windows 固有:** すべて

### 構文

**DLGCONVERT**

### 詳細

このコマンドは、**SAS/AF の BUILD** ウィンドウで OLE オブジェクトを選択しているときに使用できます。**変換** ダイアログボックスでは、選択した OLE オブジェクトの種類を可能な種類で変換できます。この操作は、そのオブジェクトに対して、どの種類が OLE サーバーアプリケーションでサポートされているかによって異なります。

### 関連項目:

“リンクされた OLE オブジェクトを使用する” (240 ページ)

## DLGENDR コマンド:Windows

**終了**ダイアログボックスを開きます。

**Windows 固有:** すべて

### 構文

**DLGENDR**

### 詳細

**終了**ダイアログボックスでは、SAS を終了するかどうかを確認します。ダイアログボックスで **OK** を選択すると、SAS セッションが終了します。**終了の確認をする** を**プリファレンス**ダイアログボックスの**全般**タブページで選択しない場合は、DLGENDR コマンドを入力すると SAS が終了します。

### 関連項目:

“セッションプリファレンスについて ” (58 ページ)

## DLGFIND コマンド:Windows

**検索**ダイアログボックスを開きます。

**Windows 固有:** すべて

### 構文

**DLGFIND**

### 詳細

**検索** ダイアログボックスでは、テキスト文字列を検索できます。

### 関連項目:

"DLGREPLACE コマンド:Windows" (341 ページ)

## DLGFONT コマンド:Windows

**フォント**ダイアログボックスを開きます。

**Windows 固有:** すべて

### 構文

**DLGFONT**

### 詳細

**フォントの選択**ダイアログボックスでは、SAS ウィンドウ環境のフォントを動的に変更できます。

## DLGLIB コマンド:Windows

**ライブラリ**ダイアログボックスを開きます。

**Windows 固有:** すべて

### 構文

**DLGLIB**

### 詳細

**ライブラリ** ダイアログボックスでは、SAS ライブラリを定義または変更します。DLGLIB コマンドは、これまでの SAS との互換性を維持するためにサポートされています。

**SAS のエクスプローラ**ウィンドウで、SAS ライブラリの表示、割り当てを行えます。

また、起動時に **自動的に割り当てる** をクリックして、新しい SAS セッションを開くたびにライブラリを開始することもできます。

### 関連項目:

SAS エクスプローラウィンドウを使用しての SAS ライブラリの管理についての詳細な情報のための SAS ヘルプとドキュメント

## DLGLINKS コマンド:Windows

**リンク**ダイアログボックスを開きます。

**Windows 固有:** すべて

### 構文

**DLGLINKS**

### 詳細

DLGLINKS コマンドで、**リンク**ダイアログボックスを開いて、リンクオブジェクトを更新できます。

リンクオブジェクトがない場合、このウィンドウには、リンクされたオブジェクトがありません。のメッセージが SAS メッセージログに表示されます。

### 関連項目:

- “リンクされた OLE オブジェクトを使用する” (240 ページ)
- “リンクされた OLE オブジェクトを使用する” (240 ページ)

## DLGOPEN コマンド:Windows

デフォルトエディタの**ファイルを開く**ダイアログボックスを開きます。

**Windows 固有:** すべて

### 構文

**DLGOPEN** <LONGFILTER="*filters*" | FILTER='*filters*'< REPLACE> > <
SUBMIT | NOSUBMIT>
<IMPORT> <VERIFY> <ALTCMD='*command*'>

#### *引数なし*

デフォルト設定で **ファイルを開く** ダイアログボックスを開きます。

#### *オプション引数*

**LONGFILTER="*filters*" | FILTER='*filters*'**

LONGFILTER="*filters*" は、**ファイルを開く** ダイアログボックスに表示するファイルの検索条件として使用するファイルのフィルタを 1 つ以上指定します。引数リスト内の 1 つ目のフィルタはデフォルトフィルタで、検索条件として使用されます。引数リストのすべてのフィルタが、**ファイルの種類**コンボボックスのフィルタリストに追加されます。さらにファイルの種類を検索するには、**ファイルの種類**コンボボックスから別のフィルタを選択します。

フィルタリストは、二重引用符で囲む必要があります。スペースや一重引用符を含む、長いファイル名を指定できます。各フィルタは、縦棒(|)で区切って指定します。たとえば、

```
dlgopen longfilter="*.text|*.Bob's work|*.*XX"
```

ダイアログボックスに現在のフォルダ内でファイル拡張子が.text であるファイルがすべて表示されるように指定すると、ダイアログボックスの**ファイルの種類***.text、*.Bob's work、および*.XX が追加されます。

DM ステートメントに DLGOPEN コマンドを使用するときは、LONGFILTER=に一重引用符を使用しないでください。DM ステートメントでは、サブミットするコマンドに一重引用符を付ける必要があるためです。LONGFILTER=に一重引用符を使用すると、コマンドの終端であるとみなされます。

FILTER='*filters*'は、**ファイルを開く** ダイアログボックスに表示するファイルの検索条件として使用するファイルのフィルタを 1 つ以上指定します。引数リスト内の 1 つ目のフィルタはデフォルトフィルタで、検索条件として使用されます。引数リストのすべてのフィルタが、**ファイルの種類**コンボボックスのフィルタリストに追加されます。さらにファイルの種類を検索するには、**ファイルの種類**コンボボックスから別のフィルタを選択します。フィルタリストは、引用符で囲む必要があります。フィルタを複数指定する場合は、スペースで区切ります。たとえば、

```
dlgopen filter='*.bak *.txt'
```

ダイアログボックスに、現在のフォルダ内でファイル拡張子が.bak であるファイルがすべて表示されるように指定する場合、**ファイルの種類**コンボボックスに*.BAK および*.TXT の両方が追加されます。

注 LONGFILTER="*filters*"と FILTER='*filters*'では、スペースと引用符の使用方法が異なります。*filters* にスペースや一重引用符が含まれる場合、LONGFILTER="*filters*"を使用してください。IFILTER='*filters*'を使用する場合、*filters* にスペースと一重引用符を含めることはできません。

**REPLACE**

フィルタリストを、デフォルトフィルタと連結させるのではなく、指定したフィルタに置き換えます。このオプションは、LONGFILTER=または FILTER=を指定したときにのみ有効です。たとえば、コマンド

```
dlgopen longfilter="*.txt" replace
```

がロードする **ファイルの種類**ボックスには、デフォルトのファイルの種類ではなく*.TXT が指定されます。

**SUBMIT | NOSUBMIT**

**サブミット** ダイアログボックスを表示するときにチェックボックスをオンにするかどうかを指定します。デフォルトでは、**サブミット** チェックボックス(開いているファイルの内容が即座に SAS プログラムとしてサブミットされることを示す)は、オンになりません。ファイルを開いたときに自動的にサブミットするには、**開いたファイルの内容をサブミットする** を、**プリファレンス**ダイアログボックスの **全般** ページで選択します。

**IMPORT**

**インポート** ダイアログボックスが開き、SAS セッションにグラフィックファイルをインポートできます。グラフィックのインポートの詳細については、“SAS/GRAPH ウィンドウのグラフィックファイルのインポート” (190 ページ)を参照してください。

**VERIFY**

アクティブウィンドウのファイルメニューに**開く**項目があるかどうかを確認します。この場合、**ファイルを開く** ダイアログボックスによって、デフォルトの INCLUDE コマンドではなく Open item コマンドが呼び出されます。

VERIFY 引数は、ALTCMD または IMPORT とともに指定できません。

**ALTCMD='*command*'**
**ファイルを開く** ダイアログボックスで選択したファイルに適用されるコマンドを指定します。たとえば、コマンド

```
dlgopen altcmd='x' longfilter="*.bat"
```

により、DOS バッチファイルを選択し、そのファイルを DOS シェルで実行できます。デフォルトのコマンドは、INCLUDE コマンドです。

## 詳細

**ファイルを開く** ダイアログを使用して、デフォルトエディタでファイルを開きます。デフォルトのエディタは、**プリファレンス** ダイアログボックスの**編集**タブにある**拡張エディタの使用** オプションの指定を基に決定されます。このオプションが選択されていると、拡張エディタがデフォルトエディタとなります。選択されていない場合は、プログラムエディタがデフォルトエディタとなります。

メニューから**ファイルを開く** ダイアログボックスにアクセスするには、[ファイル]メニューを選択し、**ファイルを開く**を選択します。

## 関連項目:

- “ファイルを開く” (108 ページ)
- “セッションプリファレンスについて ” (58 ページ)
- “他のアプリケーションからのグラフィックのインポート” (190 ページ)
- “FILEOPEN コマンド:Windows” (346 ページ)

# DLGPAGESETUP コマンド:Windows

**ページ設定**ダイアログボックスを開きます。

**Windows 固有:** すべて

## 構文

**DLGPAGESETUP**

## 詳細

**ページ設定** ダイアログボックスでは、用紙サイズ、用紙トレイ、向き、余白などのページ属性を定義できます。

## 関連項目:

“印刷ページの設定” (170 ページ)

# DLGPREF コマンド:Windows

**プリファレンス**ダイアログボックスを開きます。

**Windows 固有:** すべて

## 構文

**DLGPREF**

## 詳細

**プリファレンス**ダイアログボックスでは、SAS セッションを操作しやすいように構成できます。

## 関連項目:

“セッションプリファレンスについて ” (58 ページ)

# DLGPRT コマンド:Windows

**印刷**ダイアログボックスを開きます。

**Windows 固有:** すべて

## 構文

**DLGPRT** <NOSOURCE | ACTIVEBITMAP | SCREENBITMAP | AWSBITMAP | CLIPBITMAP | CLIPTEXT | ALTCMD='*command*' | BITMAPONLY | NODISPLAY | VERIFY>

### *引数なし*

デフォルトの印刷設定でアクティブウィンドウを印刷します。

### *オプション引数*

**ACTIVEBITMAP**
**印刷** ダイアログボックスを非表示にして、アーカイブウィンドウをビットマップとして印刷します。

**ALTCMD='*command*'**
**印刷** ダイアログボックスを使用して、PRINT 以外のコマンドを発行します。

**AWSBITMAP**
**印刷** ダイアログボックスを非表示にして、メイン SAS ウィンドウをビットマップとして印刷します。

**BITMAPONLY**
**印刷**ダイアログボックスからのビットマップ印刷のみを許可します。

**CLIPBITMAP**
**印刷** ダイアログボックスを非表示にして、Windows クリップボードの内容をビットマップとして印刷します。

**CLIPTEXT**
**印刷** ダイアログボックスを非表示にして、Windows クリップボードの内容をテキストとして印刷します。

**NODISPLAY**
**印刷** ダイアログボックスを非表示にして、デフォルト設定を使用して印刷します。

**NOSOURCE**
ユーザーが印刷ソース(アプリケーションウィンドウ)を指定できないようにします。

**SCREENBITMAP**
**印刷** ダイアログボックスを非表示にして、画面全体をビットマップとして印刷します。

**VERIFY**
アクティブなアプリケーションウィンドウで、テキスト印刷がサポートされている( **ファイル** メニューに **印刷**項目が含まれている)かどうかを確認します。テキストがサポートされない場合は、**印刷** ダイアログボックスではビットマップ印刷のみが許可されます。

### 詳細

**印刷**ダイアログボックスでは、アクティブウィンドウの内容を印刷できます。

### 関連項目:

“SAS ウィンドウからの印刷” (166 ページ)

## DLGPRTPREVIEW コマンド:Windows

**印刷プレビュー**ウィンドウを呼び出します。

**Windows 固有:** すべて

### 構文

**DLGPRTPREVIEW** <VERIFY>

#### *オプション引数*

**VERIFY**
アクティブなアプリケーションウィンドウで、印刷がサポートされている(**ファイル**メニューに**印刷**項目が含まれている)かどうかを確認します。印刷がサポートされていない場合は、**印刷プレビュー**ウィンドウは表示されません。ただし、この場合でも、ウィンドウをビットマップとして印刷することはできます。DLGPRT VERIFY コマンドを発行して、**プレビュー**をクリックして出力をプレビューします。

### 詳細

印刷プレビュー機能をサポートしない SAS アプリケーションウィンドウもあります。

### 関連項目:

- “印刷前の出力のプレビュー” (175 ページ)
- “印刷” (165 ページ)

## DLGPRTSETUP コマンド:Windows

**印刷設定**ダイアログボックスを開くか、またはプリンタ設定をプログラムで設定します。

**Windows 固有:** すべて

### 構文

**DLGPRTSETUP** <ORIENT=PORTRAIT | LANDSCAPE|> <NODISPLAY>

#### *オプション引数*

**ORIENT=PORTRAIT | LANDSCAPE**
現在のプリンタに、ページのデフォルトの印刷の向きを設定します。ORIENT=パラメータは、SAS の下位互換性をサポートするために用意されています。通常は、向きを指定するには、ORIENTATION システムオプションを使用します。

**NODISPLAY**
**印刷設定**ダイアログボックスを非表示にします。このオプションは、他のオプションを使用してプリンタ設定を明示的に設定するときにのみ使用します。

### 詳細

**印刷設定** ダイアログボックスを使用すると、印刷するプリンタの名前を指定できます。また、SAS フォームを使用して印刷することを指定し、用紙の向き、余白、フォントなど、SAS で情報を印刷する方法を制御するダイアログボックスを開くことができます。

### 関連項目:

- “プリンタの変更” (168 ページ)
- “印刷フォントの変更” (169 ページ)
- “印刷ページの設定” (170 ページ)

## DLGREPLACE コマンド:Windows

**置換**ダイアログボックスを開きます。

**Windows 固有:** すべて

### 構文

**DLGREPLACE**

### 詳細

**置換** ダイアログボックスを使用すると、テキスト文字列を検索して、別のテキスト文字列に置換できます。

### 関連項目:

“DLGFIND コマンド:Windows” (334 ページ)

## DLGRUN コマンド:Windows

**実行**ダイアログボックスを開きます。

**Windows 固有:** すべて

## 構文

**DLGRUN**

## 詳細

**実行** ダイアログボックスでは、SAS 内から別のアプリケーションを起動できます。たとえば、**実行** ダイアログボックスの**コマンド行** フィールドに excel.exe と入力すると、Microsoft Excel が起動します。

# DLGSAVE コマンド:Windows

**名前を付けて保存**ダイアログボックスを開きます。

**Windows 固有:** すべて

## 構文

**DLGSAVE** <LONGFILTER="*filters*" | FILTER='*filters*' <REPLACE>> <EXPORT> <NOPROMPT> <VERIFY> <ALTCMD='*command*'>

### *引数なし*

デフォルト設定で**名前を付けて保存** ダイアログボックスを開きます。

### *オプション引数*

**LONGFILTER="*filters*" | FILTER='*filters*'**

LONGFILTER="*filters*"は、名前を付けて保存ダイアログボックスに表示するファイルの検索条件として使用するファイルのフィルタを 1 つ以上指定します。引数リスト内の 1 つ目のフィルタはデフォルトフィルタで、検索条件として使用されます。引数リストのすべてのフィルタが、**ファイルの種類**コンボボックスのフィルタリストに追加されます。さらにファイルの種類を検索するには、**ファイルの種類**コンボボックスから別のフィルタを選択します。

フィルタは、二重引用符で囲む必要があります。フィルタには、スペースや一重引用符を含む長いファイル名も指定できます。各フィルタは、縦棒(|)で区切って指定します。たとえば、

```
dlgsave longfilter="*.text|*.Bob's work|*.**XX"
```

ダイアログボックスに現在のフォルダ内でファイル拡張子が.text であるファイルがすべて表示されるように指定すると、ダイアログボックスの**ファイルの種類**コンボボックスに*.text、*.Bob's work、および*.XX が追加されます。

DM ステートメントに DLGSAVE コマンドを使用するときは、LONGFILTER=に一重引用符を使用しないでください。DM ステートメントでは、サブミットするコマンドに一重引用符を付ける必要があるためです。LONGFILTER=に一重引用符を使用すると、コマンドの終端であるとみなされます。

FILTER='*filters*'は、**ファイルを開く** ダイアログボックスに表示するファイルの検索条件として使用するファイルのフィルタを 1 つ以上指定します。引数リスト内の 1 つ目のフィルタはデフォルトフィルタで、検索条件として使用されます。引数リストのすべてのフィルタが、**ファイルの種類**コンボボックスのフィルタリストに追加されます。さらにファイルの種類を検索するには、**ファイルの種類**コンボボックスから別のフィルタを選択します。フィルタリストは、引用符で囲む必要があります。フィルタを複数指定する場合は、スペースで区切ります。たとえば、

```
dlgsave filter='*.bak *.txt'
```

ダイアログボックスに、現在のフォルダ内でファイル拡張子が.bak であるファイルがすべて表示されるように指定する場合、**ファイルの種類**コンボボックスに*.BAK および*.TXT の両方が追加されます。

注 LONGFILTER="*filters*" と FILTER='*filters*' の違いは、LONGFILTER="*filters*" ではフィルタでスペースと引用符を使用できますが、FILTER='*filters*'ではスペースと引用符を使用することはできません。

**REPLACE**
フィルタリストを、デフォルトフィルタと連結させるのではなく、指定したフィルタに置き換えます。このオプションは、LONGFILTER=引数または FILTER=引数を指定したときにのみ有効です。たとえば、コマンド

```
dlgsave longfilter="*.txt" replace
```

がロードする **ファイルの種類**コンボボックスには、デフォルトのファイルの種類ではなく*.TXT のみが指定されます。

**EXPORT**
**エクスポート** ダイアログボックスを呼び出し、SAS セッションからグラフィックファイルをエクスポートできます。**エクスポート** ダイアログボックスの詳細については、"他のアプリケーションで使用するグラフィックのエクスポート" (192 ページ)を参照してください。

**NOPROMPT**
既存のファイルの置き換えまたは追加を行うときに、確認メッセージを表示しません。

**VERIFY**
アクティブウィンドウのファイルメニューに**保存**項目が存在するかどうかを確認します。存在する場合は、**名前を付けて保存** ダイアログボックスでは、デフォルトの FILE コマンドではなく**保存** 項目を呼び出します。

VERIFY 引数は、ALTCMD または EXPORT とともに指定できません。

**ALTCMD='*command*'**
**名前を付けて保存** ダイアログボックスから選択されたファイルに適用されるコマンドを指定します。たとえば、コマンド

```
dlgsave altcmd='prtfile'
```

は、**名前を付けて保存** d ダイアログボックスから選択されたファイルを現在の印刷ファイルとして設定します。デフォルトのコマンドは、FILE コマンドです。

## 詳細

**名前を付けて保存** ダイアログボックスでは、アクティブウィンドウの内容をファイルに保存します。メニューから **名前を付けて保存** ダイアログボックスにアクセスするには、ファイルメニューを選択し、**名前を付けて保存** を選択します。

## 関連項目:

- 拡張エディタのセクション"ファイルの保存" (82 ページ)
- プログラムエディタのセクション "ファイルの保存" (114 ページ)

## DLGSMAIL コマンド:Windows

**メールの送信**ダイアログボックスを開きます。

**Windows 固有:** すべて

### 構文

**DLGSMAIL**

### 詳細

DLGSMAIL コマンドは、EMAILDLG システムオプションの値に基づいて、電子メールダイアログボックスを開きます。EMAILDLG オプションの値が `sas` の場合は、DLGSMAIL では**メールの送信** ダイアログボックスを表示します。EMAILDLG オプションの値が**ネイティブ**の場合 DLGSMAIL は、MAPI に準拠する電子メールダイアログボックスを開きます。

*注:* 電子メールがインターネット接続に設定されていない場合、ウィザードウィンドウが開いて電子メールを設定できます。

### 関連項目:

- “SAS を使用して電子メールの送信” (44 ページ)
- “WEMAILFMT コマンド:Windows” (362 ページ)
- “EMAILDLG システムオプション:Windows” (503 ページ)

## FILE コマンド:Windows

現在のウィンドウの内容を外部ファイルに書き込みます。

**Windows 固有:** 有効なオプション

### 構文

**FILE** *file-specification* <ENCODING='*encoding-value*'> <*portable-options*> <*host-options*>

#### *必須引数*

***file-specification***
ファイル参照名、ファイルのショートカット、引用符に囲まれた Windows ファイル名、環境変数または現在の作業ディレクトリに保存されている引用符で囲まれていないファイル名など、Windows で有効な外部ファイル指定を指定します。

#### *オプション引数*

**ENCODING='*encoding-value*'**
出力ファイルに書き出す際に使用するエンコーディングを指定します。ENCODING=の値により、出力ファイルのエンコーディングが現在のセッションのエンコーディングと異なることがわかります。

SAS では、出力ファイルにデータを書き込む際に、データのエンコードをセッションのエンコードから指定したエンコードに変換します。

有効なエンコードの値については、22 章: “Encoding Values in SAS Language Elements” (*SAS National Language Support (NLS): Reference Guide*)を参照してください。

***portable-options***
SAS ヘルプとドキュメントの FILE コマンドで説明されている 1 つ以上のポータブルオプションを指定します。

***host-options***

**BLKSIZE=***block-size***BLK=***block-size***
入出力操作で物理的に読み書きするバイト数を指定します。デフォルトは 8192 です。最大 1MB まで指定できます。

**IGNOREDOSEOF**
可変長レコード形式のファイルの入出力操作で使用します。このオプションを指定すると、^Z は EOF 記号ではなく文字データとして解釈されます。

**LRECL=***record-length*
レコード長をバイト単位で指定します。Windows ではデフォルト値は 32767 です。*record-length* の値は、1 から 1,073,741,823 (1 GB)の範囲から指定できます。

**RECFM=***record-format*
レコードの形式を制御します。Windows では、次の値が有効です。

| | |
|---|---|
| F | 固定長形式を指定します。 |
| N | バイナリ形式を指定し、ファイルはバイトストリームとして処理されます。LRECL を指定しない場合、デフォルト設定では、一度に 32767 バイトがファイルから読み込まれます。 |
| P | 印刷形式を指定します。 |
| S370V | 可変長 S370 レコード形式(V)を指定します。 |
| S370VB | 可変長ブロック S370 レコード形式(VB)を指定します。 |
| S370VBS | スパンレコード可変長ブロック S370 レコード形式(VBS)を指定します。 |
| V\|D | 変数の出力形式を指定します。これはデフォルトの設定です。 |

## 詳細

FILE コマンドは、ウィンドウからテキストを消去せずに、アクティブウィンドウの全内容を外部ファイルに書き込みます。

*file-specification* を指定しない場合、前の FILE コマンドまたは INCLUDE コマンドで指定したファイル名が使用されます。この場合、まずファイルを上書きするかどうかが確認され、次の操作を選択できます。

- 置換
- 付加
- キャンセル

FILE コマンドまたは INCLUDE コマンドを実行しなかった場合は、デフォルトファイルが存在しないことを示すエラーメッセージが表示されます。

拡張エディタでは、ファイル名が 8 文字以下の場合、ファイル拡張子の.SAS が *file-specification* に付け加えられます。8 文字より長い *file-specification* には、拡張子は付け加えられません。

### 関連項目:

SAS ヘルプとドキュメントの“FILE コマンド“

## FILEOPEN コマンド:Windows

拡張エディタの**ファイルを開く**ダイアログボックスを開くか、または拡張エディタにファイルを開きます。

**Windows 固有:** すべて

### 構文

**FILEOPEN** <"*file specification*">

#### *必須引数*

**“*file specification*”**
Windows の有効なパス、ファイル名およびファイル拡張子を指定します。ファイルが現在の作業フォルダにある場合は、パスは必要ありません。

### 詳細

FILEOPEN コマンドに file-specification を含めないと、**ファイルを開く** ダイアログ**ボックス**が表示されます。FILEOPEN コマンドにファイルを指定した場合は、**ファイルを開く** ダイアログボックスは開かれず、拡張エディタにファイルが開かれます。指定したファイルは、一重引用符または二重引用符で囲む必要があります。

*注:* プログラムエディタにファイルを開くには、DLGOPEN コマンドを使用します。

### 関連項目:

- “ファイルを開く” (81 ページ)
- “DLGOPEN コマンド:Windows” (336 ページ)

## FILL コマンド:Windows

埋め込み文字を指定します。

**Windows 固有:** デフォルト文字

### 構文

**FILL** *fill-character*

#### *必須引数*

***fill-character***
行を埋め込むのに使用する文字を指定します。

### 詳細

現在のカーソル位置から埋め込み文字が埋め込まれます。Windows 版では、デフォルトの埋め込み文字はアンダースコア(_)です。

### 関連項目:

SAS ヘルプとドキュメントの“FILL Command-line コマンド”

## GSUBMIT コマンド:Windows

Windows クリップボードに格納されている SAS コードをサブミットします。

**Windows 固有:** 次に対する有効な値:*paste-buffer-name*

### 構文

**GSUBMIT BUF**=*paste-buffer-name* | "*SAS-statement-1;...SAS-statement-n;*"

### 詳細

Windows 版で、*paste-buffer-name* 引数を指定する場合は、DEFAULT を指定する必要があります。Windows クリップボードがデフォルトのペーストバッファです。

拡張エディタを使用してサブミットしたプロシジャの実行中は、Windows クリップボードに格納された SAS ステートメントを、GSUBMIT コマンドを使用してサブミットすることはできません。この場合、新たな**拡張エディタ**ウィンドウを開いて SAS ステートメントをコピーし、サブミットすることができます。

### 関連項目:

外部ファイルの使用セクションの GSUBMIT コマンドの使用 *Windows 版 SAS*

## HOME コマンド:Windows

カーソルの位置を、現在の位置からホームポジションに移動します。

**Windows 固有:** キーボードと同等

### 構文

**HOME**

### 詳細

Windows 版では、HOME コマンドはキーボードの Home キーに相当します。HOME コマンドを発行すると、ウィンドウ内のカーソルの位置が直前のカーソル位置とホームポジションの間で移動します。ウィンドウにコマンド行が表示されている場合は、ホームポジションはコマンド行となります。

ファンクションキーを定義して CURSOR コマンドを実行することもできます。CURSOR コマンドは、カーソルをウィンドウ内のホームポジションに配置しますが、切り替え機能は備わっていません。

### 関連項目:
SAS ヘルプとドキュメントの「HOME コマンド」

## ICON コマンド:Windows

アクティブなウィンドウを最小化します。

**Windows 固有:** すべて

### 構文

**ICON** <ALL> , <ON> , <OFF>

#### *引数なし*
アクティブウィンドウを最小化します。

#### *オプション引数*

**ALL**
メイン SAS ウィンドウ以外のすべてのウィンドウを最小化します。

**ON**
アクティブウィンドウを最小化します。

**OFF**
アクティブウィンドウを、前のサイズに復元します。

### 詳細

ウィンドウバーがアクティブな場合、ICON コマンドによりウィンドウが最小化されウィンドウバーに収められます。ウィンドウバーがアクティブでない場合、ウィンドウは最小化されアプリケーションワークスペースに収められます。

ICON コマンド(オプションなし)は、トグルとして機能します。

*注:* ICON コマンドを ICON システムオプションと混同しないでください。ICON システムオプションではメイン SAS ウィンドウを最小化できます。

### 関連項目:
SAS ヘルプとドキュメントの「ICON コマンド」

## INCLUDE コマンド:Windows

外部ファイルの内容すべてを現在のウィンドウにコピーします。

**Windows 固有:** 有効なオプション

### 構文

**INCLUDE** *file-specification* <ENCODING='*encoding-value*'> <*portable-options*> <*host-options*>

## 必須引数

***file-specification***
ファイル参照名、ファイルのショートカット、引用符に囲まれた Windows ファイル名、環境変数または現在の作業ディレクトリに保存されている引用符で囲まれていないファイル名など、Windows で有効な外部ファイル指定を指定します。

## オプション引数

**ENCODING='*encoding-value*'**
外部ファイルから読み込む際に使用するエンコーディングを指定します。ENCODING=の値により、外部ファイルのエンコーディングが現在のセッションのエンコーディングと異なることがわかります。

SAS では、外部ファイルからデータを読み込む際には、データのエンコードを指定のエンコードからセッションのエンコードに変換します。

有効なエンコードの値については、22 章: “Encoding Values in SAS Language Elements” (*SAS National Language Support (NLS): Reference Guide*)を参照してください。

***portable-options***
SAS ヘルプとドキュメントの INCLUDE コマンドで説明されている 1 つ以上のポータブルオプションを指定します。

***host-options***

**BLKSIZE=*block-size*BLK=*block-size***
入出力操作で物理的に読み書きするバイト数を指定します。デフォルトの設定は 8K です。最大 1MB まで指定できます。

**IGNOREDOSEOF**
可変長レコード形式のファイルの入出力操作で使用します。このオプションを指定すると、^Z は EOF 記号ではなく文字データとして解釈されます。

**LRECL=*record-length***
レコード長をバイト単位で指定します。Windows ではデフォルト値は 32767 です。*record-length* の値は、1 から 1,073,741,823 (1 GB)の範囲から指定できます。

**NOTABS**
動的データ交換でのみ使用します。変数間の区切り文字に、タブ以外の文字を使用できます。詳細については、*Windows 版 SAS* の動的データ交換の使用セクションの DDE での NOTAB オプションの使用に関する説明を参照してください。

**RECFM=*record-format***
レコードの形式を制御します。Windows では、次の値が有効です。

| | |
|---|---|
| F | 固定長形式を指定します。 |
| N | バイナリ形式を指定し、ファイルはバイトストリームとして処理されます。LRECL を指定しない場合、デフォルト設定では、一度に 32767 バイトがファイルから読み込まれます。 |
| P | 印刷形式を指定します。 |
| S370V | 可変長 S370 レコード形式(V)を指定します。 |
| S370VB | 可変長ブロック S370 レコード形式(VB)を指定します。 |
| S370VBS | スパンレコード可変長ブロック S370 レコード形式(VBS)を指定します。 |

V|D　変数の出力形式を指定します。この値はデフォルトの設定です。

## 詳細

INCLUDE コマンドは、外部ファイルの内容すべてをアクティブウィンドウにコピーします。

*file-specification* を指定しない場合、前の FILE コマンドまたは INCLUDE コマンドで指定したファイル名が使用されます。FILE コマンドまたは INCLUDE コマンドを実行しなかった場合は、デフォルトファイルが存在しないことを示すエラーメッセージが表示されます。

拡張エディタでは、ファイル名が 8 文字以下の場合、ファイル拡張子の.SAS が *file-specification* に付け加えられます。8 文字より長い *file-specification* には、拡張子は付け加えられません。

## 関連項目:

- "ENCODING= Option" (*SAS National Language Support (NLS): Reference Guide*)
- "外部ファイルの参照" (148 ページ)
- "INCLUDE コマンドの使用" (161 ページ)
- "高度な外部 I/O の操作法" (162 ページ)

# PMENU コマンド:Windows

SAS アプリケーションウィンドウのコマンド行のオン／オフを切り替えます。

**Windows 固有:** コマンドの動作

## 構文

**PMENU** <ON | OFF>

### *引数なし*

コマンド行のオン／オフを切り替えます。

### *オプション引数*

**ON**
　コマンド行をオフにします。

**OFF**
　コマンド行をオンにします。

## 詳細

Windows 動作環境では、コマンド行は常に表示されています。SAS ウィンドウにコマンド行を表示するかどうかを指定するには、PMENU コマンドまたは COMMAND コマンドを使用します。

### 関連項目:

- SAS ヘルプとドキュメントの“PMENU コマンド“
- “COMMAND コマンド:Windows” (330 ページ)
- “WMENUPOP コマンド:Windows” (365 ページ)

## SAVE コマンド:Windows

拡張エディタ、プログラムエディタ、ログ、出力、メモ帳、キーの各ウィンドウの全内容をカタログエントリに書き込みます。

### 構文

**SAVE** <*catalog-entry*> <ATTR> <TABS> <APPEND | REPLACE>

#### *引数なし*

現在の SAS セッション中に、COPY コマンドまたは SAVE コマンドで最後に指定されたカタログエントリにウィンドウの内容を書き込みます。

#### *オプション引数*

***catalog-entry***
4 レベル名を指定します。

**ATTR**
属性をエントリとともに保存します。

**TABS**
デフォルトのスペースを保持してファイルを保存するのではなく、保存する際にタブに変更してスペースを圧縮します。

**APPEND**
ウィンドウの内容をカタログエントリの内容に追加します。この引数を指定すると、別のカタログエントリが指定されるまで、デフォルトのカタログエントリとして追加が続けられます。

**REPLACE**
カタログエントリの内容をウィンドウの内容に置き換えます。この引数を指定すると、別のカタログエントリが指定されるまで、デフォルトのカタログエントリとして置換が続けられます。

## STORE コマンド:Windows

選択したテキストやグラフィックを Windows クリップボードにコピーします。

**Windows 固有:** 拡張エディタではサポートされていない有効なオプション

### 構文

**STORE** <LAST | ALL>

### *オプション引数*

**LAST**
テキスト範囲が複数選択されている場合、最後に選択したテキストのみをコピーし、それ以外のテキストの選択をすべて解除します。複数のテキスト範囲が選択されている場合に 1 つの範囲をコピーするには、LAST と ALL のどちらかの引数を指定する必要があります。

**ALL**
テキスト範囲が複数選択されている場合、現在選択しているテキストをすべて格納します。

## 詳細

STORE コマンドは、アクティブウィンドウで選択したテキストやグラフィックをコピーし、Windows クリップボードに格納します。

Windows 版では、STORE コマンドの APPEND=オプションと BUFFER=オプションはサポートされません。

## 関連項目:

- SAS ヘルプとドキュメントの“STORE コマンド“
- “クリップボードの使用” (55 ページ)

# SUBTOP コマンド:Windows

SAS プログラムの最初の *n* 行をサブミットして処理します。

**Windows 固有:** 拡張エディタとプログラムエディタで有効

## 構文

**SUBTOP** <*n*>

### *引数なし*

プログラムの最初の行のみをサブミットして処理するように指定します。

### *オプション引数*

***n***
プログラムの最初の *n* 行をサブミットして処理します。

## 詳細

**Clear text on submit** チェックボックスを **拡張エディタオプション** ダイアログボックスで選択すると、SUBTOP コマンドを発行した時にウィンドウからサブミットされた行がすべて削除されます。

## 関連項目:

“プログラムのサブミット” (92 ページ)

## TOOLCLOSE コマンド:Windows

アプリケーションツールバーやツールボックスを閉じます。

**Windows 固有:** すべて

### 構文

**TOOLCLOSE**

### 詳細

TOOLCLOSE コマンドを使用して、ツールバーやツールボックスを閉じます。

## TOOLEDIT コマンド:Windows

**ツールのユーザー設定**ダイアログボックスを開きます。

**Windows 固有:** すべて

### 構文

**TOOLEDIT** <*library.catalog.entry*>

#### *引数なし*

現在ロードされているツールセットを編集します。

#### *オプション引数*

***library.catalog.entry***
編集する TOOLBOX エントリを指定します。

### 詳細

The TOOLEDIT command invokes the **Customize Tools** dialog box with the TOOLBOX entry specified by *library.catalog.entry*.TOOLBOX エントリが指定されない場合は、現在ロードされているツールのセットが使用されます。

### 関連項目:

“ツールバーのカスタマイズ” (67 ページ)

## TOOLLARGE コマンド:Windows

ツールバーまたはツールボックスのボタンのサイズを切り替えます。

**Windows 固有:** すべて

### 構文

**TOOLLARGE** <ON | OFF>

#### *引数なし*

ツールバーまたはツールボックスのボタンを大きいサイズまたは標準サイズに切り替えます。

#### *オプション引数*

**ON**
ツールバーまたはツールボックスのボタンを大きいサイズに設定します。

**OFF**
ツールバーまたはツールボックスのボタンを標準サイズに設定します。

### 詳細

TOOLLARGE コマンドで、ツールバーのボタンを標準サイズまたは大きいサイズに切り替えることができます。大きいサイズは、高解像度ディスプレイで使用しやすい大きさです。

### 関連項目:

“ツールをデフォルト設定にリセットする” (72 ページ)

## TOOLLOAD コマンド:Windows

特定のツールボックスをロードします。

**Windows 固有:** すべて

### 構文

**TOOLLOAD** <WINDOW> <BOX | BAR> <*libref.catalog.member*>

#### *引数なし*

アクティブウィンドウのツールバーをロードします。ツールは、[ツールのユーザー設定] ダイアログボックスの設定に応じて、ツールバーまたはツールボックスとして表示されます。

#### *オプション引数*

**WINDOW**
指定されたツールボックスエントリをアクティブウィンドウに割り当てます。この結果、ロードする特定のツールセットはアクティブウィンドウにのみ適用されます。この割り当ては、ウィンドウを閉じるまで有効です。後でウィンドウを開き直すと、ウィンドウのツールバーはデフォルトのツールバーに戻ります。

TOOLLOAD コマンドに WINDOW オプションを指定しない場合、ロードするツールバーまたはツールボックスが、SASUSER.PROFILE カタログに特定のツールセット定義が格納されていないすべてのウィンドウに適用されます。このツールセットには、ウィンドウと同じ名前を付ける必要があります。たとえば、**エクスプローラ**ウィンドウのツールセットの名前は、Sasuser.Profile.Explorer となります。WINDOW オ

プションを指定しないと、現在の SAS セッションの実行中は、ウィンドウを閉じたり開いたりする回数に関係なくツールセットの定義が保持されます。

**BOX | BAR**
アイコンを別のウィンドウのツールボックスとして表示するか、メイン SAS ウィンドウと統合したツールバーとして表示するかを制御します。

***libref.catalog.member***
ロードするカタログエントリを指定します。TOOLBOX は、デフォルトのカタログエントリタイプです。

### 詳細

TOOLLOAD コマンドが処理されると、指定したツールボックスがアクティブツールボックスになります。

### 関連項目:

“特定のアプリケーションまたはウィンドウで使用するようにツールバーをカスタマイズし保存する” (71 ページ)

## TOOLSWITCH コマンド:Windows

ツール切り替え機能のオン／オフを切り替えます。

**Windows 固有:** すべて

### 構文

**TOOLSWITCH** ON | OFF

#### *必須引数*

**ON**
アクティブウィンドウのツールバーが定義されている場合、自動的にロードされます。

**OFF**
他のツールバーを明示的にロードしない場合、すべてのウィンドウに対してデフォルトのツールバー(SASUSER.PROFILE.TOOLBOX)が使用されます。

### 詳細

TOOLSWITCH コマンドを使用すると、アクティブウィンドウに定義したツールバーと SAS のデフォルトツールバーを切り替えることができます。

### 関連項目:

- “セッションプリファレンスについて ” (58 ページ)
- “TOOLEDIT コマンド:Windows” (353 ページ)

## TOOLTIPS コマンド:Windows

ツールチップ機能を切り替えます。

**Windows 固有:** すべて

## 構文

**TOOLTIPS** <ON | OFF>

### *引数なし*

ツールチップ機能のオン/オフを切り替えます。

### *オプション引数*

**ON**
ツールチップ機能をオンにします。

**OFF**
ツールチップ機能をオフにします。

## 詳細

ツールチップは、ツールバーボタンやツールボックスボタン、メイン SAS ウィンドウのコントロールの一部にマウスポインタを置くと表示される便利なヒントです。

TOOLTIPS コマンドで、ツールボックスのアイコンや他のコントロールにカーソルを移動するときにツールチップテキストを表示するかどうかを指定します。ON または OFF を指定しないと、TOOLTIPS コマンドは、現在の設定に応じて、設定を切り替えます。

*注:* ツールチップをスクリーンチップと混同しないでください。スクリーンチップは、メイン SAS ウィンドウのステータスバー、ウィンドウバー、タブに、役立つヒントをが表示します。

## 関連項目:

"WSCREENTIPS コマンド:Windows" (370 ページ)

# WATTACH コマンド:Windows

SAS を使用して起動する電子メールメッセージにアクティブウィンドウの内容を添付するかどうかを切り替えます。

**Windows 固有:** すべて

## 構文

**WATTACH** <ON | OFF>

### *引数なし*

添付モードのオン／オフを切り替えます。

### *オプション引数*

**ON**
アクティブウィンドウの内容を添付します。

**OFF**
アクティブウィンドウの内容を添付しません。

### 詳細

ON を指定すると、アクティブウィンドウの内容は添付ファイルとして送られます。テキストウィンドウでは、形式はテキストまたは RTF です(WEMAILFMT コマンドまたは**プリファレンス** ダイアログボックスの設定で決定します)。グラフィックウィンドウはビットマップファイル(.BMP)として送られます。

**プリファレンス** ダイアログボックスの**全般** ページでもこの設定を切り替えられます。

### 関連項目:

- “SAS を使用して電子メールの送信” (44 ページ)
- “WEMAILFMT コマンド:Windows” (362 ページ)
- “セッションプリファレンスについて ” (58 ページ)

## WATTENTION コマンド:Windows

終了する SAS プロセスを選択するために、**タスクマネージャ**ウィンドウを表示します。

**Windows 固有:** すべて

### 構文

**WATTENTION**

### 詳細

WATTENTION コマンドを使用すると、終了する SAS プロセスを選択できます。このコマンドは、Ctrl キーを押しながら Break キーを押す操作に相当します。

## WAUTOSAVE コマンド:Windows

SAS エディタウィンドウの作業を自動保存する頻度を制御します。

**Windows 固有:** すべて

### 構文

**WAUTOSAVE** <<ON | OFF> <INTERVAL=*minutes*>>

#### *引数なし*

定義した間隔で作業が自動保存されるように、自動保存機能を有効化し自動保存タイマをリセットします。

#### *オプション引数*

**ON | OFF**
自動保存機能を有効化または無効化を指定します。

**INTERVAL=*minutes***
一定の間隔(分)ごとに作業を保存します。デフォルトの間隔は 10 分です。間隔は整数で指定します。

### 詳細

作業を自動保存する間隔をデフォルトの 10 分から変更する場合に、WAUTOSAVE コマンドを使用します。SAS は、プログラムエディタの内容を現在の作業フォルダの 'pgm.asv'、または AUTOSAVELOC システムオプションで指定したフォルダに保存します。拡張エディタウィンドウの内容は、使用している動作環境の一時フォルダに 'Autosave of *filename*.$AS'というファイル名で保存されます。**プリファレンス**ダイアログボックスの**編集** ページでも自動保存機能を設定できます。

### 関連項目:

- “セッションプリファレンスについて ” (58 ページ)
- “AUTOSAVELOC= System Option” (*SAS System Options: Reference*)

## WBROWSE コマンド:Windows

プリファレンスダイアログボックスで指定した Web ブラウザを起動します。

**Windows 固有:** すべて

### 構文

**WBROWSE** <"*URL*">

#### *引数なし*

**プリファレンス**ダイアログボックスの **Web** ページで定義されている標準 Web ブラウザを起動します。

#### *オプション引数*

***URL***
インターネットまたはローカルイントラネットでドキュメントを検索する際に必要なサーバーやパスの情報を含む URL (Uniform Resource Locator)を指定します。

### 詳細

デフォルトでは、WBROWSE コマンドは、デフォルトの Web ブラウザを起動し、SAS Institute のホームページ(Support.sas.com)を表示します。URL を指定すると、その URL の場所が表示されます。URL は二重引用符で囲んでください。Web ブラウザが開くデフォルトのページは、**プリファレンス** ダイアログボックスの **Web** ページで変更できます。

### 関連項目:

“セッションプリファレンスについて ” (58 ページ)

## WCOPY コマンド:Windows

アクティブウィンドウで選択した内容を Windows クリップボードにコピーします。

**Windows 固有:** すべて

### 構文

**WCOPY**

### 詳細

WCOPY はツールバーコマンドとともに使用します。WCOPY コマンドを入力したとき、アクティブウィンドウがテキストウィンドウである場合、アクティブウィンドウのメニューから[コピー]メニュー項目が検索されます。[コピー]メニュー項目が存在する場合は、選択した内容を Windows クリップボードにコピーします。[コピー]メニュー項目がない場合は、WCOPY は STORE コマンドを実行します。

### 関連項目:

"STORE コマンド:Windows" (351 ページ)

## WCUT コマンド:Windows

アクティブウィンドウで選択した内容を Windows クリップボードに移動します。

**Windows 固有:** すべて

### 構文

**WCUT**

### 詳細

WCUT はツールバーコマンドとともに使用します。アクティブウィンドウが**プログラムエディタ**ウィンドウなどのエディタウィンドウである場合にのみ使用できます。WCUT コマンドを入力すると、アクティブウィンドウのメニューから[切り取り]メニュー項目が検索されます。[切り取り]メニュー項目がある場合は、アクティブウィンドウで選択した内容を Windows クリップボードに移動します。[切り取り]メニュー項目がない場合、WCUT は CUT コマンドを実行します。

### 関連項目:

"CUT コマンド:Windows" (332 ページ)

## WDOCKVIEW コマンド:Windows

ドッキングビューのオン／オフを切り替えます。

**Windows 固有:** すべて

### 構文

**WDOCKVIEW** <ON | OFF>

***引数なし***
ドッキングビューのオン／オフを切り替えます。

### *オプション引数*

**ON**
ドッキングビューをオンにします。

**OFF**
ドッキングビューをオフにします。

### 詳細

ドッキングビューを使用すると、メイン SAS ウィンドウ内を簡単に移動できます。ドッキングビューを有効にすると、**SAS エクスプローラ**ウィンドウや**結果**ウィンドウなどのドッキング可能な(メイン SAS ウィンドウに組み合わせ可能な)ウィンドウが、メイン SAS ウィンドウの左側に表示されます。**結果**ウィンドウにリストされたプロシジャからの出力など、別のウィンドウを開くドッキングウィンドウのアイテムをクリックすると、そのウィンドウはメイン SAS ウィンドウの右側のウィンドウに表示されます。ドッキングされたウィンドウ間はタブを使用して移動します。

### 関連項目:

- “ドッキングビューの使用 ” (39 ページ)
- “セッションプリファレンスについて ” (58 ページ)

## WDOCKVIEWMINIMIZE コマンド:Windows

**ドッキングビュー**ウィンドウを最小化します。

**Windows 固有:** すべて

### 構文

**WDOCKVIEWMINIMIZE**

### 詳細

WDOCKVIEWMINIMIZE は、**ドッキングビュー**ウィンドウを最小化します。

### 関連項目:

- “WDOCKVIEWRESTORE コマンド:Windows” (361 ページ)
- “ドッキングビューの使用 ” (39 ページ)

## WDOCKVIEWRESIZE コマンド:Windows

ドッキングビューの分割バーを移動するために、リサイズモードを開始します。

**Windows 固有:** すべて

### 構文

**WDOCKVIEWRESIZE**

### 比較

コマンドバーに WDOCKVIEWRESIZE と入力すると、リサイズモードが開始されます。リサイズモードでは、マウスや、キーボードの左矢印キーまたは右矢印キーを使用して、ドッキングビューの分割バーを移動できます。Ctrl キーを押しながら左矢印キーまたは右矢印キーを押すと、分割バーの移動量が増加します。リサイズモードを終了するには、Enter キーを押します。

Alt+W+S を同時に押すか、または次のメニューパスを選択して、ドッキングビューのリサイズモードを開始することもできます:**ウィンドウ** ⇨ **サイズドッキングビュー**

## WDOCKVIEWRESTORE コマンド:Windows

タスクバーの**ドッキングビュー**ウィンドウを元のサイズに戻します。

**Windows 固有:** すべて

### 構文

**WDOCKVIEWRESTORE**

### 詳細

WDOCVIEWRESTORE は、**ドッキングビュー**ウィンドウをメイン SAS ウィンドウの左側に戻します。

### 関連項目:

- “WDOCKVIEWMINIMIZE コマンド:Windows” (360 ページ)
- “ドッキングビューの使用 ” (39 ページ)

## WEDIT コマンド:Windows

**拡張エディタ**ウィンドウを開いたり、拡張エディタを使用可能または使用不可にします。

**Windows 固有:** すべて

### 構文

**WEDIT** <"*filename*"> <USE | NOUSE>

#### *引数なし*

**拡張エディタ**ウィンドウを開きます。

#### *オプション引数*

**"*filename*"**
拡張エディタに開くファイル名を指定します。ファイル名は二重引用符で囲みます。指定する場合、*filename* は最初の引数としてください。

**USE**
拡張エディタを使用可能にし、**拡張エディタ**ウィンドウを開くように指定します。

**NOUSE**
拡張エディタを使用不可にします。**拡張エディタ**ウィンドウは開かなくなります。

### 詳細

WEDIT コマンドを使用して拡張エディタを使用可能にする場合、**プリファレンス**ダイアログボックスの**編集**タブで**拡張エディタの使用**チェックボックスが選択されます。同様に、WEDIT コマンドを使用して拡張エディタを使用不可にした場合、**拡張エディタの使用** チェックボックスは選択されません。

### 関連項目:

- “セッションプリファレンスについて ” (58 ページ)
- 3 章, “Windows 版での SAS エディタの使用” (79 ページ)

## WEMAILFMT コマンド:Windows

テキストウィンドウの内容を電子メールメッセージに添付する際に使用する形式を指定します。

**Windows 固有:** すべて

### 構文

**WEMAILFMT** TEXT | RTF

#### *必須引数*

**TEXT**
現在の SAS テキストウィンドウの内容をテキストファイルとして添付します。

**RTF**
現在の SAS テキストウィンドウの内容をリッチテキストフォーマットファイル(RTF)として添付します。

### 詳細

現在の SAS ウィンドウにグラフィックが含まれている場合、ウィンドウの内容は自動的にビットマップファイルとして添付されます。

WEMAILFMT コマンドを使用すると、**ウィンドウの内容を電子メールに添付する** チェックボックスは、**プリファレンス** ダイアログボックスの全般タブページで更新されます。

### 関連項目:

- “SAS を使用して電子メールの送信” (44 ページ)
- “WATTACH コマンド:Windows” (356 ページ)
- “DLGSMAIL コマンド:Windows” (344 ページ)

## WEXITSAVE コマンド:Windows

SAS 終了時の設定保存の有効化と無効化を切り替えます。

**Windows 固有:** すべて

## 構文

**WEXITSAVE** <ON | OFF>

### *引数なし*

SAS 終了時の設定保存の有効化と無効化を切り替えます。

### *オプション引数*

**ON**
SAS を終了するときに設定を保存します。

**OFF**
SAS を終了するときに設定を保存しません。

## 詳細

この設定は、**プリファレンス** ダイアログボックスの**全般** ページでも切り替えられます。

## 関連項目:

"セッションプリファレンスについて " (58 ページ)

# WFILE コマンド:Windows

アクティブウィンドウの内容を保存します。

**Windows 固有:** すべて

## 構文

**WFILE**

## 詳細

WFILE コマンドは、アクティブウィンドウの内容をファイルに保存します。

## 関連項目:

- 拡張エディタのセクション"ファイルの保存" (82 ページ)
- プログラムエディタのセクション"ファイルの保存" (114 ページ)

# WHIDECURSOR コマンド:Windows

テキストを入力できない SAS ウィンドウには、カーソルを表示しません。

**Windows 固有:** すべて

### 構文

**WHIDECURSOR** <ON | OFF>

#### *引数なし*

カーソルの表示と非表示を切り替えます。

#### *オプション引数*

**ON**
カーソルを表示しません。

**OFF**
カーソルを表示します。

### 詳細

WHIDECURSOR コマンドを発行すると、SAS/EIS や SAS/AF ソフトウェアなどの、テキストの入力を許可しないウィンドウには、デフォルトのテキストカーソルは表示されません。WHIDECURSOR 設定は、**プリファレンス** ダイアログボックスの **詳細** ページでも切り替えられます。

### 関連項目:

“セッションプリファレンスについて ” (58 ページ)

## WHSBAR コマンド:Windows

水平スクロールバーのオン／オフを切り替えます。

**Windows 固有:** すべて

### 構文

**WHSBAR** <ON | OFF>

#### *引数なし*

水平スクロールバーのオン／オフを切り替えます。

#### *オプション引数*

**ON**
水平スクロールバーを表示します。

**OFF**
水平スクロールバーを表示しません。

### 詳細

この設定は、**プリファレンス** ダイアログボックスの**表示**ページでも切り替えられます。

### 関連項目:

“セッションプリファレンスについて ” (58 ページ)

## WINSERT コマンド:Windows

挿入モードのオン／オフを切り替えます。

**Windows 固有:** すべて

### 構文

**WINSERT** <ON | OFF>

***引数なし***
挿入モードのオン／オフを切り替えます。

***オプション引数***

**ON**
挿入モードを有効化します。

**OFF**
上書きモードを有効化します。

### 詳細

この設定を切り替えるには、キーボードの Insert キーを押すか、または**プリファレンス**ダイアログボックスの**編集** タブページの**上書き入力する** オプションを変更します。

### 関連項目:

“セッションプリファレンスについて ” (58 ページ)

## WMENUPOP コマンド:Windows

SAS アプリケーションウィンドウのポップアップメニューのオン／オフを切り替えます。

**Windows 固有:** すべて

### 構文

**WMENUPOP** <ON | OFF>

***引数なし***
ポップアップメニューのオン／オフを切り替えます。

***オプション引数***

**ON**
ポップアップメニューをオンにします。

**OFF**
ポップアップメニューをオフにします。

## 詳細

デフォルトでは、ポップアップメニューは表示されます。ウィンドウのポップアップメニューを使用するには、ウィンドウのクライアント領域で右マウスボタンをクリックします。

このコマンドを-NOAWSMENU システムオプションと併用すると、ユーザーはすべてのメニュー選択項目を利用できなくなります。これは、エンドユーザーの操作を制限する必要のある SAS/AF アプリケーションを開発するときに役立ちます。

## 関連項目:

- “AWSMENU システムオプション:Windows” (494 ページ)
- “PMENU コマンド:Windows” (350 ページ)
- “WPOPUP コマンド:Windows” (369 ページ)

# WMRU コマンド:Windows

**ファイル** メニューで最近使用したファイル名を保存します。

**Windows 固有:** すべて

## 構文

**WMRU** <<ON> <NUM=*number-of-filenames*> <CASCADE> > | <OFF>

### *引数なし*

ファイルの一覧のオン／オフを切り替えます。

### *オプション引数*

**ON NUM=*number-of-filenames***
ファイルの一覧を有効化し、ファイルの一覧に *number-of-filenames* のファイル名を保持します。*number-of-filenames* 引数には、1 から 30 の整数を指定できます。*number-of-filenames* を省略する場合は、最近使用したファイルの最後に指定した数が使用されます。

**CASCADE**
**ファイル** メニューの**最近使ったファイル** サブメニューからアクセスできる最近使用されたファイルを指定します。

**OFF**
ファイルリストをオフにします。

## 詳細

**ファイルを開くまたは名前を付けて保存** ダイアログボックスを使用してファイルを開くか、または保存すると、**ファイル**メニューまたは**最近使ったファイル** サブメニューの最近使用したファイルリストにファイル名が追加されます。エディタをアクティブウィンドウにして、[ファイル]メニューまたは**最近使ったファイル** サブメニューから名前を選択して、最近使用したファイルを SAS エディタで開くことができます。デフォルトでは、ファイルの一覧には 4 つのファイル名が保持されます。

**プリファレンス**ダイアログボックスの**全般** ページでもこれらの設定を構成できます。

### 関連項目:

“セッションプリファレンスについて ” (58 ページ)

## WNAVKEYUNMARK コマンド:Windows

[ナビゲーションキーによるテキストのマーク解除を有効にする]の設定を切り替えます。

**Windows 固有:** すべて

### 構文

**WNAVKEYUNMARK** <ON | OFF>

***引数なし***

**ナビゲーションキーによるマーク解除を有効にする**設定のオン/オフを切り替えます。

***オプション引数***

**ON**
**ナビゲーションキーによるマーク解除を有効にする**設定をオンにします。

**OFF**
**ナビゲーションキーによるマーク解除を有効にする**設定をオフにします。

### 詳細

**ナビゲーションキーによるマーク解除を有効にする** 設定には、**ツール** ⇨ **オプション** ⇨ **プリファレンス** ⇨ **編集**を選択するとアクセスできます。

**ナビゲーションキーによるマーク解除を有効にする**の設定が選択されている場合、上向き、下向き、左向き、右向きのナビゲーションキーを使用してテキストのマークを解除できます。

### 関連項目:

“セッションプリファレンスについて ” (58 ページ)

## WNEWTITLE コマンド:Windows

アクティブウィンドウの内容とタイトルを消去します。

**Windows 固有:** すべて

### 構文

**WNEWTITLE**

### 詳細

SAS ウィンドウの内容をファイルに保存すると、SAS はそのファイル名をウィンドウのタイトルとして割り当てます。WNEWTITLE コマンドを使用すると、アクティブウィンドウの内容とタイトルが消去され、そのタイトルは(無題)に戻ります。

このコマンドを**ログウィンドウやアウトプット**ウィンドウで使用すると、ウィンドウの内容が消去され、ウィンドウ名が(無題)に変更されます。このコマンドを**プログラムエディタ**ウィンドウで使用する場合は、ウィンドウの内容とタイトルを消去する前に、ウィンドウの内容を保存するかどうか確認するダイアログボックスが表示されます。このコマンドを**拡張エディタ**ウィンドウで使用すると、新しい**拡張エディタ**ウィンドウが(無題)で開きます。

## WNEXTEDIT コマンド:Windows

すべての拡張エディタウィンドウ間で、アクティブな拡張エディタウィンドウを切り替えます。

**Windows 固有:** すべて

### 構文

**WNEXTEDIT**

### 詳細

WNEXTEDIT コマンドを使用して、拡張エディタウィンドウ間を移動できます。

### 関連項目:

"WPGM コマンド:Windows" (368 ページ)

## WPASTE コマンド:Windows

Windows クリップボードの内容をアクティブウィンドウに貼り付けます。

**Windows 固有:** すべて

### 構文

**WPASTE**

### 詳細

WPASTE は、ツールバーコマンドとともに使用します。WPASTE コマンドを入力すると、アクティブウィンドウのメニューから[貼り付け]メニュー項目が検索されます。[貼り付け]メニュー項目が存在し、クリップボードにテキストが保存されている場合、WPASTE を実行すると、メニューから[貼り付け]を選択したときと同じ操作が行なわれます。[貼り付け]メニュー項目がない場合は、PASTE コマンドが実行されます。

### 関連項目:

SAS ヘルプとドキュメントの PASTE コマンド

## WPGM コマンド:Windows

アクティブウィンドウを、最後に編集したエディタウィンドウに変更します。

**Windows 固有:** すべて

### 構文

**WPGM**

### 詳細

WPGM コマンドの動作は、**拡張エディタの使用** チェックボックスの設定で決定します。このチェックボックスは、**プリファレンス**ダイアログボックスの**編集**タブで利用できます。**拡張エディタの使用** チェックボックスを選択して WPGM コマンドを発行すると、アクティブウィンドウは、最後に編集した**拡張エディタ** ウィンドウになります。**拡張エディタの使用** チェックボックスを選択しないと、アクティブウィンドウがプログラムエディタになります。

WPGM コマンドを繰り返し発行すると、最近編集したウィンドウからさかのぼって編集された拡張エディタウィンドウが表示されます。

### 関連項目:

"WNEXTEDIT コマンド:Windows" (368 ページ)

## WPOPUP コマンド:Windows

ウィンドウのポップアップメニューを表示します。

**Windows 固有:** すべて

### 構文

**WPOPUP**

### 詳細

ウィンドウのポップアップメニューを使用するには、ウィンドウのクライアント領域で右マウスボタンをクリックします。Windows では、このコマンドはデフォルトで右マウスボタンに割り当てられています。

### 関連項目:

"WMENUPOP コマンド:Windows" (365 ページ)

## WRTFSAVE コマンド:Windows

現在のテキストウィンドウの内容を RTF ファイルに保存します。

**Windows 固有:** すべて

### 構文

**WRTFSAVE** "*filename*" <NOPROMPT>

### 必須引数

***filename***
必須引数です。ファイルパスを含めることができます。パスなしでファイル名を指定した場合は、ファイルは現在の SAS 作業フォルダに保存されます。WRTFSAVE コマンドは、.rtf ファイル拡張子を自動的に追加しません。ファイル名を.RTF で終わるようにする必要がある場合は、指定するファイル名に.RTF も指定する必要があります。

### オプション引数

**NOPROMPT**
同じファイル名のファイルがすでに存在する場合、確認のためのダイアログボックスを表示せずに既存ファイルを上書きします。

## 詳細

WRTFSAVE コマンドは、アクティブウィンドウの内容を.RTF 形式で保存します。アクティブウィンドウの内容は、テキストである必要があります。このコマンドは、**名前を付けて保存** ダイアログボックスで**.rtf** を、**保存するファイルの種類** リストから選択したときと同じ動作を実行します。しかし、WRTFSAVE では、途中のダイアログボックスを表示せずにファイルを直接保存します。

*注:* WRTFSAVE コマンドは、拡張エディタがある場合、機能しません。

# WSCREENTIPS コマンド:Windows

スクリーンチップのオン／オフを切り替えます。

**Windows 固有:** すべて

## 構文

**WSCREENTIPS** <ON | OFF>

### 引数なし

スクリーンチップのオン／オフを切り替えます。

### オプション引数

**ON**
メイン SAS ウィンドウ内のステータスバー、ウィンドウバー、タブに役立つヒントが表示されます。

**OFF**
スクリーンチップを非表示にします。

## 詳細

スクリーンチップは、メイン SAS ウィンドウのステータスバー、ウィンドウバー、タブにマウスポインタを置くと表示されるヒントです。

WSCREENTIPS コマンドは、メイン SAS ウィンドウのステータスバー、ウィンドウバー、タブにカーソルを移動したときにスクリーンチップを表示するかどうかを指定します。

*注:* スクリーンチップをツールチップと混同しないでください。ツールチップは、ツールの役立つヒントを表示します。**プリファレンス** ダイアログボックスの**表示**ページでもスクリーンチップ設定を切り替えられます。

### 関連項目:

- “TOOLTIPS コマンド:Windows” (355 ページ)
- “セッションプリファレンスについて ” (58 ページ)

## WSTATUSLN コマンド:Windows

ステータス行の表示と非表示を切り替え、各表示領域の割合を指定します。

**Windows 固有:** すべて

### 構文

**WSTATUSLN** <ON | OFF> <ALL | MSG*percent-msg* | CDIR *percent-cdir* <CURPOS> >

#### *引数なし*

ステータスバーの表示と非表示を切り替えます。最初の引数の指定は任意です。指定する場合は、2 番目のオプショングループの前に指定する必要があります。

#### *オプション引数*

**ON**
ステータス行が最後にアクティブだったときの状態で表示します。ステータス行にメッセージ行が表示される場合は、SAS アプリケーションウィンドウにメッセージ行は表示されません。デフォルトの設定は ON です。

**OFF**
ステータスバーを非表示にします。この場合、SAS アプリケーションウィンドウにメッセージ行が表示されます。2 番目の引数グループの指定も任意です。ON オプションを先に指定せずに 2 番目の引数グループのオプションを指定しても、ステータス行が現在無効化されている場合は自動的には有効化されないことに注意してください。

**ALL**
メッセージ行と現在のフォルダの両方をステータスバーに表示します。MSG オプションや CDIR オプションにパーセント値を指定しないと、ステータス行の割合として最後の設定が使用されます。デフォルトの設定は ALL です。

**MSG<=*percent-msg*>**
メッセージ領域をステータスバーの一部として表示します。CDIR オプションを指定せずにこのオプションを指定すると、メッセージ行がステータス行全体に表示されます。このオプションと CDIR オプションに割合を指定すると、メッセージ行は指定された割合でステータス行に表示されます。

**CDIR<=*percent-cdir*>**
現在のフォルダをステータスバーの一部として表示します。MSG オプションを指定せずにこのオプションを指定すると、作業フォルダがステータス行全体に表示されます。このオプションと MSG オプションに割合を指定すると、作業フォルダは指定された割合でステータス行に表示されます。

**CURPOS**
拡張エディタがアクティブウィンドウの場合、拡張エディタのカーソルの位置(行と列)をステータスバーに表示します。

### 詳細

WSTATUSLN コマンドは、アクティブウィンドウのステータスバーの表示と非表示、およびメッセージ行と現在のフォルダがステータスバーで占める割合を指定します。ステータスバーは、**プリファレンス**ダイアログボックスの**表示** ページでも切り替えられます。

### 例

メッセージと現在のフォルダが均等に表示されるステータスバーを表示するには、次のコマンドを発行します。

```
wstatusln on msg=50 cdir=50
```

### 関連項目:

“セッションプリファレンスについて ” (58 ページ)

## WUNDO コマンド:Windows

切り取り、コピー、貼り付けツールバーの操作を元に戻します。

**Windows 固有:** すべて

### 構文

**WUNDO**

### 詳細

WUNDO コマンドを入力すると、アクティブウィンドウのメニューから元に戻すメニュー項目が検索されます。元に戻すメニュー項目がある場合は、WUNDO はメニューから **UNDO** を選択した場合の動作を実行します。元に戻すメニュー項目がない場合は、WUNDO は UNDO コマンドを実行します。UNDO コマンドがないウィンドウも一部存在します。

## WVSBAR コマンド:Windows

垂直スクロールバーの表示と非表示を切り替えます。

**Windows 固有:** すべて

### 構文

**WVSBAR** <ON | OFF>

***引数なし***
垂直スクロールバーの表示と非表示を切り替えます。

### *オプション引数*

**ON**
垂直スクロールバーを表示します。

**OFF**
垂直スクロールバーを非表示にします。

## 詳細

この設定は、**プリファレンス** ダイアログボックスの**表示**ページでも切り替えられます。

## 関連項目:

“セッションプリファレンスについて ” (58 ページ)

---

# WWINDOWBAR コマンド:Windows

ウィンドウバーの表示と非表示を切り替えます。

**Windows 固有:** すべて

## 構文

**WWINDOWBAR** <ON | OFF>

### *引数なし*

ウィンドウバーの表示と非表示を切り替えます。

### *オプション引数*

**ON**
メイン SAS ウィンドウにウィンドウバーを表示します。

**OFF**
メイン SAS ウィンドウにウィンドウバーを表示しません。

## 詳細

ウィンドウバーは、メイン SAS ウィンドウの下部にあるステータス行の上に表示されます。SAS ウィンドウは、ウィンドウバーに最小化されます。ウィンドウバーのボタンをクリックすれば、対応するウィンドウを手前に表示できます。エディタなどの開いているアプリケーションからファイルを開くには、ファイルをウィンドウバーのアプリケーションボタンにドラッグし、そのアプリケーションを手前に表示し、そのままファイルをアプリケーションウィンドウにドラッグします。ウィンドウバーが表示されない場合は、SAS ウィンドウはタイトルバーに最小化されます。

ウィンドウバーの表示と非表示は、**プリファレンス** ダイアログボックスの **表示** ページでも切り替えられます。

## 関連項目:

- “コマンドを使用した SAS ウィンドウ環境のカスタマイズ ” (62 ページ)
- “セッションプリファレンスについて ” (58 ページ)

## X コマンド:Windows

SAS セッションを終了せずにホストシステムモードに移行するか、または Windows コマンドをサブミットできます。

**Windows 固有:** 次に対する有効な値:*コマンド*構文

### 構文

**X** <'*command* '> ;

#### *引数なし*

DOS コマンドウィンドウを開きます。

#### *オプション引数*

***command***
実行するコマンドを指定します。

### 詳細

この構文の X コマンドは、1 つのコマンドを発行します。コマンドは、動作環境に渡され、実行されます。エラーが発生した場合は、エラーメッセージが表示されます。

### 関連項目:

- SAS ヘルプとドキュメントの“X コマンド“
- “SAS から Windows または MS-DOS のコマンドを実行する ” (28 ページ)
- “XCMD システムオプション Windows” (590 ページ)
- “XSYNC システムオプション:Windows” (591 ページ)
- “XWAIT システムオプション:Windows” (592 ページ)

## ZOOM コマンド:Windows

アクティブウィンドウを最大化します。

**Windows 固有:** すべて

### 構文

**ZOOM** <ON | OFF>

#### *オプション引数*

**ON**
アクティブウィンドウを最大化します。

**OFF**
アクティブウィンドウをデフォルトのサイズに戻します。

## 詳細

1 つのアプリケーションウィンドウを最大化すると、SAS ウィンドウ環境は最大化モードに切り替わります。アクティブウィンドウ間の切り替えを行うと、選択するウィンドウすべてが最大化されます。アプリケーションウィンドウのいずれかを元のサイズに戻すと、すべてのウィンドウが元のサイズに戻ります。

## 関連項目:

SAS ヘルプとドキュメントの ZOOM コマンド

*17 章*

# Windows 版に固有の SAS 出力形式

**Windows 版に固有の SAS 出力形式** . . . . . **377**
**バイナリデータの書き込み** . . . . . **377**
**以前のリリースのユーザー定義入力形式を SAS 9.4 で利用する** . . . . . **378**
**ディクショナリ** . . . . . **378**
HEXw.出力形式:Windows . . . . . 378
$HEXw.出力形式:Windows . . . . . 379
IBw.d 出力形式:Windows . . . . . 380
PDw.d 出力形式:Windows . . . . . 382
PIBw.d 出力形式:Windows . . . . . 383
RBw.d 出力形式:Windows . . . . . 384
ZDw.d 出力形式:Windows . . . . . 386

## Windows 版に固有の SAS 出力形式

SAS 出力形式は、データ値を書き込むために SAS が使用する命令またはテンプレートです。ほとんどの SAS 形式は完全に *SAS 出力形式と入力形式: リファレンス*で記述されます。ここでは、Windows 版固有の動作を含む出力形式について説明します。

Windows 動作環境に固有の SAS 出力形式の多くは、バイナリデータの書き出しで使用されます。次の出力形式を使うには、“バイナリデータの書き込み” (377 ページ) に示されている概念を理解することが重要です。

SAS の以前のバージョンで作成した出力形式がある場合は、SAS 9.4 で使うためにそれらの出力形式を変換する方法の詳細については、“以前のリリースのユーザー定義入力形式を SAS 9.4 で利用する” (378 ページ) を参照してください。

## バイナリデータの書き込み

IBM メインフレームや Hewlett-Packard 9000 を始めとする多くの UNIX システムでは、バイトはビッグエンディアンと呼ばれる順序で格納されます。Intel ベースシステムや IBM 互換マイクロコンピュータ、旧 Compaq 製の VAX と Alpha コンピュータでは、バイト反転やリトルエンディアンと呼ばれる別の順序でバイトは格納されます。

ある順序で格納されているバイナリデータを、別の順序でバイナリデータを格納するコンピュータで読み取るには、追加処理が必要です。SAS アプリケーションを設計すると

きは、データがどのように読み取られるかを考慮し、それにしたがって出力形式および入力形式を選択するようにします。

SAS には、バイナリデータの読み取り用に 2 種類の入力形式と、それに対応するバイナリデータ書き込み用の出力形式が用意されています。

- IB*w.d*、PD*w.d*、PIB*w.d*、RB*w.d* はネイティブモードでの読み書きの入力形式と出力形式で、つまり、コンピュータ標準であるバイト順序システムを使用します。
- S370FIB*w.d*、S370FPD*w.d*、S370FRB*w.d*、S370FPIB*w.d* コンピュータのネイティブモードにかかわらず、IBM 370 標準に基づく読み書きの入力形式と出力形式を使用します。これらの入力形式と出力形式を使用すると、数値データがどれだけ保存されているかにかかわらず、どの SAS 環境でも実行できる SAS プログラムを記述できます。

バイナリデータの読み取りと書き込みを行う SAS プログラムが 1 種類のコンピュータ上だけで実行されている場合は、ネイティブモードの入力形式と出力形式を使用することができます。それに対し、異なるバイト格納方式を使用する複数のコンピュータ上で実行できる SAS プログラムを記述する場合は、IBM 370 出力形式および入力形式を使用します。IBM 370 入力形式と出力形式の目的は、数値データの格納に用いるバイト格納方式に関係なく、どのような SAS 環境でも実行できる SAS プログラムを記述できるようにすることです。

たとえば、PIB*w.d* 形式でデータを書き込むプログラムがあるとします。バイト反転モードでデータを保存できるように、このプログラムをマイクロコンピュータで実行します。そのとき、PC 上では、PIB*w.d* 情報を使用してデータを読み込む別の SAS プログラムを実行します。両方のプログラムとも、PC 上で実行されるので、データは正しく読み取られます。ただし、データはマイクロコンピュータに最適化されてはいても、Hewlett Packard 9000 には最適化されてはいないため、Hewlett Packard 9000 シリーズのコンピュータにデータをアップロードして正しく読み込むことはできません。この問題を回避するには、S370FPIB*w* を使用します。*d* 形式を使用してデータを書き込んでください。マイクロコンピュータ上でも、IBM370 モードでデータの保存が可能です。それから、S370FPIB*w.d* 入力形式を使用してデータを読み込んでください。データを読み取るときにどの種類のコンピュータを使用しても、データは正しく読み取られます。

# 以前のリリースのユーザー定義入力形式を SAS 9.4 で利用する

64 ビットの Windows 版 SAS を使用している場合、PROC CPORT と CIMPORT を使用して変換する必要があります。PROC CPORT と CIMPORT に関する詳細については、*Base SAS プロシジャガイド*を参照してください。

32 ビットの Windows 版 SAS を使用し、SAS ライブラリを移行する必要がある場合は、PROC MIGRATE メソッドを使用することをお勧めします。

*注:* ユーザー定義の出力形式はカタログエントリーとして保存されます。

# ディクショナリ

## HEX*w.*出力形式:Windows

バイナリ実数(浮動小数点数)を 16 進値に変換します。

**カテゴリ:** 数値
**配置:** 左
**Windows 固有:** ネイティブ浮動小数点表現
**参照項目:** “HEXw. Format” (*SAS Formats and Informats: Reference*)

## 構文

**HEX***w*.

### *必須引数*

***w***
出力フィールドの幅を指定します。*w* に 1 から 15 の値を指定すると、バイナリ実数は、固定小数点整数に切り捨てられてから 16 進表記に変換されます。*ww* の値に 16 を指定する場合、その数の浮動小数点値が使われます。すなわち、その数は切り捨てられません。

**デフォルト** 8

**範囲** 1–16

## 例

次の例では 123 の入力値を使います。

```
data _null_; x=123; put x=hex8.; run;
```

```
x=0000007B
```

## 関連項目:

**出力形式:**

- “$HEXw.出力形式:Windows” (379 ページ)

**入力形式:**

- “HEXw.入力形式:Windows” (417 ページ)

# $HEX*w*.出力形式:Windows

文字値を 16 進値に変換します。

**カテゴリ:** 文字
**配置:** 左
**Windows 固有:** ASCII 文字エンコーディングシステム
**参照項目:** $HEXw.

## 構文

**$HEX***w*.

### *必須引数*

***w***
出力フィールドの幅を指定します。

**デフォルト** 2

**範囲** 1-32767

## 詳細

$HEX*w*.出力形式は HEX*w*.出力形式に類似しており、1 バイトが 2 カラムで表現される 16 進表記に文字値を変換します。Windows では、$HEX*w* 出力形式は、文字を ASCII コードの 16 進表現で出力します。

## 例

次の例では 123 の入力値を使います。

```
data _null_; x='123'; put x=$hex8.; run;
```

```
x=313233
```

## 関連項目:

**出力形式:**

- “HEXw.出力形式:Windows” (378 ページ)

**入力形式:**

- “$HEXw.入力形式:Windows” (418 ページ)

# IB*w.d* 出力形式:Windows

値をバイナリ整数(固定小数点数)形式で書き出します。

**カテゴリ:** 数値
**配置:** 左
**Windows 固有:** ネイティブ浮動小数点表現
**参照項目:** “IBw.d Format” (*SAS Formats and Informats: Reference*)

## 構文

**IB***w.d*

### 必須引数

***w***
出力フィールドの幅を桁ではなくバイトで指定します。

**デフォルト** 4

**範囲** 1-8

***d***
倍率を指定します。*d* 値を指定すると、IB*w.d* 出力形式は、数を $10^d$ で乗算し、バイナリ整数出力形式をその値に適用します。

**範囲** 0-10

## 詳細

IB*w.d* 出力形式は、倍精度数値を変換し、バイナリ整数(固定小数点数)値として書き出します。整数は、バイナリ整数(固定小数点数)形式で格納されます。

マイクロコンピュータの固定小数点数の詳細については、Intel 社の開発者向け Web サイトを参照してください。

## 例

### 例 1: 正数の処理

1.0 を倍精度数値としてフォーマットする場合は、整数として格納されます。

```
01 00 00 00 00 00 00 00
```

Windows は、バイナリデータのバイト順序を逆にして格納することを忘れないでください。書き出される値は、ユーザーが指定する *w* 値によって変わります。

IB4.出力形式を指定すると、次の値が書き出されます。

```
01 00 00 00
```

IB2.出力形式を指定すると、次の値が書き出されます。

```
01 00
```

### 例 2: 負数の処理

-1 を IB4.出力形式でフォーマットしようとすると、次の値が書き出されます。

```
FF FF FF FF
```

IB2.出力形式を指定すると、次の値が書き出されます。

```
FF FF
```

### 例 3: 大きすぎてフォーマットできない数の処理

数値が大きすぎてフォーマットできない場合、結果は、4 バイトで格納できる最大整数値 2,147,483,647 になります。

次のコードでは

```
data a;
   x = 9999999999999999999;
   y = put(x, IB8.);
   put y = hex16.;
```

```
run;
```

SAS は 2147483647 の 16 進表現を返します。

```
 y=FFFFFFFFFFFFFF7F
```

### 関連項目:

**入力形式:**

- “IBw.d 入力形式:Windows” (419 ページ)
- “バイナリデータの書き込み” (377 ページ)

## PD*w.d* 出力形式:Windows

値をパック 10 進データで書き出します。

**カテゴリ:** 数値

**配置:** 左

**Windows 固有:** 値が負または正の数として解釈される仕組み

**参照項目:** “PDw.d Format” (*SAS Formats and Informats: Reference*)

### 構文

**PD***w.d*

#### *必須引数*

***w***

出力フィールドの幅を桁ではなくバイトで指定します。

**デフォルト** 1

**範囲** 1–16

***d***

倍率を指定します。*d* 値を指定すると、PD*w.d* 出力形式は、数を $10^d$ で乗算し、パック 10 進出力形式をその値に適用します。

**範囲** 1-31

### 詳細

PD*wd* 出力形式はパック 10 進出力形式で倍精度数を書き込みます。パック 10 進データでは、1 バイトに 2 桁の数字を含みます。*w* 値は、桁数ではなくバイト数を表します。値の符号は、最初のバイト全体が符号を表すために使用されている場合でも、最初のバイトの最上位ビットで示されます。

## 例

### *例 1: 正数の処理*

1143.0 を PD2.出力形式でフォーマットすると、次の値が書き出されます。

```
00 99
```

PD4.を指定すると、次の値が書き出されます。

```
00 00 11 43
```

### *例 2: 負数の処理*

-1143.0 を PD2.出力形式でフォーマットすると、次の値が書き出されます。

```
80 43
```

PD4.出力形式を指定すると、次の値が書き出されます。

```
80 00 11 43
```

### *例 3: 大きすぎてフォーマットできない数の処理*

この例のように数値が大きすぎてフォーマットできない場合

```
data a;
   x = 1e308;
   y = put(x, PD16.2);
   put y = hex16.;
run;
```

結果

```
y=0099999999999999
```

## 関連項目:

**入力形式:**

- “PDw.d 入力形式:Windows” (420 ページ)
- “バイナリデータの書き込み” (377 ページ)

# PIB*w.d* 出力形式:Windows

値を正のバイナリ整数(固定小数点数)形式で書き出します。

**カテゴリ:** 数値

**配置:** 左

**Windows 固有:** ネイティブバイトスワップ整数

**参照項目:** PIBw.d

## 構文

**PIB***w.d*

### 必須引数

***w***
出力フィールドの幅を桁ではなくバイトで指定します。

**デフォルト** 1

**範囲** 1-18

***d***
倍率を指定します。*d* 値を指定すると、PIB*w.d* 出力形式は、数を $10^d$ で乗算し、正のバイナリ整数出力形式をその値に適用します。

**範囲** 0-10

## 詳細

PIB*w.d* 出力形式は固定点値を整数バイナリ値へと変換します。固定点値が負の場合、PIB*wd* 出力形式は-1 の表示に整数を書き込みます。

マイクロコンピュータの固定小数点数の詳細については、Intel 社の開発者向け Web サイトを参照してください。

## 例: 大きすぎてフォーマットできない数の処理

数値が大きすぎてフォーマットできない場合、結果は、4 バイトで格納できる最大整数値 2,147,483,647 になります。

次のコードでは

```
data a;
   x = 999999999999999999;
   y = put(x, PIB8.);
   put y = hex16.;
run;
```

SAS は 2147483647 の 16 進表現を返します。

```
 y=0000E8890423C78A
```

## 関連項目:

- “Byte Ordering for Integer Binary Data on Big Endian and Little Endian Platforms” (*SAS Formats and Informats: Reference* 1 章)
- “Byte Ordering for Integer Binary Data on Big Endian and Little Endian Platforms” (*SAS Formats and Informats: Reference* 3 章)

### 入力形式:

- “PIBw.d 入力形式:Windows” (421 ページ)
- “バイナリデータの書き込み” (377 ページ)

# RB*w.d* 出力形式:Windows

値をバイナリ実数(浮動小数点数)形式で書き出します。

**カテゴリ:** 数値
**配置:** 左
**デフォルト:** 4
**範囲:** 2-8, 0-10
**Windows 固有:** ネイティブ浮動小数点表現
**参照項目:** “RBw.d Format” (*SAS Formats and Informats: Reference*)

## 構文

**RB***w.d*

### *必須引数*

***w***
出力フィールドの幅を指定します。

**デフォルト** 4

**範囲** 2-8

***d***
倍率を指定します。*d* 値を指定すると、RB*w.d* 出力形式は、数を $10^d$ で乗算し、バイナリ実数出力形式をその値に適用します。

**範囲** 0-10

## 詳細

RB*w.d* 出力形式は、数値データをバイナリ実数(浮動小数点)表記で書き出します。一般に、科学計算では数値データは浮動小数点表記で表現されます(SAS はすべての数値を浮動小数点表記で保存します)。浮動小数点数は、次の 2 つの部分から構成されます。すなわち、値を表す仮数部と、値の大きさを表す指数部です。

バイナリ実数は、数値を表す最も効率的な形式です。SAS はもともと、この形式で数値を表すので、変換が必要ありません。

Windows の浮動小数点表記の詳細については、Intel 社の開発者向け Web サイトを参照してください。

## 例: 大きすぎてフォーマットできない数の処理

この例のように数値が大きすぎてフォーマットできない場合

```
data a;
   x = 1e308;
   y = put(x, RB8.2);
   put y = hex16.;
run;
```

結果

```
y=0000000000D1FFFF
```

### 関連項目:

- “Byte Ordering for Integer Binary Data on Big Endian and Little Endian Platforms” (*SAS Formats and Informats: Reference* 1 章)

#### 入力形式:

- “Byte Ordering for Integer Binary Data on Big Endian and Little Endian Platforms” (*SAS Formats and Informats: Reference* 3 章)
- “RBw.d 入力形式:Windows” (422 ページ)
- “バイナリデータの書き込み” (377 ページ)

## ZD*w.d* 出力形式:Windows

ゾーン 10 進でデータを書き出します。

**カテゴリ:** 数値

**配置:** 左

**Windows 固有:** 最後のバイトには、符号が含まれます。

**参照項目:** “ZDw.d Format” (*SAS Formats and Informats: Reference*)

### 構文

**ZD***w.d*

#### *必須引数*

***w***

桁数ではなくバイト数を指定します。

**デフォルト** 1

**範囲** 1-32

***d***

数値の小数点部の桁数を指定します。

**範囲** 1-10

### 詳細

ZD*wd* 出力形式はゾーン 10 進でデータを書き出します。これは、オーバプリント後置数値形式としても知られています。Windows 動作環境では、フィールドの最後のバイトには、数値の符号情報が含まれます。次の表は、最後のバイトの変換方法を示しています。

***表 17.1*** *最後のバイト変換*

| 数字 | ASCII 文字 | 数字 | ASCII 文字 |
|---|---|---|---|
| 0 | { | -0 | } |

| 数字 | ASCII 文字 | 数字 | ASCII 文字 |
|---|---|---|---|
| 1 | A | -1 | J |
| 2 | B | -2 | K |
| 3 | C | -3 | L |
| 4 | D | -4 | M |
| 5 | E | -5 | N |
| 6 | F | -6 | O |
| 7 | G | -7 | P |
| 8 | H | -8 | Q |
| 9 | I | -9 | R |

## 例: 大きすぎてフォーマットできない数の処理

この例のように数値が大きすぎてフォーマットできない場合

```
data a;
   x = 1e308;
   y = put(x, ZD32.2);
   put y = hex16.;
run;
```

結果は 39 の連続になります。

```
y=3939393939393939
```

## 関連項目:

“ZDw.d 入力形式:Windows” (423 ページ)

*18 章*

# Windows 版に固有の SAS 関数と CALL ルーチン

**Windows 版に固有の SAS 関数と CALL ルーチン** . . . . . **389**
概要:Windows 版に固有の SAS 関数と CALL ルーチン . . . . . 389

**ディクショナリ** . . . . . **390**
BYTE 関数:Windows . . . . . 390
CALL SOUND ルーチン:Windows . . . . . 390
CALL SYSTEM ルーチン:Windows . . . . . 391
COLLATE 関数:Windows . . . . . 393
DINFO 関数:Windows . . . . . 394
DOPEN 関数:Windows . . . . . 395
DOPTNAME 関数:Windows . . . . . 395
DOPTNUM 関数:Windows . . . . . 396
FDELETE 関数:Windows . . . . . 397
FEXIST 関数:Windows . . . . . 397
FILEEXIST 関数:Windows . . . . . 398
FILENAME 関数:Windows . . . . . 398
FILEREF 関数:Windows . . . . . 399
FINFO 関数:Windows . . . . . 400
FOPTNAME 関数:Windows . . . . . 402
FOPTNUM 関数:Windows . . . . . 403
LIBNAME 関数:Windows . . . . . 404
MCIPISLP 関数:Windows . . . . . 405
MCIPISTR 関数:Windows . . . . . 406
MODULE 関数:Windows . . . . . 407
PEEKLONG 関数:Windows . . . . . 409
RANK 関数:Windows . . . . . 410
SLEEP 関数:Windows . . . . . 410
TRANSLATE 関数:Windows . . . . . 411
WAKEUP 関数:Windows . . . . . 412

## Windows 版に固有の SAS 関数と CALL ルーチン

### *概要:Windows 版に固有の SAS 関数と CALL ルーチン*

SAS 関数は、演算またはシステム操作の結果の値を返します。SAS CALL ルーチンは、変数値を変更するため、または他のシステム関数を実行するために使用します。大半の SAS 関数と CALL ルーチンについては、*SAS 関数と CALL ルーチン: リファレンス*の SAS 関数と CALL ルーチン部分に詳細に記述されています。

# ディクショナリ

## BYTE 関数:Windows

指定した値と対応する ASCII 照合順序の文字を返します。

**カテゴリ:** 文字

**Windows 固有:** ASCII コード順序を使用します。

**参照項目:** “BYTE Function” (*SAS Functions and CALL Routines: Reference*)

### 構文

**BYTE**(*n*)

#### *必須引数*

***n***
特定の ASCII 文字を表す整数を指定します。*n* の値は、0 から 255 の範囲で指定できます。

### 詳細

BYTE 関数から、長さがまだ割り当てられていない変数に値が返される場合、デフォルトでは変数に 1 の長さが割り当てられます。

Windows は ASCII システムを採用しているため、BYTE 関数は、ASCII 照合順序で *n* 番目の文字を返します。*n* の値は、0 から 255 の範囲で指定できます。

プログラムに ASCII 127 (16 進数表記では`'7F'x`)以降の文字を返す BYTE 関数を使用する場合、各国で異なる値が返される可能性があります。ASCII 127 以降の文字は各国語文字であり、国ごとに文字が異なるからです。

## CALL SOUND ルーチン:Windows

特定の周波数と期間で音声を生成します。

**カテゴリ:** 特殊

**Windows 固有:** すべて

### 構文

**CALL SOUND**(*frequency*,*duration*);

#### *必須引数*

***frequency***
音声周波数をヘルツ単位で指定します。20 から 20,000 までの範囲で周波数を指定できます。

***duration***
音声の期間をミリ秒単位で指定します。デフォルトは-1 です。

### 例: トーンの生成

次のステートメントは、周波数が 523 ヘルツ(中央ハ)で、周期が 2 秒のトーンを作成します。

```
data _null_;
   call sound(523,2000);
run;
```

## CALL SYSTEM ルーチン:Windows

オペレーティングシステムコマンドまたは Windows アプリケーションをサブミットして実行します。

**カテゴリ:** 特殊

**Windows 固有:** *command* には、有効な Windows コマンドを指定してください。

**参照項目:** “CALL SYSTEM Routine” (*SAS Functions and CALL Routines: Reference*)

### 構文

**CALL SYSTEM**(*command*);

#### *必須引数*

***command***
次のいずれかを指定できます。

- 引用符で囲んだオペレーティングシステムコマンドまたは引用符で囲んだ Windows アプリケーション名。
- 値がオペレーティングシステムコマンドまたは Windows アプリケーション名の式。
- 値がオペレーティングシステムコマンドまたは Windows アプリケーション名の文字変数名。

### 詳細

SAS を対話的に実行している場合、コマンドプロンプトウィンドウでコマンドを実行します。デフォルトでは、SAS セッションに戻るために **exit** と入力する必要があります。

*注:* NOXCMD で SAS を起動した場合、CALL SYSTEM 関数は使用できません。

### 比較

CALL SYSTEM ルーチンは、X コマンドと同様です。ただし、CALL SYSTEM ルーチンの場合は呼び出すことができ、したがって条件付きで実行できます。

XSYNC システムオプションと XWAIT システムオプションの値は、CALL SYSTEM ルーチンの処理に影響します。

## 例

### *例 1: オペレーティングシステムコマンドを条件付きで実行する*

条件付きのオペレーティングシステムコマンドを実行するには、CALL SYSTEM ルーチンを使用します。

```
options noxwait;
data _null_;
   input flag $ name $8.;
   if upcase(flag)='Y' then
      do;
         command='md c:\'||name;
         call system(command);
      end;
   datalines;
Y mydir
Y junk2
N mydir2
Y xyz
;
```

この例では、変数 FLAG の値を使用して条件付きでディレクトリを作成します。DATA ステップが実行されると、次の 3 つのディレクトリが作成されます。C:\MYDIR、C:\JUNK2、および C:\XYZ.ディレクトリ C:\MYDIR2 は、対応するオブザベーションの FLAG の値が **Y** ではないので、作成されません。

X コマンドは、グローバル SAS ステートメントです。したがって、条件付きで X コマンドを実行できないので注意してください。たとえば、次のコードをサブミットするには、X ステートメントを実行します。

```
data _null_;
   answer='n';
   if upcase(answer)='y' then
      do;
         x 'md c:\extra';
      end;
run;
```

この場合、ANSWER の値が**'n'**または**'y'**と等しいかどうかにかかわらず、ディレクトリ C:\EXTRA が作成されます。

### *例 2: ディレクトリリストの取得*

CALL SYSTEM ルーチンを使用して、次に示すディレクトリを取得できます。

```
data _null_;
   call system('dir /w');
run;
```

この例では、Windows で DIR コマンドの/W オプションを使用して、縦方向の一覧表示形式ではなくワイド形式でディレクトリを出力します。

## 関連項目:

- “X コマンド:Windows” (374 ページ)
- “XSYNC システムオプション:Windows” (591 ページ)
- “XWAIT システムオプション:Windows” (592 ページ)

# COLLATE 関数:Windows

ASCII 照合順序と対応する文字列を返します。

**カテゴリ:** 文字

**Windows 固有:** ASCII コード順序を使用します。

**参照項目:** “COLLATE Function” (*SAS Functions and CALL Routines: Reference*)

## 構文

**COLLATE** (*start-position*< ,*end-position*> )|(*start-position*<,,*length*> )

### *必須引数*

***start-position***
返すべき 1 文字目の照合順序における数値的位置を指定します。

### *オプション引数*

***end-position***
返すべき最後の照合順序における数値的位置を指定します。

***length***
返される照合順序の文字列の文字数を指定します。

## 詳細

COLLATE 関数は、値が 0 から 255 までの ASCII 文字列を返します。COLLATE 関数で返される文字列は、*start-position* 引数で指定した ASCII 文字で始まります。*end-position* 引数を指定した場合は、*start-position* 引数と *end-position* 引数の間の ASCII 文字をすべて含む文字列が COLLATE 関数によって返されます。*end-position* 引数の代わりに*長さ*引数を指定した場合は、COLLATE 関数は、*長さ*の値を含む文字列を返します。255 を超える文字数が含まれる文字列を要求した場合、255 文字で終わる、つまり切り捨てられた文字列を返します。

返される文字列の値のデフォルト長は 200 文字です。201 から 256 の長さの ASCII 文字を返すには、返される文字列変数に$256 などの出力形式を使用するか、または **length y $260** などの変数の長さを明示的に定義します。

プログラムに ASCII 127 (16 進表記では**'7F'x**)以降の文字を返す COLLATE 関数を使用する場合、各国で異なる値が返される可能性があります。ASCII127 以降の文字は各国語文字であり、国ごとに文字が異なるからです。

## 例

### *例 1: 戻り変数のデフォルト文字列長を使用して ASCII 文字列を返す*

この例では、リターンコード変数 **y** は、返される文字列のデフォルトの長さ 200 を使用します。したがって、COLLATE 関数は 200 文字の照合順序文字列を返します。

```
data _null_;
   y = collate(1,256);
   put y;
run;
```

### 例 2: 戻り変数のデフォルト文字列長よりも長い ASCII 文字列を返す場合

リターンコード変数の長さを 256 文字より長くすると、COLLATE 関数は 256 文字の照合順序文字列を返します。

```
data _null_;
   format y $260.;
   y = collate(1,256);
   put y;
run;
```

### 例 3: 特定の長さの ASCII 文字列を返す場合

この例では、リターンコード変数 y は返される文字列の長さに 56 を使用し、COLLATE 関数は、照合順序の最初の 56 文字を返します。

```
data _null_;
   y = collate(,,56);
   put y;
run;
```

## DINFO 関数:Windows

ディレクトリに関する情報を返します。

**カテゴリ:** 外部ファイル

**Windows 固有:** ディレクトリのパス名のみ取得できます。

**参照項目:** “DINFO Function” (*SAS Functions and CALL Routines: Reference*)

### 構文

**DINFO**(*directory-id*, *info-item*)

#### 必須引数

***directory-id***
ディレクトリ ID を指定します。ディレクトリ ID は、通常、DOPEN 関数からディレクトリを開くときに割り当てられます。

***info-item***
取得する情報項目を指定します。*info-item* の値が無効の場合、DINFO 関数は空白を返します。

### 詳細

DOPEN 関数を使用して開いたディレクトリは、directory-id を使用して指定します。取得できるシステム依存のディレクトリの情報項目名を調べるには、DOPTNAME 関数を使用します。取得できるディレクトリの情報項目数を調べるには、DOPTNUM 関数を使用します。

Windows 版では、*info-item* として、*directory-id* のパス名である Directory のみ指定できます。*directory-id* が連結ディレクトリのリストを参照する場合、Directory は連結ディレクトリ名のリストになります。

### 例: ディレクトリ情報の取得

```
data a; rc=filename("tmpdir", "c:"); put "rc = 0 if the directory exists:" rc=; did=dope
```

***ログ 18.1*** *SAS ログに表示されるディレクトリ情報*

```
NOTE:PROCEDURE PRINTTO used (Total process time): real time 0.03 seconds cpu
time 0.00 seconds 446 data a; 447 rc=filename("tmpdir", "c:"); 448 put "rc = 0
if the directory exists:" rc=; 449 did=dopen("tmpdir"); 450 put did=; 451
numopts=doptnum(did); 452 put numopts=; 453 do i = 1 to numopts; 454 optname =
doptname(did,i); 455 put i= optname=; 456 optval=dinfo(did,optname); 457 put
optval=; 458 end; 459 run; rc = 0 if the directory exists:rc=0 did=1 numopts=1
i=1 optname=Directory optval=C:\TEMP\elimal NOTE:The data set WORK.A has 1
observations and 6 variables.NOTE:DATA statement used (Total process time): real
time 0.08 seconds cpu time 0.04 seconds 460 proc printto; run;
```

### 関連項目:

"DOPEN 関数:Windows" (395 ページ)

## DOPEN 関数:Windows

ディレクトリを開き、ディレクトリ ID の値を返します。

**カテゴリ:** 外部ファイル

**Windows 固有:** 環境変数を使用して *fileref* を割り当てられます。

**参照項目:** "DOPEN Function" (*SAS Functions and CALL Routines: Reference*)

### 構文

**DOPEN**(*"fileref"*)

#### *必須引数*

***fileref***
ディレクトリに割り当てるファイル参照名を指定します。

### 詳細

DOPEN はディレクトリを開き、ディレクトリ ID の値(0 より大きい数値)を返します。この値は、開いたディレクトリを他の SAS 外部ファイルアクセス関数で指定する際に使用されます。ディレクトリを開けない場合は、DOPEN は 0 を返します。開くディレクトリは、ファイル参照名で指定してください。

## DOPTNAME 関数:Windows

ディレクトリの情報項目名を返します。

**カテゴリ:** 外部ファイル

**Windows 固有:** directory のみ指定できます。

**参照項目:** "DOPTNAME Function" (*SAS Functions and CALL Routines: Reference*)

### 構文

**DOPTNAME**(*directory-id*, *nval* )

#### *必須引数*

***directory-id***
ディレクトリ ID を指定します。ディレクトリ ID は、通常、DOPEN 関数からディレクトリを開くときに割り当てられます。

***nval***
オプションのシーケンス番号を指定します。

### 詳細

Windows 版では、ディレクトリの情報項目として、*directory-id* のパス名である Directory のみを指定できます。Directory の *nval* (シーケンス番号)は、1 です。*directory-id* が連結ディレクトリのリストを参照する場合、Directory は連結ディレクトリ名のリストになります。

### 例

DOPTNAME の使用例については、“例: ディレクトリ情報の取得” (395 ページ)を参照してください。

## DOPTNUM 関数:Windows

指定できるディレクトリの情報項目数を返します。

**カテゴリ:** 外部ファイル

**Windows 固有:** directory のみ指定できます。

**参照項目:** “DOPTNUM Function” (*SAS Functions and CALL Routines: Reference*)

### 構文

**DOPTNUM**(*directory-id*)

#### *必須引数*

***directory-id***
ディレクトリ ID を指定します。ディレクトリ ID は、通常、DOPEN 関数からディレクトリを開くときに割り当てられます。

### 詳細

Windows 版では、ディレクトリの情報項目として、Directory という名前の項目のみ指定できます。その値は、*directory-id* のパス名またはパス名のリストとなります。また、項目のシーケンス番号は 1 です。1 つのディレクトリでは、1 つの情報項目だけが使用できるので、この関数は 1 の値を返します。

### 例

DOPTNAME の使用例については、“例: ディレクトリ情報の取得” (395 ページ)を参照してください。

## FDELETE 関数:Windows

外部ファイルまたは空のディレクトリを削除します。

**カテゴリ:** 外部ファイル

**Windows 固有:** 環境変数を使用して *fileref* を割り当てられます。

**参照項目:** “FDELETE Function” (*SAS Functions and CALL Routines: Reference*)

### 構文

**FDELETE**("*fileref*")

#### *必須引数*

***fileref***
外部ファイルまたはディレクトリに割り当てるファイル参照名を指定します。ファイル参照名は、連結されたファイル名またはディレクトリのリストには割り当てられません。ファイル参照名をディレクトリに割り当てる場合、ディレクトリは空である必要があります。ファイルを削除するには、権限が必要です。Windows 版では、*fileref* に環境変数を指定できます。指定するファイル参照名または環境変数は、引用符で囲む必要があります。

### 詳細

FDELETE は、操作が正常に行われた場合は 0 を返し、正常に行われなかった場合は 0 以外の数を返します。

## FEXIST 関数:Windows

ファイル参照名を使用して外部ファイルの存在を確認します。

**カテゴリ:** 外部ファイル

**Windows 固有:** *fileref* に環境変数を割り当てられます。

**参照項目:** “FEXIST Function” (*SAS Functions and CALL Routines: Reference*)

### 構文

**FEXIST**("*fileref*")

#### *必須引数*

***fileref***
外部ファイルに割り当てるファイル参照名を指定します。Windows 版では、*fileref* に環境変数を指定することもできます。指定するファイル参照名または環境変数は、引用符で囲む必要があります。

### 詳細

FEXIST は、ファイル参照名が割り当てられた外部ファイルが存在する場合は 1 を返し、存在しない場合は 0 を返します。

### 例

FINFO 関数の使用例については、“例: ディレクトリ情報の取得” (395 ページ)を参照してください。

## FILEEXIST 関数:Windows

ファイルの物理名を使用して外部ファイルの存在を確認します。

**カテゴリ:** 外部ファイル

**Windows 固有:** *filename* に環境変数を割り当てられます。

**参照項目:** “FILEEXIST Function” (*SAS Functions and CALL Routines: Reference*)

### 構文

**FILEEXIST**("*filename*")

#### *必須引数*

***filename***
外部ファイルの完全な形の物理ファイル名を指定します。DATA ステップでは、*filename* に、文字式、引用符で囲んだ文字列、DATA ステップ変数を指定できます。マクロでは、*filename* に式を指定できます。

Windows 版では、*filename* に環境変数を指定することもできます。指定するファイル名または環境変数は、引用符で囲む必要があります。

注 FILEEXIST 関数でディレクトリの存在を確認することもできます。

### 詳細

FILEEXIST は、外部ファイルが存在する場合は 1 を返し、存在しない場合は 0 を返します。

## FILENAME 関数:Windows

外部ファイル、ディレクトリまたは出力デバイスにファイル参照名を割り当てたり、割り当てを解除したりします。

**カテゴリ:** 外部ファイル

**Windows 固有:** デバイスタイプとホストオプション

**参照項目:** “FILENAME Function” (*SAS Functions and CALL Routines: Reference*)

### 構文

**FILENAME** ("*fileref*", "*filename*" <,*device-type*<,*host-options*<,*dir-ref*>>>)

### 必須引数

***fileref***
DATA ステップでは、外部ファイルに割り当てるファイル参照名を指定します。マクロ(%SYSFUNC 関数など)では、ファイル参照名にアンパサンドを付けずにマクロ変数を指定します。このマクロ変数の値には、外部ファイルに割り当てるファイル参照名が含まれます。

Windows 版では、*fileref* に Windows 環境変数を指定することもできます。指定するファイル参照名または環境変数は、引用符で囲む必要があります。

***filename***
外部ファイルを指定します。ファイル名に空白を指定すると、以前に割り当てたファイル参照名が消去されます。

### オプション引数

***device-type***
ファイル参照名が入力デバイス、出力デバイスまたは物理ファイルではない場所を参照する場合、デバイスのタイプまたは使用するアクセス方法を指定します。FILENAME ステートメントの device-type 引数 (448 ページ)に示されているデバイスのいずれか 1 つを指定できます。DISK が、デフォルトのデバイスタイプです。

***host-options***
Windows 版に固有のオプションです。FILENAME ステートメントで使用できるオプションを使用できます。FILENAME ステートメントの host-option-list (448 ページ)を参照してください。

***dir-ref***
dir-ref

## 詳細

外部ファイルが常駐するディレクトリに割り当てるファイル参照名を指定します。FILENAME は、操作が正常に行われた場合は 0 を返し、正常に行われなかった場合は 0 以外の数を返します。

## 例

FILENAME 関数の使用例については、“例: ディレクトリ情報の取得” (395 ページ)を参照してください。

# FILEREF 関数:Windows

現在の SAS セッションにファイル参照名が割り当てられていることを確認します。

**カテゴリ:** 外部ファイル

**Windows 固有:** fileref 引数に Windows 環境変数を指定できます。

**参照項目:** “FILEREF Function” (*SAS Functions and CALL Routines: Reference*)

## 構文

**FILEREF**("*fileref*")

### 必須引数

***fileref***
確認するファイル参照名を指定します。Windows 版では、*fileref* に Windows 環境変数を指定することもできます。指定するファイル参照名または環境変数は、引用符で囲む必要があります。

## 詳細

リターンコードが負の場合、ファイル参照名は存在しますが、ファイル参照名に割り当てられた物理ファイルは存在しないことになります。値が正の場合、ファイル参照名は割り当てられていません。値が 0 場合、ファイル参照名と外部ファイルの両方が存在します。

## 例

FILEREF 関数の使用例については、“例: ディレクトリ情報の取得” (395 ページ)を参照してください。

# FINFO 関数:Windows

外部ファイルの情報項目の値を返します。

**カテゴリ:** 外部ファイル

**Windows 固有:** 指定できる *info-items*

**参照項目:** “FINFO Function” (*SAS Functions and CALL Routines: Reference*)

## 構文

**FINFO**(*file-id*, *info-item*)

### 必須引数

***file-id***
ファイル ID を指定します。ファイル ID は、通常、FOPEN 関数からファイルを開くときに割り当てられます。

***info-item***
取得するファイルの情報項目名を指定します。文字値を指定します。*Info-item* には、ファイル情報名を含む変数または引用符で囲まれているファイル情報名を指定します。

*ディスクファイルの info-item* として、次のファイル情報項目のいずれかを指定できます。

- 作成日:*ddmmmyyyy:hh:mm:ss*

  *注:* 作成日の日時の情報項目は、サイトのロケールに合わせてローカライズされます。日時形式は、ロケールによって多少異なる場合があります。
- 最終更新日:*ddmmmyyyy:hh:mm:ss*
- ファイル名
- ファイルのサイズ(バイト)
- RECFM

- LRECL

*パイプファイルの info-item* として、次のファイル情報項目のいずれかを指定できます。

- 名前の付いていないパイプアクセスデバイス
- PROCESS
- RECFM
- LRECL

## 詳細

FINFO 関数を使用すると、FOPEN 関数によりすでに開かれ、ファイル ID を割り当てられている外部ファイルが持つシステム依存の情報項目の値が返されます。*info-item* に指定した値が無効な場合、FINFO は空白を返します。

## 例: ファイル情報の取得

```
data a; /* Does fileref "curdirfl" exist?No = 0 */ rc=fexist ("curdirfl"); put; put "Fil
```

***ログ 18.2*** *結果の SAS ログ*

```
NOTE:PROCEDURE PRINTTO used (Total process time): real time           0.36
seconds cpu time            0.00 seconds 291  data a; 292 293  /* Does fileref
"curdirfl" exist?No = 0 */ 294 295  rc=fexist ("curdirfl"); 296  put; t297  put
"Fileref curdirfl exist? rc should be 0 (no); " rc=; 298 299  /* assign fileref
*/ 300 301  rc=filename("curdirfl", "c:\tmp333"); 302 303  /* RC=0 indicates
success in assigning fileref */ 304 305  put "Fileref assigned - rc should be 0;
" rc=; 306  rc=fexist ("curdirfl"); 307 308  /* Does file which "curdirfl"
points to exist?No = 0 */ 309  /* Assigning a fileref doesn't create the file.*/
310 311  put "File still doesn't exist - rc should be 0; " rc=; 312  rc=fileref
("curdirfl"); 313 314  /* Does fileref "curdirfl" exist?*/ 315  /* Negative
means fileref exists, but file does not */ 316  /* Positive means fileref does
not exist            */ 317  /* Zero means both fileref and file exist
*/ 318 319  put "Fileref now exists - rc should be negative; " rc=; 320  put;
321 322  /* Does the file that the fileref points to exist?Should be no. */ 323
324  if ( fileexist ("./tmp333") ) then 325      /* if it does, open it for
input */ 326        do; 327          put "Open file for input"; 328
fid=fopen ("curdirfl", "i") ; 329        end; 330     else       /* most likely
scenario */ 331        do; 332          put "Open file for output"; 333
fid=fopen ("curdirfl", "o"); 334        end;
```

```
335 336 /* fid should be non-zero. 0 indicates failure. */ 337 put "File id is:0
indicates failure.*/ 337  put "File id is:" fid=; 338  numopts = foptnum(fid);
339  put "Number of information items should be 6; " numopts=; 340  do i = 1 to
numopts; 341  optname = foptname (fid,i); 342  put i= optname=; 343  optval  =
finfo (fid, optname); 344  put optval= ; 345  end; 346  rc=fclose (fid); 347
rc=fdelete ("curdirfl"); 348  put "Closing the file, rc should be 0; " 349  rc=;
run; Fileref curdirfl exist? rc should be 0 (no); rc=0 Fileref assigned - rc
should be 0; rc=0 File still doesn't exist - rc should be 0; rc=0 Fileref now
exists - rc should be negative; rc=-20006 Open file for output File id is: fid=1
Number of information items should be 6; numopts=6 i=1 optname=Filename optval=c:
\tmp333 i=2 optname=RECFM optval=V i=3 optname=LRECL optval=32767 i=4
optname=File Size (bytes) optval=0 i=5 optname=Last Modified
optval=13Mar2007:13:12:23 i=6 optname=Create Time optval=13Mar2007:13:12:23
Closing the file, rc should be 0; rc=0 NOTE:The data set WORK.A has 1
observations and 6 variables.NOTE:DATA statement used (Total process time): real
time 0.12 seconds cpu time 0.09 seconds 350 proc printto; run;
```

### 関連項目:

“FOPEN Function” (*SAS Functions and CALL Routines: Reference*)

## FOPTNAME 関数:Windows

外部ファイルの情報項目名を返します。

**カテゴリ:** 外部ファイル

**Windows 固有:** 指定できる情報項目

**参照項目:** “FOPTNAME Function” (*SAS Functions and CALL Routines: Reference*)

### 構文

**FOPTNAME**(*file-id*, *nval*)

#### *必須引数*

***file-id***
ファイル ID を指定します。ファイル ID は、通常、FOPEN 関数からファイルを開くときに割り当てられます。

***nval***
取得するファイルの情報項目数を指定します。次の表には、Windows 動作環境で単一ファイルや連結ファイルに対して指定できる *nval* の値を示します。

***表 18.1*** *ファイルの情報項目*

| *nval* | 単一ファイル | パイプファイル | 連結ファイル |
|---|---|---|---|
| 1 | ファイル名 | 名前の付いていないパイプアクセスデバイス | ファイル名 |
| 2 | RECFM | PROCESS | RECFM |
| 3 | LRECL | RECFM | LRECL |
| 4 | | LRECL | |

### 詳細

エラーが発生した場合、FOPTNAME 関数は空白を返します。

### 例

次の例では、パイプ使用時に FOPTNAME 関数により返された名前属性や値属性を含むデータセットが作成されます。

```
data fileatt;
   filename mypipe pipe 'dir';
   fid=fopen("mypipe","s");
   /* fid should be non-zero. 0 indicates failure */
```

```
   put "File id is: " fid=;
   numopts=foptnum(fid);
   put "Number of information items should be 4; " numopts=;
   do i=1 to numopts;
      optname=foptname(fid,i);
      put i= optname=;
      optval=finfo(fid,optname);
      put optval=;
   end;
rc=fclose(fid);
run;
```

***ログ 18.3*** *SAS ログに表示されるパイプファイル情報*

```
NOTE:PROCEDURE PRINTTO used (Total process time): real time           0.03
seconds cpu time            0.01 seconds 6    data fileatt; 7       filename
mypipe pipe 'dir'; 8       fid=fopen("mypipe","s"); 9       /* fid should be non-
zero.0 indicates failure */ 10      put "File id is:" fid=; 11
numopts=foptnum(fid); 12 put "Number of information items should be 4; "
numopts=; 13 do i=1 to numopts; 14 optname=foptname(fid,i); 15 put i= optname=;
16 optval=finfo(fid,optname); 17 put optval=; 18 end; 19 20 rc=fclose(fid); 21
run; File id is:fid=1 Number of information items should be 4; numopts=4 i=1
optname=Unnamed Pipe Access Device optval= i=2 optname=PROCESS optval=dir i=3
optname=RECFM optval=V i=4 optname=LRECL optval=32767 NOTE:The data set
WORK.FILEATT has 1 observations and 6 variables.NOTE:DATA statement used (Total
process time): real time 9.64 seconds cpu time 1.16 seconds 22 proc printto; run;
```

# FOPTNUM 関数:Windows

指定できるファイルの情報項目数を返します。

**カテゴリ:** 外部ファイル

**Windows 固有:** 指定できる情報項目

**参照項目:** “FOPTNUM Function” (*SAS Functions and CALL Routines: Reference*)

## 構文

**FOPTNUM**(*file-id*)

### *必須引数*

***file-id***
ファイル ID を指定します。ファイル ID は、通常、FOPEN 関数からファイルを開くときに割り当てられます。

## 詳細

指定できるファイルの情報項目は、次の 6 個です。

- ファイル名
- RECFM
- LRECL
- ファイルのサイズ(バイト)

- 最終更新日
- 作成日

次に、指定できるパイプの情報項目を示します。

- 名前の付いていないパイプアクセスデバイス
- PROCESS
- RECFM
- LRECL

FOPTNUM 関数は、次の値を返します。

ファイルの場合
6

パイプの場合
4

### 例

FOPTNUM 関数の使用例については、“例: ディレクトリ情報の取得” (395 ページ)を参照してください。

## LIBNAME 関数:Windows

SAS ライブラリにライブラリ参照名を割り当てたり、割り当てを解除したりします。

**カテゴリ:** SAS ファイル I/O

**Windows 固有:** ' 'libref (一重引用符に囲まれたスペース)の動作

**参照項目:** “LIBNAME Function” (*SAS Functions and CALL Routines: Reference*)

### 構文

**LIBNAME**(*libref*<,*SAS-data-library*<,*engine*<,*options*>>>)

#### *必須引数*

***libref***
SAS ライブラリに割り当てるライブラリ参照名を指定します。Windows 版では、*libref* の値に環境変数を指定できます。

#### *オプション引数*

***SAS-data-library***
ライブラリ参照名を割り当てる SAS ライブラリの物理名を指定します。

***engine***
データライブラリで開かれた SAS ファイルにアクセスするために使用するエンジンを指定します。

***options***
指定したエンジンで実行できるオプションを指定します。複数指定する場合は空白で区切って指定します。

## 詳細

正常に実行されると、LIBNAME 関数は 0 を返します。ただし、関数が正常に実行されたとしても、0 以外の値が返されることもあります。エラー、警告、注意が発生した場合は、0 以外の値が返されます。関数が正常に実行されたかどうかを確認するには、次のガイドラインを参照し、SAS ログを調べてください。

- 警告または注意が表示されている場合、関数は正常に実行されています。
- エラーが表示されている場合、関数は正常に実行されていません。

Windows 版では、*SAS-data-library* を指定しない場合や、*SAS-data-library* に' ' (一重引用符で囲まれたスペース)または" (間にスペースのない一重引用符)を指定した場合、ライブラリ参照名の割り当てが解除されます。

# MCIPISLP 関数:Windows

マルチメディア機器がアクティブになるまで SAS を待機させます。

**カテゴリ:** 特殊

**Windows 固有:** すべて

## 構文

**MCIPISLP**(*number-of-seconds*)

### *必須引数*

***number-of-seconds***
待機時間を秒単位で指定します。整数を指定する必要があります。

## 詳細

MCIPISLP 関数は、MCIPISTR 関数を使用してマルチメディア機器を開いた後、使用可能になるまでしばらく時間がかかることがわかっている場合に、特に便利です。

*number-of-seconds* 引数は、整数で、待機する秒数を表します。戻り値は、アクティブになるまでの秒数です。

MCIPISLP 関数は、DATA ステップおよび SCL コードの中で使用できます。

## 例

この例では、MCIPISTR 関数と MCIPISLP 関数の両方を使用して CD とビデオを再生しています。これらの関数の戻り値は、PUT ステートメントで表示されます。SAS ログに表示される戻り値を参照することにより、問題のある機器の有無を調べることができます。

```
data _null_;
  /* Open a CD player. */
  msg=mcipistr("open cdaudio alias mytunes");
  put msg=;
  /* Wait one second for the CD player     */
  /* to become active.                     */
  slept=mcipislp(1);
  /* Begin playing your favorite tunes     */
```

```
  /* from the beginning of the CD.         */
  msg=mcipistr("play mytunes");
  put msg=;
  /* Now open a video file. */
  msg=mcipistr("open c:\movies\amovie.avs
               alias myshow");
  put msg=;
  /* Begin the show and wait for it to     */
  /* complete.                             */
  msg=mcipistr("play myshow wait");
  put msg=;
  /* When the show is complete,            */
  /* close the instance.                   */
  msg=mcipistr("close myshow");
  put msg=;
  /* Stop and close the instance of the CD */
  /* player.                               */
  msg=mcipistr("stop mytunes");
  put msg=;
  msg=mcipistr("close mytunes");
  put msg=;
run;
```

## 関連項目:

"MCIPISTR 関数:Windows" (406 ページ)

# MCIPISTR 関数:Windows

MCI 文字列コマンドをマルチメディア機器にサブミットします。

**カテゴリ:** 特殊

**Windows 固有:** すべて

## 構文

**MCIPISTR**(*MCI-string-command*)

### *必須引数*

***MCI-string-command***
有効な SAS 文字列、つまり、文字変数、引用符で囲んだ文字定数またはその他の文字式を指定します。

## 詳細

MCIPISTR 関数は、MCI (Media Control Interface)文字列コマンドをサブミットします。

MCI を使用すると、CD プレーヤ、ミキサー、ビデオディスクプレーヤなど、さまざまな種類のマルチメディア機器を制御できます。Windows は、MCI をサポートしています。有効な MCI 文字列コマンドの詳細については、Windows multimedia SDK のドキュメント、および使用している MCI 準拠デバイスのドキュメントを参照してください。

戻り値は、MCI 文字列コマンドから返される文字列情報です。たとえば、"invalid instance"や"1"などの情報が返されます。

*注:* なお、SAS で使用可能なリターンコードを返さない MCI コマンドも一部あります。

MCIPISTR 関数は、DATA ステップおよび SCL コードの中で使用できます。

### 例

CD プレーヤを使用するには、DATA ステップの中で次のステートメントをサブミットします。

```
msg=mcipistr("open cdaudio alias cd");
msg=mcipistr("play cd");
msg=mcipistr("stop cd");
msg=mcipistr("close cd");
```

### 関連項目:

"MCIPISLP 関数:Windows" (405 ページ)

## MODULE 関数:Windows

外部動的リンクライブラリ(DLL)内にある特定のルーチンまたはモジュールを呼び出します。

**カテゴリ:** 外部ルーチン

**Windows 固有:** すべて

**注意:** **必ず正しい引数と属性を使用してください。** DLL 関数に誤った引数または属性を使用した場合、SAS およびオペレーティングシステムに障害が発生する可能性があります。

### 構文

**CALL MODULE**(<*control string*>,*module*,*argument-1, ..., argument-n*);

*num*=**MODULEN**(<*control-string*>,*module*,*argument-1, ..., argument-n*);

*char*=**MODULEC**(<*control-string*> ,*module*,*argument-1, ..., argument-n*);

**CALL MODULEI**(<*control-string*> ,*moduleargument-1, ..., argument-n*);

*num*=**MODULEIN**(<*control-string*> ,*module*,*argument-1, ..., argument-n*)

*char*=**MODULEIC**(<*control-string*> ,*module*,*argument-1, ..., argument-n*);

#### *必須引数*

***control-string***

任意指定の制御文字列です。先頭の文字はアスタリスク(*)にする必要があります。アスタリスクの後ろに、次の文字を自由に組み合わせて指定できます。

I DLL ルーチンを呼び出す前と呼び出した後で、MODULE 関数と呼び出した DLL ルーチンの引数をすべて 16 進数表現で出力します。このオプションを使用すると、引数または属性テーブルの誤りが原因で発生した問題を診断できます。`I` オプションを指定すると、`E` オプションが暗黙に指定されます。

E 詳細なエラーメッセージを出力します。`E` オプション(またはそれに代わる `I` オプション)を指定しない場合、MODULE 関数が生成する唯一のエラーメッセージは「Invalid argument to function」になり、通常はエラーの原因を判断できません。

S*x* フィールド定義の区切り文字として *x* を使用します。引数リスト内の *x* を独自の文字引数として指定することにより、x は、1 つの構造としてまとめたい引数リストのデリミタとして機能します。このオプションは、SASCBTBL 属性テーブルにエントリを登録しない場合にのみ使用します。SASCBTBL 属性テーブルにこのモジュールのエントリを登録する場合は、SASCBTBL テーブルで ARG ステートメントの FDSTART オプションを使用して各構造を区切る必要があります。

H MODULE ルーチン、属性のファイル形式、および推奨される SAS 形式と入力形式のヘルプ情報を表示します。

たとえば、制御文字列が`'*IS/'`の場合、パラメータリストを出力し、文字列`'/'`を引数リスト内の区切り文字として扱うことが指定されます。

***module***
は、DLL 名と、カンマ区切りのルーチン名または序数値として指定された使用外部モジュール名です。モジュールは、動的リンクライブラリ(DLL)内に存在し、外部から呼び出し可能である必要があります。たとえば、値が`'KERNEL32,GetProfileString'`の場合、KERNEL32.DLL をロードし、GetProfileString ルーチンを呼び出します。DLL 名は、大文字と小文字が区別されませんが、ルーチン名は、ルーチンの開発言語の制約に従うため、大文字と小文字が区別されます。

DLL が、順序値を使用した名前の指定をサポートしている場合は、DLL 名の後に 10 進数を指定できます。たとえば、`'XYZ,30'`と指定できます。

SASCBTBL 属性テーブルでルーチンに MODULE 属性を指定した場合は、ルーチン名が一意である限り、つまり属性ファイル内に同じ名前のルーチンが他になければ、DLL 名を指定する必要はありません。

*module* は、定数ではなく SAS の文字式として指定することができます。ただし、通常は、定数として渡します。

***引数***
は、要求されているルーチンに渡される引数です。適切な引数属性を使用してください。つまり、数値属性には数値引数、文字属性には文字引数を指定します。

## 詳細

MODULE 関数は、外部(SAS 内にない)動的リンクライブラリ内にある *module* というルーチンを、*arg-1* から *arg-n* までの指定された引数を使用して実行します。

MODULE 呼び出しルーチンは値を返しません。MODULEN 関数と MODULEC 関数は、それぞれ数値 *num* または文字 *char* を返します。実行する DLL 関数の戻り値に応じて、使用するルーチンを決めます。

MODULEI、MODULEIC および MODULEIN は、引数にベクトルおよび行列を指定できる、特殊なバージョンの MODULE 関数です。これらの関数の戻り値も、スカラです。これらの関数は、IML プロシジャからのみ呼び出すことができます。

これら 6 つのルーチンは、名前が異なることを除き、構文の形式は同一です。

MODULE 関数は、*arg-1* から *arg-n* の情報と、ユーザーが別のファイルに定義したルーチン記述および引数属性テーブルを使用して、パラメータリストを作成します。MODULE ルーチンを呼び出す前に、この外部ファイルを参照するファイル参照名として SASCBTBL を定義する必要があります。ファイルを作成するときは、自由に名前を付けることができます。

この結果、MODULE 関数への引数として SAS 変数と出力形式を使用でき、これらの引数が正しく変換されてから DLL ルーチンに渡されます。

CALL MODULEI、MODULEIN および MODULEIC を使用してベクトルと行列の引数を指定でき、IML プロシジャ内で使うことができます。

### 関連項目:

"SASCBTBL 属性テーブル" (292 ページ)

## PEEKLONG 関数:Windows

32 ビットプラットフォームおよび 64 ビットプラットフォームで、数値変数にメモリアドレスの内容を保存します。

**カテゴリ:** 特殊

**操作:** SAS サーバーがロックダウン状態の場合は、PEEKLONG 関数は実行されません。詳細については、"SAS Processing Restrictions for Servers in a Locked-Down State" (*SAS Language Reference: Concepts* 2 章)を参照してください。

**Windows 固有:** すべて

**参照項目:** "PEEKLONG Function" (*SAS Functions and CALL Routines: Reference*)および "PEEKCLONG Function" (*SAS Functions and CALL Routines: Reference*)

**注意:** **PEEKLONG 関数を使用すると、メモリアドレスに直接アクセスできます。PEEKLONG 関数の使用方法を誤ると、SAS およびオペレーティングシステムで障害が発生する場合があります。** PEEKLONG 関数は、MODULE 関数の 1 つから返される情報にアクセスする場合にのみ使用します。

### 構文

**PEEKLONG**(*address*<,*length*)>

#### *必須引数*

**address**
メモリアドレスにする文字式を指定します。

#### *オプション引数*

**length**
文字データの長さを指定します。

### 詳細

PEEKLONG 関数は、メモリアドレスの *address* から始まり、長さが *length* のデータを含む値を返します。

PEEKLONG 関数には、次の種類があります。

PEEKCLONG
文字列にアクセスします。

PEEKLONG
数値にアクセスします。

通常、PEEKLONG 関数のいずれかを使用する必要がある場合は、文字列にアクセスする PEEKCLONG を使用します。

## RANK 関数:Windows

ASCII 照合順序の文字位置を返します。

**カテゴリ:** 文字

**Windows 固有:** ASCII 順序を使用します。

**参照項目:** “RANK Function” (*SAS Functions and CALL Routines: Reference*)

### 構文

**RANK**(*x*)

#### *必須引数*

***x***

ASCII 照合順序の文字を含む文字式です。*x* の長さが 1 より長い場合、文字式での最初の文字の順位が返されます。

### 詳細

Windows では ASCII 文字セットを使用するため、RANK 関数は ASCII 照合順序の文字位置を表す整数を返します。

*注:* ASCII 127(16 進数表記では`'7F'x`)以降の文字で RANK 関数を使用するプログラムは、これらの文字が各国文字であり、国によって変わるため、移植はできません。

## SLEEP 関数:Windows

指定された時間、SAS DATA ステップの実行を停止します。

**カテゴリ:** 特殊

**Windows 固有:** すべて

**参照項目:** “SLEEP Function” (*SAS Functions and CALL Routines: Reference*)

### 構文

**SLEEP**(*n*<,*unit*>)

#### *必須引数*

***n***

DATA ステップの実行を停止する秒数を指定します。*n* 引数には、0 以上の数値定数を指定する必要があります。*n* を負または欠損値にすると、無効になります。

### オプション引数

***unit***
*n* に適用される単位を 10 の累乗秒に指定します。たとえば、1、10、0.01、または 0.001 のように指定します。このデフォルト値は 1 です。

## 詳細

SLEEP 関数は、指定された秒数の間、DATA ステップの実行を停止します。SLEEP 関数に、デフォルト値の *unit* を使用すると、ポップアップウィンドウが表示され、SAS を停止する時間が表示されます。

*n* 引数の戻り値は、停止していた秒数です。SLEEP 関数の最大停止期間は、およそ 46 日間です。

SLEEP 関数を呼び出すプログラムをサブミットすると、再始動するときに **SLEEP** ウィンドウにメッセージが表示されます。NOSLEEPWINDOW システムオプションを指定して SAS を起動すると、**SLEEP** ウィンドウは表示されません。この停止期間が終了するまで、SAS セッションはアクティブになりません。SLEEP 関数の呼び出しを取り消す場合は、Ctrl キーを押しながら Break キーを押します。

SLEEP 関数を呼び出す場合は、null の DATA ステップを使用し、この DATA ステップの後に SAS プログラムの残りを記述します。**SLEEP** 関数をこのように使用すると、Ctrl キーを押しながら Break キーを押して SLEEP 関数に割り込み、残りの SAS プログラムの実行を続行することができます。

## 例

この例の SLEEP 関数では、12 時間 15 分、プログラムの実行を延期するように SAS に通知します。

```
data _null_;
   /* argument to sleep must be expressed in seconds */
   slept= sleep((60*60*12)+(60*15));
run;
data monthly;
   /*... more data lines */
run;
```

## 関連項目:

“SLEEPWINDOW システムオプション:Windows” (563 ページ)

# TRANSLATE 関数:Windows

文字式に含まれる特定の文字を置き換えます。

**カテゴリ:** 文字

**Windows 固有:** 必須の構文、任意指定の *to* 引数と *from* 引数

**参照項目:** “TRANSLATE Function” (*SAS Functions and CALL Routines: Reference*)

## 構文

**TRANSLATE**(*source*,*to-1*,*from-1* <,…*to-n*,*from-n*>)

### 構文記述

***source***
元の文字値が含まれている SAS 式を指定します。

***to***
置き換え後の文字を指定します。

***from***
置き換えられる文字を指定します。

## 詳細

Windows 版では、*to* 引数と *from* 引数のペアを指定する必要はありません。省略する場合は、カンマを指定する必要があります。

# WAKEUP 関数:Windows

SAS DATA ステップの実行を再開する時間を指定します。

**カテゴリ:** 特殊

**Windows 固有:** すべて

## 構文

**WAKEUP**(*until-when*)

### 必須引数

***until-when***
WAKEUP 関数で実行を再開する時間を指定します。

## 詳細

WAKEUP 関数を使って、DATA ステップの実行を再開する時間を指定します。戻り値は、アクティブになるまでの秒数です。

*until-when* 引数は、次に示す SAS 日時値、SAS 時間値または数値定数のいずれかにすることができます。

- *until-when* が日時値の場合、WAKEUP 関数は、指定された日時まで実行を延期します。過去の日付と時間を指定した場合、WAKEUP 関数は実行の延期を行わず、値 0 を返します。
- *until-when* が時間値の場合、WAKEUP 関数は指定された時刻まで実行を延期します。指定された時刻が該当する 24 時間内で既に経過している場合は、WAKEUP 関数は再度指定された時刻になるまで実行を延期します。
- *until-when* の値が数値定数の場合、WAKEUP 関数は、次の午前 0 時が到来する前または後の、指定された秒数だけ実行を延期します。*until-when* の値が正の数値定数の場合、WAKEUP 関数は、午前 0 時を過ぎてから *until-when* 秒の間実行を延期します。*until-when* の値が負の数値定数の場合、WAKEUP 関数は、午前 0 時になる前の *until-when* 秒になるまで実行を延期します。

*until-when* 引数には負の値を指定できますが、欠損値にすることはできません。WAKEUP 関数の最大停止期間は、約 46 日間です。

WAKEUP 関数を呼び出すプログラムをサブミットすると、再始動するときに **SLEEP** ウィンドウにメッセージが表示されます。NOSLEEPWINDOW システムオプションを指定して SAS を起動すると、**SLEEP** ウィンドウは表示されません。この待機期間が終了するまで、SAS セッションはアクティブになりません。WAKEUP 関数の呼び出しを取り消す場合は、Ctrl キーを押しながら Break キーを押します。

WAKEUP 関数を呼び出す場合は、null の DATA ステップを使用し、この DATA ステップの後に残りの SAS プログラムを記述します。WAKEUP 関数をこのように使用すると、Ctrl キーを押しながら Break キーを押して待機期間に割り込み、残りの SAS プログラムの実行を続行することができます。

## 例

### *例 1: 指定した日付または時刻までプログラムの実行を延期*

この例のコードでは、2004 年 1 月 1 日午後 1:00 まで、プログラムの実行を延期します。

```
data _null_;
   slept=wakeup('01JAN2004:13:00:00'dt);
run;
data compare;
   /* ...more data lines */
run;
```

次の例では、午後 10:00 まで、プログラムの実行を延期します。

```
data _null_;
   slept=wakeup("22:00:00"t);
run;
data compare;
   /* ...more data lines */
run;
```

### *例 2: 午前 0 時から指定した時間だけプログラムの実行を延期*

次の例では、次の午前 0 時から 35 秒間、プログラムの実行を延期します。

```
data _null_;
   slept=wakeup(35);
run;
data compare;
   /* ...more data lines */
run;
```

### *例 3: WAKEUP 関数の引数として変数を使用*

この例では、WAKEUP 関数の引数として変数を使用します。

```
data _null_;
   input x;
   slept=wakeup(x);
   datalines;
1000
;
data compare;
   input article1 $ article2 $ rating;
   /* ...more data lines */
run;
```

ストリーム内のデータに X の値が 1000 であると示されているため、WAKEUP 関数は午前 0 時から 1,000 秒間実行を延期します。

## 関連項目:

“SLEEPWINDOW システムオプション:Windows” (563 ページ)

*19 章*

# Windows 版に固有の SAS 入力形式

**Windows 版に固有の SAS 入力形式** . . . . . **415**
Windows 版に固有の SAS 入力形式の概要 . . . . . 415
**バイナリデータの読み取り** . . . . . **416**
**以前のリリースのユーザー定義入力形式を SAS 9.4 用に変換する** . . . . . **416**
以前のリリースのユーザー定義入力形式から SAS 9.4 用への変換について . 416
バージョン 6 のユーザー定義入力形式の変換 . . . . . 417
バージョン 5 のユーザー定義入力形式の変換 . . . . . 417
**ディクショナリ** . . . . . **417**
HEXw.入力形式:Windows . . . . . 417
$HEXw.入力形式:Windows . . . . . 418
IBw.d 入力形式:Windows . . . . . 419
PDw.d 入力形式:Windows . . . . . 420
PIBw.d 入力形式:Windows . . . . . 421
RBw.d 入力形式:Windows . . . . . 422
ZDw.d 入力形式:Windows . . . . . 423

## Windows 版に固有の SAS 入力形式

### *Windows 版に固有の SAS 入力形式の概要*

SAS 入力形式は、SAS がデータ値を変数へと読み込むために使用する命令またはテンプレートです。ほとんどの SAS 形式は完全に *SAS 出力形式と入力形式: リファレンス*で記述されます。ここでは、Windows 版 SAS 固有の動作を含む入力形式について説明します。

Windows 動作環境に固有の SAS 入力形式の多くは、バイナリデータの読み込みに使用されます。これらの入力形式を使うには、“バイナリデータの読み取り” (416 ページ) に示されている概念を理解することが重要です。

SAS の以前のバージョンで使用するために作成した入力形式がある場合は、それらを SAS 9.4 用に変換する方法については、“以前のリリースのユーザー定義入力形式を SAS 9.4 用に変換する” (416 ページ) を参照してください。

# バイナリデータの読み取り

IBM メインフレームや Hewlett-Packard 9000 を始めとする多くの UNIX システムでは、バイトはビッグエンディアンと呼ばれる順序で格納されます。Intel ベースシステムや IBM 互換マイクロコンピュータ、旧 Compaq 製の VAX と Alpha コンピュータでは、バイト反転やリトルエンディアンと呼ばれる別の順序でバイトは格納されます。

ある順序で格納されているバイナリデータを、別の順序でバイナリデータを格納するコンピュータで読み取るには、追加処理が必要です。SAS アプリケーションを設計するときは、データがどのように読み取られるかを考慮し、それにしたがって出力形式および入力形式を選択するようにします。

SAS には、バイナリデータの読み取り用に 2 種類の入力形式と、それに対応するバイナリデータ書き込み用の出力形式が用意されています。

- IB*w.d*、PD*w.d*、PIB*w.d*、RB*w.d* はネイティブモードでの読み書きの入力形式と出力形式で、つまり、コンピュータ標準であるバイト順序システムを使用します。
- S370FIB*w.d*、S370FPD*w.d*、S370FRB*w.d*、S370FPIB*w.d* コンピュータのネイティブモードにかかわらず、IBM 370 標準に基づく読み書きの入力形式と出力形式を使用します。これらの入力形式と出力形式を使用すると、数値データがどれだけ保存されているかにかかわらず、どの SAS 環境でも実行できる SAS プログラムを記述できます。

バイナリデータの読み取りと書き込みを行う SAS プログラムが 1 種類のコンピュータ上だけで実行されている場合は、ネイティブモードの入力形式と出力形式を使用することができます。それに対し、異なるバイト格納方式を使用する複数のコンピュータ上で実行できる SAS プログラムを記述する場合は、IBM 370 出力形式および入力形式を使用します。IBM 370 入力形式と出力形式の目的は、数値データの格納に用いるバイト格納方式に関係なく、どのような SAS 環境でも実行できる SAS プログラムを記述できるようにすることです。

たとえば、PIB*w.d* 形式でデータを書き込むプログラムがあるとします。バイト反転モードでデータを保存できるように、このプログラムをマイクロコンピュータで実行します。そのとき、PC 上では、PIB*w.d* 情報を使用してデータを読み込む別の SAS プログラムを実行します。両方のプログラムとも、PC 上で実行されるので、データは正しく読み取られます。ただし、データはマイクロコンピュータに最適化されてはいても、Hewlett Packard 9000 には最適化されてはいないため、Hewlett Packard 9000 シリーズのコンピュータにデータをアップロードして正しく読み込むことはできません。この問題を回避するには、S370FPIB*w* を使用します。*d* 形式を使用してデータを書き込んでください。マイクロコンピュータ上でも、IBM370 モードでデータの保存が可能です。それから、S370FPIB*w.d* 入力形式を使用してデータを読み込んでください。データを読み取るときにどの種類のコンピュータを使用しても、データは正しく読み取られます。

# 以前のリリースのユーザー定義入力形式を SAS 9.4 用に変換する

### *以前のリリースのユーザー定義入力形式から SAS 9.4 用への変換について*

リリース 6.04、6.06、6.08 のユーザー定義入力形式と出力形式は、SAS 9.4 形式に変換してから SAS 9.4 のプログラムで使用できるようになります。Windows 版のリリース 6.08 以降で作成されたユーザー定義入力形式と出力形式のみ、変換は不要

です。これらの入力形式と出力形式は、Windows の SAS セッションで直接読み込むことができます。[1]

### バージョン 6 のユーザー定義入力形式の変換

リリース 6.04、6.06、6.08 のユーザー定義入力形式および出力形式を含む SAS カタログを変換するには、次のいずれかの方法を使用します。

リリース 6.04 のカタログの変換
: リリース 6.04 で、PROC FORMAT ステートメントに CNTLOUT=オプションを指定し、出力データセットを作成します。次に、SAS 9.4 で、PROC FORMAT ステートメントに CNTLIN=オプションを指定し、SAS 9.4 の入力形式または出力形式を作成します。データセットを読み込むには、SAS 9.4 のセッションで、V604 Engine を使用する必要があります。この方法は、リリース 6.06 またはリリース 6.08 から変換する場合にも使用できます。

リリース 6.06 またはリリース 6.08 のカタログの変換
: CPORT プロシジャおよび CIMPORT プロシジャを使用して入力形式と出力形式を変換します。CPORT プロシジャと CIMPORT プロシジャに関する詳細については、*Base SAS プロシジャガイド*を参照してください。この方法は、リリース 6.06 またはリリース 6.08 形式を変換する場合にのみ使用することができます。リリース 6.04 形式には使用できません。

### バージョン 5 のユーザー定義入力形式の変換

SAS 9.4 のプログラムでバージョン 5 のユーザー定義入力形式と出力形式を使用する場合も、SAS 9.4 形式に変換する必要があります。(これらのファイルを変換するだけでなく、リモートオペレーティングシステムから PC へファイルを転送する必要があります)。ファイルを変換するには、次のいずれかの方法を使用します。

- リモートオペレーティング環境で V5TOV6 プロシジャを使用し、入力形式と出力形式をバージョン 6 形式に変換します。このためには、リモートオペレーティングシステムからバージョン 6 の SAS にアクセスできる必要があります。次に、変換されたユーザー定義入力形式と出力形式を、バイナリファイルとして Windows オペレーティングシステムに転送し、CIMPORT プロシジャを使用して変換を完了します。

  *注:* V5TOV6 プロシジャは、SAS 9.4 では利用できません。このプロシジャは SAS リリース 6 で使用する必要があります。

- リモートオペレーティング環境上のバージョン 5 で、SAS Global Forum サプリメンタルプロシジャの FMTLIB を使用して出力データセットを作成し、PC にそのデータセットを転送します。次に、SAS 9.4 で PROC FORMAT ステートメントに CNTLIN=オプションを指定して、SAS 9.4 の入力形式または出力形式を作成します。

# ディクショナリ

## HEX*w*.入力形式:Windows

16 進の正のバイナリ値を固定小数点または浮動小数点のバイナリ値に変換します。

[1] ただし、ユーザー定義入力形式と出力形式を含む古い Windows カタログが過去のリリースで使用されない場合は、CPORT プロシジャおよび CIMPO RT プロシジャを使用して SAS 9.4 形式に変換することをお勧めします。

**カテゴリ:** 数値

**配置:** 左

**Windows 固有:** ネイティブ浮動小数点表現

**参照項目:** “HEXw. Informat” (*SAS Formats and Informats: Reference*)

## 構文

**HEX***w*.

### *必須引数*

***w***
入力が、バイナリ整数(固定小数点数)またはバイナリ実数(浮動小数点数)のどちらを表すかを指定します。*w* に 1 から 15 の値を指定すると、16 進の入力値はバイナリ整数を表します。*w* に 16 を指定すると、16 進の入力値は浮動小数点数を表します。

**デフォルト** 8

**範囲** 1–16

## 詳細

HEX*w*.入力形式は、バイト順序が逆になっていない、つまり Windows 形式ではない入力を想定しています。(バイナリ数値用の IB、PIB、RB 入力形式は、バイト順序が逆になっていることを想定しています)。HEX*w*.入力形式を使用すると、Windows 以外の環境で作成された SAS プログラムから 16 進リテラルを読み込むことができます。

## 関連項目:

- “$HEXw.入力形式:Windows” (418 ページ)
- “HEXw.出力形式:Windows” (378 ページ)

# $HEX*w*.入力形式:Windows

16 進データを文字データに変換します。

**カテゴリ:** 文字

**配置:** 左

**Windows 固有:** ASCII 文字エンコーディングシステム

**参照項目:** “$HEXw. Informat” (*SAS Formats and Informats: Reference*)

## 構文

**$HEX***w*.

### 必須引数

*w*
入力値の幅を指定します。

デフォルト 2

範囲 1-132767

## 詳細

$HEX*w*.出力形式は HEX*w*.入力形式と似ており、各 16 進数文字が 1 バイトを占める値が読み込まれます。$HEX*w*.入力形式を使用すると、入力データが印刷可能な文字に限定されている場合、16 進情報を文字変数にエンコードできます。変換は、ASCII 文字セットに基づいて行われます。

## 関連項目:

- “HEXw.入力形式:Windows” (417 ページ)
- “$HEXw.出力形式:Windows” (379 ページ)

# IB*w.d* 入力形式:Windows

バイナリ整数(固定小数点数)を読み込みます。

**カテゴリ:** 数値

**Windows 固有:** ネイティブ浮動小数点表現

**参照項目:** “IBw.d Informat” (*SAS Formats and Informats: Reference*)

## 構文

**IB***w.d*

### 必須引数

*w*
入力フィールドの幅を指定します。

デフォルト 4

範囲 1-8

*d*
入力値を除算する 10 の累乗数を指定します。入力データに小数点が含まれる場合でも *d* 値で除算されます。

範囲 0-10

## 詳細

バイナリ整数データの場合、高順位ビットが値の符号を表します。正の数の場合は 0、負の数の場合は 1 になります。負の数は、2 の補数表記で表します。入力形式に *d* 値が含まれる場合は、データ値は $10^d$。

IB*w.d* 入力形式を使用するには、2 の補数とバイトスワップデータ形式を理解する必要があります。

マイクロコンピュータの固定小数点数の詳細については、Intel 社の開発者向け Web サイトを参照してください。

### 比較

IB*w.d* 入力形式および PIB*w.d* 入力形式は異なる結果を返します。IB*w.d* 入力形式は、正の数と負の数の両方を処理し、高順位ビットを符号ビットとして使用します。対照的に、PIB*w.d* 入力形式は正の数のみに使われ、符号ビットを検索しません。たとえば、データに次の 2 バイトのバイトスワップ値が含まれるとします。

```
01 80
```

この値を IB2.入力形式で読み込むと、IB2.入力形式は符号ビットを検索し設定を確認するため、値を-32,767 として読み込みます。この値を PIB2.入力形式で読み込むと、符号ビットは確認されないため、結果は 32,769 になります。

### 例

データに次の 6 バイトのバイトスワップ値が含まれるとします。

```
64 00 00 00 00 00
```

この値を IB6 入力形式を使用して読み込む場合は、固定小数点値の 100.0 として読み込まれます。データにのバイトスワップ値が含まれるとします。

```
01 80
```

符号ビットが設定されるため、その値は−32,767 として読み込まれます。

### 関連項目:

- “IBw.d 出力形式:Windows” (380 ページ)
- “バイナリデータの読み取り” (416 ページ)

## PD*w.d* 入力形式:Windows

パック 10 進データを読み込みます。

**カテゴリ:** 数値

**Windows 固有:** 値が負または正の数として解釈される仕組み

**参照項目:** “PDw.d Informat” (*SAS Formats and Informats: Reference*)

### 構文

**PD***w.d*

#### *必須引数*

***w***
入力フィールドの幅を指定します。

**デフォルト** 1

範囲 1–16

*d*
入力値を除算する 10 の累乗数を指定します。データに小数点が含まれる場合、*d* 値は無視されます。

範囲 0-31

## 詳細

パック 10 進データでは、1 バイトに 2 桁の数字を含みます。値の符号は、最初のバイト全体が符号を表すために使用されている場合でも、最初のバイトの最上位ビットで示されます。パック 10 進データを端末から直接キー入力することは、通常は不可能ですが、多くのプログラムはパック 10 進データを書き出します。小数部の範囲は、1 から 31 です。

## 例

データに次のパック 10 進数が含まれるとします。

```
80 00 11 43
```

PD4 入力形式を使用する場合は、この値は倍精度値の-1143.0 として読み込まれます。同様に、次の値は 1500.0 として読み込まれます。

```
00 00 15 00
```

## 関連項目:

- “PDw.d 出力形式:Windows” (382 ページ)
- “バイナリデータの読み取り” (416 ページ)

# PIB*w.d* 入力形式:Windows

正のバイナリ整数(固定小数点数)を読み込みます。

**カテゴリ:** 数値

**Windows 固有:** ネイティブバイトスワップ整数

**参照項目:** “PIBw.d Informat” (*SAS Formats and Informats: Reference*)

## 構文

**PIB***w.d*

### *必須引数*

*w*
入力フィールドの幅を指定します。

デフォルト 1

範囲 1-8

*d*
入力値を除算する 10 の累乗数を指定します。入力データに小数点が含まれる場合でも *d* 値で除算されます。

**範囲** 0-10

## 詳細

正のバイナリ整数値は、すべての値が正として扱われる以外は、バイナリ整数と(入力形式“IBw.d 入力形式:Windows” (419 ページ)を参照してください)同じです。したがって、高順位ビットは、値の符号ではなく、値の一部です。

## 比較

PIB*w.d* 入力形式および IB*w.d* 入力形式は異なる結果を返します。これら 2 つの入力形式の差異に注意する必要があります。IB*w.d* 入力形式は、正の数と負の数の両方を処理し、高順位ビットを符号ビットとして使用します。対照的に、PIB*w.d* 入力形式は正の数のみに使われ、符号ビットを検索しません。たとえば、データに次の 2 バイトのバイトスワップ値が含まれるとします。

```
01 80
```

この値を IB2.入力形式で読み込むと、IB2.入力形式は符号ビットを検索し設定を確認するため、値を-32,767 として読み込みます。ただし、この値を PIB2.入力形式でよみこむと、符号ビットが使用されず、結果は 32,769 になります。

## 例

データに次の 1 バイト値が含まれるとします。

```
FF
```

この値を PIB1 入力形式を使用して読み込んだ場合は、倍精度値の 255.0 として読み込まれます。この入力形式を使用するには、2 の補数とバイトスワップデータ形式を理解する必要があります。

## 関連項目:

- “PIBw.d 出力形式:Windows” (383 ページ)
- “バイナリデータの読み取り” (416 ページ)

# RB*w.d* 入力形式:Windows

バイナリ実数(浮動小数点数)データを読み込みます。

**カテゴリ:** 数値

**Windows 固有:** ネイティブ浮動小数点表現

**参照項目:** “RBw.d Informat” (*SAS Formats and Informats: Reference*)

## 構文

**RB***w.d*

### 必須引数

*w*
入力フィールドの幅を指定します。

デフォルト 4

範囲 2-8

*d*
入力値を除算する 10 の累乗数を指定します。入力データに小数点が含まれる場合でも *d* 値で除算されます。

範囲 0-10

## 詳細

RB*w.d* 入力形式は、PC にバイナリ実数(浮動小数点)表記で保存されている数値データを読み込みます。多くの場合、科学計算の数値データは、浮動小数点表記で保存されます(SAS はすべての数値を浮動小数点表記で保存します)。浮動小数点数は、次の 2 つの部分から構成されます。すなわち、値を表す仮数部と、値の大きさを表す指数部です。浮動小数点バイナリデータを端末から直接キー入力することは、通常は不可能ですが、多くのプログラムは浮動小数点バイナリデータを書き出します。

## 関連項目:

- “RBw.d 出力形式:Windows” (384 ページ)
- “バイナリデータの読み取り” (416 ページ)

# ZD*w.d* 入力形式:Windows

ゾーン 10 進データを読み込みます。

**カテゴリ:** 数値

**Windows 固有:** 最後のバイトには、次の符号が含まれます。

**参照項目:** “ZDw.d Informat” (*SAS Formats and Informats: Reference*)

## 構文

**ZD***w.d*

### 必須引数

*w*
入力フィールドの幅を指定します。

デフォルト 1

範囲 1-32

*d*
入力値を除算する 10 の累乗数を指定します。データに小数点が含まれる場合、*d* 値は無視されます。

範囲 1-10

## 詳細

これは、オーバプリント後置数値形式としても知られています。Windows 版では、フィールドの最後のバイトには、数値の符号情報が含まれます。次の表は、最後のバイトの変換方法を示しています。

*表 19.1 末尾の数字入力形式の重ね打ち*

| 数字 | ASCII 文字 | 数字 | ASCII 文字 |
|---|---|---|---|
| 0 | { | -0 | } |
| 1 | A | -1 | J |
| 2 | B | -2 | K |
| 3 | C | -3 | L |
| 4 | D | -4 | M |
| 5 | E | -5 | N |
| 6 | F | -6 | O |
| 7 | G | -7 | P |
| 8 | H | -8 | Q |
| 9 | I | -9 | R |

## 関連項目:

“ZDw.d 出力形式:Windows” (386 ページ)

*20 章*

# Windows 版に固有の SAS プロシジャ

**Windows 版に固有の SAS プロシジャ** . . . . . **425**

**ディクショナリ** . . . . . **425**
CATALOG プロシジャ:Windows . . . . . 425
CIMPORT プロシジャ:Windows . . . . . 426
CONTENTS プロシジャ:Windows . . . . . 427
CONVERT プロシジャ:Windows . . . . . 428
CPORT プロシジャ:Windows . . . . . 431
DATASETS プロシジャ:Windows . . . . . 432
OPTIONS プロシジャ:Windows . . . . . 433
PMENU プロシジャ:Windows . . . . . 434
PRINTTO プロシジャ:Windows . . . . . 435
SORT プロシジャ:Windows . . . . . 436

## Windows 版に固有の SAS プロシジャ

Base SAS プロシジャを使用して、統計計算、レポート作成およびデータ管理を実行できます。Base SAS プロシジャの概要については、*Base SAS プロシジャガイド*を参照してください。

## ディクショナリ

### CATALOG プロシジャ:Windows

SAS カタログ内のエントリを管理します。

**Windows 固有:** CONTENTS ステートメントの FILE=オプション

**参照項目:** 9 章: “CATALOG プロシジャ” (*Base SAS Procedures Guide*)

#### 構文

**PROC CATALOG** CATALOG=<*libref.*> *catalog* <ENTRYTYPE=*etype*> <KILL> ;
**CONTENTS** <OUT=*SAS-data-set*> <FILE=*fileref*;>

### *必須引数*

この構文は CATALOG プロシジャ構文を単純化したものです。完全な構文およびその説明については、9 章: “CATALOG プロシジャ” (*Base SAS Procedures Guide*)を参照してください。

***fileref***
ファイル参照名を指定します。Windows 動作環境に固有の名前を指定します。

## 詳細

CATALOG プロシジャは、SAS カタログ内のエントリを管理します。

CATALOG プロシジャの CONTENTS ステートメントの FILE=オプションには、Windows 動作環境に特有のファイル指定を使用できます。FILE=オプションでファイル参照名を引用符で囲まずに指定し、そのファイル指定が FILENAME ステートメント、SET システムオプションまたは Windows 環境変数で定義されていない場合、ファイルは、*file-specification*.LST という名前で作業ディレクトリに保存されます。たとえば、MYFILE が、FILENAME ステートメント、SET システムオプション、Windows 環境変数によって定義されたファイル参照名でない場合に、次のステートメントをサブミットすると、ファイル MYFILE.LST(SASUSER.PROFILE の内容のリストを含む)が作成され、SASINITIALFOLDER の場所に保存されます。

```
proc catalog catalog=sasuser.profile;
   contents file=myfile;
run;
```

# CIMPORT プロシジャ:Windows

CPORT プロシジャによって作成された移送ファイルを復元します。

**Windows 固有:** 移送ファイルの名前および保存場所

**参照項目:** 11 章: “CIMPORT プロシジャ” (*Base SAS Procedures Guide*)

## 構文

**PROC CIMPORT** *destination=libref| <libref.> member-name <option(s)>* ;

### *必須引数*

この構文は CIMPORT プロシジャ構文を単純化したものです。完全な構文 およびその説明については、11 章: “CIMPORT プロシジャ” (*Base SAS Procedures Guide*) を参照してください。

***destination***
トランスポートファイル内のファイルが、単一の SAS データセット、単一の SAS カタログあるいは SAS ライブラリの複数のメンバであるかを指定します。

***libref | <libref.>member-name***
移送ファイルから作成する SAS データセット名、カタログ名あるいはライブラリ名を指定します。

## 詳細

CIMPORT プロシジャは、CPORT プロシジャによって作成(*エクスポート*)された移送ファイルをインポートします。

CIMPORT プロシジャと CPORT プロシジャを組み合わせて使用すると、カタログおよびデータセットを動作環境間で移動することができます。

*注:* CIMPORT プロシジャは、CPORT プロシジャによって生成されたファイルを処理しますが、XPORT エンジンによって生成されたトランスポートファイルは処理しません。

*注:* SAS 9.1 から、MIGRATE プロシジャを使用して SAS ライブラリをこれまでのリリースから移行できるようになりました。

Windows 版の CIMPORT プロシジャを使用するときは、次の点に注意してください。

- INFILE=オプションの値は、FILENAME ステートメントで定義されたファイル参照名、引用符で囲まれた Windows パス名または環境変数のいずれかにすることができます。
- INFILE=オプションを省略し、予約済みファイル参照名 SASCAT が定義されていない場合、作業ディレクトリにある SASCAT.DAT というファイルから読み込まれます。その名前によるファイルが存在しない場合、C:\SAS が作業ディレクトリとして定義されたと仮定して、次のエラーメッセージが発行されプロシジャは終了します。

  エラー:ファイルが存在しません。C:\SAS\SASCAT.DAT

- CPORT プロシジャによって作成されたファイルがバイナリ形式で転送されていない場合、CIMPORT プロシジャはそのファイルを読み込むことができません。次のメッセージが表示されます。

  エラー:移送ファイルが損傷しています。

## 関連項目:

- “CPORT プロシジャ:Windows” (431 ページ)
- MIGRATE プロシジャとリリース間の互換性については、http://support.sas.com/migration/planning/files/regression.html*Base SAS プロシジャガイド*を参照してください。

# CONTENTS プロシジャ:Windows

SAS ライブラリファイルの内容に関する説明を出力します。

**Windows 固有:** Engine/Host Dependent Information の出力

**参照項目:** 13 章: “CONTENTS プロシジャ” (*Base SAS Procedures Guide*)

## 構文

**CONTENTS プロシジャ** <*option(s)*> ;

### *必須引数*

***option(s)***
利用可能なオプションの説明については、CONTENTS プロシジャ *Base SAS プロシジャガイド*を参照してください。

## 詳細

CONTENTS プロシジャは、SAS セットの内容を表示し、SAS ライブラリのディレクトリを出力します。

CONTENTS プロシジャで 生成される出力のほとんどの部分は、どの動作環境でも同じです。Engine/Host Dependent Information の出力は、動作環境およびエンジンによって異なります。ここでは、次のステートメントの結果として V9 エンジンのために生成される Engine/Host Dependent Information の例を示します。

```
DATA SCHOOL;
   INPUT NAME $ Y GRADE CLASS $ ID;
   DATALINES;
 PHIL 1 85 MATH 234107589
 ROBERTO 1 90 ENGLISH 190873452
 CAROL 2 70 MATH 257902348
 THOMAS 2 71 ENGLISH 234567823
 JUANITA 3 98 FRENCH 876345290
 CEDRIC 3 75 HISTORY 231987222
 MARIA 4 89 PE 87654321
 ;
 PROC CONTENTS DATA=SCHOOL OUT=SCHOUT(DROP=CRDATE MODATE);
    TITLE 'SCHOOL DATASET';
    RUN;
```

***ログ 20.1*** *V9 Engine を使用する CONTENTS プロシジャからの Engine/Host Dependent Information*

```
                                                    SCHOOL DATASET 07:52
Wednesday, October 17, 2007 1 The CONTENTS Procedure Data Set Name WORK.SCHOOL
Observations 7 Member Type DATA Variables 5 Engine V9 Indexes 0 Created
Wednesday, October 17, 2007 00:39:17 Observation Length 40 Last Modified
Wednesday, October 17, 2007 00:39:17 Deleted Observations 0 Protection
Compressed NO Data Set Type Sorted NO Label Data Representation WINDOWS_32
Encoding wlatin1 Western (Windows) Engine/Host Dependent Information Data Set
Page Size 4096 Number of Data Set Pages 1 First Data Page 1 Max Obs per Page 101
Obs in First Data Page 7 Number of Data Set Repairs 0 File Name C:
\DOCUME~1\sasusr1\LOCALS~1\Temp\SAS Temporary Files\_TD1904\school.sas7bdat
Release Created 9.9901B0 Host Created XP_PRO Alphabetic List of Variables and
Attributes # Variable Type Len 4 CLASS Char 8 3 GRADE Num 8 5 ID Num 8 1 NAME
Char 8 2 Y Num 8
```

エンジン名(V9)は、ヘッダー情報に示されています。Engine/Host Dependent Information は、データセットのページサイズやページごとのオブザベーション最大数など、データセットの属性の説明です。データセットサイズ情報がどのように解釈されるかについての情報は、“データセットのサイズの計算” (207 ページ)を参照してください。

## CONVERT プロシジャ:Windows

BMDP システムファイル、OSIRIS システムファイルおよび SPSS エクスポートファイルを SAS データセットに変換します。

**Windows 固有:** すべて

### 構文

**PROC CONVERT** *product-specification* <*option(s)*>

## 必須引数

***product-specification***

必須の引数です。次のいずれかを指定します。

**BMDP=*fileref* <(CODE=*code*CONTENT=*content-type*)>**

DOS 環境で作成された BMDP 保存ファイルの最初のメンバを、SAS データセットに変換します。次に、例を示します。

```
filename save 'c:\myidr\bmdp.dat';
proc convert bmdp=save;
run;
```

*fileref* 引数によって参照される BMDP ファイルに複数の保存ファイルがある場合は、*fileref* の後ろにかっこで囲んで 2 つのオプションを使用できます。CODE=オプションに希望の保存ファイルのコードを指定し、CONTENT=オプションに保存ファイルの内容を指定します。たとえば、CODE=JUDGES のファイルの内容が DATA の場合、次のステートメントを使用することができます。

```
filename save 'c:\mydir\bmdpl.dat';
proc convert bmdp=save(code=judges
                       content=data);
run;
```

**OSIRIS=*fileref***

SAS データセットに変換する OSIRIS ファイルのファイル参照名を指定します。このプロダクト指定をする場合、DICT=オプションによって、使用する OSIRIS ディクショナリも指定する必要があります。

**SPSS=*fileref***

SAS データセットに変換する SPSS エクスポートファイルのファイル参照名を指定します。SPSS エクスポートファイルは、いずれの動作環境でも、SPSS EXPORT コマンドを使用して作成する必要があります。

***option-list***

**DICT=*fileref***

OSIRIS ファイルのディクショナリファイルのファイル参照名を指定します。DICT=オプションは、プロダクト指定が OSIRIS の場合のみ有効です。

**FIRSTOBS=*n***

変換を開始するオブザベーションの番号を指定します。このオプションで OSIRIS または SPSS/PC システムファイルの先頭にあるオブザベーションを処理から除外する場合に指定します。

**OBS=*n***

変換を終了するオブザベーションの番号を指定します。ファイルの最後にあるオブザベーションを処理から除外する場合に指定します。

**OUT=*SAS-data-set***

変換されたデータを保存するために作成される SAS データセットの名前を指定します。OUT=オプションを省略した場合でも、DATA ステートメントでデータセット名を省略したときと同じように、WORK データセットが作成され、DATA*n* という名前が自動的に付けられます。たとえば、データセットがジョブまたはセッションで変換される最初のデータセットであれば DATA1、2 番目のデータセットであれば DATA2 というように名前が付けられます。OUT=オプションを省略する場合、または OUT=オプションにライブラリ参照名を含む 2 レベル名を指定しない場合は、変換されたデータセットは WORK データライブラリに保存されます。WORK データライブラリは、デフォルトでは永久ライブラリではありません。

## 詳細

### CONVERT プロシジャの概説

CONVERT プロシジャは、BMDP システムファイル、OSIRIS システムファイル、SPSS エクスポートファイルを SAS データセットに変換します。その結果、出力データセットが 1 つ作成されます。出力は表示されません。新しいデータセットには入力システムと同じ情報が含まれます。例外については、“出力データセット” (430 ページ)を参照してください。BMDP、OSIRIS、SPSS の各エンジンは、より拡張された機能を提供しています。

BMDP、OSIRIS、SPSS プロダクトは他の会社や組織によってメンテナンスされるため、各システムファイルと、現バージョンの PROC CONVERT との互換性が失われる可能性があります。SAS Institute では、新しいバージョンの SAS の提供時にのみ、これらのプロダクトに加えられた変更に対応するために CONVERT プロシジャをアップグレードします。

### 欠損値

入力データセット内の数値変数に値が存在しないか、システム欠損値が存在する場合、CONVERT プロシジャはその変数に欠損値を割り当てます。

### 出力データセット

このセクションでは、*product-specification* の各値に対応する出力 SAS データセットの属性について説明します。

***注意:***

**変換後の名前が重複しないようにしてください。** SAS では、変数名が変換される場合もあります。プロシジャを正しく動作させるために、変換後の名前が重複しないように変数名を設定してください。

### BMDP の出力

BMDP 保存ファイルに含まれる変数名は、SAS データセット内で使用されます。ただし、末尾以外のブランクおよびすべての特殊文字は、SAS の変数名ではアンダースコアに変換されます。x(1)など、BMDP 変数名に含まれる添字は、かっこを取り除いた X1 が SAS 変数名の一部として使用されます。BMDP アルファベット変数の X1.は、同じ長さの SAS 文字変数になります。BMDP のカテゴリレコードは、変換されません。

### OSIRIS の出力

単一応答変数の場合、V1-V9999 の名前は SAS 変数名になります。複数応答変数の場合、接尾語 R*n* が変数名に追加されます。*n* は、その応答を表します。たとえば、V25R1 は、複数応答変数 V25 の最初の応答です。V1000 以降の変数に 100 以上の応答が含まれる場合、100 以上の応答は削除されます。OSIRIS で、文字モード、固定小数点バイナリモード、または浮動小数点バイナリモードで保存されている数値変数は、SAS の数値変数になります。アルファベット変数は、SAS の文字変数になります。長さが 200 を超えるアルファベット変数は、200 の長さに切り捨てられます。OSIRIS の変数説明は、SAS の変数ラベルになります。OSIRIS のプリント形式は、SAS の出力形式になります。

### SPSS の出力

SPSS の変数名および変数ラベルは、そのまま変更されずに変数名およびラベルになります。SPSS のアルファベット変数は、SAS の文字変数になります。SPSS のブランクの値は、SAS の欠損値に変換されます。SPSS のプリント形式は SAS 出力形式になります。小数点以下がない数値の SPSS のデフォルトの精度は、変数の出力形式の一

部になります。SPSS の値のラベルは、コピーされません。DOCUMENT データはコピーされます。したがって、CONTENTS プロシジャで表示できます。

### 比較

CONVERT プロシジャは、BMDP、OSIRIS、SPSS の各インターフェイスライブラリエンジンと密接に関係しています。実際、CONVERT プロシジャではこれらのエンジンが使用されます。たとえば、次の 2 つのコードセクションの実行結果はまったく同じになります。

- ```
  filename myfile 'myspss.por';
  proc convert spss=myfile out=temp;
   run;
  ```
- ```
  libname myfile spss 'myspss.por';
  data temp;
      set myfile._first_;
   run;
  ```

ただし、BMDP、OSIRIS、SPSS エンジンは、CONVERT プロシジャより拡張された機能を提供します。

## CPORT プロシジャ:Windows

SAS データセットおよび SAS カタログを、特殊形式の移送ファイルに書き込みます。

**Windows 固有:** 移送ファイルの名前および保存場所

**参照項目:** 15 章: “CPORT プロシジャ” (*Base SAS Procedures Guide*)

### 構文

**PROC CPORT** *source-type=libref| <libref.> member-name<option(s)>* ;

#### *必須引数*

この構文は CPORT プロシジャ構文を単純化したものです。完全な構文と説明については、CPORT プロシジャを参照してください。参照先: 15 章: “CPORT プロシジャ” (*Base SAS Procedures Guide*)

***libref***
変換するファイルの名前と保存場所を指定します。

### 詳細

CPORT プロシジャは、SAS データセット、SAS カタログ、または SAS ライブラリを順次ファイル形式(移送ファイル)に書き込みます。CPORT プロシジャを CIMPORT プロシジャとともに使用して、ファイルを異なる OS 間で移動することができます。

FILE=オプションの値には、FILENAME ステートメントで定義されたファイル参照名、引用符で囲まれた Windows パス名または環境変数を設定できます。

FILE=オプションを使用せず、予約済みファイル参照名 SASCAT が定義されていない場合、SASCAT.DAT というファイルが作業ディレクトリに作成されます。

*注:* SAS 9.1 から、MIGRATE プロシジャを使用して SAS ライブラリをこれまでのリリースから移行できるようになりました。

### 関連項目:

- “CIMPORT プロシジャ:Windows” (426 ページ)
- MIGRATE プロシジャとリリース間の互換性については、http://support.sas.com/rnd/migration/planning/files/regression.html を参照してください。

## DATASETS プロシジャ:Windows

SAS ファイルをリスト表示、コピー、名前変更、および削除します。SAS ライブラリのインデックスの管理、SAS データセットの追加もできます。

**Windows 固有:** ディレクトリ情報; CONTENTS ステートメントの出力

**参照項目:** 16 章: “DATASETS プロシジャ” (*Base SAS Procedures Guide*)

### 構文

**PROC DATASETS** <*オプション*> ;
**CONTENTS** <*オプション*> ;

#### *必須引数*

#### *オプション*

ここでは、DATASETS プロシジャの構文を簡単に説明します。完全な構文については、DATASETS プロシジャ *Base SAS プロシジャガイド*を参照してください。

### 詳細

DATASETS プロシジャは、SAS ファイルを管理するユーティリティプロシジャです。

DATASETS プロシジャによって SAS ログに表示される SAS ライブラリ情報は、使用している動作環境とエンジンによって異なります。次の例には、Windows 版で DATASETS プロシジャが SAS ログに出力する(V9 Engine の)情報を示します。

***アウトプット 20.1*** *DATASETS プロシジャから出力される SAS データライブラリ情報*

```
                PROC DATASETS library=work;
                                          Directory
                       Libref         WORK
                       Engine         V9
                       Physical Name  C:\DOCUME~1\sasusr\LOCALS~1\Temp\SAS
Temporary Files\_TD2663
                       File Name      C:\DOCUME~1\sasusr\LOCALS~1\Temp\SAS
Temporary Files\_TD2663
                                    Member     File
                       #  Name      Type       Size  Last Modified
                       1  GSEG      CATALOG   54272   17MAR2003:13:20:23
                       2  HAT       DATA      46080   17MAR2003:13:20:20
                       3  SASGOPT   CATALOG    5120   17MAR2003:13:20:21
```

この出力には、ライブラリ参照名、エンジン、ライブラリに割り当てられた物理名およびそのライブラリに含まれる SAS ファイルの名前とその他のプロパティが示されています。

DATASETS プロシジャの CONTENTS ステートメントは、CONTENTS プロシジャと同じ Engine/Host Dependent Information を出力します。

## 関連項目:

"CONTENTS プロシジャ:Windows" (427 ページ)

# OPTIONS プロシジャ:Windows

すべての SAS システムオプションの現在の値のリストを表示します。

**Windows 固有:** ホストオプション

**参照項目:** 5 章: "OPTIONS プロシジャ" (*SAS System Options: Reference*)

## 構文

**PROC OPTIONS** <*オプション*> ;

### *必須引数*

***オプション***

この構文は OPTIONS プロシジャ構文を単純化したものです。完全な構文と説明については、OPTIONS プロシジャ *Base SAS プロシジャガイド*を参照してください。

## 詳細

OPTIONS プロシジャは、現在設定されている SAS システムオプションをリスト表示します。

OPTIONS プロシジャで表示される動作環境に依存しないオプションは、すべての動作環境(セッションと構成)で使用できます。ただし、デフォルト値は多少異なることがあります。また、OPTIONS プロシジャで表示される環境固有のオプションは、動作環境ごとに異なります。次のコードをサブミットすると、Windows 環境固有のオプションの一部が表示されます。

```
proc options host;
run;
```

***ログ 20.2*** *OPTIONS プロシジャによって表示される Windows 動作環境のオプション*

```
Host Options:ACCESSIBILITY=STANDARD Enable Extended Accessibility
ALTLOG=           Specifies the destination for a copy of the SAS log
ALTPRINT=         Specifies the destination for a copy of the SAS procedure
output file AUTHPROVICERDOMAIN= Authentication providers associated with domain
suffixes AUTHSERVER=       Specify the authentication server or
domain.AUTOEXEC=         Specifies the autoexec file to be used
AWSCONTROL=(SYSTEMMENU MINMAX TITLE) Used to customize the appearance for the
SAS AWS.Valid parameters are:TITLE/NOTITLE SYSTEMMENU/NOSYSTEMMENU MINMAX/
NOMINMAX AWSDEF=(  0   0  79 79) Specify the initial size and position of the
SAS AWS.This should be specified as follows:0 0 100 100 AWSMENU           Show
the main window's (AWS) menu.AWSMENUMERGE      Add host specific menu items to
the main window's (AWS) menu....NOTE:PROCEDURE OPTIONS used:( Total process
time) real time 0.01 seconds cpu time 0.01 seconds
```

ここで示されているオプション値は、単なる例です。OPTIONS プロシジャの出力は、さまざまな要件に応じて異なります。一部のオプション値は、SAS の実行方式によって異なります。たとえば、SAS ウィンドウ環境におけるデフォルトの行サイズは、VGA ディスプレイでは 75 行です。ただし、バッチモードでは 132 行です。プロセスの設定方法によって、システムオプションのデフォルト値が変わることもあります。たとえば、

SASAUTOS=オプションのデフォルト値は、自動呼び出しマクロの保存場所に応じて異なります。

OPTIONS プロシジャを使用すると、すべてのシステムオプションの値を確認することができます。特定の動作環境オプションについての情報は、変数属性(“Windows 版に固有の SAS システムオプション” (473 ページ))を参照するか、または SAS ヘルプとドキュメント内の、ユーザーの動作環境での SAS ソフトウェアの使用を参照してください。

### 関連項目:

“Windows 版に固有の SAS システムオプション” (473 ページ)

## PMENU プロシジャ:Windows

SAS ソフトウェアの各種ウィンドウで表示できるメニューの機能を定義します。

**Windows 固有:** ACCELERATE=オプションに指定できる数種類のキーの組み合わせ

**参照項目:** 40 章: “PMENU プロシジャ” (*Base SAS Procedures Guide*)

### 構文

**PROC PMENU** <CATALOG=<*libref.*> *catalog*> <DESC '*entry-description*'> ;

**ITEM** *command* <*option(s)*> ;

**ITEM** '*menu-item*' <*オプション*> ;

ACCELERATE=*name-of-key*;

#### *必須引数*

この構文は PMENU プロシジャ構文を単純化したものです。完全な構文については、40 章: “PMENU プロシジャ” (*Base SAS Procedures Guide*)を参照してください。

**ACCELERATE=*name-of-key***

メニュー項目を選択するかわりに使用できるキー順を定義します。このショートカットキーを押すと、メニューバーまたはプルダウンメニューの項目を選択した場合と同じ処理が行われます。

Windows 版では、ITEM ステートメントの ACCELERATE=オプションで次のキーの組み合わせのみを指定できます。

- Ctrl+A (すべて選択)
- Ctrl+C (コピー)
- Ctrl+F (検索)
- Ctrl+N (新規作成)
- Ctrl+O (インクルード)
- Ctrl+P (印刷)
- Ctrl+S (ファイル)
- Ctrl+V (貼り付け)
- Ctrl+X (切り取り)
- Ctrl+Z (元に戻す)
- Del (消去)

## 詳細

PMENU プロシジャでは、DATA ステップウィンドウ、マクロウィンドウ、SAS/AF と SAS/FSP のウィンドウまたはカスタマイズしたメニューを指定できる SAS アプリケーションで使用できるメニューを定義します。

SAS プログラムで代替のキーの組み合わせを使用する場合、**編集**メニューには標準のキーの組み合わせが表示されます。ただし、メニュー項目を有効にするには、標準または代替のキーの組み合わせのどちらかひとつのみ使用できます。

# PRINTTO プロシジャ:Windows

SAS プロシジャ出力および SAS ログの出力先を定義します。

**Windows 固有:** *file-specification* の有効な値; UNIT=オプション

**参照項目:** 43 章: “PRINTTO プロシジャ” (*Base SAS Procedures Guide*)

## 構文

**PROC PRINTTO** <*option(s)*> ;

### *必須引数*

### *オプション*

**LOG=*file-specification* PRINT=*file-specification***
次のいずれかを指定できます。

- FILENAME ステートメントまたは FILENAME 関数で定義されたファイル参照名。SAS 出力またはログを直接プリンタへ送るには、デバイスタイプキーワード PRINTER を指定して FILENAME ステートメントまたは FILENAME 関数を使用します。
- 引用符で囲まれた Windows パス名。
- 英数文字列。出力先ファイル名は、*file-specification*.LOG または *file-specification*.LST となり、現在のディレクトリに保存されます。

  *注:* AUX、CON、NUL、PRN、LPT1 - LPT9、COM1 - COM9 は Windows の予約語です。これらの用語をファイル参照名として使用しないでください。
- SAS 環境変数または Windows 環境変数。

**UNIT=*nn***
SAS プロシジャ出力を FT*nn*F001.LST というファイルに出力します。*nn* は UNIT=値を表し、1 から 99 の範囲で指定できます。ファイルは、SAS 作業ディレクトリに置かれます。

## 詳細

ここでは、PRINTTO プロシジャの構文を簡単に説明します。完全な構文および その説明については、43 章: “PRINTTO プロシジャ” (*Base SAS Procedures Guide*) を参照してください。

PRINTTO プロシジャは、SAS プロシジャ出力と SAS ログの出力先を定義します。

## 例

### *例 1: SAS のログ出力の出力先の変更*

次のステートメントでは、RUN ステートメントの実行後に生成される SAS ログのエントリを、TEST というファイル参照名を持つ出力ファイルへ出力します。TEST は、LPT1: デバイスに割り当てられています。

```
filename test printer 'lpt1:';
proc printto log=test;
run;
```

これらのステートメントが発行されると、ダイアログボックスが開いて PRINTTO プロシジャが実行中であると表示されます。すべての SAS ログエントリーは、指定されたとおりに TEST 出力ファイルに出力されます。ただし、SAS ログエントリーをデフォルトの出力先または他のファイルへと出力することにより、SAS ログエントリーは出力ファイルが閉じられてから LPT1:デバイスに表示されます。

次のステートメントでは、RUN ステートメントの実行後に生成される SAS ログのエントリを、ファイル参照名 MYFILE に割り当てられた外部ファイルに送ります。

```
filename myfile 'c:\mydir\mylog.log';
proc printto log=myfile;
run;
```

### *例 2: SAS のプロシジャ出力の出力先の変更*

次のステートメントでは、SAS プロシジャ出力を、作業ディレクトリ内の MYPRINT.LST というファイルに送ります。MYPRINT は、以前に定義されたファイル参照名または環境変数ではないと仮定します。

```
proc printto print=myprint;
run;
```

次のステートメントでは、SAS プロシジャ出力をプリンタポートに送ります。通常、プリンタポートはシステムにより LPT1:として定義されています。

```
proc printto print='lpt1:';
run;
```

### *例 3: 出力先をデフォルトへと変更*

オプションなしの PROC PRINTTO ステートメントを含めて、次のステートメントでは、SAS ログやプロシジャ出力をデフォルトの宛先に転送します。

```
proc printto;
run;
```

## SORT プロシジャ:Windows

SAS データセット内のオブザベーションを 1 つ以上の変数に基づきソートし、ソートしたオブザベーションを新しい SAS データセットに保存するか、または元のデータセットに上書きします。

**Windows 固有:** 使用可能なソートユーティリティ、SORTSIZE=ステートメントオプション、TAGSORT ステートメントオプション

**参照項目:** 55 章: “SORT プロシジャ” (*Base SAS Procedures Guide*)

# 構文

**PROC SORT** <*option(s)*> <*collating-sequence-option*> ;

## *必須引数*

**SORTSIZE=*memory-specification***
SORT プロシジャに使用できるメモリの最大量を指定します。SORTSIZE=オプションの詳細については、次の「詳細」を参照してください。

**TAGSORT**
BY 変数とオブザベーション番号だけを一時ファイルに格納します。TAGSORT を指定すると、シングルスレッドでソートされます。マルチスレッドでソートする場合は、TAGSORT を指定しないでください。TAGSORT オプションの詳細については、次の「詳細」を参照してください。

# 詳細

## *SORT プロシジャ構文*

ここでは、SORT プロシジャの構文を簡単に説明します。完全な構文と説明については、SORT プロシジャを参照してください。参照先: 55 章: “SORT プロシジャ” (*Base SAS Procedures Guide*)

SORT プロシジャを使用して、1 つ以上の文字変数または数値変数に基づき SAS データセットのオブザベーションをソートし、元のデータセットを上書きするか、またはソートされたデータセットを新規に作成します。Windows 環境のデフォルトでは、ASCII 照合順序が使用されます。

SORT プロシジャでは、SORTPGM システムオプションに指定されたソートユーティリティが使用されます。ソートは、SAS、データベース、Windows の SyncSort ユーティリティで行うことができます。SAS ソートユーティリティで利用できるオプション(SORTSEQ、NODUPKEY など)は、すべて使用することができます。利用できるオプションの完全なリストについては、「関連項目」を参照してください。

## *SORTSIZE=オプション*

Windows 環境では、PROC SORT ステートメントの SORTSIZE=オプションを使用して、SORT プロシジャで利用できるメモリの量を制限できます。このオプションによって、データセットのソートの際に行われるスワッピングの回数を減らすことができます。SORT プロシジャでは、指定したメモリ量よりも多くのメモリが必要な場合、SASWORK ディレクトリに、データを格納するための一時ユーティリティファイルが作成されます。SORT プロシジャのアルゴリズムでは、Windows よりも効率的にデータをスワップできます。

SORTSIZE=オプションの構文は次のとおりです。

## *構文*

SORTSIZE=*memory-specification*

ここでは、*memory-specification* には、次のいずれかを指定します。

*n*
メモリの量をバイト単位で指定します。

*n*K
メモリの量を 1KB 単位で指定します。

*n*M
メモリの量を 1MB 単位で指定します。

デフォルトの SAS 構成ファイルで SORTSIZE=システムオプションにより、このオプションは 1G に設定されます。

SORTSIZE=システムオプションのデフォルト値をオーバーライドするには、PROC SORT ステートメントで SORTSIZE=に別の値を指定するか、OPTIONS ステートメントをサブミットして SORTSIZE=システムオプションに新しい値を設定します。

### *TAGSORT オプション*

PROC SORT ステートメントの TAGSORT オプションは、大きな SAS データセットをソートするのに十分なディスク容量がない場合に役立ちます。TAGSORT を指定すると、シングルスレッドでソートされます。マルチスレッドでソートする場合は、TAGSORT を指定しないでください。

TAGSORT オプションを指定すると、ソートキー(BY ステートメントに指定された変数)と各オブザベーションのオブザベーション番号だけが、一時ファイルに格納されます。ソートキーとオブザベーション番号を合わせて、タグと呼ばれます。ソート処理が完了すると、入力データセットのレコードが、タグに基づいてソート順に取り出されます。このため、ソートキーの合計バイト数がレコード長に比べて小さい場合は、一時ディスクの使用量が大幅に減少します。もう 1 つのデータ(出力データセット)または 2 つのタグセットのうち、いずれか大きい方を格納できるだけのディスク容量を確保する必要があります。TAGSORT オプションを使用すると、一時ディスクの使用量は減少しますが、処理時間は大幅に伸びる可能性があります。ただし、空きディスク容量の限られたコンピュータでは、通常はソートできない状況でも、TAGSORT オプションを使用するとソートできる場合があります。

### *独自の照合順序の作成*

独自の照合順序を作成したり、または提供された照合順序を変更したい場合は、TRANTAB プロシジャを使用して変換テーブルを作成または変更します。TRANTAB プロシジャの詳細については、*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。独自の変換テーブルを作成すると、変換テーブルは SASUSER.PROFILE カタログに格納され、HOST カタログに格納されている同じ名前の変換テーブルより優先されます。

*注:* システム管理者は、新しく作成されたテーブルを SASUSER.PROFILE カタログから HOST カタログにコピーすることによって、HOST カタログを変更できます。その後、すべてのユーザーは新規のまたは変更された変換テーブルにアクセスすることができます。

SAS エクスプローラを用いて HOST カタログに格納されている照合順序名を参照する場合、次のステートメントをサブミットしてください。

```
dm 'catalog sashelp.host' catalog;
```

また、**ビュー**メニューから **explorer** 項目を選択し、Sashelp ライブラリをダブルクリックして、HOST カタログをダブルクリックすることもできます。バッチモードの場合は、次のステートメントを使用して、HOST カタログの内容のリストを生成できます。

```
proc catalog catalog=sashelp.host;
   contents;
run;
```

TRANTAB の入力エントリーは照合順序です。

特定の変換テーブルを内容を参照する場合は、次のステートメントを使用してください。

```
proc trantab table=table-name;
   list;
run;
```

照合順序の内容は SAS ログに表示されます。

### SAS での SyncSort の使用

サイトに SyncSort がインストールされている場合、データベースソートや SAS ソートのかわりに、ソートアルゴリズムとして SyncSort を使用できます。SORTPGM システムオプション、SORTCUT システムオプション、SORTCUTP システムオプションに設定されている値によって、使用するソート方法が決まります。

SyncSort のインストール時に、SyncSort ディレクトリが Windows の PATH ステートメントに追加されます。SyncSort ディレクトリが Windows の PATH ステートメントに追加されると、SAS で SyncSort を起動することができます。SyncSort は Syncsort, Inc により開発されました。

### ソートアルゴリズムとして SyncSort を設定

常に SyncSort のソートルーチンを使用してソートするには、SORTPGM システムオプションの値を HOST にする必要があります。このオプションを設定するには、次の OPTIONS ステートメントをサブミットします。

```
options sortpgm=host;
```

*注:* SORTPGM オプションは、**システムオプション**ウィンドウから、SAS 構成ファイルに設定したり、SAS の起動時に設定したりすることもできます。SAS の起動時または SAS 構成ファイルに SORTPGM システムオプションを指定する方法を次に示します。

```
-sortpgm host
```

### サイズまたはオブザベーションを基準にした並べ替え

SAS が使用するソートルーチンは、データセットのオブザベーション数またはサイズに基づきます。SORTPGM オプションを BEST に設定すると、この優先順位を基に、最初に利用できる適切なソートアルゴリズムが使用されます。

- データベースソートユーティリティ
- ホストソートユーティリティ
- SAS ソートユーティリティ

データベースソートを使用しない場合、SAS は SORTCUT オプションと SORTCUTP オプションの値を調べ、使用するソート方法を決定します。

オブザベーション数が sortcut の値よりも大きいか、または等しい場合に、SyncSort が使われます。SORTCUTP オプションは、SyncSort を使用する時のデータセットのバイト数を指定します。

SORTCUT と SORTCUTP を 0 に設定すると、SAS ソートルーチンが使用されます。両方のオプションが設定されて、どちらの条件も満たされるならば、SAS は SyncSort を使用します。

次の OPTIONS ステートメントが有効な場合、オブザベーション数が 501 以上であると、SyncSort のルーチンが使用されます。

```
options sortpgm=best sortcut=500;
```

次の例では、データセットのサイズが 40MB を超える場合は、SyncSort のルーチンが使用されます。

```
options sortpgm=best sortcutp=40M;
```

これらのソートオプションに関する詳細については、“SORTPGM システムオプション:Windows” (568 ページ)、“SORTCUT システムオプション:Windows” (564 ペー

ジ) 、および“SORTCUTP システムオプション:Windows” (565 ページ)を参照してください。

### SyncSort 一時ファイルの場所の変更

デフォルトでは SyncSort は WORK オプションに指定された一時ファイルの場所を使用します。SyncSort の一時ファイルの場所を変更するには、SORTDEV オプションを使用して新しい場所を指定します。次に、例を示します。

```
options sortdev="c:\temp\sortsync";
```

SORTDEV オプションの詳細については、“SORTDEV システムオプション:Windows” (566 ページ)を参照してください。

### SyncSort へのオプションの受け渡し

SORTANOM オプションを使用して、SyncSort に使用するオプションを指定します。

*表 20.1 SyncSort の SORTANOM オプション*

| タスク | SORTANOM オプション |
|---|---|
| 単一呼び出しモードではなく複数呼び出しモードで実行 | SORTANOM=b |
| SAS ログに、ソートプロセスに関する統計量を出力 | SORTANOM=t |
| SAS ログに、Syncsort に渡されたコマンドを出力 | SORTANOM=v |

次のように、オプションを連結して、複数のオプションを指定することができます。

```
options sortdev=btv;
```

SORTANOM オプションの詳細については、“SORTANOM システムオプション:Windows” (563 ページ)を参照してください。

### SyncSort へのパラメータの受け渡し

SORTPARM オプションを使用して、SyncSort に Syncsort オプションを渡します。この OPTIONS ステートメントでは、次のようにオプションを引用符で囲みます。

```
options sortparm="SyncSort-options";
```

SORTPARM オプションの詳細については、“SORTPARM システムオプション:Windows” (567 ページ)を参照してください。SyncSort オプションの説明については、SyncSort のドキュメントを参照してください。

### ホストソートユーティリティを使用して SORTSEQ=オプションを指定

SORTSEQ=オプションを使用すると、ソートの照合順序を指定できます。有効な値のリストについては、*Base SAS プロシジャガイド*の SORT プロシジャを参照してください。

***注意:***

**データのソートにホストソートユーティリティを使用し、SORTSEQ=オプションを指定する際に、ソート順序の変換テーブルとその逆テーブルのマッピングが 1 対 1 でない場合は、文字変数の BY 変数が破損する可能性があります。** 変換テーブルでは、1 つの重みに対して 1 つの文字を割り当てる必要があります。また、逆テーブルでは、1 つの文字変数に対して 1 つの重みを割り当てる必要があります。

変換テーブルが 1 対 1 のマッピングでない場合、次の方法のいずれかを用いることでソートを実行できます。

- 1 対 1 にマッピングされた変換テーブルを作成します。1 対 1 にマッピングされた変換テーブルを作成すると、TRANTAB プロシジャを使用して、簡単に、対応する逆テーブルを作成できます。テーブルが 1 対 1 のマッピングでない場合、逆テーブルを作成しようとすると、SAS ログに次のメッセージが表示されます。

  ```
  NOTE:  This table cannot be mapped one to one.
  ```

  詳細については、TRANTAB プロシジャ(*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。

- SAS ソートを使用します。SORTPGM システムオプションを使用して、SAS ソートを指定できます。詳細については、"SORTPGM システムオプション:Windows" (568 ページ)を参照してください。

- ホストソートユーティリティの照合順序オプションを指定します。詳細については、ホストソートユーティリティのドキュメントを参照してください。

- ダミーの BY 変数を使用してビューを作成します。

*注:* これらの方法のいずれかを実行した後に、NOTSORTED オプションまたは NOBYSORTED システムオプションを使用した BY 処理が必要な場合もあります。NOTSORTED オプションの詳細については、BY ステートメント(*SAS ステートメント: リファレンス*)を参照してください。NOBYSORTED システムオプションの詳細については、BYSORTED 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。

### *例:ダミーの BY 変数を使用してビューを作成する*

次に、ダミーの BY 変数を使用してビューを作成するコードの例を示します。

```
options no date nostimer ls-78 ps-60;
options sortpgm=host msglevel=i;
data one;
   input name $ age;
   datalines;
   anne 35
   ALBERT 10
   JUAN 90
   janet 5
   bridget 23
   BRIAN 45
   ;
run;
data oneview / view=oneview;
   set one;
   name1=upcase(name);
run;
proc sort data=oneview out=final(drop=name1);
   by name1;
run;
proc print data=final;
run;
```

出力は、次のようになります。

***アウトプット 20.2*** *ダミーの BY 変数を使用してビューを作成する*

```
  The SAS System
Obs         name        age
 1          ALBERT       10
 2          anne         35
 3          BRIAN        45
 4          bridget      23
 5          janet         5
 6          JUAN         90
```

## 関連項目:

- 19 章: “TRANTAB プロシジャ” (*SAS National Language Support (NLS): Reference Guide*)
- “SORTANOM システムオプション:Windows” (563 ページ)
- “SORTCUT システムオプション:Windows” (564 ページ)
- “SORTCUTP システムオプション:Windows” (565 ページ)
- “SORTDEV システムオプション:Windows” (566 ページ)
- “SORTPARM システムオプション:Windows” (567 ページ)
- “SORTPGM システムオプション:Windows” (568 ページ)
- “SORTSIZE システムオプション:Windows” (569 ページ)
- “SORT プロシジャのパフォーマンス向上” (205 ページ)

*21 章*

# Windows 版に固有の SAS ステートメント

**Windows 版に固有の SAS ステートメント** . . . . . **443**

**ディクショナリ** . . . . . **443**
ABORT ステートメント:Windows . . . . . 443
ATTRIB ステートメント:Windows . . . . . 444
FILE ステートメント:Windows . . . . . 445
FILENAME ステートメント:Windows . . . . . 448
FOOTNOTE ステートメント:Windows . . . . . 454
%INCLUDE ステートメント:Windows . . . . . 455
INFILE ステートメント:Windows . . . . . 457
LENGTH ステートメント:Windows . . . . . 459
LIBNAME ステートメント:Windows . . . . . 460
SYSTASK ステートメント:Windows . . . . . 463
TITLE ステートメント:Windows . . . . . 466
WAITFOR ステートメント:Windows . . . . . 466
X ステートメント:Windows . . . . . 467

## Windows 版に固有の SAS ステートメント

SAS ステートメントは SAS への一種のディレクティブであり、SAS に対して特定の操作の実行を要求したり、後の操作に必要な情報を提供したりします。

すべての SAS ステートメントについては、*SAS ステートメント: リファレンス*を参照してください。

## ディクショナリ

### ABORT ステートメント:Windows

現在の DATA ステップ、SAS ジョブまたは SAS セッションの実行を中止します。

**該当要素:** DATA ステップ

**Windows 固有:** ABEND および RETURN オプションの操作、次の最大値:*condition-code*

**参照項目:** “ABORT Statement” (*SAS Statements: Reference*)

## 構文

**ABORT** <ABEND | RETURN | CANCEL | NOLIST> <*n*>;

### *オプション引数*

**ABEND**
現在のプロセスの現在の SAS ジョブまたは SAS セッションを異常終了させます。以後の動作については、使用している動作環境およびサイトにおける、異常終了したジョブの処理方法によって異なります。

**RETURN**
現在の SAS ジョブまたは SAS セッションを即座に正常終了させます。ジョブが異常終了した場合は、エラーを示す条件コードが返されます。

***n***
呼び出しプログラムに返される条件コードを指定します。*n* の値は、整数である必要があります。0 から 6 のリターンコードと、997 より大きいリターンコードは、SAS で使用されます。

**CANCEL <FILE>**
サブミットしたステートメントの実行をキャンセルさせます。結果は、処理方法によって異なります。

**NOLIST**
SAS ログへのすべての変数の出力を抑制します。

***n***
整数。これを使用すると条件コードを指定できます。

- CANCEL 引数を指定して使用する場合、値は SYSINFO 自動マクロ変数に配置されます。
- CANCEL 引数を指定して使用しない場合、SAS によって返されるエラーコードは ERROR です。ERROR の値はオペレーティングシステムに依存します。条件コード *n* は、SAS システム終了コードとしてオペレーティングシステムに返されます。

## 詳細

ABORT ステートメントは、現在の DATA ステップの処理を中止させます。

ABEND オプションと RETURN オプションは、どちらも SAS のプロセス、ジョブまたはセッションを終了させます。

## 関連項目:

- 詳細については、“ABORT Statement” (*SAS Statements: Reference*)を参照してください。
- “リターンコードと完了ステータス” (617 ページ)

# ATTRIB ステートメント:Windows

変数に出力形式、入力形式、ラベル、長さを関連付けます。

**該当要素:** DATA ステップ

**Windows 固有:** 長さの指定

**参照項目:** “ATTRIB Statement” (*SAS Statements: Reference*)

## 構文

**ATTRIB** *variable-list-1 attribute-list-1*…<*variable-list-n attribute-list-n*> ;

### *構文記述*

ここでは、ATTRIB ステートメントの構文を簡単に説明します。詳しい構文とその説明については、“ATTRIB Statement” (*SAS Statements: Reference*)の ATTRIB ステートメントを参照してください。

### *属性リスト*

**LENGTH=<$>*長さ***
*variable-list* の変数の長さを指定します。Windows 版では、数値変数に指定できる長さは 3 から 8 バイトです。

**variable-list**
属性に関連付ける変数の名前を付けます。

## 詳細

DATA ステップで ATTRIB ステートメントを使用すると、変数が含まれている SAS データセット内のディスクリプタ情報が変更されるため、属性が変数に永久に関連付けられます。

# FILE ステートメント:Windows

PUT ステートメントの現在の出力ファイルを指 定します。

**該当要素:** DATA ステップ

**制限事項:** SAS がロックダウンされた状態にある場合、FILENAME ステートメントはロックダウンパスリストにないファイルに対して使用できません。詳細については、“SAS Processing Restrictions for Servers in a Locked-Down State” (*SAS Language Reference: Concepts* 2 章).を参照してください。

**Windows 固有:** 次に対する有効な値:*ファイル指定*; *encoding-value*。次に対する有効なオプション *host-option-list*

**参照項目:** “FILE Statement” (*SAS Statements: Reference*)

## 構文

**FILE** *file-specification*<PERMISSION='*permission-value*'><ENCODING='*encoding-value*'> <*option-list*> <*host-option-list*> ;

### *必須引数*

***file-specification***
ファイル指定形式として、*Windows 版 SAS* の「外部ファイルの使用」セクションの「外部ファイルの参照」に示されているいずれかを使用できます。

AUX、CON、NUL、PRN、LPT1 - LPT9、COM1 - COM9 は Windows の予約語です。これらの用語をファイル名として使用しないでください。

## *オプション引数*

**PERMISSION='*permission-value*'**
指定されたファイル参照名に設定する権限を指定します。

'***permission-value***'

```
A::<trustee_type>::<permissions>
```

**A**
アクセス権限です。その他の値はサポートされていません。

***trustee_type***

u ユーザー

g グループ(すべてのグループ)

o その他(ファイルを生成したユーザーを含むその他すべてのユーザー)

権限の値としては、r (読み取り)、w (書き込み)、x (実行)をこの順序で指定します。付与しない権限がある場合は、その位置に代わりに「-」を入力します(例、r-x、rw-)。

複数セットの権限値を指定する場合は、それぞれを引用符で囲んでカンマで区切ります。

自分の所属するすべてのグループに読み取りおよび実行権限を指定するには、権限値として A::g::r-x と指定します。権限値として A::o::r--と指定すると、すべてのユーザーにファイルへの読み取りアクセスが許可されます。次のコードは、権限値を指定しています。

```
permission='A::o::r--'
```

**ENCODING='*encoding-value*'**
出力ファイルに書き出す際に使用するエンコーディングを指定します。ENCODING=の値により、出力ファイルのエンコーディングが現在のセッションのエンコーディングと異なることがわかります。

SAS では、出力ファイルにデータを書き込む際に、データのエンコードをセッションのエンコードから指定したエンコードに変換します。

有効なエンコードの値については、SAS 言語要素中のエンコーディング値(*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。

**option-list**
すべての動作環境で有効な FILE ステートメントのいずれかのオプションです。

***host-option-list***
Windowsd 動作環境に固有の外部入出力ステートメントオプションを指定します。次のオプションを指定できます。

**BLKSIZE=*block-size*BLK=*block-size***
入出力操作で物理的に読み書きするバイト数を指定します。デフォルトの設定は 8K です。最大 1MB まで指定できます。

**BLOCK | NOBLOCK**
名前付きパイプを指定しているときにのみ使用します。使用可能なデータがない場合にクライアントが待機するかどうかを指定します。デフォルト値は BLOCK です。

**BYTE | MESSAGE**
名前付きパイプを指定しているときにのみ使用します。このオプションでは、パイプのタイプを指定します。デフォルト値は BYTE です。

**COMMAND**
DDE (動的データ交換)を指定しているときにのみ使用します。SYSTEM トピック名を使用しないアプリケーションに対し、リモートコマンドを発行します。詳細については、*Windows 版 SAS* の「動的データ交換の使用」セクションの「DDE 外部ファイルの参照」と「DDE を使用して他のアプリケーションを制御する」を参照してください。

**EOFCONNECT**
名前付きパイプを指定しているときにのみ使用します。サーバーを定義しているときのみ有効です。ファイル終端文字(EOF)をクライアントから受け取ったときに、サーバーが次のクライアントに接続します。

**HOTLINK**
DDE (動的データ交換)を指定しているときにのみ使用します。詳細とこのオプションの使用例については、*Windows 版 SAS* の「動的データ交換の使用」セクションの「DDE HOTLINK の使用」を参照してください。

**IGNOREDOSEOF**
可変長レコード形式のファイルの入出力操作で使用します。このオプションを指定すると、^Z は EOF 記号ではなく文字データとして解釈されます。

**LRECL=*record-length***
レコード長をバイト単位で指定します。Windows ではデフォルト値は 32767 です。*record-length* の値は、1 から 1,073,741,823 (1 GB)の範囲から指定できます。

また、LRECL=システムオプションを使用して論理レコード長の値を指定することもできます。

**MOD**
出力を既存ファイルに追加します。

**NOTAB**
DDE (動的データ交換)を指定しているときにのみ使用します。変数間の区切り文字に、タブ以外の文字を使用できます。詳細については、*Windows 版 SAS* の動的データ交換の使用セクションの DDE での NOTAB オプションの使用に関する説明を参照してください。

**RECFM=*record-format***
レコードの形式を制御します。Windows の場合、次の値を指定できます。

| | |
|---|---|
| F | 固定長形式を指定します。 |
| N | バイナリ形式を指定し、ファイルはバイトストリームとして処理されます。LRECL を指定しない場合、デフォルト設定では、一度に 32767 バイトのデフォルト値がファイルから読み込まれます。 |
| P | 印刷形式を指定します。 |
| S370V | 可変長 S370 レコード形式(V)を指定します。 |
| S370VB | 可変長ブロック S370 レコード形式(VB)を指定します。 |
| S370VBS | スパンレコード可変長ブロック S370 レコード形式(VBS)を指定します。 |
| V \| D | 変数の出力形式を指定します。この形式はデフォルトです。 |

S370 値は、z/OS タイプのファイルにのみ有効です。つまり、可変長レコードを含む EBCDIC 形式のバイナリファイルに有効です。固定長形式の z/OS ファイルを使用するには、最初にファイルを可変長のバイナリ形式の z/OS ファイルにコピーします。

**RETRY=*seconds***
名前付きパイプを指定しているときにのみ使用します。名前付きパイプのクライアントがビジー状態のパイプを待つ時間の長さを指定します。*seconds* の最小(デフォルト)値は 10 です。

**SERVER | CLIENT**
名前付きパイプを指定しているときにのみ使用します。名前付きパイプのモードを指定します。デフォルト値は SERVER です。

**TERMSTR=**
ファイルの行末文字(EOL)を指定します。このオプションを使用して、UNIX 動作環境と Windows 動作環境間でファイルを共有します。有効な値は、次のとおりです。

**CRLF**
キャリッジリターンとラインフィード。TERMSTR=CRLF を使用すると、Windows 形式のファイルが書き込まれます。CRLF はデフォルトの設定です。

**LF**
ラインフィード。TERMSTR=LF を使用すると、UNIX 形式のファイルが書き込まれます。

**NL**
改行。TERMSTR=NL を使用すると、UNIX 形式のファイルが書き込まれます。

### 詳細

FILE ステートメントを使用すると、PUT ステートメントからの出力を、プロシジャ出力の出力先と同じ外部ファイルまたは別の外部ファイルに送ることができます。

FILE ステートメントに、ENCODING 引数が含まれ、ファイルに予約済みのファイル参照名 LOG または PRINT が指定されている場合、エラーメッセージが表示されます。FILE ステートメントの ENCODING 値は、ENCODING システムオプションの値より優先されます。

### 関連項目:

- “名前付きパイプの例” (284 ページ)
- “DDE の例” (270 ページ)

## FILENAME ステートメント:Windows

SAS ファイル参照名と外部ファイルまたは出力デバイスとの関連付け、ファイル参照名と外部ファイルの関連付けの解除、または外部ファイルの属性の表示を行います。

**該当要素:** SAS プログラム内の任意の場所

**制限事項:** SAS がロックダウン状態にある場合は、FILENAME ステートメントで次のアクセス方式を使用できません。サーバー管理者は、ロックダウン状態でもアクセス可能になるよう、これらのアクセス方式を再有効化できます。

- “ EMAIL” (450 ページ)
- “ FTP” (450 ページ)
- “Hadoop” (450 ページ)
- “ SOCKET” (451 ページ)
- “URL” (451 ページ)

詳細については、“SAS Processing Restrictions for Servers in a Locked-Down State” (*SAS Language Reference: Concepts* 2 章).を参照してください。

**Windows 固有:** *access-method* に有効な値; *device-type* に有効な値; *外部ファイル*に有効なファイル名; *エンコーディング*に有効な値; 次に対する有効なオプション *host-option-list*

**参照項目:** “FILENAME Statement” (*SAS Statements: Reference*)

## 構文

**FILENAME** *fileref* <*device-type*> '*external-file*' <PERMISSION='*permission-value*' > <ENCODING='*encoding-value*'>
<*host-option-list*> ;

**FILENAME** *fileref device-type*<'*external-file*'> <ENCODING='*encoding-value*>
<*host-option-list*> ;

**FILENAME** *fileref* <*device-type*>('*directory-1*'<,…*directory-n*'> )
<ENCODING='*encoding-value*>
<*host-option-list*> ;

### *システム説明*

ここでは、FILENAME ステートメントの構文を簡単に説明します。完全な構文およびその説明については、“FILENAME Statement” (*SAS Statements: Reference*) の FILENAME ステートメントを参照してください。

***fileref***

有効なファイル参照名です。詳細については、*Windows 版 SAS* の「外部ファイルの使用」セクションの「ファイル参照名の使用」を参照してください。

AUX、CON、NUL、PRN、LPT1 - LPT9、COM1 - COM9 は Windows の予約語です。これらの用語をファイル参照名として使用しないでください。

ファイル参照名の member-name 構文(集計構文)での使用例については、“ディレクトリへのファイル参照名の割り当て” (151 ページ) を参照してください。SAS がファイル参照名でファイルにアクセスする場合に適用される規則については、“連結ディレクトリのアクセス法について” (156 ページ)を参照してください。

***device-type***

ファイルではなくデバイスからデータを読み込み、書き込みできます。SFTP アクセスメソッドの詳細については次を参照してください。UNIX と Windows での SSH クライアントソフトウェアの設定。次の値が有効です。

**CATALOG**
SAS カタログを外部フラットファイルとして読み込みます。

**CLIPBOARD**
ホストコンピュータのクリップボードからテキストデータを読み込んだり、そこへテキストデータを書き込んだりします。

**DATAURL**
DATAURL アクセスメソッドを使用し、ユーザー指定テキストを読み込みます。

**DDE**
動的データ交換を使用して、別のアプリケーションからデータを読み込み、書き込みます。詳細については、“SAS での DDE 構文” (268 ページ)を参照してください。

**DISK**
ディスクファイルからデータを読み込み、書き込みます。Windows では、DISK が *デバイスタイプ*のデフォルト値です。

**DRIVEMAP**
使用可能なハードドライブ(ローカルとネットワーク)の情報を表示します。

**DUMMY**
ヌルな出力デバイスを指定します。この値は、テストを行うときに特に便利です。

**EMAIL**
SAS から電子メールを自動的に送信します。詳細については、“SAS を使用して電子メールの送信” (44 ページ)を参照してください。

**制限事項** SAS がロックダウン状態にある場合は、FILENAME ステートメントで EMAIL アクセス方式を使用できません。サーバー管理者は、ロックダウン状態でもアクセス可能になるよう、このアクセス方式を再有効化できます。詳細については、“SAS Processing Restrictions for Servers in a Locked-Down State” (*SAS Language Reference: Concepts* 2 章).を参照してください。

**FTP**
TCP/IP を使用して、他のマシン上の情報にアクセスします。使用しているローカルコンピュータに TCP/IP ソフトウェアと WINSOCK.DLL がインストールされている必要があります。また、FTP サーバーとして使用できるコンピュータに接続できるように設定する必要もあります。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*の FILENAME ステートメントを参照してください。

**Hadoop**
Hadoop Distributed File System (HDFS)でファイルにアクセスします。その場所は設定ファイルで指定されます。

**NAMEPIPE**
名前付きパイプにデータを書き込みます。詳細については、“名前付きパイプの使用 ” (282 ページ)を参照してください。

**PIPE**
名前の付いていないパイプにデータを書き込みます。詳細については、“名前の付いていないパイプの使用 ” (280 ページ)を参照してください。

**PLOTTER**
プロッタにアクセスします。Windows のプリントマネージャは使用されません。このデバイスタイプキーワードは、SAS/GRAPH と併用するときにのみ使用できます。

**PRINTER**
プリンタファイルまたはデバイスにアクセスします。デフォルトでは、このデバイスタイプキーワードを使用すると、出力は Windows のプリントマネージャを経由します。デフォルトプリンタの変更の詳細については、“SYSPRINT システムオプション:Windows” (578 ページ)を参照してください。

**SFTP**
SFTP プロトコルを使用してリモートファイルにアクセスします。

**SOCKET**
TCP/IP ソケットを使用して、情報を読み込み、書き込みます。使用しているローカルコンピュータに TCP/IP ソフトウェアと WINSOCK.DLL がインストールされている必要があります。SOCKET アクセス方式では、非ブロッキング方式でソケット要求が発行されます。SOCKET 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 *SAS 関数と CALL ルーチン: リファレンス*の FILENAME ステートメントおよび FILENAME 関数を参照してください。

**TEMP**
ファイル名が割り当てられている間だけ存在する一時ファイルを作成します。この一時ファイルは、論理名によってのみアクセスでき、論理名が存在する間だけ使用できます。物理パス名はユーザーには表示されません。物理パス名を指定した場合、エラーが返されます。TEMP デバイスによって扱われるファイルは、DISK ファイルと同じ属性を持ち、同じように操作できます。

**TERMINAL**
出力がある場合のみ有用です。出力は**メッセージログ**ウィンドウに送られます。

FILENAME ステートメントのデバイスタイプの指定例については、“高度な外部 I/O の操作法” (162 ページ)を参照してください。TAPE デバイスタイプキーワード(*SAS ステートメント: リファレンス*を参照)は、Windows 動作環境では使用できません。Windows 版 SAS プログラムで TAPE デバイスタイプキーワードを使用すると、エラーメッセージが返されます。Windows 版では、DISK デバイスタイプキーワードが受け入れられます。ただし、Windows ではディスクファイルが標準であるため、無視されます。

**URL**
URL アクセスメソッドを使用してリモートファイルにアクセスします。

**WebDAV**
WebDAV プロトコルを使用してリモートファイルにアクセスします。

**ZIP**
ZIP ファイルにアクセスします。

**ディレクトリ**
アクセスするファイルが格納されるディレクトリを指定します。

***external-file***
有効な Windows ファイルを引用符で囲んで指定します。詳細については、“外部ファイルの参照” (148 ページ)を参照してください。

**PERMISSION='*permission-value*'**
指定されたファイル参照名に設定する権限を指定します。

***'permission-value'***

```
A::<trustee_type>::<permissions>
```

**A**
アクセス権限です。その他の値はサポートされていません。

***trustee_type***

u ユーザー

g グループ(すべてのグループ)

o その他(ファイルを生成したユーザーを含むその他すべてのユーザー)

権限の値としては、r (読み取り)、w (書き込み)、x (実行)をこの順序で指定します。付与しない権限がある場合は、その位置に代わりに「-」を入力します(例、r-x、rw-)。

複数セットの権限値を指定する場合は、それぞれを引用符で囲んでカンマで区切ります。

自分の所属するすべてのグループに読み取りおよび実行権限を指定するには、権限値として **A::g::r-x** と指定します。すべてのユーザーにファイルに対する読み取りアクセスを付与するには、権限値として **A::o::r--**を指定します。たとえば、このオプションは、コードに次のように指定します。

```
permission='A::o::r--'
```

**ENCODING='*encoding-value*'**
外部ファイルの読み込みまたは書き込みに使用するエンコードを指定します。ENCODING=の値により、外部ファイルのエンコーディングが現在のセッションのエンコーディングと異なることがわかります。

SAS では、外部ファイルからデータを読み込む際には、データのエンコードを指定のエンコードからセッションのエンコードに変換します。また、外部ファイルにデータを書き込む際には、データのエンコードをセッションのエンコードから指定のエンコードに変換します。

有効なエンコードの値については、SAS 言語要素中のエンコーディング値(*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。

***host-option-list***
Windows に固有の外部 I/O ステートメントオプションを指定します。次のオプションを指定できます。

**ALTDEST=*ファイル名***
PRINTER デバイスタイプ指定時のみ使用します。*ファイル名*には、ファイル参照名に直接出力するときのファイル出力先を指定します。出力は、プリンタではなくディスクに書き込まれますが、*外部ファイル*引数で指定したプリンタに関連付けられているプリンタドライバによって処理されます。たとえば、

```
filename groupHP printer
   "HP LaserJet 4si, 1st floor"
   altdest=
      "C:\My SAS Files\Printer output\out.prn";
```

このように指定すると、**out.prn** への出力の作成に、指定したプリンタ(HP LaserJet 4si)に関連付けられたプリンタドライバが使用されます。このオプションでファイル参照名を指定すると、出力は実際にはプリンタに送られません。

**BLKSIZE=*block-size*BLK=*block-size***
入出力操作で物理的に読み書きするバイト数を指定します。デフォルトの設定は 8K です。最大 1MB まで指定できます。

**BLOCK | NOBLOCK**
名前付きパイプを指定しているときにのみ使用します。使用可能なデータがない場合にクライアントが待機するかどうかを指定します。デフォルト値は BLOCK です。

**BYTE | MESSAGE**
名前付きパイプを指定しているときにのみ使用します。このオプションでは、パイプのタイプを指定します。デフォルト値は BYTE です。

**COMMAND**
DDE (動的データ交換)を指定しているときにのみ使用します。SYSTEM トピック名を使用しないアプリケーションに対し、リモートコマンドを発行します。詳細については、“DDE 外部ファイルの参照” (269 ページ)および“DDE を使用して他のアプリケーションを制御する” (270 ページ)を参照してください。

**CONSOLE=*状態***
パイプを使用してアプリケーションを開く際の **DOS** ウィンドウの状態を指定します。指定できる値は、次のとおりです。

| | |
|---|---|
| MAX | **DOS** ウィンドウを最大化にして開きます |
| MIN | **DOS** ウィンドウを最小化にして開きます |
| NORMAL | **DOS** ウィンドウをマシンのデフォルトサイズで開きます。 |

このホストオプションは、PIPE キーワードを指定しているときにのみ有効です。

**EOFCONNECT**
名前付きパイプを指定しているときにのみ使用します。サーバーを定義しているときのみ有効です。ファイル終端文字(EOF)をクライアントから受け取ったときに、サーバーが次のクライアントに接続します。

**HOTLINK**
DDE (動的データ交換)を指定しているときにのみ使用します。詳細とこのオプションの使用例については、“DDE HOTLINK の使用” (274 ページ)を参照してください。

**IGNOREDOSEOF**
可変長レコード形式のファイルの入出力操作で使用します。このオプションを指定すると、^Z は EOF 記号ではなく文字データとして解釈されます。

**LRECL=*record-length***
レコード長をバイト単位で指定します。Windows ではデフォルト値は 32767 です。*record-length* の値は、1 から 1,073,741,823 (1 GB)の範囲から指定できます。

また、“LRECL= System Option” (*SAS System Options: Reference*) を使用して論理レコード長の値を指定することもできます。

**MOD**
出力を既存ファイルに追加します。

**NOTAB**
DDE (動的データ交換)を指定しているときにのみ使用します。変数間の区切り文字に、タブ以外の文字を使用できます。詳細については、“DDE で NOTAB オプションを使用する” (273 ページ)を参照してください。

**RECFM=*record-format***
レコードの形式を制御します。Windows の場合、次の値を指定できます。

| | |
|---|---|
| F | 固定長形式を指定します。 |
| N | バイナリ形式を指定し、ファイルはバイトストリームとして処理されます。N は、PIPE デバイスタイプおよび NAMEPIPE デバイスタイプでは使用できません。LRECL を指定しない場合、一度に 32767 バイトのデフォルト値がファイルから読み込まれるか、ファイルに書き込まれます。 |
| P | 印刷形式を指定します。 |
| S370V | 可変長 S370 レコード形式(V)を指定します。 |
| S370VB | 可変長ブロック S370 レコード形式(VB)を指定します。 |
| S370VBS | スパンレコード可変長ブロック S370 レコード形式(VBS)を指定します。 |
| V \| D | 変数の出力形式を指定します。この形式はデフォルトです。 |

S370 値は、z/OS タイプのファイルにのみ有効です。つまり、可変長レコードを含む EBCDIC 形式のバイナリファイルに有効です。固定長形式の z/OS ファイルを使用するには、最初にファイルを可変長のバイナリ形式の z/OS ファイルにコピーします。

**RETRY=*seconds***
名前付きパイプを指定しているときにのみ使用します。名前付きパイプのクライアントがビジー状態のパイプを待つ時間の長さを指定します。*seconds* の最小(デフォルト)値は 10 です。

**SERVER | CLIENT**
名前付きパイプを指定しているときにのみ使用します。名前付きパイプのモードを指定します。デフォルト値は SERVER です。

## 詳細

FILENAME ステートメントは、外部ファイルまたは出力デバイスに有効な SAS 名を関連付けます。外部ファイルは、Windows 動作環境で作成および保持されているファイルで、このファイルからデータを読み取る必要があります。

## 例: 外部ファイルの参照

ファイルまたはディレクトリの連結リストから外部ファイルを参照できます。FILENAME ステートメントには、次のようにワイルドカード文字*を使用できます。

```
filename read ('c:\myfiles\*.*','c:\myotherfiles\abc.dat');
data new;
infile read;
input;
run;
```

## 関連項目:

“高度な外部 I/O の操作法” (162 ページ)

# FOOTNOTE ステートメント:Windows

プロシジャ出力の一番下に最大 10 行のテキストを出力します。

**該当要素:** SAS プログラム内の任意の場所

**Windows 固有:** フットノートの最大長

**参照項目:** “FOOTNOTE Statement” (*SAS Statements: Reference*)

## 構文

**FOOTNOTE** <*n*> <'*text*' | "*text*">;

### *オプション引数*

***n***
フットノートに使用する行を指定します。

***text***
フットノートのテキストを一重引用符または二重引用符で囲んで指定します。

## 詳細

FOOTNOTE ステートメントは、FOOTNOTE ステートメントを関連付けたステップまたは RUN グループが実行されたときに、有効になります。ある行にフットノートを指定すると、その行のフットノートを取り消すか再定義するまで、すべてのページで同じフットノートが出力されます。

Windows 版では、フットノートの最大長は 256 文字です。指定したフットノートが LINESIZE システムオプションの値よりも大きい場合は、行サイズに切り捨てられます。

# %INCLUDE ステートメント:Windows

SAS ステートメントとデータ行を読み込んで実行します。

**該当要素:** SAS プログラム内の任意の場所

**Windows 固有:** *source*(ファイル指定を使用した場合); 次に対する有効なオプション *encoding-value* および *host-options*

**参照項目:** “%INCLUDE Statement” (*SAS Statements: Reference*)

## 構文

**%INCLUDE** *source* </<ENCODING='*encoding-value*'> <*host-options*> > ;

### *必須引数*

ここでは、%INCLUDE ステートメントの構文を簡単に説明します。完全な構文およびその説明については、“%INCLUDE Statement” (*SAS Statements: Reference*) の%INCLUDE ステートメントを参照してください。

***source***

アクセスする情報の場所を示します。ソースは、外部ファイルを指定するか、または現在の SAS セッションで入力したテキスト行を指定します。Microsoft Windows 動作環境で拡張エディタを使う場合、キーボード入力を示すためにアスタリスク(*)を使用することはできません。ファイルの指定には、“外部ファイルの参照” (148 ページ)に示されているファイル指定形式のいずれかを使用します。

メンバ名構文や、数字で始まるメンバ名を使用する場合は、メンバ名を引用符で囲みます。メンバ名にマクロ変数参照が含まれる場合は、二重引用符で囲みます。

### *オプション引数*

**ENCODING='*encoding-value*'**

指定したソースから読み込む際に使用するエンコーディングを指定します。ENCODING=の値により、指定したソースのエンコーディングが現在のセッションのエンコーディングと異なることがわかります。

SAS では、指定したソースからデータを読み込む際に、データのエンコードを指定のエンコードからセッションのエンコードに変換します。

有効なエンコード値については、22 章: “Encoding Values in SAS Language Elements” (*SAS National Language Support (NLS): Reference Guide*)を参照してください。

***host-options***

Windows 版に有効なステートメントオプションを指定します。オプションリストの前には、必ずスラッシュ(/)を入力します。Windows 版では、次のオプションが有効です。

**BLKSIZE=*block-size*BLK=*block-size***

入出力操作で物理的に読み書きするバイト数を指定します。デフォルトの設定は 8K です。最大 1MB まで指定できます。

**BLOCK | NOBLOCK**

名前付きパイプを指定しているときにのみ使用します。使用可能なデータがない場合にクライアントが待機するかどうかを指定します。

**BYTE | MESSAGE**

名前付きパイプを指定しているときにのみ使用します。このオプションでは、パイプのタイプを指定します。デフォルト値は BYTE です。

**EOFCONNECT**

名前付きパイプを指定しているときにのみ使用します。サーバーを定義しているときのみ有効です。サーバーは、ファイル終端文字(EOF)をクライアントから受け取ったときに、次のクライアントへの接続を試行します。

**IGNOREDOSEOF**

可変長レコード形式のファイルの入出力操作で使用します。このオプションを指定すると、^Z は EOF 記号ではなく文字データとして解釈されます。

**LRECL=*record-length***

レコード長をバイト単位で指定します。Windows ではデフォルト値は 32767 です。*record-length* の値は、1 から 1,073,741,823 (1 GB)の範囲から指定できます。

**NOTAB**

動的データ交換でのみ使用します。変数間の区切り文字に、タブ以外の文字を使用できます。詳細については、“DDE で NOTAB オプションを使用する” (273 ページ)を参照してください。

**RECFM=*record-format***

レコードの形式を制御します。Windows の場合、次の値を指定できます。

| | |
|---|---|
| F | 固定長形式を指定します。 |
| N | バイナリ形式を指定し、ファイルはバイトストリームとして処理されます。LRECL を指定しない場合、デフォルト設定では、一度に 32767 バイトがファイルから読み込まれます。 |
| P | 印刷形式を指定します。 |
| S370V | 可変長 S370 レコード形式(V)を指定します。 |
| S370VB | 可変長ブロック S370 レコード形式(VB)を指定します。 |
| S370VBS | スパンレコード可変長ブロック S370 レコード形式(VBS)を指定します。 |
| V\|D | 変数の出力形式を指定します。この形式はデフォルトです。 |

S370 値は、z/OS タイプのファイル、つまり、可変長レコードを含む EBCDIC 形式のバイナリファイルに有効です。固定長形式の z/OS ファイルを使用するには、最初にファイルを可変長のバイナリ形式の z/OS ファイルにコピーします。

## 詳細

%INCLUDE ステートメントを含むプログラムを実行すると、SAS では、%INCLUDE を使用してプログラムに挿入されたステートメントやデータ行を含むコードが実行されます。

プログラムの用意のためにキーボードを使用すると、現在のプログラムの実行を中断できます。キーボードを使用しながら Microsoft Windows 動作環境で拡張エディタを使う場合、キーボード入力を示すためにアスタリスク(*)を使用することはできません。キーボード入力方法の詳細については “%INCLUDE Statement” (*SAS Statements: Reference*)を参照してください。

# INFILE ステートメント:Windows

INPUT ステートメントで読み込む外部ファイルを指定します。

**該当要素:** DATA ステップ

**Windows 固有:** 次に対する有効な値:*エンコーディング値*、*ファイル指定*、および*ホストオプション*

**参照項目:** “INFILE Statement” (*SAS Statements: Reference*)

## 構文

**INFILE** *file-specification* <ENCODING='*encoding-value*'> <*options*><*host-options*>;

### *必須引数*

***file-specification***
入力データレコードのソースを指定します。ソースは通常、外部ファイルになります。*file-specification* 引数には、“外部ファイルの参照” (148 ページ)に示されているファイル指定形式のいずれかを使用します。予約済みのファイル参照名 CARDS を指定すると、INFILE ステートメントでストリーム内データを参照できます。

AUX、CON、NUL、PRN、LPT1 - LPT9、COM1 - COM9 は Windows の予約語です。これらの用語をファイル参照名として使用しないでください。

### *オプション引数*

**ENCODING='*encoding-value*'**
外部ファイルから読み込む際に使用するエンコーディングを指定します。ENCODING=の値により、外部ファイルのエンコーディングが現在のセッションのエンコーディングと異なることがわかります。

SAS では、外部ファイルからデータを読み込む際には、データのエンコードを指定のエンコードからセッションのエンコードに変換します。

有効なエンコードの値については、22 章: “Encoding Values in SAS Language Elements” (*SAS National Language Support (NLS): Reference Guide*)を参照してください。

***host-options***
Windowsd 動作環境に固有の外部入出力ステートメントオプションを指定します。次のオプションを指定できます。

**BLKSIZE= *block-size*BLK=*block-size***
入出力操作で物理的に読み書きするバイト数を指定します。デフォルトの設定は 8K です。最大 1MB まで指定できます。

**BLOCK | NOBLOCK**
名前付きパイプを指定しているときにのみ使用します。使用可能なデータがない場合にクライアントが待機するかどうかを指定します。デフォルト値は BLOCK です。

**BYTE | MESSAGE**
名前付きパイプを指定しているときにのみ使用します。このオプションでは、パイプのタイプを指定します。デフォルト値は BYTE です。

**EOFCONNECT**
名前付きパイプを指定しているときにのみ使用します。サーバーを定義しているときのみ有効です。ファイル終端文字(EOF)をクライアントから受け取ったときに、サーバーが次のクライアントに接続します。

**HOTLINK**
DDE (動的データ交換)を指定しているときにのみ使用します。詳細とこのオプションの使用例については、“DDE HOTLINK の使用” (274 ページ)を参照してください。

**IGNOREDOSEOF**
可変長レコード形式のファイルの入出力操作で使用します。このオプションを指定すると、^Z は EOF 記号ではなく文字データとして解釈されます。

**LRECL=*record-length***
レコード長をバイト単位で指定します。Windows ではデフォルト値は 32767 です。*record-length* の値は、1 から 1,073,741,823 (1 GB)の範囲から指定できます。

**NOTAB**
DDE (動的データ交換)を指定しているときにのみ使用します。変数間の区切り文字に、タブ以外の文字を使用できます。詳細については、“DDE で NOTAB オプションを使用する” (273 ページ)を参照してください。

**RECFM=*record-format***
レコードの形式を制御します。Windows 版では、次の値が有効です。

| | |
|---|---|
| F | 固定長形式を指定します。 |
| N | バイナリ形式を指定し、ファイルはバイトストリームとして処理されます。LRECL を指定しない場合、デフォルト設定では、一度に 32767 バイトがファイルから読み込まれます。 |
| P | 印刷形式を指定します。 |
| S370V | 可変長 S370 レコード形式(V)を指定します。 |
| S370VB | 可変長ブロック S370 レコード形式(VB)を指定します。 |
| S370VBS | スパンレコード可変長ブロック S370 レコード形式(VBS)を指定します。 |
| V \| D | 変数の出力形式を指定します。この形式はデフォルトです。 |

S370 値は、z/OS タイプのファイルにのみ有効です。つまり、可変長レコードを含む EBCDIC 形式のバイナリファイルに有効です。固定長形式の z/OS ファイルを使用するには、最初にファイルを可変長のバイナリ形式の z/OS ファイルにコピーします。

**RETRY=*seconds***
名前付きパイプを指定しているときにのみ使用します。名前付きパイプのクライアントがビジー状態のパイプを待つ時間の長さを指定します。*seconds* の最小(デフォルト)値は 10 です。

**SERVER | CLIENT**
名前付きパイプを指定しているときにのみ使用します。名前付きパイプのモードを指定します。デフォルト値は SERVER です。

**TERMSTR=**
ファイルの行末文字(EOL)を指定します。このオプションを使用して、UNIX 動作環境と Windows 動作環境間でファイルを共有します。TERMSTR が指定されていない場合、1 つの LF または CRLF が行末文字(EOL)として機能します TERMSTR=CRLF の場合、CRLF が行末文字(EOL)として機能します。Windows を次に示します。

**CRLF**
キャリッジリターンとラインフィード。TERMSTR=CRLF を使用すると、Windows 形式または DOS 形式のファイルが読み込まれます。CRLF はデフォルトの設定です。

**LF**
ラインフィード。TERMSTR=LF を使用すると、UNIX 形式のファイルが読み込まれます。ファイルに CRLF 文字が含まれる場合、CR が行末文字(EOL)ではなくデータの一部として機能します。

**CR**
キャリッジリターン。行末文字(EOL)が CR の場合、TERMSTR=CR を使用してください。

## 詳細

INFILE ステートメントに、ENCODING 引数が含まれ、ファイルに CARDS、CARDS4、DATALINES、あるいは DATALINES4 が指定されている場合、エラーメッセージが表示されます。INFILE ステートメントの ENCODING 値は、ENCODING システムオプションの値より優先されます。

## 例: 外部ファイルの参照

ファイルまたはディレクトリの連結リストから外部ファイルを参照できます。INFILE ステートメントには、次のようにワイルドカード文字*を使用できます。

```
data new;
infile '("c:\myfiles\*.*","c:\myotherfiles\abc.dat")';
input;
run;
```

## 関連項目:

- “名前付きパイプの例” (284 ページ)
- “DDE の例” (270 ページ)

# LENGTH ステートメント:Windows

数値変数の保存に使用されるバイト数を指定します。

**該当要素:** DATA ステップ

**Windows 固有:** 数値変数の有効な長さ; 次に対する有効な値:*長さ*; 次に対する有効な値:*n*

**参照項目:** “LENGTH Statement” (*SAS Statements: Reference*)

## 構文

**LENGTH** <*variable-1*> <*…variable-n*> <$> <*length*> <DEFAULT=*n*>;

### *オプション引数*

**variable**
長さとして割り当てられる 1 つ以上の変数を指定します。これには、出力データセットからドロップしたものを含め、DATA ステップ内の任意の変数が含まれます。

**$**
先行する変数が文字変数であるように指定します。

***length***
Windows 版では、数値変数の長さは 3 から 8 バイトです。

**DEFAULT=*n***
新規作成される数値変数の値の保存にデフォルトで使用されるバイト数を、8 から *n* に変更します。Windows 版では、*n* の値は 3 から 8 バイトです。

## 詳細

LENGTH ステートメントは、作成されるデータセットで変数値の保存に使用されるバイト数を指定します。

***注意:***
**8 バイト未満の長さを指定すると、変数値の精度が損なわれることがあります。**

## 関連項目:

“Windows 版での変数の長さと精度” (595 ページ)

# LIBNAME ステートメント:Windows

ライブラリ参照名を SAS ライブラリに関連付けて、SAS ライブラリのファイル属性のリストを表示します。

**該当要素:** SAS プログラム内の任意の場所

**Windows 固有:** 次に対する有効な値:*engine*; 次に対する仕様:*SAS-データライブラリ*

**参照項目:** “LIBNAME Statement” (*SAS Statements: Reference*)

## 構文

**LIBNAME** *libref* <*engine-name*> '('*'SAS-data-library-1'* <,…*'SAS-data-library-n'*>')' <MEMLIB>, <FILELOCKWAIT>;

**LIBNAME** *libref* _ALL_ LIST;

**LIBNAME** *libref* _ALL_ CLEAR;

### 構文記述

ここでは、LIBNAME ステートメントの構文を簡単に説明します。完全な構文およびその説明については、LIBNAME ステートメントを参照してください。“LIBNAME Statement” (*SAS Statements: Reference*)

***libref***
有効なライブラリ参照名を指定します。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。

***engine-name***
Windows 版でサポートされている次のライブラリエンジンの 1 つを指定します。

| | |
|---|---|
| V9 | SAS 9、SAS 9.1、SAS 9.2、SAS9.3、SAS9.4 のデータセットにアクセスできます。このエンジンに対しては、ニックネーム BASE を使用できます。 |
| V8 | バージョン 8、リリース 8.1 およびリリース 8.2 のデータセットにアクセスできます。 |
| V7 | バージョン 7 のデータセットにアクセスできます。 |
| V6 | リリース 6.08 から 6.12 のデータセットにアクセスできます。V604 Engine を使用すると、32 ビット Windows 版 SAS 9.2 のセッションから直接、リリース 6.03 とリリース 6.04 の SAS データセットを読み込むことができます。リリース 6.03 とリリース 6.04 の SAS データセットは、x64 64 ビット環境と Itanium 64 ビット環境と互換性がありません。 |
| V604 | リリース 6.03 およびリリース 6.04 のデータセットにアクセスできます。 |
| XML | SAS データセットから XML ドキュメントを生成します。 |
| XPORT | 移送形式ファイルにアクセスできます。 |
| BMDP | 32 ビット動作環境の BMDP データファイルにアクセスできます。 |
| OSIRIS | OSIRIS データファイルにアクセスできます。 |
| SPSS | SPSS エクスポートファイルにアクセスできます。 |

これらのエンジンの詳細については、“マルチエンジンアーキテクチャ” (122 ページ)を参照してください。

***SAS-data-library***
Windows での SAS ライブラリの物理名を指定します。Windows で有効なパス名、または Windows で有効なパス名に設定された環境変数を指定する必要があります。複数のディレクトリを連結して、単一の SAS ライブラリとして機能させることもできます。複数のライブラリを指定する場合、最初と最後のライブラリのパス名の前後をかっこで囲みます。連結 SAS ライブラリの詳細については、“複数のフォルダが格納された SAS ライブラリへのアクセス方法について” (132 ページ)を参照してください。

**MEMLIB**
このライブラリに対して、ESMA (Extended Server Memory Architecture)メモリを使用することを指定します。拡張メモリの使用の詳細については、“メモリベースのライブラリ” (200 ページ)を参照してください。

**FILELOCKWAIT=*n***
ロックされたファイルが別プロセスで使用できるようになるまで待機する秒数を指定します。ロックされたファイルが *n* に指定した秒数より前に解放された場合、この解放されたファイルは現在のプロセスに対してロックされて処理が続行されます。

指定した秒数を経過してもファイルがロックされたままの場合、"Locked File"エラーがログに書き込まれ、DATA ステップは失敗します。

| | |
|---|---|
| デフォルト | 0 |
| 範囲 | 0 – 600 |
| 操作 | FILELOCKWAIT=システムオプションを指定すると、別のプロセスで、ロックされている SAS ファイルの解放を待機している SAS/SHARE Server やクライアントセッションに悪影響を与えることがあります。待機条件によっては、SAS/SHARE Server やクライアントの処理でエラーが発生する場合があります。<br><br>SAS/SHARE プロセスでエラーが発生しないようにするには、FILELOCKWAIT=0 に指定します。この値は、SAS/SHARE Server やクライアントが、ロックされたファイルの解放を待機するのをキャンセルします。待機をキャンセルすると、プロセスの失敗が避けられます。詳細については、“FILELOCKWAITMAX= システムオプション:Windows” (507 ページ)を参照してください。 |

## 詳細

### *LIBNAME ステートメントの概要*

LIBNAME ステートメントは、ライブラリ参照名を永久 SAS ライブラリに関連付けます。また、SAS ライブラリのファイル属性を表示するときにも使用できます。LIBNAME ステートメントは、ライブラリ参照名を消去するためにも使用します。詳細については、“ライブラリ参照名のクリア” (131 ページ)を参照してください。

*注:* AUX、CON、NUL、PRN、LPT1 - LPT9、COM1 - COM9 は Windows の予約語です。これらの用語をライブラリ参照名として使用しないでください。

### *ライブラリ参照名の関連付け*

ライブラリ参照名またはエンジンを SAS ライブラリに関連付けるには、次の形式の LIBNAME ステートメントを使用します。

**LIBNAME** *libref* <*engine-name*> '*SAS-data-library*'

**LIBNAME** *libref* <*engine-name*>('*SAS-data-library-1*' <,…'*SAS-data-library-n*')> ;

*SAS-データライブラリ*が物理パスの場合は、引用符で囲みます。ライブラリ参照名を連結する場合は、引用符は不要です。

この形式の LIBNAME ステートメントには、LIBNAME ステートメント構文で示した同じ引数を使用できます。

### *データライブラリ属性のリスト表示*

LIBNAME ステートメントに LIST オプションを指定すれば、SAS ライブラリの属性をリスト表示できます。次の LIBNAME ステートメントはデータライブラリ属性のリスト表示をします。

```
libname sashelp
list;
```

***アウトプット 21.1*** *LIBNAME ステートメントで表示されるデータライブラリ属性のリスト*

```
5    libname sashelp list;
1    libname sashelp list;
NOTE: Libref=   SASHELP
      Scope=    Kernel
      Levels=   27
        -Level 1-
      Engine=   V9
      Physical Name= C:\Program Files\SASHome\SASFoundation\9.4\nls\en\SASCFG
      File Name= C:\Program Files\SASHome\SASFoundation\9.4\nls\en\SASCFG
        -Level 2-
      Engine=   V9
      Physical Name= C:\Program Files\SASHome\SASFoundation\9.4\core\sashelp
      File Name= C:\Program Files\SASHome\SASFoundation\9.4\core\sashelp
       . . .
      -Level 27-
      Engine=   V9
      Physical Name= C:\Program Files\SASHome\SASFoundation\9.4\webhound\sashelp
      File Name= C:\Program Files\SASHome\SASFoundation\9.4\webhound\sashelp
2    run;
```

## 関連項目:

- “LIBNAME Statement, SASEDOC” (*SAS Output Delivery System: User's Guide*)
- “LIBNAME Statement” (*SAS Statements: Reference*)

# SYSTASK ステートメント:Windows

非同期タスクを実行、リスト表示または終了します。

**該当要素:** SAS プログラム内の任意の場所

**Windows 固有:** すべて

## 構文

**SYSTASK COMMAND** "*operating system command*"<WAIT | NOWAIT>
<TASKNAME=*taskname*> <MNAME=*name-variable*>
<STATUS=*stat-variable*>
<SHELL<="*shell-command*"> > ;

**SYSTASK LIST** <_ALL_ | *taskname*> <STATE> <STATVAR>;

**SYSTASK KILL** *taskname* <*taskname*...>;

### *構文記述*

**COMMAND**
*オペレーティングシステムコマンド*を実行します。

**LIST**
特定のアクティブなタスクまたはシステム内のすべてのアクティブなタスクをリスト表示します。

**KILL**
指定されたタスクを強制終了します。

***operating system command***
Windows コマンド名とコマンド固有のオプションを指定します。コマンドは一重引用符または二重引用符で囲みます。コマンドオプションに引用符が必要な場合は、引用符を続けて入力します。例を次に示します。

```
systask command "find ""my text"" c:\mydir\myfile.sas"
```

ここに指定したオペテーティングシステムコマンドには、キーボードからの入力は使用できません。

**WAIT | NOWAIT**
タスクが完了するまで、SYSTASK COMMAND が現在の SAS セッションの実行を中断するかどうかを指定します。NOWAIT が、デフォルトの設定です。NOWAIT 引数を指定してタスクを起動した場合に、タスクが終了するまで SAS セッションの実行を中断するには、WAITFOR ステートメントを使用します。

**TASKNAME=*taskname***
タスクを識別する名前を指定します。タスク名は、すべてのアクティブなタスクで一意である必要があります。タスクは、実行中であるとアクティブとして認識されます。また、実行が完了し、WAITFOR ステートメントによる待機が存在しないタスクも、アクティブとして認識されます。タスク名が重複している場合は、SAS ログにエラーが出力されます。タスク名を指定しない場合は、SYSTASK が自動的に名前を生成します。タスク名に空白文字が含まれる場合は、タスク名を引用符で囲みます。

**MNAME=*name-variable***
SYSTASK が、自動的に生成したタスク名を格納するマクロ変数を指定します。TASKNAME オプションと MNAME オプションを両方とも指定した場合は、MNAME で指定した変数に TASKNAME で指定した名前がコピーされます。

**STATUS=*stat-variable***
SYSTASK がタスクのステータスを保存するマクロ変数を指定します。ステータス変数名は、すべてのアクティブなタスクで一意である必要があります。

**SHELL<="*shell-command*">**
*オペテーティングシステムコマンド*を、オペレーティングシステムのシェルコマンドと一緒に実行します。シェルコマンドを指定した場合、SYSTASK はそのシェルコマンドを使用してシェルを呼び出します。指定しないと、SYSTASK はデフォルトのシェルを使用します。シェルコマンドは、引用符で囲みます。

**_ALL_**
システム内のすべてのアクティブなタスクを指定します。

**STATE**
タスクのステータスを、起動失敗(Start Failed)、実行中(Running)または完了(Complete)の中から表示することを指定します。

**STATVAR**
タスクに関連するステータス変数を表示することを指定します。ステータス変数は、SYSTASK COMMAND ステートメントの STATUS オプションで割り当てた変数です。

## 詳細

SYSTASK を使用すると、SAS セッションまたは SAS アプリケーション内からオペレーティングシステム固有のコマンドを実行できます。X ステートメントとは異なり、SYSTASK はこれらのコマンドを非同期タスクとして実行します。つまり、現在実行されている他のタスクとは無関係に実行されます。非同期タスクはバックグラウンドで実行されるので、非同期タスクの実行中に別のタスクを実行できます。

たとえば、SAS プログラムをコピーするには、次のステートメントを使用できます。

```
systask command "copy myprog.sas myprog1.sas"
        taskname="copyfile" status=copystat;
```

**copy** コマンドからの戻り値は、マクロ変数の COPYSTAT に保存されます。

*注:* Windows コマンド出力は、SAS ログには書き込まれません。

SAS アプリケーション内で SYSTASK ステートメントに続くプログラムステップは、SYSTASK ステートメントの実行が成功すると実行されます。したがって、SYSTASK ステートメントに構文エラーがあると、SAS アプリケーションは停止します。

SYSTASK と一緒に実行できるタスクには、次の 2 つのタイプがあります。

タスク
: SYSTASK COMMAND で起動したタスクのタイプは、すべてタスクです。タスクタイプのタスクの場合、STATVAR または STATE を指定しないと、SYSTASK LIST はタスクの名前、タイプ、および状態に加えてステータスマクロ変数の名前を表示します。タスクタイプのタスクを停止するには、SYSTASK KILL を使用します。

SAS/CONNECT プロセス
: SAS/CONNECT から SIGNON ステートメント、SIGNON コマンド、RSUBMIT ステートメントで起動されるタスクは、SAS/CONNECT プロセスタイプです。SAS/Connect プロセスを表示するには、タスクの名前、タイプ、状態を表示する LISTTASK ステートメントを使用します。SAS/CONNECT プロセスを停止するには、KILLTASK ステートメントを使用します。SAS/CONNECT プロセスの詳細については、*SAS/CONNECT User's Guide* を参照してください。

  *注:* SAS/Connect プロセスに限らず、タスクを表示するには、SYSTASK LIST ではなく LISTTASK ステートメントを使用する方法をお勧めします。タスクを終了するには、SYSTASK KILL ではなく KILLTASK ステートメントを使用する方法をお勧めします。

SYSRC マクロ変数には、SYSTASK ステートメントの戻り値が格納されます。STATUS オプションで指定したステータス変数には、SYSTASK COMMAND で起動したプロセスの戻り値が格納されます。SYSTASK ステートメントのステータスと、SYSTASK ステートメントで起動されたプロセスのステータスの両方をモニタリングして、タスクの実行成功を確認するようにしてください。

SYSTASK ステートメントの実行が失敗した場合、SYSRC マクロ変数にはゼロでない値が格納されます。たとえば、タスクを完了するためのリソースが十分でない、または SYSTASK ステートメントに構文エラーがあるなどの原因が考えられます。SYSTASK KILL ステートメントを使用した場合にプロセスを終了できない場合、SYSRC はゼロでない値に設定されます。

タスクが起動されると、そのステータス変数は NULL に設定されます。各タスクのステータス変数を使用すれば、どのタスクが完了しなかったかを調べることができます。ステータス変数が NULL のタスクは、実行が完了していません。ステータス変数の詳細は、WAITFOR を参照してください。

X ステートメントとは異なり、SYSTASK ステートメントを使用して新しい対話型のセッションを開始することはできません。

## 関連項目:

- “WAITFOR ステートメント:Windows” (466 ページ)
- “X ステートメント:Windows” (467 ページ)

## TITLE ステートメント:Windows

SAS 出力のタイトル行を指定します。

**該当要素:** SAS プログラム内の任意の場所

**Windows 固有:** タイトルの最大長

**参照項目:** “TITLE Statement” (*SAS Statements: Reference*)

### 構文

**TITLE** <*n*> <'*テキスト*' | "*テキスト*">;

#### *オプション引数*

***n***
タイトル行を表示する行を指定します。

**'*テキスト*' | "*テキスト*"**
テキストを一重引用符または二重引用符で囲んで指定します。

### 詳細

TITLE ステートメントは、プロシジャ出力ファイルやその他の SAS 出力に表示するタイトル行を指定します。TITLE ステートメントは、TITLE ステートメントを割り当てた DATA ステップ、PROC ステップ、または RUN グループが実行されたときに、有効になります。行にタイトルを指定すると、そのタイトルを取り消すか、別のタイトルをその行に定義するまで、以後のすべての出力でそのタイトルが使用されます。

Windows 版では、タイトルの最大長は 256 文字です。指定したタイトルが LINESIZE システムオプションの値よりも大きい場合は、行サイズまで切り捨てられます。

## WAITFOR ステートメント:Windows

指定したタスクの実行が終了するまで、現在の SAS セッションを中断します。

**該当要素:** SAS プログラム内の任意の場所

**Windows 固有:** すべて

### 構文

**WAITFOR**<_ANY_ | _ALL_> *taskname* <*taskname*...> <TIMEOUT=*seconds*>;

#### *構文記述*

***taskname***
待機対象のタスクの名前を指定します。タスク名の詳細については、“SYSTASK ステートメント:Windows” (463 ページ)を参照してください。指定するタスク名は、SYSTASK COMMAND ステートメントで指定したタスク名と完全に一致する必要があります。ワイルドカードを使用してタスク名を指定することはできません。

**_ANY_ | _ALL_**
指定された 1 つまたはすべてのタスクの実行が完了するまで、現在の SAS セッションの実行を中断します。デフォルトは_ANY_です。この設定では、指定したタスクの 1 つの実行が完了すると、WAITFOR ステートメントの実行が終了します。

**TIMEOUT=*seconds***
WAITFOR が現在の SAS セッションを中断する最大秒数を指定します。TIMEOUT オプションを指定しないと、WAITFOR は SAS セッションの実行を無期限に中断します。

## 詳細

WAITFOR ステートメントは、指定したタスクの実行が終了するまでまたは TIMEOUT 間隔(指定した場合)が経過するまで、現在の SAS セッションの実行を中断します。指定したタスクが XWAIT オプションで起動されたタスクの場合、WAITFOR ステートメントはそのタスクを無視します。

たとえば、次のステートメントは、3 つの異なる SAS ジョブを起動し、3 つのジョブの実行が終了するまで、現在の SAS セッションの実行を中断します。

```
systask command "sas myprog1.sas" taskname=sas1;
systask command "sas myprog2.sas" taskname=sas2;
systask command "sas myprog3.sas" taskname=sas3;
waitfor _all_ sas1 sas2 sas3;
```

SYSRC マクロ変数には、WAITFOR ステートメントの戻り値が格納されます。WAITFOR ステートメントの実行が失敗した場合、SYSRC マクロ変数にはゼロでない値が格納されます。たとえば、WAITFOR ステートメントに構文エラーがあるなどの原因が考えられます。TIMEOUT オプションで指定した秒数が経過すると WAITFOR ステートメントは実行を終了します。この時、次のいずれかが当てはまる場合に、SYSRC にゼロでない値が設定されます。

- 1 つのタスクを指定したが実行が終了していない。
- 複数のタスクとデフォルト設定の_ANY_オプションを指定したが、どのタスクも実行が終了していない。
- 複数のタスクと_ALL_オプションを指定したが、実行が終了していないタスクがある。

WAITFOR ステートメントの実行後にステータス変数が NULL のタスクは、実行が完了していません。

## 関連項目:

- “SYSTASK ステートメント:Windows” (463 ページ)
- “X ステートメント:Windows” (467 ページ)
- “XWAIT システムオプション:Windows” (592 ページ)

# X ステートメント:Windows

SAS セッションからオペレーティングシステムのコマンドを発行するか、または Windows アプリケーションを実行します。

**該当要素:** SAS プログラム内の任意の場所

**Windows 固有:** 次に対する有効な値:*コマンド*

**参照項目:** “X Statement” (*SAS Statements: Reference*)

## 構文

**X** <'*command*'> ;

### *引数なし*

コマンドプロンプトセッションが表示され、オペレーティングシステムのプロンプトが表示されます。ここで、SAS 作業ディレクトリのコンテキストで、Windows コマンドを実行できます。SAS セッションで使用される環境変数の定義など、この状態のコマンドプロンプトから実行できない操作があります。たとえば、環境変数は、SAS を起動する前に定義する必要があります。SAS セッションに戻るには、コマンドプロンプトに EXIT と入力して Enter キーを押します。

### *オプション引数*

***command***

Windows コマンドまたは Windows アプリケーションを指定します。この引数には、SAS コマンドをはじめ、DOS プロンプトで使用できる任意のコマンドを指定できます。したがって、X ステートメントを使用して Windows アプリケーションを実行できます。コマンドは引用符で囲むことができますが、構文上必須ではありません。

コマンドは Windows に渡され、作業ディレクトリのコンテキストで実行されます。エラーが発生した場合は、エラーメッセージが表示されます。

デフォルトでは、コマンドの実行が完了した後で SAS セッションに戻るには、EXIT と入力します。またデフォルトでは、メモ帳など、Windows アプリケーションを実行した場合は、アプリケーションを終了しないと SAS セッションに戻れません。EXIT と入力しなくても SAS セッションに戻れるようにする場合は、OPTIONS ステートメントで NOXWAIT を指定します。NOXWAIT が有効な場合、コマンドの実行が終了すると直ちに SAS セッションに制御が戻ります。ただし、X ステートメントを使用して Windows アプリケーションを実行する場合は、NOXWAIT を指定しても、そのアプリケーションを閉じないと SAS セッションに戻れません。

システムオプションの XSYNC を使用すると、SAS セッションに戻る前に、コマンドの実行が終了するまで待つかどうかを制御できます。NOXSYNC を指定すると、X ステートメントで Windows アプリケーションを開始し、そのアプリケーションを閉じなくても SAS セッションに戻れます。これら 2 つのシステムオプションの詳細については、“XWAIT システムオプション:Windows” (592 ページ)および“XSYNC システムオプション:Windows” (591 ページ)を参照してください。

## 詳細

X ステートメントは、ウィンドウモードで SAS を実行しているときに、SAS セッションの内部からホストコマンドを発行します。SAS は X ステートメントを即座に実行します。

Windows 環境では、*コマンド*引数なしの X ステートメントを発行できます。

Windows 版で動作環境のコマンドを実行する方法は、X ステートメント(および X コマンド)のほかにもあります。

## 関連項目:

- “X コマンド:Windows” (374 ページ)
- “XSYNC システムオプション:Windows” (591 ページ)
- “XWAIT システムオプション:Windows” (592 ページ)
- “CALL SYSTEM ルーチン:Windows” (391 ページ)

- SYSEXEC ステートメント“マクロステートメント” (601 ページ)
- “SAS から Windows または MS-DOS のコマンドを実行する ” (28 ページ)
- “ツールメニューにアプリケーションの追加” (66 ページ)

*22 章*

# Windows 版に固有の SAS システムオプション

**Windows 版に固有の SAS システムオプション** . . . . . **473**
Windows 環境下での SAS システムオプションの概要 . . . . . 473
制限オプション . . . . . 474

**SAS システムオプション設定の表示** . . . . . **474**

**SAS システムオプション設定の変更** . . . . . **475**
概要:SAS システムオプション設定の変更 . . . . . 475
SAS 起動または SAS 構成ファイルのシステムオプションの構文 . . . . . 475
SAS システムオプションでライブラリを連結する構文 . . . . . 476
OPTIONS ステートメントのシステムオプションの構文 . . . . . 477

**複数の場所に設定されたシステムオプションの処理** . . . . . **477**

**カテゴリごとの SAS システムオプション** . . . . . **477**

**ディクショナリ** . . . . . **486**
ACCESSIBILITY システムオプション:Windows . . . . . 486
ALIGNSASIOFILES システムオプション:Windows . . . . . 486
ALTLOG システムオプション:Windows . . . . . 487
ALTPRINT システムオプション:Windows . . . . . 488
APPEND システムオプション:Windows . . . . . 489
AUTHPROVIDERDOMAIN システムオプション:Windows . . . . . 490
AUTHSERVER システムオプション:Windows . . . . . 490
AUTOEXEC システムオプション:Windows . . . . . 491
AWSCONTROL システムオプション:Windows . . . . . 492
AWSDEF システムオプション:Windows . . . . . 493
AWSMENU システムオプション:Windows . . . . . 494
AWSMENUMERGE システムオプション:Windows . . . . . 495
AWSTITLE システムオプション:Windows . . . . . 495
BUFNO システムオプション:Windows . . . . . 496
BUFSIZE システムオプション:Windows . . . . . 497
CATCACHE システムオプション:Windows . . . . . 498
CLEANUP システムオプション:Windows . . . . . 499
COMDEF システムオプション:Windows . . . . . 500
CONFIG システムオプション Windows . . . . . 501
DEVICE システムオプション:Windows . . . . . 501
ECHO システムオプション:Windows . . . . . 502
EMAILDLG システムオプション:Windows . . . . . 503
EMAILSYS システムオプション:Windows . . . . . 504
ENGINE システムオプション:Windows . . . . . 505
ENHANCEDEDITOR システムオプション:Windows . . . . . 506
FILELOCKWAIT システムオプション:Windows . . . . . 506
FILELOCKWAITMAX= システムオプション:Windows . . . . . 507

FILTERLIST システムオプション:Windows . . . . . 508
FONT システムオプション:Windows . . . . . 509
FONTALIAS システムオプション:Windows . . . . . 510
FONTSLOC システムオプション:Windows . . . . . 511
FORMCHAR システムオプション:Windows . . . . . 512
FULLSTIMER システムオプション:Windows . . . . . 513
HELPHOST システムオプション:Windows . . . . . 515
HELPINDEX システムオプション:Windows . . . . . 515
HELPLOC システムオプション:Windows . . . . . 516
HELPREGISTER システムオプション:Windows . . . . . 517
HELPTOC システムオプション:Windows . . . . . 519
HOSTINFOLONG システムオプション:Windows . . . . . 520
HOSTPRINT システムオプション:Windows . . . . . 521
ICON システムオプション:Windows . . . . . 521
INITSTMT システムオプション:Windows . . . . . 522
INSERT システムオプション:Windows . . . . . 523
JREOPTIONS システムオプション:Windows . . . . . 524
LINESIZE システムオプション:Windows . . . . . 525
LOADMEMSIZE システムオプション:Windows . . . . . 526
LOG システムオプション:Windows . . . . . 526
MAPS システムオプション:Windows . . . . . 528
MAXMEMQUERY システムオプション:Windows . . . . . 529
MEMBLKSZ システムオプション:Windows . . . . . 529
MEMCACHE システムオプション:Windows . . . . . 531
MEMLIB システムオプション:Windows . . . . . 532
MEMMAXSZ システムオプション:Windows . . . . . 532
MEMSIZE システムオプション:Windows . . . . . 533
MSG システムオプション:Windows . . . . . 534
MSGCASE システムオプション:Windows . . . . . 535
MSYMTABMAX システムオプション:Windows . . . . . 536
MVARSIZE システムオプション:Windows . . . . . 537
NEWS システムオプション:Windows . . . . . 538
NUMKEYS システムオプション:Windows . . . . . 538
NUMMOUSEKEYS システムオプション:Windows . . . . . 539
OBS システムオプション:Windows . . . . . 540
OPLIST システムオプション:Windows . . . . . 540
PAGENO システムオプション:Windows . . . . . 541
PAGESIZE システムオプション:Windows . . . . . 542
PAPERTYPE システムオプション:Windows . . . . . 543
PATH システムオプション:Windows . . . . . 543
PFKEY システムオプション:Windows . . . . . 544
PRIMARYPROVIDERDOMAIN=システムオプション:Windows . . . . . 547
PRINT システムオプション:Windows . . . . . 547
PRNGETLIST システムオプション:Windows . . . . . 548
PRTABORTDLGS システムオプション:Windows . . . . . 549
PRTPERSISTDEFAULT システムオプション]Windows . . . . . 549
PRTSETFORMS システムオプション:Windows . . . . . 550
REALMEMSIZE システムオプション Windows . . . . . 551
REGISTER システムオプション:Windows . . . . . 552
RESOURCESLOC システムオプション:Windows . . . . . 553
RSASUSER システムオプション:Windows . . . . . 553
RTRACE システムオプション:Windows . . . . . 554
RTRACELOC システムオプション:Windows . . . . . 555
SASAUTOS システムオプション:Windows . . . . . 556
SASCONTROL システムオプション:Windows . . . . . 557
SASHELP システムオプション:Windows . . . . . 558

SASINITIALFOLDER システムオプション:Windows . . . . . 558
SASUSER システムオプション n:Windows . . . . . 559
SCROLLBARFLASH システムオプション:Windows . . . . . 560
SET システムオプション:Windows . . . . . 561
SGIO システムオプション:Windows . . . . . 562
SLEEPWINDOW システムオプション:Windows . . . . . 563
SORTANOM システムオプション:Windows . . . . . 563
SORTCUT システムオプション:Windows . . . . . 564
SORTCUTP システムオプション:Windows . . . . . 565
SORTDEV システムオプション:Windows . . . . . 566
SORTNAME System Option:Windows . . . . . 567
SORTPARM システムオプション:Windows . . . . . 567
SORTPGM システムオプション:Windows . . . . . 568
SORTSIZE システムオプション:Windows . . . . . 569
SPLASH システムオプション:Windows . . . . . 570
SPLASHLOC システムオプション:Windows . . . . . 571
STIMEFMT システムオプション:Windows . . . . . 571
STIMER システムオプション Windows . . . . . 574
SYSGUIFONT システムオプション:Windows . . . . . 576
SYSIN システムオプション:Windows . . . . . 577
SYSPARM システムオプション:Windows . . . . . 577
SYSPRINT システムオプション:Windows . . . . . 578
SYSPRINTFONT システムオプション:Windows . . . . . 579
TOOLDEF システムオプション:Windows . . . . . 581
TOOLSMENU システムオプション . . . . . 582
EMAILDLG システムオプション . . . . . 583
UPRINTMENUSWITCH システムオプション:Windows . . . . . 584
USER システムオプション:Windows . . . . . 585
USERICON システムオプション:Windows . . . . . 585
VERBOSE システムオプション:Windows . . . . . 586
VIEWMENU システムオプション . . . . . 587
WEBUI システムオプション:Windows . . . . . 587
WINDOWSMENU システムオプション:Windows . . . . . 588
WORK システムオプション:Windows . . . . . 589
XCMD システムオプション Windows . . . . . 590
XMIN システムオプション:Windows . . . . . 591
XSYNC システムオプション:Windows . . . . . 591
XWAIT システムオプション:Windows . . . . . 592

# Windows 版に固有の SAS システムオプション

## *Windows 環境下での SAS システムオプションの概要*

SAS システムオプションは、出力先、プログラムの実行効率、SAS ファイルおよびデータライブラリの属性など、SAS セッションの多くの要素を制御するために使用します。システムオプションは、SAS コマンド、SAS 構成ファイル、PROC OPTLOAD または DMOPTLOAD コマンドの 使用(PROC OPTLOAD と DMOPTLOAD コマンドは、SAS データセットに前回保存されたオプション設定をロード)、OPTIONS ステートメント(SAS プログラムまたは SAS 自動実行ファイル)、**システムオプション**ウィンドウ、または SCL プログラムで指定できます。

データを構成ファイルに保存する場合、デフォルトの場所は **C:\Program Files \SASHome2\SASFoundation\9.4\nls\language\sasv9.cfg** です。language は使用ロケールの言語で、たとえば英語の場合は en です。

設定したシステムオプションは、再設定しない限り、プログラムや SAS セッション中のすべての DATA ステップおよび PROC ステップに影響します。たとえば、CENTER システムオプションは、プログラムのステップ数に関係なく、設定した後で発生するすべての出力に影響します。

SAS システムオプションの中には、データセットオプションやステートメントオプションと同じ効果(通常は同じオプション名)を持つものがあります。たとえば、BUFSIZE システムオプションは、BUFSIZE=データセットオプションに類似しています。オプションが重複して指定された場合、次の優先規則が適用されます。

- データセットオプション値(最優先)
- ステートメントオプション値(システムオプションより優先)
- システムオプション値(最も低い優先度)

### *制限オプション*

制限オプションはサイト管理者で決定された値を持ち、オーバーライドできないシステムオプションです。サイト管理者は制限オプション表を作成して、SAS が起動したときに制限オプション値を指定することができます。制限オプション表にリストアップされたシステムオプションを変更しようとすると、システムオプションはサイト管理者のみに限定されていて変更できないというメッセージを SAS ログに記録します。制限オプションの詳細については、“Restricted Options” (*SAS System Options: Reference* 1 章)を参照してください。

グローバル制限オプションの指定は、**!SASROOT\rstropts\rsasv9.cfg** パスにあるファイルに含まれます。ユーザー制限オプションの指定は、**!SASROOT \rstropts\userid_rsasv9.cfg** パスにあるファイルに含まれます。userid はユーザーのシステム ID(jomaxw)です。

- COMDEF
- FILELOCKWAITMAX
- MEMCACHE
- MEMLIB
- PATH
- RESOURCESLOC
- SASCONTROL
- SGIO
- TOOLDEF
- XCMD

## SAS システムオプション設定の表示

SAS システムオプションはデフォルト値に設定されます。SAS システムオプションの設定を SAS ログに表示するには、OPTIONS プロシジャを使用します。たとえば、次のス

テートメントによって、1 行に 1 つのオプションを表示するオプションのリストが表示されます。リストには、それぞれのオプションの簡単な説明も表示されます。

```
proc options; run;
```

PROC OPTIONS ステートメントに SHORT オプションステートメントを指定すると、オプション設定のリストから、各オプションの説明を省略できます。詳細については、*Base SAS プロシジャガイド*の OPTIONS プロシジャを参照してください。

対話型の SAS セッションでは、**システムオプション**ウィンドウにて、起動と設定オプションを含む多くの SAS システムオプションの設定を表示します。**オプション** ウィンドウではこれらのオプションを変更することはできません。**システムオプション**ウィンドウを開くには、**ツール** ⇨ **オプション** ⇨ **システム**の順に選択するか、コマンドエリアで **options** と入力し、コマンドをサブミットします。

# SAS システムオプション設定の変更

## *概要:SAS システムオプション設定の変更*

SAS システムオプションではいくつかの方法で値を指定することができます。

- SAS を呼び出すコマンドの一部として指定
- SAS の初期化時に処理される SAS 構成ファイルの一部として指定
- SAS の初期化時に処理される Windows 環境変数(SAS_OPTIONS)で指定
- SAS データセットに前回保存されたオプション値セットをロードする PROC OPTLOAD または DMOPTLOAD コマンドを使用して指定
- SAS セッション中に実行する OPTIONS ステートメントの一部として指定
- 対話型の **SAS システムオプション**ウィンドウを使用して指定
- SCL プログラムまたは SAS/AF プログラム内で、OPTSETC および OPTSETN SCL 関数を使用して指定

システムオプションには、SAS セッションまたは SAS プロセスの初期化時(起動時)にのみ指定できるものと、SAS セッション中に必要に応じて変更できるものがあります。

なお、SAS を起動する SAS コマンドにシステムオプションを指定したり、SAS 構成ファイルにシステムオプションを指定したりする場合と、OPTIONS ステートメントにシステムオプションを指定する場合では、構文が異なるため注意が必要です。構文は状況に応じて異なり、間違えて使用するとエラーメッセージが表示されます。OPTIONS ステートメントの詳細については、*SAS ステートメント: リファレンス*を参照してください。

BUFNO オプションと MEMSIZE オプションは構成ファイルで設定できます。MEMSIZE を SAS の起動後に設定することはできません。

```
-BUFNO 10
-MEMSIZE 4G
```

BUFNO オプションは、OPTION ステートメントを使用して設定することもできます。

```
OPTIONS BUFNO=10;
```

## *SAS 起動または SAS 構成ファイルのシステムオプションの構文*

初期化するときにシステムオプションを指定するときはハイフン(-)を先に付けてください。オン/オフを切り替えるオプションでは、目的の設定に対応するキーワードを指定し

ます。たとえば、次のコマンドでは、SAS を起動し、SAS 出力を中央揃えにしないことが指定されます。

```
c:\sas\sas.exe -nocenter
```

値を指定するオプションでは、等号(=)は使用しません。オプション名の後にスペースを 1 個入れ、その後に値を指定します。たとえば、次の SAS コマンドでは、SAS を起動し、1 行の長さを 132 にすることが指定されます。

```
c:\sas\sas.exe -linesize 132
```

SAS コマンドや SAS 構成ファイルに物理名(ディレクトリ名やファイル名)を指定するには、二重引用符で囲む必要があります。特に、Windows ファイル名に有効な、スペースや一重引用符が入ったファイル名またはパス名を指定するときは、引用符を忘れずに付けてください。たとえば、次の OPTIONS ステートメントでは、自動呼び出しマクロが **C:\SAS\CORE\SASMACRO** ディレクトリに格納されていることが示されます。

```
c:\sas\sas.exe -sasautos "c:\sas\core\sasmacro"
```

次の例のように、オプションの値に等号(=)が入っているときも、値を二重引用符で囲む必要があります。

```
c:\sas\sas.exe -set fruit "navel=orange"
```

SAS コマンドに複数のオプションを指定するには、オプションの間にスペースを 1 つ挿入します。たとえば、次の SAS コマンドでは、これまでに使用した 3 つのオプションすべてが指定されます。

```
c:\sas\sas.exe -linesize 132 -nocenter
               -sasautos "c:\sas\core\sasmacro"
```

SAS 構成ファイルには、オプション設定だけを指定します。SAS ステートメントは指定できません。ファイルには、SAS コメントを含めることができます。たとえば、MySASConfig.CFG という名前の構成ファイルに次のようにオプションを指定できます。

```
-nocenter
-noxwait
-pagesize 60
```

SAS 構成ファイルには、すべての SAS システムオプションを使用できます。SAS 構成ファイルの詳細については、“SAS 構成ファイル ” (15 ページ) を参照してください。

## *SAS システムオプションでライブラリを連結する構文*

ハードディスクやネットワーク上の複数の論理ドライブなどのように、異なったドライブにまたがって SAS ファイルを保管するためにより多くの柔軟性を持たせるために、SAS は SAS ライブラリを連結することを可能にします。SAS においては、連結とは、SAS 構成ファイルまたは SAS コマンドに特定のシステムオプションを指定するときに、複数のドライブやディレクトリを指定することを意味します。連結されたディレクトリを指定するには、各ディレクトリ名を二重引用符で囲み、それぞれをスペースで区切り、全体をかっこで囲みます。

連結ライブラリの有効な使用方法の 1 つに、SAS ヘルプカタログの格納があります。SAS プロダクトを複数のディレクトリに分割する場合、SAS 構成ファイルの SASHELP オプションに、次の例のように複数のディレクトリを指定します。

```
-sashelp ("c:\sas\core\sashelp"
          "d:\sas\stat\sashelp")
```

### *OPTIONS ステートメントのシステムオプションの構文*

同じ SAS セッション内であれば、いつでも OPTIONS ステートメントにて多数の SAS システムオプションを指定することが可能です。オプションの設定は、SAS セッションの間、別の OPTIONS ステートメントでオプションを変更するまで、または PROC OPTLOAD または DMOPTLOAD コマンドを使用して前回保存されたオプション値セットをロードするまで有効です。OPTIONS ステートメントの詳細については、*SAS ステートメント: リファレンス*を参照してください。

OPTIONS ステートメントにシステムオプションを指定するときは、システムオプション名の前にハイフン(-)は付けません。また、値を指定するシステムオプションでは、スペースではなく、等号(=)を使用して値を指定します。たとえば、次のステートメントでは、出力に日付のラベルを付けず、行サイズを 132 にすることを指定します。

```
options nodate linesize=132;
```

OPTIONS ステートメントに物理名(ディレクトリ名やファイル名)を指定するときは、名前を引用符で囲む必要があります。たとえば、次の OPTIONS ステートメントでは、自動呼び出しマクロが C:\SAS\CORE\SASMACRO ディレクトリに格納されていることが示されます。

```
options sasautos="c:\sas\core\sasmacro";
```

OPTIONS ステートメントで引用符を使用せずにファイル名を指定すると、そのファイル名は、論理名、つまりファイル参照名または環境変数名と見なされます。この論理名が見つからない場合は、エラーメッセージが表示されます。

システムオプションの中には、OPTIONS ステートメントでは指定できないものもあります。システムオプションが OPTIONS ステートメントで指定できるかどうかを見破るには、PROC OPTIONS を使用します。

```
option=optname define;
run;
```

## 複数の場所に設定されたシステムオプションの処理

同じシステムオプションが複数の場所に設定されている場合、最も最新の指定が使われます。したがって、**SAS システムオプション**ウィンドウまたは OPTIONS ステートメントは、SAS autoexec ファイルの指定よりも優先されます。また、SAS autoexec ファイルは、SAS コマンドよりも優先されます。SAS コマンドは、SAS 構成ファイルと環境変数の設定値よりも優先されます。

## カテゴリごとの SAS システムオプション

SAS システムオプションのカテゴリは、SAS システムオプションとサブグループに対応します:

| | |
|---|---|
| 通信:電子メール | SAS を使用した電子メールの送受信に関連付けられたオプション |
| 通信:ネットワークと暗号化 | リモート通信、共有設定、および暗号化に関連するオプション |

| | |
|---|---|
| 通信:メタデータ | メタデータを使用するために SAS を設定するオプション |
| 環境制御:表示 | SAS ウィンドウと表示プリファレンスを設定するオプション |
| 環境制御:エラー処理 | エラー条件とエラーメッセージに関連付けられたオプション |
| 環境制御:ファイル | SAS ライブラリとファイルの場所のプリファレンスを設定するオプション |
| 環境制御:ヘルプ | SAS ヘルプの設定に使用するオプション |
| 環境制御:初期化と操作 | SAS 動作環境を確立するオプション |
| 環境制御:言語制御 | 言語と変換プリファレンスを設定するオプション |
| ファイル:外部ファイル | SAS によって作成されないファイルを処理する方法を定義するオプション |
| ファイル:SAS ファイル | SAS ファイルを処理する方法を定義するオプション |
| 入力制御:データ処理 | データ入力とデータ処理プリファレンスに関するオプション |
| 入力制御:データ品質 | SAS データ品質サーバーを設定するオプション |
| グラフィックス:ドライバ設定 | デバイス、グラフィックス、およびマッププリファレンスを定義するオプション |
| ログとプロシジャ出力制御:SAS ログ | SAS ログに書き込まれるメッセージの表示を制御するオプション |
| ログとプロシジャ出力制御:プロシジャ出力 | プロシジャ出力と表示プリファレンスを定義するオプション |
| ログとプロシジャ出力制御:SAS ログとプロシジャ出力 | SAS ログとプロシジャ出力プリファレンスを制御するオプション |
| ログとプロシジャ出力制御:ODS 出力 | ODS 出力先への出力プリファレンスを定義するオプション |
| ログとプロシジャ出力制御:PDF | PDF ファイルのプリファレンスを定義するオプション |
| ログとプロシジャ出力制御:SVG | SVG ファイルのプリファレンスを定義するオプション |
| ログとプロシジャ出力制御:アニメーション | アニメーション SVG ファイルのプリファレンスを定義するオプション |
| マクロ:SAS マクロ | SAS マクロプリファレンスを定義するオプション |
| ソート:プロシジャオプション | SAS ファイルのソートに関するプリファレンスを定義するオプション |
| システム管理:インストール | サイトインストール設定を定義するためのオプション |
| システム管理:メモリ | コンピュータメモリプリファレンスを定義するオプション |
| システム管理:パフォーマンス | パフォーマンスプリファレンスを定義するオプション |

| | |
|---|---|
| システム管理:コード生成 | SAS 言語ステートメントの生成に関するプリファレンスを定義するオプション |
| システム管理:セキュリティ | セキュリティ設定を定義するためのオプション |
| システム管理:SQL | SQL プロシジャ y の設定を定義するオプション |

| カテゴリ | 言語要素 | 説明 |
|---|---|---|
| SAS ファイル | FILELOCKWAIT システムオプション:Windows (p. 506) | ロックされたファイルが使用可能になるまで待機する秒数を設定します。 |
| 環境制御:エラー処理 | CLEANUP システムオプション:Windows (p. 499) | リソース不足状態の対処方法を指定します。 |
| 環境制御:初期化と操作 | AUTHPROVIDERDOMAIN システムオプション:Windows (p. 490) | ドメインサフィックスを認証プロバイダに割り当てます。 |
| | AUTHSERVER システムオプション:Windows (p. 490) | セキュリティで保護されたサーバーログインを検索するドメイン認証サーバーを指定します。 |
| | INITSTMT システムオプション:Windows (p. 522) | autoexec ファイルのステートメントの実行後、また SYSIN=ファイルのステートメントの前に実行する SAS ステートメントを指定します。 |
| | JREOPTIONS システムオプション:Windows (p. 524) | SAS で使用する Java Runtime Environment (JRE)オプションを特定します。 |
| | PRIMARYPROVIDERDOMAIN=システムオプション:Windows (p. 547) | 基本の認証プロバイダのドメイン名を 指定します。 |
| 環境制御:表示 | AWSCONTROL システムオプション:Windows (p. 492) | メイン SAS ウィンドウに、タイトルバー、システム/コントロールメニュー、最大化/最小化ボタンを含めるかどうかを指定します。 |
| | AWSDEF システムオプション:Windows (p. 493) | SAS の起動時の、メイン SAS ウィンドウの位置および大きさを指定します。 |
| | AWSMENU システムオプション:Windows (p. 494) | メイン SAS ウィンドウに、メニューバーを表示するかどうかを指定します。 |
| | AWSMENUMERGE システムオプション:Windows (p. 495) | メインメニューに、Windows 固有のメニュー項目を組み込むかどうかを指定します。 |
| | AWSTITLE システムオプション:Windows (p. 495) | メイン SAS タイトルバーの、デフォルトのテキストを置き換えます。 |
| | COMDEF システムオプション:Windows (p. 500) | SAS コマンドウィンドウを表示する場所を指定します。 |
| | ENHANCEDEDITOR システムオプション:Windows (p. 506) | SAS 起動時に拡張エディタを使用可能にするかどうかを指定します。 |
| | FILTERLIST システムオプション:Windows (p. 508) | 開くと名前を付けて保存ダイアログボックスで使用する、ファイルフィルタの代替セットを指定します。 |

| カテゴリ | 言語要素 | 説明 |
|---|---|---|
| | FONT システムオプション:Windows (p. 509) | SAS ウィンドウで使用するフォントを指定します。 |
| | FONTSLOC システムオプション:Windows (p. 511) | SAS セッション時にロードされる SAS フォントの保存場所を指定します。 |
| | ICON システムオプション:Windows (p. 521) | SAS ウィンドウを最小化します。 |
| | PRTABORTDLGS システムオプション:Windows (p. 549) | 印刷中止ダイアログボックスをいつ表示するかを指定します。 |
| | PRTSETFORMS システムオプション:Windows (p. 550) | フォームを使用するチェックボックスを印刷設定ダイアログボックスに含めるかどうかを指定します。 |
| | REGISTER システムオプション:Windows (p. 552) | メイン SAS ウィンドウのツールメニューにアプリケーションを追加します。 |
| | RESOURCESLOC システムオプション:Windows (p. 553) | SAS のリソースが格納されるファイルのディレクトリを指定します。 |
| | SASCONTROL システムオプション:Windows (p. 557) | SAS アプリケーションウィンドウにてシステムと設定メニュそして最小化と最大化のボタンを含むかどうかを指定します。 |
| | SCROLLBARFLASH システムオプション:Windows (p. 560) | マウスまたはキーボードを使用してスクロールバーにフォーカスできるかどうかを指定します。 |
| | SLEEPWINDOW システムオプション:Windows (p. 563) | SLEEP ウィンドウの表示と非表示を切り替えます。 |
| | SPLASH システムオプション:Windows (p. 570) | SAS の起動時にスプラッシュ画面(ロゴ画面)を表示するかどうかを指定します。 |
| | SPLASHLOC システムオプション:Windows (p. 571) | SAS の起動時に表示されるスプラッシュ画面のビットマップの保存場所を指定します。 |
| | SYSGUIFONT システムオプション:Windows (p. 576) | ボタンテキストとその説明テキストに使用するフォントを指定します。 |
| | TOOLDEF システムオプション:Windows (p. 581) | ツールボックスの表示位置を指定します。 |
| | TOOLSMENU システムオプション (p. 582) | SAS ウィンドウにツールメニューが含まれるかどうかを指定します。 |
| | USERICON システムオプション:Windows (p. 585) | ユーザー定義のアイコンに対応したリソースファイルのパス名を指定します。 |
| | VIEWMENU システムオプション (p. 587) | SAS ウィンドウに表示メニューが含まれるかどうかを指定します。 |
| | WINDOWSMENU システムオプション:Windows (p. 588) | メニューを表示するウィンドウに、ウィンドウメニューを表示するかどうかを指定します。 |

| カテゴリ | 言語要素 | 説明 |
|---|---|---|
| | XCMD システムオプション Windows (p. 590) | 現在の SAS セッションで X コマンドが有効になるようにします。 |
| | XMIN システムオプション:Windows (p. 591) | X コマンドに指定したアプリケーションを最小化したウィンドウの状態で開くか、またはアクティブウィンドウの状態(デフォルト)で開くかを指定します。 |
| | XSYNC システムオプション:Windows (p. 591) | X コマンドまたは X ステートメントの実行を同期とするか非同期とするかを制御します。 |
| | XWAIT システムオプション:Windows (p. 592) | DOS シェルを閉じる前に、DOS プロンプトに対して EXIT を入力するかどうかを指定します。 |
| 環境制御:ファイル | ALTLOG システムオプション:Windows (p. 487) | SAS ログのコピー先を指定します。 |
| | ALTPRINT システムオプション:Windows (p. 488) | SAS プロシジャ出力ファイルのコピー先を指定します。 |
| | AUTOEXEC システムオプション:Windows (p. 491) | SAS autoexec ファイルを指定します。 |
| | LOG システムオプション:Windows (p. 526) | SAS をバッチモードで実行している場合の SAS ログのコピー先を指定します。 |
| | MSG システムオプション:Windows (p. 534) | SAS エラーメッセージが格納されているライブラリを指定します。 |
| | NEWS システムオプション:Windows (p. 538) | SAS ログに書き込まれるメッセージが保存されているファイルを指定します。 |
| | PRINT システムオプション:Windows (p. 547) | バッチモードで実行している際の SAS 出力先を指定します。 |
| | RSASUSER システムオプション:Windows (p. 553) | SASUSER データライブラリのメンバを更新モードで開くか、読み込み専用モードで開くかを制御します。 |
| | RTRACELOC システムオプション:Windows (p. 555) | SAS のひとつのセッションでリードまたはロードするリソースのリストを書き込むファイルのパス名を指定します。 |
| | SASHELP システムオプション:Windows (p. 558) | SASHELP カタログ内のエントリ(SAS デフォルトフォーム、デバイスリスト、ディクショナリなど)を検索するディレクトリを指定します。 |
| | SASINITIALFOLDER システムオプション:Windows (p. 558) | SAS の初期化が完了した後、ファイルを開くおよび名前を付けて保存ダイアログボックスの作業フォルダおよびデフォルトフォルダを、指定したフォルダに変更します。 |
| | SASUSER システムオプション:Windows (p. 559) | SASUSER ライブラリの名前を指定します。 |
| | SET システムオプション:Windows (p. 561) | SAS 環境変数を定義します。 |

| カテゴリ | 言語要素 | 説明 |
|---|---|---|
| | SYSIN システムオプション:Windows (p. 577) | バッチモードのソースファイルを指定します。 |
| | SYSPARM システムオプション:Windows (p. 577) | SAS プログラムに渡される文字列を指定します。 |
| | USER システムオプション:Windows (p. 585) | デフォルトの永久 SAS データライブラリ名を指定します。 |
| | WORK システムオプション:Windows (p. 589) | Work ライブラリの場所を指定します。 |
| 環境制御:ファイル;マクロ:SAS マクロ | SASAUTOS システムオプション:Windows (p. 556) | 自動呼び出しマクロライブラリを指定します。 |
| 環境制御:ヘルプ | HELPINDEX システムオプション:Windows (p. 515) | SAS ヘルプとドキュメントに対する 1 つ以上のインデックスファイルを指定します。 |
| | HELPLOC システムオプション:Windows (p. 516) | Microsoft HTML Help を使用して SAS ヘルプとドキュメントを表示する場合に使用されるヘルプファイルの場所を指定します。 |
| | HELPREGISTER システムオプション:Windows (p. 517) | メイン SAS ウィンドウのヘルプメニューからアクセスするヘルプファイルを登録します。 |
| | HELPTOC システムオプション:Windows (p. 519) | SAS ヘルプとドキュメントで使用される目次ファイルを指定します。 |
| グラフィックス:ドライバ設定 | DEVICE システムオプション:Windows (p. 501) | SAS/GRAPH ソフトウェアのグラフィックス出力デバイスドライバを指定します。 |
| | FONTALIAS システムオプション:Windows (p. 510) | 1 つの SAS フォントに対して、1 つの Windows フォントを割り当てます。 |
| | MAPS システムオプション:Windows (p. 528) | SAS/GRAPH マップデータセットが保存されている SAS ライブラリの名前を指定します。 |
| システム管理:インストール | CONFIG システムオプション Windows (p. 501) | SAS システムオプションの値を初期化または上書きする際に使用する構成ファイルを指定します。 |
| | PATH システムオプション:Windows (p. 543) | SAS 実行ファイルを検索するパスを 1 つ以上指定します。 |
| システム管理:メモリ | LOADMEMSIZE システムオプション:Windows (p. 526) | SAS がロードする実行プログラムに必要な推奨メモリ量を指定します。 |
| | MAXMEMQUERY システムオプション:Windows (p. 529) | プロシジャに割り当てる最大メモリ量を指定します。 |
| | MEMBLKSZ システムオプション:Windows (p. 529) | Windows 動作環境で、メモリベースのライブラリに使用されるメモリのブロックサイズを指定します。 |
| | MEMMAXSZ システムオプション:Windows (p. 532) | Windows 動作環境で、メモリベースのライブラリの使用に割り当てる最大メモリ量を指定します。 |

| カテゴリ | 言語要素 | 説明 |
|---|---|---|
| | MEMSIZE システムオプション:Windows (p. 533) | SAS セッション時に使用できる仮想メモリ量の上限を指定します。 |
| | REALMEMSIZE システムオプション Windows (p. 551) | 割り当てることができる実メモリの量を指定します。 |
| | SORTSIZE システムオプション:Windows (p. 569) | SORT プロシジャで使用できるメモリ量を指定します。 |
| ソート:プロシジャオプション | SORTANOM システムオプション:Windows (p. 563) | ホストソートユーティリティのオプションを指定します。 |
| | SORTCUT システムオプション:Windows (p. 564) | このデータサイズを上回ると内部 SAS ソートではなくホストソートを使用する上限をオブザベーション数として指定します。 |
| | SORTCUTP システムオプション:Windows (p. 565) | このデータサイズを上回ると内部 SAS ソートではなくホストソートを使用する上限をバイト数で指定します。 |
| | SORTDEV システムオプション:Windows (p. 566) | ホストソートユーティリティによって作成される一時ファイルのパス名を指定します。 |
| | SORTPARM システムオプション:Windows (p. 567) | ホストソートユーティリティのパラメータを指定します。 |
| | SORTPGM システムオプション:Windows (p. 568) | SAS ソートユーティリティとホストソートユーティリティのいずれを使用するかを指定します。また、使用するソートユーティリティを SAS に選択させることもできます。 |
| | SORTSIZE システムオプション:Windows (p. 569) | SORT プロシジャで使用できるメモリ量を指定します。 |
| 通信:電子メール | EMAILDLG システムオプション:Windows (p. 503) | ユーザーが使用している電子メールアプリケーションのダイアログボックスか、または SAS が提供する電子メールのダイアログボックスを使用するかを指定します。 |
| | EMAILSYS システムオプション:Windows (p. 504) | 電子メールの送信に使用する電子メールプロトコルを指定します。 |
| 入力制御:データ処理 | ACCESSIBILITY システムオプション:Windows (p. 486) | ツールのユーザー設定ダイアログボックスでユーザー補助機能を使用できるようにします。 |
| | ALIGNSASIOFILES システムオプション:Windows (p. 486) | SAS データセットでデータのページを調整します。 |
| | NUMKEYS システムオプション:Windows (p. 538) | 使用可能なファンクションキーの数を制御します。 |
| | NUMMOUSEKEYS システムオプション:Windows (p. 539) | KEYS ウィンドウに表示されるマウスボタンの数を指定します。 |
| | PFKEY システムオプション:Windows (p. 544) | ファンクションキーの基本セットを指定します。 |

| カテゴリ | 言語要素 | 説明 |
|---|---|---|
| | WEBUI システムオプション:Windows (p. 587) | Web 拡張機能を有効にします。 |
| ファイル:SAS ファイル | BUFNO システムオプション:Windows (p. 496) | SAS データセットの処理に割り当てるバッファ数を指定します。 |
| | BUFSIZE システムオプション:Windows (p. 497) | 出力 SAS データセットの永久的なバッファページサイズを指定します。 |
| | CATCACHE システムオプション:Windows (p. 498) | 開いたままにしておける SAS カタログの数を指定します。 |
| | ENGINE システムオプション:Windows (p. 505) | SAS ライブラリに使用するデフォルトのアクセス方式を指定します。 |
| | FILELOCKWAITMAX= システムオプション:Windows (p. 507) | ロックされたファイルの待機時間の上限を設定します。 |
| | MEMCACHE システムオプション:Windows (p. 531) | メモリベースのライブラリを、SAS のファイルキャッシュとして使用するように指定します。 |
| | MEMLIB システムオプション:Windows (p. 532) | WORK ライブラリを、メモリベースのライブラリとして処理するように指定します。 |
| | OBS システムオプション:Windows (p. 540) | どこまでのオブザベーション(レコード)を処理するか指定します。 |
| | SGIO システムオプション:Windows (p. 562) | Scatter/Gather I/O 機能を有効化します。 |
| マクロ:SAS マクロ | MSYMTABMAX システムオプション:Windows (p. 536) | マクロ変数のシンボルテーブルに割り当てるメモリの最大量を指定します。 |
| | MVARSIZE システムオプション:Windows (p. 537) | メモリ内マクロ変数の最大サイズを指定します。 |
| ログおよびプロシジャ出力制御:ODS 印刷 | EMAILDLG システムオプション (p. 583) | ユニバーサル印刷のメニューを有効化するかどうかと、印刷の初期設定を設定するかどうかを指定します。 |
| ログとプロシジャ出力制御:ODS 出力 | PAPERTYPE システムオプション:Windows (p. 543) | 印刷に使用する用紙の種類をプリンタに指定します。 |
| | PRTPERSISTDEFAULT システムオプション]Windows (p. 549) | SAS セッション間で同じ出力先プリンタを使用するように指定します。 |
| | UPRINTMENUSWITCH システムオプション:Windows (p. 584) | ファイルメニューでユニバーサル印刷コマンドを有効化します。 |
| ログとプロシジャ出力制御:SAS ログ | ECHO システムオプション:Windows (p. 502) | SAS の起動時に、SAS ログに表示するメッセージを指定します。 |

| カテゴリ | 言語要素 | 説明 |
|---|---|---|
| | FULLSTIMER システムオプション:Windows (p. 513) | すべてのシステム性能 統計を SAS ログに書き出すかどうかを指定します。 |
| | HOSTINFOLONG システムオプション:Windows (p. 520) | SAS 起動時に SAS ログの追加動作環境情報の記述を指定します。 |
| | MSGCASE システムオプション:Windows (p. 535) | SAS によって生成される説明、警告、エラーメッセージを大文字で表示するかどうかを指定します。 |
| | OPLIST システムオプション:Windows (p. 540) | SAS システムオプションの設定を SAS ログに表示するには、OPTIONS プロシジャを使用します。 |
| | RTRACE システムオプション:Windows (p. 554) | SAS セッション時に読み込まれるリソースリストを生成します。 |
| | STIMEFMT システムオプション:Windows (p. 571) | FULLSTIMER と STIMER 出力を表示する時間に使用されるフォーマットを指定します。 |
| | STIMER システムオプション Windows (p. 574) | システムパフォーマンス統計量の一部を SAS ログに書き出します。 |
| | VERBOSE システムオプション:Windows (p. 586) | 構成ファイルに指定されているすべてのシステムオプションの設定を、端末とバッチログのどこに書き出すかを制御します。 |
| ログとプロシジャ出力制御:SAS ログとプロシジャ出力 | LINESIZE システムオプション:Windows (p. 525) | SAS のログウィンドウおよび出力ウィンドウの行サイズを指定します。 |
| | PAGESIZE システムオプション:Windows (p. 542) | SAS 出力のページ構成行数を指定します。 |
| ログとプロシジャ出力制御:プロシジャ出力 | FORMCHAR システムオプション:Windows (p. 512) | デフォルトの出力罫線文字を指定します。 |
| | HOSTPRINT システムオプション:Windows (p. 521) | Windows プリントマネージャを印刷に使用するように指定します。 |
| | PAGENO システムオプション:Windows (p. 541) | ページ番号をリセットします。 |
| | PRNGETLIST システムオプション:Windows (p. 548) | システムに割り当てられたプリンタを 認識するかどうかを指定します。 |
| | SYSPRINT システムオプション:Windows (p. 578) | SAS 出力を印刷するための出力先プリンタを指定します。 |
| | SYSPRINTFONT システムオプション:Windows (p. 579) | 現在のデフォルトプリンタに、印刷時に使用するフォントを指定します。 |

# ディクショナリ

## ACCESSIBILITY システムオプション:Windows

**ツールのユーザー設定**ダイアログボックスでユーザー補助機能を使用できるようにします。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 入力制御:データ処理

**PROC OPTIONS GROUP=** INPUTCONTROL

**デフォルト:** STANDARD

**Windows 固有:** すべて

### 構文

**-ACCESSIBILITY** STANDARD | EXTENDED

#### *必須引数*

**STANDARD**
標準の**ツールのユーザー設定**ダイアログボックスと**プロパティ**ダイアログボックスの使用を指定します。

**EXTENDED**
アクセス機能が**ツールのユーザー設定**ダイアログボックスと一部の**プロパティ**ダイアログボックスで有効であることを指定します。

### 詳細

ACCESSIBILITY オプションを EXTENDED に設定すると、ツールのユーザー設定ダイアログボックスのユーザー設定タブと、一部の**プロパティ**ダイアログボックスで、ユーザー補助機能を使用できるように変更されます。

**カスタマイズ**タブには、**ファイルメニュー**と**編集メニュー**の 2 つのボタンが追加されます。これらのメニューボタンによって、ツールバーボタンから使用するコマンドへのユーザー補助機能がサポートされます。

これらのダイアログボックスのタブは、ユーザー補助に関する一部の**プロパティ**ダイアログボックスを有効にするためのボタンです。Ctrl キーを押しながら Page Up キーを押すか、Ctrl キーを押しながら Page Down キーを押すことによって、該当する **プロパティ**ダイアログボックスのすべての部分にアクセスすることができます。

このシステムオプションを EXTENDED に設定すると、上書きカーソルと挿入カーソルを切り替えることができます。デフォルトは挿入カーソルです。これは一部のユーザー補助ユーティリティで、挿入カーソルが要求されるためです。

## ALIGNSASIOFILES システムオプション:Windows

SAS データセットでデータのページを調整します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 入力制御:データ処理

**PROC OPTIONS GROUP=** SASFILES

**デフォルト:** OFF

**Windows 固有:** ALL

## 構文

**-ALIGNSASIOFILES**

**-NOALIGNSASIOFILES**

### *引数なし*

必要な引数はありません。

## 詳細

SAS データセットは、ヘッダーと、それに続くデータの一枚以上の ページからなります。通常は、Windows マシン上ではヘッダーのサイズは 1K で、UNIX では 8K です。ALIGNSASIOFILES により、ヘッダーのサイズが強制的にデータ ページと同じサイズに設定されるため、ページは境界にアラインされ、これにより I/O はより効率的に行われるようになります。

# ALTLOG システムオプション:Windows

SAS ログのコピー先を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES、LOGCONTROL

**デフォルト:** NOALTLOG

**Windows 固有:** *destination* は、有効な Windows パスまたはファイル名に解決される必要があります。

## 構文

**-ALTLOG** *destination*

**-NOALTLOG**

### *必須引数*

**ALTLOG *destination***

SAS ログのコピー先を指定します。*destination* 引数には、Windows の有効なパス名やファイル名(デバイス名を含む)、またはパス名に関連付けられた環境変数を指定できます。パス名のみを指定した場合、指定されたディレクトリに *filename*.LOG という名前でコピーされます。*filename* は、SAS ジョブの名前です。SAS を対話的に実行しているときに、パス名のみを指定した場合、そのパスに SAS.LOG という名前でコピーされます。

**NOALTLOG**
SAS ログをコピーしないように指定します。

## 詳細

ALTLOG システムオプションは、SAS ログのコピーの書き込み先を指定します。印刷目的でログ出力を取り込むには、ALTLOG システムオプションを使用します。

SAS ログを、デフォルト以外のプリンタに送信するには、Windows の有効なプリンタ名を、*destination* の値に使用します。

ALTLOG システムオプションの値にディレクティブを使用すると、ログの表示/非表示のタイミング、ログの命名方法を、時間、月、曜日などのリアルタイムイベントに基づいて制御できます。ディレクティブのリストについては、*SAS システムオプション: リファレンス*の LOGPARM=システムオプションを参照してください。

OBJECTSERVER および NOTERMINAL システムオプションを使用して SAS が起動され、ログが指定されていない場合、ログやログメッセージはすべて破棄されます。

*注:* ALTLOG は旧バージョンの SAS のシステムオプションである LDISK、LPRINT、LTYPE の代わりに使用されます。

## 関連項目:

- “ファイルへのプロシジャの出力と SAS ログの送付について” (183 ページ)
- SAS ログ

# ALTPRINT システムオプション:Windows

SAS プロシジャ出力ファイルのコピー先を指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | 環境制御:ファイル |
| **PROC OPTIONS GROUP=** | ENVFILES |
| **デフォルト:** | NOALTPRINT |
| **Windows 固有:** | *destination* には、有効な Windows パスまたはファイル名を指定する必要があります。 |

## 構文

**-ALTPRINT** *file-specification*

**-NOALTPRINT**

### *必須引数*

**ALTPRINT** ***file-specification***
SAS プロシジャ出力ファイルのコピー先を指定します。*file-specification* 引数は、Windows の有効なパス名やファイル名(デバイス名を含む)、またはパス名を割り当てた環境変数にすることができます。パス名のみを指定した場合、指定されたディレクトリに *filename*.LST という名前でコピーされます。*filename* は、SAS ジョブ名です。SAS を対話的に実行しているときに、パス名のみを指定した場合、ファイル名は SAS となります。

**NOALTPRINT**
SAS プロシジャ出力ファイルのコピーを作成しません。

## 詳細

ALTPRINT システムオプションは、SAS プロシジャからの出力ファイルのコピーの書き込み先を指定します。印刷目的で、プロシジャ出力を取り込むには、ALTPRINT システムオプションを使用します。

プロシジャ出力を、デフォルト以外のプリンタに送信するには、Windows の有効なプリンタ名を、*destination* の値に使用します。

*注:* ALTPRINT は、旧バージョンの SAS のシステムオプションである PDISK、PPRINT、PTYPE の代わりに使用されます。

## 関連項目:

- “ファイルへのプロシジャの出力と SAS ログの送付について” (183 ページ)
- “印刷” (165 ページ)

# APPEND システムオプション:Windows

SAS 起動時に使用して、指定したシステムオプションの終端に値を追加します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**システムオプション**ウィンドウ

**PROC OPTIONS GROUP=** ENVFILES

**デフォルト:** なし

**Windows 固有:** SAS 起動構文

## 構文

**-APPEND** *system-option 引数*

### *必須引数*

***system-option***
AUTOEXEC、CMPLIB、FMTSEARCH、HELPLOC、MAPS、MSG、SAMPLOC、SASAUTOS、SASHELP、SASSCRIPT、SET を指定できます。

***引数***
*system-option* の現在の値に追加する、新しいパス名や環境設定を指定します。次に、FMTSEARCH オプションにライブラリを追加する例を示します。

```
-set APFMTLIB "SASEnvironment/SASFormats"
-append fmtsearch APFMTLIB
```

## 詳細

デフォルトでは、AUTOEXEC、CMPLIB、FMTSEARCH、HELPLOC、MAPS、MSG、SASAUTOS、SASSCRIPT、SASHELP、SET のシステムオプションを複数回指定すると、最後に指定した値が SAS にて使用されます。これらのオプションのいずれかによってすでにパス名が指定されている場合にパス名を追加するには、APPEND または INSERT システムオプションを使用して新しいパス名を追加します。たとえば、次の

SAS コマンドを入力すると、SAS が検索するヘルプファイルの唯一の場所は **c:\app2\help** になり、OPTIONS プロシジャの出力では **C:\app2\help** のみが表示されます。

```
sas -helploc "c:\app1\help" -helploc "c:\app2\help"
```

SAS に両方の場所にてヘルプファイルを検索させるには APPEND オプションを使って HELPOC オプションの値に新しい場所を追加します。

```
sas -helploc "c:\app1\help" -append helploc "c:\app2\help"
```

HELPLOC オプションの値は、OPTIONS プロシジャで次のように示されます。

```
("c:\app1\help" "c:\app2\help")
```

### 関連項目:

- "Changing an Option Value By Using the INSERT and APPEND System Options" (*SAS System Options: Reference* 1 章)
- "APPEND= System Option" (*SAS System Options: Reference*)
- "INSERT= System Option" (*SAS System Options: Reference*)

## AUTHPROVIDERDOMAIN システムオプション:Windows

ドメインサフィックスを認証プロバイダに割り当てます。

**該当要素:** 構成ファイル、SAS 起動
**カテゴリ:** 環境制御:初期化と操作
**PROC OPTIONS GROUP=** EXECMODES
**別名:** AUTHPD
**参照項目:** "AUTHPROVIDERDOMAIN System Option" (*SAS System Options: Reference*)

### 構文

**AUTHPROVIDERDOMAIN** <プロバイダ:ドメイン>

**AUTHPROVIDERDOMAIN** <(provider-1:domain-1<, ...provider:domain-n>)>

## AUTHSERVER システムオプション:Windows

セキュリティで保護されたサーバーログインを検索するドメイン認証サーバーを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ
**カテゴリ:** 環境制御:初期化と操作
**PROC OPTIONS GROUP=** EXECMODES
**デフォルト:** 信頼できるローカルサーバー
**Windows 固有:** すべて

## 構文

**-AUTHSERVER** <", " | '*domain-name*' | '.'>

**AUTHSERVER** <", " | '*domain-name*' | '.'>

### *オプション引数*

" "
有効なユーザーログインを、最初にローカルサーバーで、次に信頼されたサーバーで検索するように指定します。

***'domain-name'***
有効なユーザーログインを検索するための特定の domain-name を指定します。一重引用符は必須です。

'.'
有効なユーザーログインをローカルサーバーのみで検索します。一重引用符は必須です。

## 詳細

AUTHSERVER システムオプションは、ユーザーログインの確認時に検索するサーバーを指定します。

## 比較

AUTHSERVER システムオプションは、単一のドメイン認証サーバーを指定する場合に使用します。AUTHPROVIDERDOMAIN システムオプションは、複数の認証プロバイダと、そのプロバイダに割り当てられたドメインを指定する場合に使用します。

## 関連項目:

"AUTHPROVIDERDOMAIN システムオプション:Windows" (490 ページ)

# AUTOEXEC システムオプション:Windows

SAS autoexec ファイルを指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | 環境制御:ファイル |
| **PROC OPTIONS GROUP=** | ENVFILES |
| **デフォルト:** | ファイルがあれば AUTOEXEC.SAS、なければ、なし |
| **Windows 固有:** | *file-specification* には、Windows で有効なファイル名を指定する必要があります。 |

## 構文

**-AUTOEXEC** *file-specification*

**-NOAUTOEXEC**

### *必須引数*

**AUTOEXEC** ***(file-specification1<...file-specification-n>)***
デフォルトの AUTOEXEC.SAS ファイルの代わりに使用する、SAS autoexec ファイルを指定します。*file-specification* 引数には、有効な Windows ファイル名またはパス名を割り当てた環境変数を指定します。SAS autoexec ファイルの詳細については、*Windows 版 SAS* の「はじめに」セクションの「SAS autoexec ファイル」を参照してください。

**NOAUTOEXEC**
SAS autoexec ファイルが 1 つでも存在するときに、SAS autoexec ファイルを処理しないことを指定します。

## 詳細

AUTOEXEC システムオプションは、autoexec ファイルを指定します。autoexec ファイルには、SAS を起動したとき、または別の SAS プロセスを開始したときに、自動的に実行される SAS ステートメントを含めます。autoexec ファイルには、有効な SAS ステートメントであれば何でも記述することができます。たとえば、SAS セッションで日常的にアクセスする、SAS ライブラリの LIBNAME ステートメントを記述することができます。

AUTOEXEC.SAS ファイルが見つからない場合、このオプションには、デフォルトの NOAUTOEXEC が指定されます。

APPEND と INSERT システムオプションを使ってファイル指定を追加することができます。

## 関連項目:

- “SAS 自動実行ファイル ” (20 ページ)
- “APPEND= System Option” (*SAS System Options: Reference*)
- “INSERT= System Option” (*SAS System Options: Reference*)

# AWSCONTROL システムオプション:Windows

メイン SAS ウィンドウに、タイトルバー、システム/コントロールメニュー、最大化/最小化ボタンを含めるかどうかを指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動、OPTIONS ステートメント、**システムオプション**ウィンドウ |
| **カテゴリ:** | 環境制御:表示 |
| **PROC OPTIONS GROUP=** | ENVDISPLAY |
| **デフォルト:** | TITLE SYSTEMMENU MINMAX |
| **Windows 固有:** | すべて |

## 構文

**-AWSCONTROL** <TITLE | NOTITLE> <SYSTEMMENU | NOSYSTEMMENU> <MINMAX | NOMINMAX>

**AWSCONTROL**= <TITLE | NOTITLE> <SYSTEMMENU | NOSYSTEMMENU> <MINMAX | NOMINMAX>

### *オプション引数*

**AWSCONTROL**
メイン SAS ウィンドウに、タイトルバー、システムメニュー、最大化/最小化ボタンを表示することを指定します。

**TITLE | NOTITLE**
メイン SAS ウィンドウに、タイトルバーを表示するかどうかを指定します。NOTITLE を指定すると、システムメニュー、最大化/最小化ボタンも自動的に非表示になります。

**SYSTEMMENU | NOSYSTEMMENU**
メイン SAS ウィンドウのタイトルバーに、システムメニューを表示するかどうかを指定します。NOSYSTEMMENU を指定すると、最大化/最小化ボタンも非表示になります。

**MINMAX | NOMINMAX**
メイン SAS ウィンドウのタイトルバーに、最大化/最小化ボタンを表示するかどうかを指定します。

## 詳細

AWSCONTROL システムオプションでは、メイン SAS ウィンドウだけが制御され、メイン SAS ウィンドウに含まれるウィンドウは制御されません。SASCONTROL システムオプションでは、このような SAS プロセスウィンドウが制御されます。

このシステムオプションは、SAS/AF アプリケーションのインターフェイスをカスタマイズするための、プログラマ向けのオプションです。

## 関連項目:

"SASCONTROL システムオプション:Windows" (557 ページ)

# AWSDEF システムオプション:Windows

SAS の起動時の、メイン SAS ウィンドウの位置および大きさを指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ |
| **カテゴリ:** | 環境制御:表示 |
| **PROC OPTIONS GROUP=** | ENVDISPLAY |
| **デフォルト:** | 80%の画面の高さと幅 |
| **Windows 固有:** | すべて |

## 構文

**-AWSDEF** *row-percent-position column-percent-position height-percent width-percent*

**AWSDEF=***row-percent-position column-percent-position height-percent width-percent*

### *必須引数*

***row-percent-position* と *column-percent-position***
メイン SAS ウィンドウの左上隅の位置を、スクリーンに対するパーセント単位で指定します。たとえば、これらの値にそれぞれ 50 を指定した場合、SAS ウィンドウの左上隅が、ディスプレイの中央に配置されます。

これらのパラメータの有効範囲は、0 から 95 です。

***height-percent* と *width-percent***
メイン SAS ウィンドウのサイズを、スクリーンに対するパーセント単位で指定します。たとえば、これらの値にそれぞれ 100 を指定した場合、SAS ウィンドウが全画面表示になります。たとえば、これらの値にそれぞれ 50 を指定した場合、SAS ウィンドウは、画面の半分の領域を使用します。

これらのパラメータの有効範囲は、40 から 100 です。

## 詳細

AWSDEF システムオプションは、SAS の起動時の、メイン SAS ウィンドウの位置および大きさを指定します。AWSDEF システムオプションの使用方法の例については、“メイン SAS ウィンドウのサイズと配置の変更” (65 ページ)を参照してください。

# AWSMENU システムオプション:Windows

メイン SAS ウィンドウに、メニューバーを表示するかどうかを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** AWSMENU

**Windows 固有:** すべて

## 構文

**-AWSMENU** | **-NOAWSMENU**

**AWSMENU** | **NOAWSMENU**

### *必須引数*

**AWSMENU**
メイン SAS ウィンドウにメニューバーを表示します。

**NOAWSMENU**
メイン SAS ウィンドウにメニューバーを表示しません。

## 詳細

AWSMENU システムオプションは、SAS/AF アプリケーションのインターフェイスをカスタマイズするための、プログラマ向けのオプションです。

## AWSMENUMERGE システムオプション:Windows

メインメニューに、Windows 固有のメニュー項目を組み込むかどうかを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** AWSMENUMERGE

**Windows 固有:** すべて

### 構文

**-AWSMENUMERGE** | **-NOAWSMENUMERGE**

**AWSMENUMERGE** | **NOAWSMENUMERGE**

#### *構文記述*

**AWSMENUMERGE**
Windows 固有のメニュー項目を組み込むことを指定します。

**NOAWSMENUMERGE**
Windows 固有のメニュー項目を組み込まないことを指定します。

### 詳細

AWSMENUMERGE システムオプションは、Window 動作環境に固有のメニュー項目を、メイン SAS ウィンドウのメニューに追加するかどうかを指定します。

このシステムオプションは、SAS/AF アプリケーションのインターフェイスをカスタマイズするための、プログラマ向けのオプションです。SAS をバッチモードで起動した場合、このシステムオプションは NOAWSMENUMERGE に設定されます。

### 関連項目:

“WINDOWSMENU システムオプション:Windows” (588 ページ)

## AWSTITLE システムオプション:Windows

メイン SAS タイトルバーの、デフォルトのテキストを置き換えます。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** なし

**Windows 固有:** すべて

## 構文

**-AWSTITLE** "*title-text*"

### *必須引数*

**“*title-text*”**
メイン SAS ウィンドウのタイトルバーに表示するテキストを指定します。テキストは、一重引用符または二重引用符で囲みます。

## 詳細

AWSTITLE システムオプションを使用すると、メイン SAS ウィンドウのタイトルバーに表示されるデフォルトのテキストを、指定したタイトルで置き換えることができます。

このシステムオプションは、SAS/AF アプリケーションのインターフェイスをカスタマイズするための、プログラマ向けのオプションです。

# BUFNO システムオプション:Windows

SAS データセットの処理に割り当てるバッファ数を指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** ファイル:SAS ファイル

**PROC OPTIONS GROUP=** SASFILES、PERFORMANCE

**デフォルト:** 1

**Windows 固有:** デフォルト値

**参照項目:** “BUFNO= System Option” (*SAS System Options: Reference*)

## 構文

**-BUFNO** *n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

**BUFNO=** *n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

### *システム引数*

***n* | *n*K | *n*M | *n*G**
割り当てるバッファ数をそれぞれ 1、1(バイト)、1,024 (キロバイト)、1,048,576 (メガバイト)または 1,073,741,824 (ギガバイト)の倍数で指定します。キロバイト、メガバイト、ギガバイト単位の数は、10 進値で指定することができます。たとえば、`8` はバッファ数 8、`.782k` はバッファ数 801、`3m` はバッファ数 3,145,728 を示します。

1G より大きい値を指定するには、*n*M オプションを使用するか、MAX を指定します。

***hex*X**
バッファ数を、16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、値 `2dx` はバッファ数 45 を指定します。

**MIN**
バッファ数を 0 に設定し、デフォルト値 1 を使用することを SAS に要求します。

**MAX**
バッファ数を 2,147,483,647 に設定します。

## 詳細

バッファ数は、データセットの永久的な属性ではありません。現在の SAS セッションまたはジョブについてのみ有効です。

BUFNO=は、入力、出力、あるいは更新のために開いた SAS データセットに適用されます。

BUFNO=を使用すると、特定の SAS データセットの処理に要求される入出力操作の数を制限することによって、実行時間を短縮することができます。ただし、実行時間が短縮されるかわりに、メモリの使用量が増加します。

Windows 版では、割り当て可能なバッファの最大数は、使用可能なメモリの量によって決定されます。データセットのページ数に基づいてバッファ数が割り当てられるようにするには、SASFILE ステートメントを使用します。

## 関連項目:

- “BUFSIZE システムオプション:Windows” (497 ページ)
- 12 章: “Optimizing System Performance” (*SAS Language Reference: Concepts*)

# BUFSIZE システムオプション:Windows

出力 SAS データセットの永久的なバッファページサイズを指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ |
| **カテゴリ:** | ファイル:SAS ファイル |
| **PROC OPTIONS GROUP=** | SASFILES、PERFORMANCE |
| **デフォルト:** | 0 |
| **Windows 固有:** | 次に対する有効な値:*n* |
| **参照項目:** | “BUFNO= System Option” (*SAS System Options: Reference*) |

## 構文

**-BUFSIZE** *n* | *n*K | *n*M | *n*G | *hex*X | MAX

**BUFSIZE**=*n* | *n*K | *n*M | *n*G | *hex*X | MAX

### *必須引数*

***n* | *n*K | *n*M | *n*G**
バッファページサイズをそれぞれ 1、1,024 (キロバイト)、1,048,576 (メガバイト)および 1,073,741,824 (ギガバイト)の倍数で指定します。キロバイト、メガバイト、ギガバイト単位の数は、10 進値で指定することができます。たとえば、`8` はバッファ数 8、`.782k` はバッファ数 801、`3m` はバッファ数 3,145,728 を示します。

***hex*X**
バッファページサイズを、16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、値 `2dx` は、バッファページサイズを 45 バイトに設定します。

**MAX**
バッファページサイズを、2,147,483,647 バイトに設定します。

## 詳細

BUFSIZE システムオプションを使用すると、出力 SAS データセットの永久的なバッファページサイズを指定できます。Windows 版では、512 - 2,147,483,647 バイトの値を指定できます。デフォルト値の 0 を使用すると、エンジンによってページサイズがオブザベーションのサイズに合わせて選択されるため、バッファページサイズが最適化されます。

経験が豊富なユーザーによって、メモリ使用量つまりページあたりのオブザベーション数を最大にするために、BUFSIZE システムオプションの値は変更されます。

## 関連項目:

“BUFNO システムオプション:Windows” (496 ページ)

# CATCACHE システムオプション:Windows

開いたままにしておける SAS カタログの数を指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | ファイル:SAS ファイル |
| **PROC OPTIONS GROUP=** | SASFILES |
| **デフォルト:** | 0 |
| **Windows 固有:** | 次に対する有効な値:*n* |
| **参照項目:** | “CATCACHE= System Option” (*SAS System Options: Reference*) |

## 構文

**-CATCACHE** *n* | *n*K | MIN | MAX

### *必須引数*

***n* | *n*K**
キャッシュメモリに保存できるオープンファイルディスクリプタの数を、1 (*n*)または 1,024 (*n*K)の倍数で指定します。キロバイト単位の数は、10 進値で指定することができます。たとえば、`8` はオープンファイルディスクリプタ数 8、`.782k` はオープンファイルディスクリプタ数 801、`3k` はオープンファイルディスクリプタ数 3,072 を示します。

*n* > 0 の場合、カタログを閉じずに、この数までのオープンファイルディスクリプタがキャッシュメモリに保存されます。

**MIN**
キャッシュメモリに保存されるオープンファイルディスクリプタの数を、0 に設定します。

**MAX**
キャッシュメモリに保存されるオープンファイルディスクリプタの数を、32,767 に設定します。

## 詳細

CATCACHE システムオプションを使用して、開いたままにする SAS カタログの数を指定すると、同じカタログを開いたり閉じたりする手間が省けます。

SAS が z/OS サーバーで実行され、MINSTG システムオプションが指定されている場合は、CATCACHE の値は自動的に 0 に設定されます。

# CLEANUP システムオプション:Windows

リソース不足状態の対処方法を指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** 環境制御:エラー処理

**PROC OPTIONS GROUP=** ERRORHANDLING

**デフォルト:** CLEANUP

**Windows 固有:** バッチモードで実行されている場合の動作

**参照項目:** “CLEANUP System Option” (*SAS System Options: Reference*)

## 構文

**-CLEANUP** | **-NOCLEANUP**

**CLEANUP** | **NOCLEANUP**

### *必須引数*

**CLEANUP**
セッション全体を通じて、実行に不要なリソースのクリーンアップが、自動的に継続して実行されるように指定します。不要なリソースには、キャッシュメモリのようなユーザーには見えないリソースと、KEYS ウィンドウのようなユーザーに見えるリソースがあります。

CLEANUP を指定すると、ディスク領域不足状態以外のリソース不足状態についてはメッセージが表示されません。ディスク領域不足状態でメッセージが表示されないようにするには、CLEANUP オプションと NOTERMINAL オプションを併用します。

**NOCLEANUP**
リソース不足状態の対処方法をユーザーが選択できるようにします。
NOCLEANUP が指定されている場合、リソース不足で SAS が実行できないときには、自動的にキャッシュメモリのようなユーザーには見えないリソースのクリーンアップが試行されます。ただし KEYS ウィンドウのようなユーザーに見えるリソースは、自動的にクリーンアップされません。その代わり、SAS がリソースの回復を試みる前に、メッセージが表示されます。

## 詳細

CLEANUP システムオプションでは、SAS でリソース不足の状態が発生した場合に、クリーンアップする項目を選択するメッセージを表示するかどうかを指定します。

NOCLEANUP を指定し、入力を求めるメッセージが表示された場合、ディスク領域不足以外のメニューでは、**継続**を選択できます。**継続** を選択すると、CLEANUP オプションが有効化され、SAS で使用できるディスク領域が残っている限り、リソース不足の状態が発生しても再度選択を求められることはありません。

# COMDEF システムオプション:Windows

**SAS コマンド**ウィンドウを表示する場所を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** BOTTOM CENTER

**Windows 固有:** すべて

## 構文

**-COMDEF** TOP | CENTER | BOTTOM
<LEFT | CENTER | RIGHT>

### *構文記述*

**TOP | CENTER | BOTTOM**
**コマンド**ウィンドウの垂直方向の位置を指定します。デフォルト値は、BOTTOM です。

**LEFT | CENTER | RIGHT**
**コマンド**ウィンドウの水平方向の位置を指定します。デフォルト値は、CENTER です。

## 詳細

最初に垂直方向の位置を指定する必要があります。水平方向の位置は任意で指定します。省略すると、CENTER が使用されます。

*注:* **コマンド**ウィンドウは、メイン SAS ウィンドウを基準にしてではなく、ディスプレイ全体を基準に位置が決定されます。また、COMDEF システムオプションはコマンドバーがメイン SAS ウィンドウにドッキングされていない場合のみ適用されます。

## 関連項目:

- “セッションプリファレンスについて ” (58 ページ)
- “コマンドバーを使用してコマンドの発行” (43 ページ)

## CONFIG システムオプション Windows

SAS システムオプションの値を初期化または上書きする際に使用する構成ファイルを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** システム管理:インストール

**PROC OPTIONS GROUP=** INSTALL

**デフォルト:** !*sasroot*\SASV9.CFG

**Windows 固有:** すべて

### 構文

**-CONFIG** *file-specification*

#### *必須引数*

***file-specification***
使用する SAS 構成ファイルのファイル名または有効なファイル名を参照する Windows 環境変数を指定します。*file-specification* には、有効な Windows ファイル名を指定する必要があります。*file-specification* にスペースが含まれる場合は、引用符で囲む必要があります。

### 詳細

CONFIG システムオプションには、構成ファイルの完全なファイル名を指定します。このファイルには、SAS の起動時に自動的に実行される、SAS のオプションが含まれます。SAS ではデフォルトの構成ファイルも提供されていますが、ユーザーが独自に構成ファイルを作成し、任意の場所に格納することもできます。

### 関連項目:

“SAS 構成ファイル ” (15 ページ)

## DEVICE システムオプション:Windows

SAS/GRAPH ソフトウェアのグラフィックス出力デバイスドライバを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** グラフィックス:ドライバ設定

**PROC OPTIONS GROUP=** GRAPHICS

**別名:** -DEV

**デフォルト:** なし

**Windows 固有:** 次に対する有効な値:*device-driver-name*;デフォルト値

**参照項目:** “DEVICE= System Option” (*SAS/GRAPH: Reference*)

## 構文

**-DEVICE** *device-driver-name*

**DEVICE=** *device-driver-name*

### 必須引数

***device-driver-name***
グラフィックス出力デバイスドライバの名前を指定します。

## 詳細

Windows で使用可能なデバイスドライバのリストを表示するには、GDEVICE プロシジャを使用します。SAS ウィンドウ環境を使用する場合は、次のステートメントをサブミットします。

```
proc gdevice catalog=sashelp.devices;
run;
quit;
```

SAS ログにこのデバイスのリストを書き込むには、次のステートメントをサブミットします。

```
proc gdevice catalog=sashelp.devices nofs;
    list _all_;
run;
quit;
```

使用しているサイトによっては、GDEVICE0 ライブラリ参照名によって参照される追加デバイスカタログが定義されている場合があります。詳細は、SAS サポートコンサルタントに問い合わせてください。

# ECHO システムオプション:Windows

SAS の起動時に、SAS ログに表示するメッセージを指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | ログとプロシジャ出力制御:SAS ログ |
| **PROC OPTIONS GROUP=** | LOGCONTROL |
| **デフォルト:** | NOECHO |
| **Windows 固有:** | すべて |

## 構文

**-ECHO** "*message*" | -NOECHO

### 必須引数

**ECHO "*message*"**
SAS ログに表示するメッセージテキストを指定します。テキストが複数の単語で構成される場合、一重引用符または二重引用符で囲む必要があります。1 語のみの場合は、引用符は不要です。

**NOECHO**
SAS ログにメッセージを表示しないことを指定します。

## 詳細

autoexec ファイルにエラーが検出された場合は、ECHO システムオプションの設定にかかわらず、メッセージは SAS ログに出力されます。

## 例: 例

たとえば、次のように指定できます。

```
-echo "SAS System under Windows
       is initializing."
```

SAS の起動時、**ログ**ウィンドウに、このメッセージが表示されます。

## 関連項目:

- “ECHOAUTO System Option” (*SAS System Options: Reference*)
- “The SAS Log” (*SAS Language Reference: Concepts* 9 章)

# EMAILDLG システムオプション:Windows

ユーザーが使用している電子メールアプリケーションのダイアログボックスか、または SAS が提供する電子メールのダイアログボックスを使用するかを指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | 通信:電子メール |
| **PROC OPTIONS GROUP=** | E-MAIL |
| **デフォルト:** | NATIVE |
| **Windows 固有:** | すべて |

## 構文

**-EMAILDLG** NATIVE | Windows 版 SAS

### *必須引数*

**NATIVE**
使用している電子メールシステムのベンダが提供する、電子メールのダイアログボックスを使用することを指定します。SAS で電子メールシステムのダイアログボックスを使用するには、電子メールシステムで MAPI インターフェイスがサポートされている必要があります。

**Windows 版 SAS**
SAS が提供する、電子メールのダイアログボックスを使用することを指定します。

## 詳細

EMAILDLG システムオプションは、ユーザーが使用している電子メールアプリケーションの対話型ダイアログボックスを使用するか、または SAS が提供するダイアログボ

ックスを使用するかを指定します。デフォルトでは、電子メールアプリケーションのダイアログボックスが使用されます。

## 関連項目:

"SAS を使用して電子メールの送信" (44 ページ)

# EMAILSYS システムオプション:Windows

電子メールの送信に使用する電子メールプロトコルを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTION ステートメント、**SAS システムオプション** ウィンドウ

**カテゴリ:** 通信:電子メール

**PROC OPTIONS GROUP=** EMAIL

**デフォルト:** MAPI

**Windows 固有:** すべて

## 構文

**-EMAILSYS** MAPI | VIM | SMTP

### *必須引数*

**MAPI**
電子メールインターフェイスに、MAPI (Messaging Application Program Interface)を使用することを指定します。この値はデフォルトの設定です。

**VIM**
電子メールインターフェイスに、VIM (Vendor Independent Mail)を使用することを指定します。

**SMTP**
電子メールインターフェイスに、SMTP (Simple Mail Transport Protocol)を使用することを指定します。

## 詳細

SAS では、3 種類のプロトコルがサポートされています。MAPI (Microsoft Exchange など)、Vendor Independent Mail (VIM)、SMTP です。デフォルトの値は MAPI です。SMTP を指定する場合、EMAILHOST システムオプションと EMAILPORT システムオプションも設定する必要があります。SMTP は、電子メールを自動的に送信する場合にのみ利用できます。電子メールアプリケーションのダイアログボックス、または SAS による電子メールのダイアログボックスを使用する場合、SMTP は使用できません。

## 関連項目:

- "SAS を使用して電子メールの送信" (44 ページ)
- *SAS システムオプション: リファレンス*
- EMAILID システムオプション
- EMAILPW システムオプション
- EMAILAUTHPROTOCOL システムオプション

- 38 章: “The SMTP E-Mail Interface” (*SAS Language Reference: Concepts*)

## ENGINE システムオプション:Windows

SAS ライブラリに使用するデフォルトのアクセス方式を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** ファイル:SAS ファイル

**PROC OPTIONS GROUP=** SASFILES

**デフォルト:** V9

**Windows 固有:** 有効な値

### 構文

**-ENGINE** *engine-name*

#### *必須引数*

***engine-name***
Windows 版では、次のいずれかの値を指定できます。

**BASE | V9**
SAS System 9 ファイル用のデフォルトの SAS エンジンを指定します。

**BMDP**
BMDP データファイル用のエンジンを指定します。

**OSIRIS**
OSIRIS データファイル用のエンジンを指定します。

**SPSS**
SPSS データファイル用のエンジンを指定します。

**V8**
バージョン 8 ファイル用の SAS エンジンを指定します。

**V7**
バージョン 7 ファイル用の SAS エンジンを指定します。

**V6**
リリース 6.08 - 6.12 用のデフォルトのエンジンを指定します。V6 エンジンは、32 ビットの動作環境でのみサポートされます。

**V604**
リリース 6.04 およびリリース 6.03 用のデフォルトのエンジンを指定します。

**XML**
XML ファイル用のデフォルトのエンジンを指定します。

**XPORT**
ランスポートエンジンを指定します。

### 詳細

SAS ライブラリが空のディレクトリまたは新しいファイルを参照する場合は、デフォルトのエンジンが使用されます。SAS/SHARE Engine と SAS/ACCESS Engine の詳細については、対応するマニュアルを参照してください。

### 関連項目:

- “ライブラリエンジンの種類” (123 ページ)
- *SAS 言語リファレンス: 解説編*
- *SAS/ACCESS for Relational Databases: Reference*
- *Communications Access Methods for SAS/CONNECT and SAS/SHARE*

## ENHANCEDEDITOR システムオプション:Windows

SAS 起動時に拡張エディタを使用可能にするかどうかを指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | 環境制御:表示 |
| **PROC OPTIONS GROUP=** | ENVDISPLAY |
| **デフォルト:** | ENHANCEDEDITOR |
| **Windows 固有:** | すべて |

### 構文

**-ENHANCEDEDITOR** | **-NOENHANCEDEDITOR**

#### *必須引数*

**ENHANCEDEDITOR**
SAS 起動時に拡張エディタを使用可能にします。

**NOENHANCEDEDITOR**
SAS 起動時に拡張エディタを使用可能にしません。

### 詳細

デフォルトでは、SAS の起動時に拡張エディタは使用可能になります。SAS の起動時に拡張エディタを使用可能にしない場合は、NOENHANCEDEDITOR システムオプションを使用します。

### 関連項目:

“WEDIT コマンド:Windows” (361 ページ)

## FILELOCKWAIT システムオプション:Windows

ロックされたファイルが使用可能になるまで待機する秒数を設定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** SAS ファイル

**PROC OPTIONS GROUP=** SASFILES

**デフォルト:** 0

**Windows 固有:** すべて

## 構文

**FILELOCKWAIT=** *wait-time*

### *必須引数*

***wait-time***
ロックされたファイルが使用できるようになるまで待機する時間(秒)を指定します。

## 詳細

通常、ロックされたファイルにアクセスしようとすると、エラーが返されます。FILELOCKWAIT=システムオプションでは、ロックされたファイルが利用可能になるまでの待機時間を制限したり、オンにしたりできます。FILELOCKWAIT= として *wait-time* の値を設定すると、SAS はファイルが利用可能になるのを指定時間待機してから 失敗を報告します。指定時間が過ぎると、SAS はファイルがロックされていることを示す locked-file エラーを報告し、DATA ステップは失敗します。locked-file エラーが報告されるまでの待機時間の許容設定値は最大 10 分です。FILELOCKWAIT=を 0 設定した場合、SAS はすぐに失敗します。

FILELOCKWAIT システム オプションは主に、FILELOCKWAITMAX システム オプションを使用して FILELOCKWAIT=のデフォルト値を編集する権限があるシステム管理者が設定します。また、システム管理者はこのオプションを制限することもできます。

FILELOCKWAIT= ステートメントオプションを LIBNAME ステートメントで指定して ライブラリ内のファイルに対して制限時間を設定することもできます。

# FILELOCKWAITMAX= システムオプション:Windows

ロックされたファイルの待機時間の上限を設定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** ファイル:SAS ファイル

**PROC OPTIONS GROUP=** SASFILES

**デフォルト:** 600

**Windows 固有:** すべて

## 構文

**FILELOCKWAITMAX=** *wait-time*

### 必須引数

***wait-time***
ロックされたファイルが使用できるようになるまで待機する時間(秒)を指定します。

デフォルト 600

範囲 0-600

操作 FILELOCKWAITMAX=システムオプションを指定すると、別のプロセスで、ロックされている SAS ファイルの解放を待機している SAS/SHARE サーバーやクライアントセッションに悪影響を及ぼす可能性があります。待機条件によっては、SAS/SHARE Server やクライアントの処理でエラーが発生する場合があります。

SAS/SHARE プロセスでエラーが発生しないようにするには、FILELOCKWAITMAX=0 に指定します。この値は、SAS/SHARE Server やクライアントが、ロックされたファイルの解放を待機するのをキャンセルします。待機をキャンセルすると、プロセスの失敗が避けられます。

## 詳細

FILELOCKWAITMAX=システムオプションでは、ロックされたファイルの待機時間を制限したり、キャンセルしたりできます。ファイルが利用可能になるまで待機するには、FILELOCKWAIT= LIBNAME オプションを使用します。管理者は、FILELOCKWAITMAX=システムオプションを使用して、待機を制限したり、キャンセルしたりできます。通常、ロックされたファイルにアクセスしようとすると、エラーが返されます。ユーザーが FILELOCKWAITMAX=を 0 に設定すると、ファイルがロックされている場合すぐにエラーが返されます。このオプションは、主にシステム管理者が設定します。

## 関連項目:

“LIBNAME ステートメント:Windows” (460 ページ)

# FILTERLIST システムオプション:Windows

**開くと名前を付けて保存**ダイアログボックスで使用する、ファイルフィルタの代替セットを指定します。

**該当要素:** 構成ファイル、SAS 起動
**カテゴリ:** 環境制御:表示
**PROC OPTIONS GROUP=** ENVDISPLAY
**デフォルト:** なし
**Windows 固有:** すべて

## 構文

**-FILTERLIST** "*filter1* | *filter2*|... | *filter-n*"

### 必須引数

***filter1...filter n***
"*.Bob's work | SAS*.*"のように、二重引用符で囲まれ、"|"によって区切られた、1つ以上のテキスト文字列を指定します。スペースや一重引用符を含む、長いファイル名の拡張子を指定できます。

## 詳細

FILTERLIST のすべてのフィルタが、**ファイルの種類**ボックス**ファイルを開く**ダイアログボックス)と**保存の種類** ボックス(**名前を付けて保存**ダイアログボックス内)に表示されるアプリケーション指定のフィルタリストに追加されます。FILTERLIST で先頭に指定されたフィルタが、デフォルトのフィルタになります。FILTERLIST は、二重引用符で囲む必要があります。

## 関連項目:

- “DLGOPEN コマンド:Windows” (336 ページ)
- “DLGSAVE コマンド:Windows” (342 ページ)

# FONT システムオプション:Windows

SAS ウィンドウで使用するフォントを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ
**カテゴリ:** 環境制御:表示
**PROC OPTIONS GROUP=** ENVDISPLAY
**デフォルト:** Sasfont 8
**Windows 固有:** すべて

## 構文

**-FONT** "*font-name*" <BOLD | NORMAL> <REGULAR | ITALIC> <*font-size*> <*character-set*>

**FONT**= "*font-name*" <BOLD | NORMAL> <REGULAR | ITALIC> <*font-size*> <*character-set*>

### 必須引数

***"font-name"***
SAS ウィンドウ環境でテキストに使用するフォント名を指定します。有効なフォント名を指定する必要があります("SAS Monospace"や"Courier"など)。*font-name* 引数は、二重引用符で囲む必要があります。これは、必須引数です。

**BOLD | NORMAL**
フォントの太さを指定します。デフォルトの設定は NORMAL です。

**REGULAR | ITALIC**
フォントのスタイルを指定します。デフォルトの設定は REGULAR です。

***font-size***
印刷に使用するフォントサイズを指定します。1 から 7200 の整数で指定する必要があります。この引数を省略した場合は、最後に選択されたサイズが使用されます。ただし、以前に使用されたサイズがない場合は、8 が使用されます。

***character-set***
使用する文字セットを指定します。デフォルトは、"Western"です。有効な値は、Western、Central European、Cyrillic、Greek、Turkish、Arabic、Baltic、Thai です。指定したフォントでサポートされない文字セットを指定すると、デフォルトの文字セットが使用されます。デフォルトの文字セットがそのフォントでサポートされていない場合、そのフォントにおけるデフォルトの文字セットが使用されます。

## 詳細

指定できるフォント名は、フォントフォルダに表示されます。フォントフォルダを開くには、**font** と**実行**ダイアログボックスに入力します。たとえば、SAS コマンドで次のオプションを使用できます。

```
-font "sas monospace bold" 12
```

SAS による出力は、固定幅(固定ピッチ)フォントで最適に表示されます。プロポーショナル(可変ピッチ)フォントを使用すると、テキストが正しく表示されない場合があります。フォントに対して無効な *point-size* を指定した場合、SAS はそのフォントに最も近いポイントサイズを適用します。

## 関連項目:

- “SYSGUIFONT システムオプション:Windows” (576 ページ)
- “SYSPRINTFONT システムオプション:Windows” (579 ページ)

# FONTALIAS システムオプション:Windows

1 つの SAS フォントに対して、1 つの Windows フォントを割り当てます。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | グラフィックス:ドライバ設定 |
| **PROC OPTIONS GROUP=** | GRAPHICS |
| **デフォルト:** | 可変(詳細セクションの表を参照) |
| **Windows 固有:** | すべて |

## 構文

**-FONTALIAS** "*SAS-font*" "*host-specific-font*"

### *必須引数*

**“*SAS-font*”**
置き換える SAS フォントを指定します。フォント名は、二重引用符で囲む必要があります。

"*host-specific-font*"
割り当てる Windows フォントを指定します。フォント名は、二重引用符で囲む必要があります。

## 詳細

置き換えるフォントごとに、FONTALIAS システムオプションを使用してください。

次に、Windows のデフォルトのフォント別名を示します。

***表 22.1*** *デフォルトのフォント名*

| SAS フォント名 | Windows フォント名 |
|---|---|
| Times | Times New Roman |
| Helvetica | Arial |
| Courier | Courier New |
| Symbol | Symbol |
| Script | Script |
| AvantGarde | Arial |
| Bookman | Times New Roman |
| Schoolbook | Times New Roman |
| Palatino | Times New Roman |
| Dingbats | Symbol |

## 例

システムオプション **`-fontalias "Times" "Courier New"`**を指定することにより、SAS に対して、Times という SAS フォントが要求された場合は、必ず Courier New を使用するように指定できます。

# FONTSLOC システムオプション:Windows

SAS セッション時にロードされる SAS フォントの保存場所を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** !*sasroot*\core\resource

**Windows 固有:** すべて

**参照項目:** “FONTSLOC= System Option” (*SAS System Options: Reference*)

### 構文

**-FONTSLOC** *directory-specification*

#### *必須引数*

***directory-specification***
SAS セッション時にロードされる SAS フォントが存在するディレクトリを指定します。*directory-specification* にスペースが含まれる場合は、引用符で囲む必要があります。

### 詳細

ディレクトリには、有効な Windows パス名を指定する必要があります。

## FORMCHAR システムオプション:Windows

デフォルトの出力罫線文字を指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** ログとプロシジャ出力制御:プロシジャ出力

**PROC OPTIONS GROUP=** LISTCONTROL

**デフォルト:** (SAS 構成ファイルを参照してください。)

**Windows 固有:** 次に対する有効な値:*formatting-characters*

**参照項目:** “FORMCHAR= System Option” (*SAS System Options: Reference*)

### 構文

**-FORMCHAR** "*formatting-characters*"

**FORMCHAR**="*formatting-characters*"

#### *必須引数*

***formatting-characters***
64 バイトまでの任意の文字列または文字列のリストを指定します。長さが 64 バイトに満たない場合は、文字列の右側にブランクが埋め込まれます。文字列は、二重引用符で囲む必要があります。

### 詳細

フォーマット文字は、CALENDAR、FREQ、TABULATE プロシジャなどさまざまなプロシジャで、表出力の輪郭と仕切りの作成に使用されます。プロシジャのオプションで罫線文字を省略した場合は、FORMCHAR=システムオプションで指定されたデフォルトの設定が使用されます。罫線文字として、16 進文字の定数を指定することもできます。このオプションで 16 進文字を使用すると、SAS によって 16 進文字の定数が、Windows 環境に適した値に解釈されます。

SAS の構成ファイルには、2 つの FORMCHAR システムオプション指定が含まれています。そのうちの 1 つは、コメントアウトされています。デフォルトの FORMCHAR は、フォントとして SAS Monospace と Sasfont の文字を使用します。標準のコードページ以外のコードページを使用する場合は、SAS 出荷時の FORMCHAR システムオプションをコメント化し、もう 1 つの FORMCHAR システムオプションを使用します。

*注:* 標準形式の文字を使用して、行と列の区切りや枠付きの表形式のレポートを読みやすく印刷するには、次のフォントとプリンタを使用します。

- SAS Monospace フォントまたは SAS Monospace Bold フォント
- TrueType フォントをサポートするプリンタ

## FULLSTIMER システムオプション:Windows

すべてのシステム性能 統計を SAS ログに書き出すかどうかを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** ログとプロシジャ出力制御:SAS ログ

**PROC OPTIONS GROUP=** LOGCONTROL

**デフォルト:** NOFULLSTIMER

**Windows 固有:** すべて

### 構文

**-FULLSTIMER** | **-NOFULLSTIMER**

**FULLSTIMER** | **NOFULLSTIMER**

#### *必須引数*

**FULLSTIMER**
各ステップおよび SAS セッション全体で使用されたコンピュータリソースの詳細なリストを、SAS ログに書き出すことを指定します。

**NOFULLSTIMER**
コンピュータリソースの詳細なリストを SAS ログに書き出さないことを指定します。これはデフォルトの設定です。

### 詳細

FULLSTIMER システムオプションは、SAS で利用できるコンピュータシステムのすべてのパフォーマンス統計量を、SAS ログに書き出すことを指定します。

STIMER オプションを有効にすると、経過時間による統計量も表示されます。STIMER オプションを無効にすると、FULLSTIMER オプションは、経過時間による統計量を生成しません。

SAS ウィンドウ環境などのタスクの統計量が必要な場合は(ウィンドウ環境の統計量は、SAS システムの終了時にのみ表示可能)、ALTLOG システムオプションを使用して、SAS ログのコピー先を指定します。詳細については、“ALTLOG システムオプション:Windows” (487 ページ)を参照してください。SAS セッションを終了する前に FULLSTIMER システムオプションを指定すると、指定したコピー先に、SAS ウィンドウ環境の統計量が表示されます。

FULLSTIMER オプションを有効にすると、次のような統計量が SAS ログに表示されます。

```
NOTE: There were 5 observations read from the data set MYSAS.DEPART1.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.96 seconds
      user cpu time       0.01 seconds
      system cpu time     0.15 seconds
      Memory              83k
      OS Memory           4648k
```

FULLSTIMER により、次の統計量が表示されます。

*表 22.2 FULLSTIMER 統計量の説明*

| 統計量 | 説明 |
|---|---|
| リアルタイム | SAS ジョブの処理にかかった時間。実時間は、経過時間とも呼ばれます。 |
| ユーザー CPU 時間 | SAS コードの実行にかかった CPU 時間。 |
| システム CPU 時間 | SAS コードの実行をサポートするオペレーティングシステムタスク(システムオーバーヘッドタスク)の実行にかかった CPU 時間。 |
| メモリ | ステップの実行に必要なメモリの量。 |
| OS メモリ | システムから要求されたステップの実行に必要なメモリの最大量。 |

*注:* SAS 9 以降では、一部のプロシジャはマルチスレッドを使用します。複数の CPU を使用するコンピュータでは、オペレーティングシステムで同時に複数のスレッドを実行できます。したがって、CPU 時間は FULLSTIMER 出力の実時間を超える場合があります。

たとえば、SAS プロシジャでは、2 つの異なる CPU で同時に 2 つのスレッドを実行できます。CPU 時間の値は、次のように計算します。

```
CPU1 time + CPU2 time = total CPU time
1 second + 1 second = 2 seconds
```

CPU1 でスレッドを実行し、同時に、CPU2 で同一の SAS プロセスの別のスレッドを実行できるため、理論上は、実時間 1 秒あたり 2CPU 秒が費やされます。

## 比較

FULLSTIMER システムオプションでは、参照できるすべてのパフォーマンス統計量を SAS ログに書き出すかどうかを指定します。STIMER システムオプションでは、DATA ステップまたは PROC ステップの経過時間による統計量を SAS ログに書き出すかどうかを指定します。

## 関連項目:

- “STIMER システムオプション Windows” (574 ページ)
- 12 章: “Optimizing System Performance” (*SAS Language Reference: Concepts*)

- “The SAS Log” (*SAS Language Reference: Concepts* 9 章)

## HELPHOST システムオプション:Windows

リモートブラウジングシステムを表示するコンピュータ名を指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**デフォルト:** NULL

**参照項目:** “HELPHOST System Option” (*SAS System Options: Reference*)

### 比較

HELPHOST の値は、リモートデスクトップクライアントを実行しているコンピュータのアドレスに基づいて設定します。ホストにログインすると、エラーメッセージが表示されます。

## HELPINDEX システムオプション:Windows

SAS ヘルプとドキュメントに対する 1 つ以上のインデックスファイルを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:ヘルプ

**PROC OPTIONS GROUP=** HELP

**デフォルト:** /help/common.hlp/index.txt、/help/common.hlp/keywords.htm、common.hhk

**Windows 固有:** *HTML-HELP-index-pathname*

### 構文

**-HELPINDEX** <(> "*index-pathname-1*" < " *index-pathname-2*" " *index-pathname-n*")>

#### *構文記述*

***index-pathname***

SAS ヘルプとドキュメントで使用されるインデックスの一部パス名を指定します。*index-pathname* は、Windows の有効なパス名である必要があります。パス名は、引用符で囲む必要があります。複数のパス名を指定する場合は、パス名をスペースで区切り、パス名のリストをカッコで囲みます。

*index-pathname* には、次のいずれかまたはすべてのパス名を指定することができます。

**/help/*applet-index-filename***

UNIX 環境の SAS ヘルプとドキュメントの Java アプレットで使用されるインデックスファイルの一部パス名を指定します。*applet-index-filename* は、ファイル拡張子.txt を持ち、HELPLOC システムオプションで指定されたパスに存在する必要があります。デフォルトのパス名は、/help/common.hlp/index.txt です。

インデックスファイルに要求される形式については、デフォルトのインデックスファイルを参照してください。

**/help/*accessible-index-filename***
UNIX、OpenVMS または z/OS 環境の SAS ヘルプとドキュメントで使用される、アクセス可能なインデックスファイルの一部パス名を指定します。アクセス可能なインデックスファイルとは、Web ブラウザで使用可能な HTML ファイルです。*accessible-index-filename* は、ファイル拡張子.htm を持ち、HELPLOC システムオプションで指定されたパスに存在する必要があります。デフォルトのパス名は、/help/common.hlp/keywords.htm です。

インデックスファイルに要求される形式については、デフォルトのインデックスファイルを参照してください。

***HTML-Help-index-pathname***
Windows 環境の SAS ヘルプとドキュメントで使用される、Microsoft HTML Help のインデックスファイルのパス名を指定します。デフォルトのパス名は、common.hhk です。Microsoft HTML Help のキーワードの作成に関する情報については、Microsoft HTML Help のドキュメントを参照してください。

## 詳細

HELPINDEX オプションは、SAS から提供されるインデックスファイルの代わりに、カスタマイズされたインデックスファイルを使用する場合に使用します。複数の動作環境で 1 つの構成ファイルを使用して SAS を起動する場合は、すべての相対パス名を HELPINDEX オプションで指定することができます。指定できるパス名は各タイプにつき 1 つですが、パス名の順序は重要ではありません。

HELPINDEX オプションで UNIX、OpenVMS または z/OS 動作環境のパス名を指定すると、一部パス名に含まれる**`/help/`**が HELPLOC オプションで指定されたパス名に置換され、完全パスが決定されます。HELPLOC オプションに複数のパス名が含まれている場合は、パスごとに指定されたキーワードファイルが検索されます。

たとえば、HELPINDEX の値が**`/help/common.hlp/myindex.htm`** で、HELPLOC の値が**`/u/myhome/myhelp`** の場合、インデックスの完全なパスは**`/u/myhome/myhelp/common.hlp/myindex.htm`** になります。

## 関連項目:

“HELPLOC システムオプション:Windows” (516 ページ)

# HELPLOC システムオプション:Windows

Microsoft HTML Help を使用して SAS ヘルプとドキュメントを表示する場合に使用されるヘルプファイルの場所を指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | 環境制御:ヘルプ |
| **PROC OPTIONS GROUP=** | HELP |
| **デフォルト:** | (“!MYSASFILES\classdoc” “!*sasroot*\nls\en\help” “!*sasroot*\core\help”) |
| **Windows 固有:** | 次に対する有効な値:*pathname* |

## 構文

**-HELPLOC** <(> "*pathname-1*" <,*pathname-2*" "*pathname-n*")>

### 構文記述

***pathname***
SAS ヘルプとドキュメントのファイルが保存されている、1 つ以上のディレクトリパス名を指定します。*パス名*には、インストールされた Microsoft HTML ヘルプファイルが格納されている、有効な Windows のパス名にする必要があります。パス名は、引用符で囲む必要があります。複数のパス名を指定する場合は、パス名のリスト全体をかっこで囲む必要があります。

## 詳細

### HELPLOC の値の指定

HELPLOC システムオプションに指定した値は、値の連結リストの先頭に追加されます。この結果、SAS ヘルプとドキュメントへのアクセスを維持したまま、サイト独自のヘルプ情報にアクセスすることができます。

SAS/AF アプリケーションのヘルプおよび SAS ヘルプとドキュメントで使用するデフォルトのフォルダは、それぞれ!MYSASFILES\classdoc と!*sasroot*\core\help となります。

APPEND と INSERT システムオプションを使ってファイル指定を追加することができます。

### 例

次のコマンドには、HELPLOC の指定が 2 つ含まれています。

```
sas -helploc "c:\app1\help" -helploc "c:\app2\help"
```

システムオプションの値は、次のようになります。

```
("c:\app2\help" "c:\app1\help" "!sasuser\classdoc" "!sasroot\nls\en\help"
"!sasroot\core\help")
```

## 関連項目:

- “SAS 自動実行ファイル ” (20 ページ)
- “APPEND= System Option” (*SAS System Options: Reference*)
- “INSERT= System Option” (*SAS System Options: Reference*)

# HELPREGISTER システムオプション:Windows

メイン SAS ウィンドウのヘルプメニューからアクセスするヘルプファイルを登録します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | 環境制御:ヘルプ |
| **PROC OPTIONS GROUP=** | HELP |
| **デフォルト:** | なし |
| **Windows 固有:** | すべて |

## 構文

**-HELPREGISTER** "*menu string*" *help file location* <"*help string*"> <*topic*>
<CHM | HLP | HTML>

### *構文記述*

**"*menu string*"**
**ヘルプ**メニューに表示するテキスト文字列です。

***help file location***
ヘルプファイルが格納されているフォルダとファイル名を指定します。*help file location* は、ファイルが HELPLOC システムオプションによって指定されているフォルダに格納されている場合は指定する必要はありません。また、*help file location* は、!*sasroot* を使用して省略されます。*help file location* にスペースが含まれる場合は、引用符で囲む必要があります。

***"ヘルプ文字列"***
*メニュー文字列*にマウスを置くと、ステータスバーに表示されるテキストです。

***topic***
SAS ヘルプメニューから *menu string* (メニュー文字列)を選択したときに表示されるヘルプファイル内のトピックです。HTML ファイルの場合、トピックはドキュメント内のアンカー(#で始まる)になります。CHM ファイルの場合、トピックは CHM ファイル内のページです。HLP ファイルの場合、トピックは、WinHelp が検索するファイル内のキーワードです。*topic* にスペースが含まれる場合は、引用符で囲む必要があります。

**CHM**
ローカルシステムまたはネットワーク上の HtmlHelp の CHM ファイルを指定します。

**HLP**
ローカルシステムまたはネットワーク上の WinHelp ファイルを指定します。

**HTML**
ローカルファイルシステムまたはネットワーク上の HTML ファイル、あるいは有効な URL を指定します。

## 詳細

HELPREGISTER システムオプションを使用すると、メイン SAS ウィンドウのヘルプメニューからアクセスできるようにするヘルプファイルを 20 個まで指定することができます。スペースを含む文字列はすべて、二重引用符で囲む必要があります。任意指定の引数は、1 個のピリオド(.)または空の二重引用符("")で置き換えることにより省略できます。次に引数を指定する必要がない場合は、プレースホルダーは必要ありません。

**ヘルプ**メニューに複数のヘルプファイルを追加するには、構成ファイルまたは SAS 起動時のコマンドプロンプトで、複数の HELPREGISTER システムオプションを指定します。

## 例

### *例 1: HTML ページと URL*

```
sas -helpregister "SAS Institute Inc" http://www.sas.com "SAS's homepage on the web" . h
```

### 例 2: HTML ヘルプファイル (.CHM)

```
sas -helpregister "My CHM file" \\server\share\HelpStuff.chm ."InternalFile.htm" sas -he
```

### 例 3: WinHelp ファイル (.HLP)

```
sas -helpregister "A WinHelp File" c:\somefile.hlp "simply an .hlp file" sas -helpregist
```

## 関連項目:

"ヘルプメニューにヘルプを追加" (65 ページ)

# HELPTOC システムオプション:Windows

SAS ヘルプとドキュメントで使用される目次ファイルを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:ヘルプ

**PROC OPTIONS GROUP=** HELP

**デフォルト:** /help/common.hlp/index.txt、/help/common.hlp/keywords.htm、common.hhk

**Windows 固有:** *HTML-Help-TOC-pathname*

## 構文

**-HELPTOC** <(> "*TOC-pathname-1*" <"*TOC-pathname-2*" " *TOC-pathname-3*")>

### 構文記述

***TOC-pathname***
SAS ヘルプとドキュメントで使用される、目次ファイルの一部パス名を指定します。*TOC-pathname* には、有効な Windows のパス名を指定する必要があります。パス名は、引用符で囲む必要があります。複数のパス名を指定する場合は、パス名のリスト全体をかっこで囲む必要があります。

*TOC-pathname* には、次のいずれかまたはすべてとなります。

**/help/*applet-TOC-filename***
UNIX 環境の SAS ヘルプとドキュメントの Java アプレットで使用される目次ファイルの一部パス名を指定します。*applet-index-filename* は、ファイル拡張子.txt を持ち、HELPLOC システムオプションで指定されたパスに存在する必要があります。デフォルトのパス名は、/help/helpnav.hlp/config.txt です。

索引ファイルに要求される形式については、デフォルトの目次ファイルを参照してください。

**/help/*accessible-TOC-filename***
UNIX、OpenVMS または z/OS 環境の SAS ヘルプとドキュメントで使用される、アクセス可能な目次ファイルの一部パス名を指定します。アクセス可能な目次ファイルとは、Web ブラウザで使用可能な HTML ファイルです。*accessible-TOC-filename* は、ファイル拡張子.htm を持ち、HELPLOC システムオプションで指定されたパスに存在する必要があります。デフォルトのパス名は、/help/common.hlp/toc.htm です。

目次ファイルに要求される形式については、デフォルトの目次ファイルを参照してください。

*HTML-Help-TOC-pathname*
Windows 環境の SAS ヘルプとドキュメントで使用される、Microsoft HTML Help の目次ファイルの完全パス名を指定します。デフォルトのパス名は、common.hhc です。Microsoft HTML Help のキーワードの作成に関する情報については、Microsoft HTML Help のドキュメントを参照してください。

## 詳細

HELPTOC オプションは、SAS から提供される目次ファイルの代わりに、カスタマイズされた目次ファイルを使用する場合に指定します。複数の動作環境で 1 つの構成ファイルを使用して SAS を起動する場合は、すべての相対パス名を HELPTOC オプションで指定することができます。指定できるパス名は各タイプにつき 1 つですが、パス名の順序は重要ではありません。

HELPINDEX オプションで UNIX、OpenVMS または z/OS 動作環境のパス名を指定すると、一部パス名に含まれる`/help/`が HELPLOC オプションで指定されたパス名に置換され、完全パスが決定されます。HELPLOC オプションに複数のパス名が含まれている場合は、パスごとに指定された目次ファイルが検索されます。

たとえば、HELPINDEX の値が`/help/common.hlp/myindex.htm` で、HELPLOC の値が`/u/myhome/myhelp` の場合、目次の完全なパスは`/u/myhome/myhelp/common.hlp/myindex.htm` になります。

## 関連項目:

"HELPLOC システムオプション:Windows" (516 ページ)

# HOSTINFOLONG システムオプション:Windows

SAS 起動時に SAS ログの追加動作環境情報の記述を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** ログとプロシジャ出力制御:SAS ログ

**PROC OPTIONS GROUP=** LOGCONTROL

## 構文

**HOSTINFOLONG** | **NOHOSTINFOLONG**

### *構文記述*

**HOSTINFOLONG**
SAS 起動時に SAS ログの追加動作環境情報の記述を指定します。

**NOHOSTINFOLONG**
SAS 起動時に動作環についての追加境情報を SAS ログに記録しないことを指定します。

## 詳細

起動時に HOSTINFOLONG を指定すると、SAS は動作環境 についての追加情報を SAS ログに記録します。システムのデフォルト設定は HOSTINFOLONG です。次に、例を示します。

```
NOTE:Copyright (c) 2002-2012 by SAS Institute Inc., Cary, NC, USA.NOTE:SAS (r)
Proprietary Software 9.4 (TS1M0) Licensed to SAS Institute Inc., Site
1.NOTE:This session is executing on the W32_7PRO  platform.NOTE:Updated
analytical products:SAS/STAT 12.3 (maintenance) SAS/ETS 12.3 (maintenance)
SAS/OR 12.3 (maintenance) SAS/IML 12.3 (maintenance) SAS/QC 12.3 (maintenance)
NOTE:Additional host information:W32_7PRO DNTHOST 6.1.7601 Service Pack 1
Workstation NOTE:SAS initialization used:real time 1.42 seconds cpu time 0.70
seconds
```

## HOSTPRINT システムオプション:Windows

Windows プリントマネージャを印刷に使用するように指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** ログとプロシジャ出力制御:プロシジャ出力

**PROC OPTIONS GROUP=** LISTCONTROL

**デフォルト:** HOSTPRINT

**Windows 固有:** すべて

### 構文

**-HOSTPRINT** | **-NOHOSTPRINT**

**HOSTPRINT** | **NOHOSTPRINT**

#### *構文記述*

**HOSTPRINT**
Windows の印刷機能を使用するように指定します。これはデフォルトの設定です。

**NOHOSTPRINT**
SAS フォームを印刷に使用するように指定します。

### 詳細

NOHOSTPRINT オプションは、バッチ SAS セッションでフォームを印刷に使用する場合に指定します。NOHOSTPRINT を指定すると、**印刷設定**ダイアログボックスで**フォームを使用する**チェックボックスが選択され、SAS フォームで指定された文字数、行数、フォントの値が使用されます。

### 関連項目:

“フォーム使用関連の印刷オプションの設定” (176 ページ)

## ICON システムオプション:Windows

SAS ウィンドウを最小化します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** NOICON

**Windows 固有:** すべて

## 構文

**-ICON** | **-NOICON**

**ICON** | **NOICON**

### *構文記述*

**ICON**
メイン SAS ウィンドウをただちに最小化します。

**NOICON**
メイン SAS ウィンドウをただちに復元します。

## 詳細

SAS コマンドまたは SAS 構成ファイルで ICON システムオプションを設定した場合、SAS は初期化が終ると最小化されます。OPTIONS ステートメントで ICON システムオプションをサブミットすると、SAS はただちに最小化されます。これは、最小化ボタンをクリックすることと同じです。

このシステムオプションは、Windows の起動直後に SAS セッションを最小化して表示する場合に使用します。たとえば、SAS コマンドで ICON システムオプションを次のように指定できます。

```
c:\sas\sas.exe -icon
```

# INITSTMT システムオプション:Windows

autoexec ファイルのステートメントの実行後、または SYSIN=ファイルのステートメントの前に実行する SAS ステートメントを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:初期化と操作

**PROC OPTIONS GROUP=** EXECMODES

**別名:** IS

**デフォルト:** なし

**Windows 固有:** *拡張エディタを使用する場合は、DATA ステップまたは PROC ステップを終了する statement* を指定する必要があります。

**参照項目:** “INITSTMT= System Option” (*SAS System Options: Reference*)

## 構文

**INITSTMT** '*statement*'

### 必須引数

**'*statement*'**
1 つ以上の SAS ステートメントを指定します。拡張エディタを使用する場合は、DATA ステップまたは PROC ステップを終了する *statement* を指定する必要があります。

## INSERT システムオプション:Windows

SAS 起動時に使用して、指定したシステムオプションの先頭に値を挿入します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ
**PROC OPTIONS GROUP=** ENVFILES
**デフォルト:** なし
**Windows 固有:** SAS 起動構文

### 構文

**-INSERT** *system-option 引数*

### 必須引数

***system-option***
AUTOEXEC、CMPLIB、FMTSEARCH、HELPLOC、MAPS、MSG、SASSCRIPT、SASAUTOS、SASHELP、SET。

***引数***
*system-option* の現在の値の先頭に挿入する、別のパス名や環境設定を指定します。次に、FMTSEARCH オプションの前にライブラリを挿入する例を示します。

```
-set APFMTLIB "SASEnvironment/SASFormats"
-insert fmtsearch APFMTLIB
```

### 詳細

デフォルトでは、AUTOEXEC、CMPLIB、FMTSEARCH、HELPLOC、MAPS、MSG、SASSCRIPT、SASAUTOS、SASHELP または SET のシステムオプションを複数指定すると、最後に指定した値が使用されます。これらのオプションのいずれかによってすでに指定されている検索パスの先頭にパス名を挿入するには、INSERT または APPEND システムオプションを使用して新しいパス名を追加します。たとえば、次の SAS コマンドを入力すると、SAS が検索するヘルプファイルの唯一の場所は **c:\app2\help** になり、OPTIONS プロシジャの出力では **C:\app2\help** が表示されます。

```
sas -helploc "c:\app2\help"
```

SAS でヘルプファイルの現在のパスと **c:\app2\help** を両方検索し、そのうえで **c:\app2\help** を先に検索したい場合は、INSERT オプションを使用して HELPLOC オプションの値に新しい場所を追加します。

```
sas -insert helploc "c:\app2\help"
```

もし ヘルプファイルの現在のパスが **!sasroot\nls\en\help** の場合、HELPLOC オプションの値として OPTIONS プロシジャは次を示します。

```
("c:\app2\help" "!sasroot\nls\en\help")
```

### 関連項目:

- "Changing an Option Value By Using the INSERT and APPEND System Options" (*SAS System Options: Reference* 1 章)
- "APPEND= System Option" (*SAS System Options: Reference*)
- "INSERT= System Option" (*SAS System Options: Reference*)

## JREOPTIONS システムオプション:Windows

SAS で使用する Java Runtime Environment (JRE)オプションを特定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:初期化と操作

**PROC OPTIONS GROUP=** EXECMODES

**デフォルト:** -Djava.security.policy=<*pathname**filename*> -Dsas.jre=(private| public) -Dsas.jre.home=!*sasroot**pathname* -Djava.ext.dirs=*pathname**filename*

**Windows 固有:** すべて

**注意:** **SAS の動作に影響を与える Java オプションを変更すると、SAS が強制終了する場合があります。** JREOPTIONS オプションの設定を変更する前に、SAS の技術サポートに連絡し、予定している Java 設定の変更により SAS が強制終了しないことをあらかじめ確認することをお勧めします。自分の Java コードの Java プロパティだけを変更するのが、一番 安全です。

### 構文

**-JREOPTIONS** (-*JRE-option-1* <-*JRE-option-n*> )

JREOPTIONS (-*JRE-option-1* <-*JRE-option-n*> )

#### *構文記述*

***-JRE-option***
1 つ以上の JRE オプションを指定します。JRE オプションは、ハイフン( - )で始めます。複数の JRE オプションを区切るには、スペースを使用します。*JRE-option* の有効な値は、インストールされた Java ランタイム環境によって異なります。JRE オプションの詳細については、インストールされた Java のドキュメントを参照してください。

### 詳細

JRE オプションのセットを使用する場合は、かっこで囲みます。複数の JREOPTIONS システムオプションを指定すると、現在定義されている JRE オプションに新たな JRE オプションが追加されます。無効な JRE オプションは無視されます。

### 例: 例

- `-jreoptions \(-Dmy.java.property\)`

- `-jreoptions \(-Xmx512m -Xms256m\)`

## LINESIZE システムオプション:Windows

SAS のログウィンドウおよび出力ウィンドウの行サイズを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** ログとプロシジャ出力制御:SAS ログとプロシジャ出力

**PROC OPTIONS GROUP=** LOG_LISTCONTROL

**デフォルト:** 表示設定により異なります。

**Windows 固有:** デフォルト値

**参照項目:** “LINESIZE= System Option” (*SAS System Options: Reference*)

### 構文

**-LINESIZE** *n* | MIN | MAX

**LINESIZE=** *n* | MIN | MAX

#### *必須引数*

***n***
行サイズを文字数で指定します。64 - 256 を指定できます。

**MIN**
行サイズを 64 文字にします。

**MAX**
行サイズを 256 文字にします。

### 詳細

デフォルト値は、生成されるレポートが正しく出力されるように、プリンタの解像度とフォントに基づいて設定されます。

***注意:***
**Windows の印刷ダイアログボックスを使用して印刷オプションを変更すると、SAS の印刷システムオプションの値が変更され、これが原因で予想した出力が得られない可能性があります。** LINESIZE や PAGESIZE など、SAS システムオプションを使用して印刷オプションを設定した後、Windows の印刷ダイアログボックスを使用して印刷オプションを設定した場合、SAS システムオプションは Windows の印刷ダイアログボックスに指定した値に設定されます。

### 関連項目:

- “PAGESIZE システムオプション:Windows” (542 ページ)
- “ORIENTATION= System Option” (*SAS System Options: Reference*)
- “PAPERSIZE= System Option” (*SAS System Options: Reference*)

## LOADMEMSIZE システムオプション:Windows

SAS がロードする実行プログラムに必要な推奨メモリ量を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** システム管理:メモリ

**PROC OPTIONS GROUP=** MEMORY

**デフォルト:** 0

**Windows 固有:** すべて

### 構文

**-LOADMEMSIZE***n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

#### *必須引数*

***n* | *n*K | *n*M | *n*G**
メモリサイズをそれぞれ 1、1,024 (キロバイト)、1,048,576 (メガバイト)および 1,073,741,842 (ギガバイト)の倍数で指定します。キロバイト、メガバイト、ギガバイト単位の数は、10 進値で指定することができます。たとえば、`8` はバッファ数 8、`.782k` はバッファ数 801、`3m` はバッファ数 3,145,728 を示します。

***hex*X**
メモリ量を、16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、値 `2dx` は、メモリ量を 45 バイトに設定します。

**MIN**
0 バイトを意味します。使用可能なメモリの総量は制限されません。

**MAX**
実行プログラムの最大メモリ量は、使用可能なメモリ量までとなります。

### 詳細

LOADMEMSIZE を 0 に設定すると、SAS がロードした実行プログラムに使用されるメモリ量は、使用可能なシステムメモリ量までとなります。LOADMEMSIZE を 1 に設定すると、実行プログラムは使用されなくなるとメモリから削除されます。

値が 2 以上の場合は、SAS はまず SAS 実行プログラムで使用可能なメモリ量を確認します。使用可能なメモリの総量が LOADMEMSIZE の値より大きい場合は、SAS がロードした実行プログラムから使用されていないプログラムが削除されます。この削除処理は、使用されるメモリ量が LOADMEMSIZE オプションの値より小さくなるまで、または SAS がロードした実行プログラムのうち削除可能な実行プログラムをすべて削除するまで続けられます。すべての実行プログラムを削除した後にメモリの追加が必要な場合、使用できるシステムメモリがあれば追加されます。

## LOG システムオプション:Windows

SAS をバッチモードで実行している場合の SAS ログのコピー先を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES、LOGCONTROL

**デフォルト:** *filename*.LOG バッチモード上では *filename* は SAS ジョブの名前となります。

**Windows 固有:** *destination* は、有効な Windows ファイル名である必要があります。

## 構文

**-LOG** "*destination*" | -NOLOG

### *必須引数*

**LOG "*destination*"**
SAS ログの出力先を指定します。*destination* 引数には、Windows の有効なパス名やファイル名(LPT1 のようなデバイス名を含む)またはパス名を割り当てた環境変数を指定できます。パス名だけを指定した場合、指定されたディレクトリに *filename*.LOG というデフォルトのファイル名でログファイルが作成されます。*filename* は、対応する SAS ジョブ名です。

**NOLOG**
各ログメッセージをメッセージボックスに送ります。各メッセージボックスに、1 つのメッセージが表示されます。

## 詳細

LOG システムオプションは、バッチモードで実行している場合の SAS ログの書き込み先を指定します。

このシステムオプションは、バッチモードでのみ有効です。

SAS を対話的に実行しているときには、ログは**ログ**ウィンドウに表示されます。バッチモードのときには、ログは現在の SAS 作業ディレクトリにある *filename.LOG* というファイルに保存されます。*filename* は、対応する SAS ジョブの名前です。LOG システムオプションを使用して、別の出力先を指定することができます。

SAS ログの表示を無効化するには、NOTERMINAL システムオプションを使用します。

SAS の起動時に OBJECTSERVER および NOTERMINAL システムオプションが指定され、ログが指定されていない場合、ログメッセージはすべて廃棄されます。

OBJECTSERVER および NOTERMINAL システムオプションを有効にして SAS が起動され、ログが指定されていない場合、ログやログメッセージはすべて破棄されます。

LOG システムオプションの値にディレクティブを使用すると、ログの表示/非表示のタイミング、ログの命名方法を、時間、月、曜日などのリアルタイムイベントに基づいて制御できます。ディレクティブのリストについては、*SAS システムオプション: リファレンス* の LOGPARM=システムオプションを参照してください。

SAS をバッチモードやサーバーモードで起動し、LOGCONFIGLOC=オプションが指定されている場合、SAS ログ機能によってログが行われます。従来の SAS ログオプション LOGPARM=は、無視されます。従来の SAS ログオプション LOG=は、%S{App.Log}変換文字がログ構成ファイルに指定されている場合のみ適用されます。詳細については、*SAS ログ機能: 構成とプログラミングリファレンス*を参照してください。

### 関連項目:

- “TERMINAL System Option” (*SAS System Options: Reference*)
- “LOGPARM= System Option” (*SAS System Options: Reference*)
- “The SAS Log” (*SAS Language Reference: Concepts* 9 章)

## MAPS システムオプション:Windows

SAS/GRAPH マップデータセットが保存されている SAS ライブラリの名前を指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** グラフィックス:ドライバ設定

**PROC OPTIONS GROUP=** GRAPHICS

**デフォルト:** !*sasroot*\maps

**Windows 固有:** デフォルト値および *location-of-maps* には、有効な Windows パス名を指定する必要があります。

**参照項目:** “MAPS= System Option” (*SAS/GRAPH: Reference*)

### 構文

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

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

#### *必須引数*

***location-of-maps***
ライブラリ参照名、有効な Windows パス名またはパス名を割り当てた環境変数を指定します。パス名は、ディレクトリまたはサブディレクトリのレベルまでであることに注意してください。パス名にスペースが含まれる場合は、パス名を引用符で囲みます。

### 詳細

#### *APPEND と INSERT システムオプション*

APPEND と INSERT システムオプションを使ってファイル指定を追加することができます。

### 関連項目:

- “SAS 自動実行ファイル ” (20 ページ)
- “APPEND= System Option” (*SAS System Options: Reference*)
- “INSERT= System Option” (*SAS System Options: Reference*)

## MAXMEMQUERY システムオプション:Windows

プロシジャに割り当てる最大メモリ量を指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** システム管理:メモリ

**PROC OPTIONS GROUP=** MEMORY

**デフォルト:** 0

**Windows 固有:** すべて

### 構文

**-MAXMEMQUERY***n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

**MAXMEMQUERY=***n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

#### *必須引数*

***n* | *n*K | *n*M | *n*G**
最大メモリ量をそれぞれ 1、1,024 (キロバイト)、1,048,576 (メガバイト)および 1,073,741,842 (ギガバイト)の倍数で指定します。キロバイト、メガバイト、ギガバイト単位の数は、10 進値で指定することができます。たとえば、`8` はバッファ数 8、`.782k` はバッファ数 801、`3m` はバッファ数 3,145,728 を示します。

***hex*X**
メモリ量を、16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、値 `2dx` は、メモリ量を 45 バイトに設定します。

**MIN**
メモリ量を最小値、つまり 0 バイトに設定します。これは、各プロシジャで使用可能なメモリの総量に制限がないことを示します。

**MAX**
メモリ量を最大値、つまり 2,147,483,647 バイトに設定します。

### 詳細

一部の SAS プロシジャでは、MAXMEMQUERY オプションを使用して、SAS が一度に要求できる仮想メモリの最大ブロックを指定します。一方、MEMSIZE オプションは、SAS が常時、動的に割り当てられる仮想メモリの総量に制限を設定します。仮想メモリは、実メモリとページングスペースを組み合わせてサポートされます。必要な仮想メモリの量が、利用可能な実メモリの量を超えると、動作環境でページングが開始されます。ページングを行わないようにしたり、関連するパフォーマンスの問題を回避するには、MAXMEMQUERY および MEMSIZE システムオプションに実メモリを超えない値を設定します。

## MEMBLKSZ システムオプション:Windows

Windows 動作環境で、メモリベースのライブラリに使用されるメモリのブロックサイズを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** システム管理:メモリ

**PROC OPTIONS GROUP=** MEMORY

**デフォルト:** 16MB

**Windows 固有:** すべて

## 構文

**-MEMBLKSZ** *n* | *n*K | *n*M | *n*G | *n*T | *hex*X

### *必須引数*

***n* | *n*K | *n*M | *n*G | *n*T**
メモリのブロックサイズをそれぞれ 1、1,024 (キロバイト)、1,048,576 (メガバイト)、1,073,741,824 (ギガバイト)および 1,099,511,627,776 (テラバイト)の倍数で指定します。キロバイト、メガバイト、ギガバイト、テラバイト単位の数は、10 進値で指定することができます。たとえば、`8` はバッファ数 8、`.782k` はバッファ数 801、`3m` はバッファ数 3,145,728 を示します。

***hex*X**
メモリのブロックサイズを、16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、値 `2dx` は、メモリのブロックサイズを 45 バイトに設定します。

## 詳細

Windows 2000 以降、メモリ内で同時に複数の処理を実行できるようになりました。MEMBLKSZ システムオプションの値は、最初に割り当てられるメモリ量です。追加メモリは、MEMBLKSZ オプションに指定したメモリサイズと同じサイズで割り当てられます。追加メモリの上限は、MEMMAXSZ オプションに指定されたメモリ量です。たとえば、MEMBLKSZ が 2M の場合、追加メモリとして 2M のブロックを割り当てることができます。

メモリベースのライブラリで拡張メモリが使用されている場合は、拡張メモリにアクセスするための処理アドレススペースの容量を決定するときにも、MEMBLKSZ オプションの値が使用されます。

*注:*

- 大きすぎる値を指定すると、システム全体のパフォーマンスに悪影響を及ぼす可能性があります。MEMBLKSZ にさまざまな値を指定し、システムパフォーマンスを最適化する値を決定してください。
- 32 ビット環境で拡張メモリを使用している場合、大きすぎる値を指定すると、SAS のパフォーマンスに悪影響を及ぼす可能性があります。値を小さくすることで、最適化される場合もあります。64K からはじめるのがお勧めします。そして MEMBLKSZ にさまざまな値を指定し、SAS システムパフォーマンスを最適化する値を決定してください。

## 関連項目:


- “メモリベースのライブラリ” (200 ページ)
- “MEMMAXSZ システムオプション:Windows” (532 ページ)

# MEMCACHE システムオプション:Windows

メモリベースのライブラリを、SAS のファイルキャッシュとして使用するように指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** ファイル:SAS ファイル

**PROC OPTIONS GROUP=** SASFILES

**デフォルト:** 0

**Windows 固有:** すべて

## 構文

**-MEMCACHE** 0 | 1 | 4

**MEMCACHE=** 0 | 1 | 4

### *必須引数*

**0**
メモリキャッシュ機能を無効化します。

**1**
新しいファイルをキャッシュに追加しないように指定します。キャッシュにあるファイルに対する読み書きは、MEMCACHE が有効化されている場合と同じように続けられます。

**4**
メモリキャッシュ機能を有効化します。メモリが、SAS のファイルキャッシュとして使用されます。

## 詳細

MEMCACHE システムオプションが 4 または 1 の場合、SAS のファイルキャッシュ機能により処理データはメモリに格納されます。一度データがメモリに格納されると、SAS からの参照が可能になります。キャッシュ内のファイルは、SAS またはキャッシュ機能が終了するまでか、または新規ファイル用にさらに領域が必要になるまで維持されます。メモリは、使用率が最も低い部分から再利用されます。キャッシュに格納されたデータは、ハードディスクに書き出されます。OPTIONS ステートメントの MEMCACHE システムオプションを使用して、キャッシュを使用できる SAS ライブラリを制御できます。メモリの使用状況は、パフォーマンスツールを使用して監視できます。

## 関連項目:

- “メモリベースのライブラリ” (200 ページ)
- “MEMLIB システムオプション:Windows” (532 ページ)

## MEMLIB システムオプション:Windows

WORK ライブラリを、メモリベースのライブラリとして処理するように指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** ファイル:SAS ファイル

**PROC OPTIONS GROUP=** SASFILES

**デフォルト:** NOMEMLIB

**Windows 固有:** すべて

### 構文

**-MEMLIB** | **-NOMEMLIB**

#### *必須引数*

**MEMLIB**
WORK ライブラリにメモリを使用します。

**NOMEMLIB**
メモリを使用しません。

### 詳細

MEMLIB システムオプションを指定すると、WORK ライブラリはメモリ内で処理されます。ファイルは、SAS が終了するまでか、またはファイルが削除されるまで、メモリに保存されます。メモリの使用状況は、パフォーマンスツールを使用して監視できます。

### 関連項目:

- “メモリベースのライブラリ” (200 ページ)
- “LIBNAME ステートメント:Windows” (460 ページ)
- “MEMCACHE システムオプション:Windows” (531 ページ)
- “パフォーマンスツール” (217 ページ)

## MEMMAXSZ システムオプション:Windows

Windows 動作環境で、メモリベースのライブラリの使用に割り当てる最大メモリ量を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** システム管理:メモリ

**PROC OPTIONS GROUP=** MEMORY

**デフォルト:** 2G

**Windows 固有:** すべて

## 構文

**-MEMMAXSZ** *n* | *n*K | *n*M | *n*G | *n*T | *hex*X

### *必須引数*

***n* | *n*K | *n*M | *n*G | *n*T**
割り当てるメモリ量をそれぞれ 1、1,024 (キロバイト)、1,048,576 (メガバイト)、1,073,741,824 (ギガバイト)およびまたは 1,099,511,627,776 (テラバイト)の倍数で指定します。キロバイト、メガバイト、ギガバイト、テラバイト単位の数は、10 進値で指定することができます。たとえば、`8` はバッファ数 8、`.782k` はバッファ数 801、`3m` はバッファ数 3,145,728 を示します。

***hex*X**
割り当てるメモリ量を、16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、値 `2dx` は、メモリ量を 45 バイトに設定します。

## 詳細

MEMMAXSZ システムオプションには、メモリベースのライブラリに使用可能なメモリの総量を指定します。メモリは、パフォーマンスツールを使用して監視できます。

***注意:***
**大きすぎる値を指定すると、システム全体のパフォーマンスに悪影響を及ぼす可能性があります。** MEMBLKSZ にさまざまな値を指定し、システムパフォーマンスを最適化する値を決定してください。

## 関連項目:

- “メモリベースのライブラリ” (200 ページ)
- “MEMBLKSZ システムオプション:Windows” (529 ページ)
- “MEMLIB システムオプション:Windows” (532 ページ)
- “MEMCACHE システムオプション:Windows” (531 ページ)

# MEMSIZE システムオプション:Windows

SAS セッション時に使用できる仮想メモリ量の上限を指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | システム管理:メモリ |
| **PROC OPTIONS GROUP=** | MEMORY |
| **デフォルト:** | 2G |
| **Windows 固有:** | 有効な値 |

## 構文

**-MEMSIZE** *n* | *n*K | *n*M | *n*G | *n*T | *hex*X | MAX

### *必須引数*

***n* | *n*K | *n*M | *n*G | *n*T**
メモリ量の上限を、バイト、キロバイト(1024 バイト)、メガバイト(1,048,576 バイト)、ギガバイト(1,073,741,824 バイト)、テラバイト(1,099,511,627,776 バイト)で指定します。たとえば、`0.25G` は 268,435,456 バイト、16.5M は 17,301,504 バイトを示します。

***hex*X**
メモリ量を、16 進数で指定します。数字(0-9)で始まり、16 進数(0-9、A-F)、X の順に続く値を指定します。たとえば、`0F00000x` により、MEMSIZE オプションの値は 15,728,640 バイトに設定されます。これは値として次を指定したことに相当します。`0x`

**MAX**
SAS 起動時に利用可能な物理メモリの量とページングスペースから、適切な最大値を指定します。

## 詳細

MEMSIZE システムオプションでは、各 SAS セッションで使用できるメモリの総量を指定します。指定した値が小さすぎると、メモリ不足になります。

数値 0 (または 0x)を設定すると、オプションに MAX を設定したことになります。

不当に小さい数値(6K など)を指定すると、MEMSIZE オプションの設定値は、SAS の起動が可能で基本的な機能を使用できる最小値になるよう、自動的に増加されます。

32 ビットバージョンの SAS に 4,294,967,295 を超える数値を指定すると、設定値が 4,294,967,295 に自動的に削減されます。

9,223,372,036,854,775,807 バイトを超える数値は、無効な値として拒否され、SAS を起動できません。

MEMSIZE システムオプションに指定したメモリ量を、SAS は自動的に予約したり、割り当てたりすることはありません。SAS は、プロセスの実行に必要なメモリのみを使用します。たとえば、MEMSIZE に 500M が設定されていても、DATA ステップに 20M のメモリしか必要なければ、SAS は 20M のメモリのみを使用します。

SAS ジョブの実行時に、システム監視ツールを使用して、メモリの設定値が大きい場合の影響を監視できます。

*注:* SAS の起動後に大量のメモリを消費するオブジェクトが有効化されない場合にのみ、MEMSIZE を MAX に設定することに意味があります。たとえば、複数の SAS のインスタンスが同時実行中であり、MEMSIZE が MAX に設定されてすべて起動されている場合、メモリ不足の状態に陥るインスタンスが発生したり、オペレーティングシステムに利用可能なページングスペースがなくなる可能性があります。

# MSG システムオプション:Windows

SAS エラーメッセージが格納されているライブラリを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES

**デフォルト:** !*sasroot*\core\sasmsg

**Windows 固有:** 次に対する有効な値:*library-specification*

## 構文

**-MSG** *library-specification*

### *必須引数*

***library-specification***
Windows の論理名(検索文字列も含む)やパス名を指定します。ファイル名は使用しないでください。パス名にスペースが含まれる場合は、パス名を引用符で囲みます。

## 詳細

MSG システムオプションは、SAS エラーメッセージが格納されているライブラリの名前を指定します。

APPEND と INSERT システムオプションを使ってファイル指定を追加することができます。

## 関連項目:

- “SAS 自動実行ファイル ” (20 ページ)
- “APPEND= System Option” (*SAS System Options: Reference*)
- “INSERT= System Option” (*SAS System Options: Reference*)

# MSGCASE システムオプション:Windows

SAS によって生成される説明、警告、エラーメッセージを大文字で表示するかどうかを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** ログとプロシジャ出力制御:SAS ログ

**PROC OPTIONS GROUP=** LOGCONTROL

**デフォルト:** NOMSGCASE

**Windows 固有:** すべて

## 構文

**-MSGCASE** | **-NOMSGCASE**

### *構文記述*

**MSGCASE**
メッセージを大文字で表示します。

**NOMSGCASE**
メッセージに大文字と小文字を含めることができます。これはデフォルトの設定です。

## 詳細

SAS によって生成される説明、警告、エラーメッセージを大文字で表示するかどうかを指定します。MSGCASE オプションの設定は、ユーザーが作成するメッセージおよびソース行には影響を与えません。

MSGCASE は NL フォーマットにてサポートされています。NL フォーマットの詳細については、*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。

## 関連項目:

"The SAS Log" (*SAS Language Reference: Concepts* 9 章)

# MSYMTABMAX システムオプション:Windows

マクロ変数のシンボルテーブルに割り当てるメモリの最大量を指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** マクロ:SAS マクロ

**PROC OPTIONS GROUP=** MACRO

**デフォルト:** 4194304 bytes (4MB)

**Windows 固有:** デフォルト値

**参照項目:** MSYMTABMAX、*SAS マクロ言語: リファレンス*

## 構文

**-MSYMTABMAX** *n* | *n*K | *n*M | *n*G | *n*T | *hex*X | MIN | MAX

**MSYMTABMAX**=*n* | *n*K | *n*M | *n*G | *n*T | *hex*X | MIN | MAX

### *必須引数*

***n* | *n*K | *n*M | *n*G | *n*T**
使用可能なメモリ量をそれぞれ 1、1,024 (キロバイト)、1,048,576 (メガバイト)、1,072,741,824 (ギガバイト)および 1,099,511,627,776 (テラバイト)の倍数で指定します。キロバイト、メガバイト、ギガバイト、テラバイト単位の数は、10 進値で指定することができます。たとえば、`8` はバッファ数 8、`.782k` はバッファ数 801、`3m` はバッファ数 3,145,728 を示します。

***hex*X**
使用可能なメモリ量を、16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、値 `2dx` は、メモリ量を 45 バイトに設定します。

**MIN**
使用可能なメモリ量を最小値、つまり 0 バイトに設定します。この場合、マクロ変数はすべてディスクに書き込まれます。

**MAX**
使用可能なメモリ量を最大値に設定します。

## 詳細

MSYMTABMAX の値に達すると、追加のマクロ変数はすべてディスクに書き込まれます。

# MVARSIZE システムオプション:Windows

メモリ内マクロ変数の最大サイズを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** マクロ:SAS マクロ

**PROC OPTIONS GROUP=** MACRO

**デフォルト:** 4096 bytes

**Windows 固有:** デフォルト値

**参照項目:** MVARSIZE システムオプション、*SAS マクロ言語: リファレンス*

## 構文

**-MVARSIZE** *n* | *n*K | *hex*X | MIN | MAX

**MVARSIZE**= *n* | *n*K | *hex*X | MIN | MAX

### *必須引数*

***n* | *n*K**
マクロ変数の最大サイズをそれぞれ、1 または 1,024 (キロバイト)の倍数で指定します。キロバイト単位の数は、10 進値で指定することができます。たとえば、`8` はバッファ数 8、`.782k` はバッファ数 801、`3km` はバッファ数 3,072 を示します。

***hex*X**
マクロ変数の最大サイズを、16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、値 `2dx` は、マクロ変数の最大サイズを 45 バイトに設定します。

**MIN**
マクロ変数のサイズを最小値、つまり 0 バイトに設定します。この場合、マクロ変数はすべてディスクに書き込まれます。

**MAX**
マクロ変数のサイズを最大値、つまり 65,534 バイトに設定します。

## 詳細

MVARSIZE システムオプションは、メモリに保存されるマクロ変数の最大サイズを指定します。指定した最大値より大きいサイズのマクロ変数は、ディスクに書き出されます。

MVARSIZE システムオプションの値は、システムパフォーマンスに影響を与えることがあります。プロダクション用の値を指定する前に、テストを実行して最適な値を決めます。

## NEWS システムオプション:Windows

SAS ログに書き込まれるメッセージが保存されているファイルを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES

**デフォルト:** なし

**Windows 固有:** 次に対する有効な値:*file-specification*

**参照項目:** “NEWS= System Option” (*SAS System Options: Reference*)

### 構文

**-NEWS** *file-specification*

#### *必須引数*

***file-specification***
外部ファイルを指定します。*file-specification* の値には、Windows の有効なパス名またはショートカット名を指定します。パス名にスペースが含まれる場合は、パス名を引用符で囲みます。

### 詳細

NEWS ファイルには、SAS に関するニュース項目など、ユーザー向けの情報が格納されています。指定したファイルの内容は、SAS ログのヘッダーの直後に表示されます。

## NUMKEYS システムオプション:Windows

使用可能なファンクションキーの数を制御します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 入力制御:データ処理

**PROC OPTIONS GROUP=** INPUTCONTROL

**デフォルト:** キーボードのファンクションキーの数

**Windows 固有:** すべて

### 構文

**-NUMKEYS** *number-of-keys*

### 必須引数

***number-of-keys***
アクティブにするキーボード上のファンクションキーの数を指定します。

## 詳細

SAS の初期化時に、キーボード上のファンクションキーの数が調査されます。NUMKEYS システムオプションで、別の値を指定することにより、この設定を無効にすることができます。

## 例

次のように指定した場合、**KEYS** ウィンドウに 10 個のファンクションキーが表示されます。

```
-numkeys 10
```

# NUMMOUSEKEYS システムオプション:Windows

**KEYS** ウィンドウに表示されるマウスボタンの数を指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | 入力制御:データ処理 |
| **PROC OPTIONS GROUP=** | INPUTCONTROL |
| **デフォルト:** | 3 ボタン |
| **Windows 固有:** | すべて |

## 構文

**-NUMMOUSEKEYS** *number-of-buttons*

### 必須引数

***number-of-buttons***
マウスのボタンの数を指定します。0 から 3 を指定できます。*number-of-buttons* が 0 または 1 の場合、KEYS ウィンドウにマウスボタンは表示されません。これは唯一のボタンである左マウスボタンは、SAS によって予約されているためです。*number-of-buttons* が 2 の場合、**KEYS** ウィンドウには右マウスボタン(RMB)、Ctrl+右マウスボタンおよび Shift+右マウスボタンが表示されます。*number-of-buttons* が 3 の場合、**KEYS** ウィンドウには右マウスボタンと中央マウスボタンの両方が表示されます。

## 詳細

NUMMOUSEKEYS システムオプションを指定しないと、3 つのボタンがあるマウスを使用していると見なされます。マウスボタンの数が 1 個または 2 個のマウスを使用していることを **KEYS** ウィンドウに反映するには、SAS 構成ファイルに NUMMOUSEKEYS システムオプションを指定します。

## OBS システムオプション:Windows

どこまでのオブザベーション(レコード)を処理するか指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** ファイル:SAS ファイル

**PROC OPTIONS GROUP=** SASFILES

**デフォルト:** MAX

**Windows 固有:** 有効な範囲

**参照項目:** “OBS= System Option” (*SAS System Options: Reference*)

### 構文

**-OBS** *n* | *n*K | *n*M | *n*G | *n*T | *hex*X | MIN | MAX

**OBS**=*n* | *n*K | *n*M | *n*G | *n*T | *hex*X | MIN | MAX

#### *必須引数*

***n* | *n*K | *n*M | *n*G | *n*T**
処理対象となる最大のオブザベーション番号を指定します。*n* は整数を表します。次のいずれかの文字表記を使用すると、指定した整数に特定の値が掛けられます。つまり K (キロ)を指定すると整数に 1,024 が掛けられます。同様に M (メガ)では 1,048,576、G (ギガ)では 1,073,741,824、T (テラ)では 1,099,511,627,776 が掛けられます。*n* を使用して K、M、G、T の値を指定する場合は、n に 10 進値を指定することができます。たとえば、`20` はオブザベーション(レコード)番号 20、`.782k` はオブザベーション(レコード)番号 801、`3m` はオブザベーション(レコード)番号 3,145,728 を示します。

***hex*X**
処理対象となる最大のオブザベーション番号を、16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、10 進値の 248 に相当する 16 進値の F8 を指定する場合は、`0F8X` として指定する必要があります。16 進値の `2dx` は、10 進値の 45 を示します。

**MIN**
処理する最大のオブザベーション番号を 0 に設定します。

**MAX**
処理する最大のオブザベーション番号を 2,147,483,647 に設定します。64 ビットシステムでは、最大値は 9,223,372,036,854,775,807 です。MAX は、デフォルトの設定です。

## OPLIST システムオプション:Windows

SAS システムオプションの設定を SAS ログに表示するには、OPTIONS プロシジャを使用します。

**該当要素:** 構成ファイル、SAS 起動、SASV9_OPTIONS 環境変数

**カテゴリ:** ログとプロシジャ出力制御:SAS ログ

**PROC OPTIONS GROUP=** LOGCONTROL

**デフォルト:** NOOPLIST

**Windows 固有:** すべて

## 構文

**-OPLIST** | **-NOOPLIST**

## 詳細

OPLIST システムオプションはコマンド行に指定されたシステムオプションのみ表示します。構成ファイルまたは SASV9_OPTIONS 環境変数に指定されたシステムオプションは表示しません(構成ファイルの内容を表示したい場合は、VERBOSE オプションを使用します)。たとえば、SAS をコマンド `sas -nodms -fullstimer -nonews -oplist` で起動します。SAS は、次の行を SAS ログに記録します。`NOTE:SAS command line:-nodms -fullstimer -nonews -oplist`

# PAGENO システムオプション:Windows

ページ番号をリセットします。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** ログとプロシジャ出力制御:プロシジャ出力

**PROC OPTIONS GROUP=** LISTCONTROL

**デフォルト:** 1

**Windows 固有:** 次に対する有効な値:*n*; 構文

**参照項目:** “PAGENO= System Option” (*SAS System Options: Reference*)

## 構文

**-PAGENO** *n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

**PAGENO=***n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

### *必須引数*

***n* | *n*K | *n*M | *n*G**
ページ番号をそれぞれ、1 (*n*)、1,024 (*n*K)、1,048,576 (*n*M)、1,073,741,824 (*n*G)の倍数で指定します。*n* を使用して K、M、G、T の値を指定する場合は、n に 10 進値を指定することができます。たとえば、`8` はページ番号を 8 に設定し、`.782k` はページ番号を 801 に設定し、`3k` はページ番号を 3,072 に設定します。

***hex*X**
ページ番号を 16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、値 2dx はページ番号を 45 に設定します。

**MIN**
ページ番号を最小値、つまり 1 に設定します。

**MAX**
ページ番号を最大値、つまり 2,147,483,647 に設定します。

## 詳細

PAGENO システムオプションは、SAS が次に出力するページを、指定したページ番号で開始するようにリセットします。

# PAGESIZE システムオプション:Windows

SAS 出力のページ構成行数を指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ
**カテゴリ:** ログとプロシジャ出力制御:SAS ログとプロシジャ出力
**PROC OPTIONS GROUP=** LOG_LISTCONTROL
**デフォルト:** 表示設定により異なります。
**Windows 固有:** デフォルト値
**参照項目:** “PAPERSIZE= System Option” (*SAS System Options: Reference*)

## 構文

**-PAGESIZE** *n* | MIN | MAX

**PAGESIZE**=*n* | MIN | MAX

### *必須引数*

***n***
ページの構成行数を指定します。

**MIN**
1 ページを最小値、つまり 15 行に設定します。

**MAX**
1 ページを最大値、つまり 32,767 行に設定します。

## 詳細

Windows 環境の場合、デフォルト値は、生成されるレポートが正しく出力されるように、プリンタの解像度とフォントに基づいて設定されます。

***注意:***
**Windows の印刷ダイアログボックスを使用して印刷オプションを変更すると、SAS の印刷システムオプションの値が変更され、これが原因で予想した出力が得られない可能性があります。** LINESIZE や PAGESIZE など、SAS システムオプションを使用して印刷オプションを設定した後、Windows の印刷ダイアログボックスを使用して印刷オプションを設定した場合、SAS システムオプションは Windows の印刷ダイアログボックスに指定した値に設定されます。

## 関連項目:

“LINESIZE システムオプション:Windows” (525 ページ)

## PAPERTYPE システムオプション:Windows

印刷に使用する用紙の種類をプリンタに指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** ログとプロシジャ出力制御:ODS 出力

**PROC OPTIONS GROUP=** ODSPRINT

**デフォルト:** PLAIN

**Windows 固有:** 有効な値

**参照項目:** “PAPERTYPE= System Option” (*SAS System Options: Reference*)

### 構文

**-PAPERTYPE** PLAIN | STANDARD | GLOSSY | TRANSPARENCY | *printer-defined-value*

**PAPERTYPE**=PLAIN | STANDARD | GLOSSY | TRANSPARENCY | *printer-defined-value*

#### *必須引数*

**PLAIN**
普通紙を使用します。

**STANDARD**
プリンタの標準の用紙を使用します。

**GLOSSY**
光沢紙を使用します。

**TRANSPARENCY**
透明な用紙を使用します。

***printer-definded-value***
プリンタで定義された用紙の種類を指定します。

### 詳細

プリンタで使用できる用紙の種類については、プリンタのドキュメントを参照してください。

## PATH システムオプション:Windows

SAS 実行ファイルを検索するパスを 1 つ以上指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** システム管理:インストール

**PROC OPTIONS GROUP=** INSTALL

**デフォルト:** !*sasroot*\.core\sasexe

**Windows 固有:** すべて

## 構文

**-PATH** <(>"*directory-specification-1*" <"*directory-specification-n*")>

### *構文記述*

***directory-specification***
検索するパスを指定します。*directory-specification* の値には、有効な Windows パス名またはパス名を割り当てた環境変数を指定する必要があります。パス名にスペースが含まれる場合は、パス名を引用符で囲む必要があります。複数の *directory-specification* を指定する場合は、*directory-specification* のリストをかっこで囲みます。

## 詳細

PATH オプションでは、SAS 実行ファイルを検索するパスを指定します。PATH オプションを複数指定すると、検索順序を定義することができます。パスは SAS が実行する順番で検索されます。したがって、使用頻度の最も高いプロダクトのパスをリストの最初に指定します。

# PFKEY システムオプション:Windows

ファンクションキーの基本セットを指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | 入力制御:データ処理 |
| **PROC OPTIONS GROUP=** | INPUTCONTROL |
| **デフォルト:** | WIN |
| **Windows 固有:** | すべて |

## 構文

**-PFKEY** PRIMARY | ALTERNATE | SAA | WIN

### *必須引数*

**PRIMARY**
F1 から F12 をメインフレームの PF1 から PF12 に対する基本の設定にマップし、Shift+F1 から Shift+F12 を PF13 から PF24 にマップします。右マウスボタン(RMB)は MB2 にマップされます。ファンクションキーが 10 個のみの場合、F11、F12、Shift+F11 および Shift+F12 は使用できず、**KEYS** ウィンドウに表示されません。

次の表は、メインフレームの基本のキー定義を示しています。

*表 22.3 主メインフレームキー定義*

| PC のキー | メインフレーム定義 | キー | メインフレーム定義 |
|---|---|---|---|
| F1 | 選択 | Shift+F1 | ヘルプ |
| F2 | smark | Shift+F2 | ズーム |
| F3 | 選択解除 | Shift+F3 | 縮小;サブミット |
| F4 | 削除 | Shift+F4 | pgm; リコール |
| F5 | 貼り付け | Shift+F5 | rfind |
| F6 | 保存 | Shift+F6 | rchange |
| F7 | prevwind | Shift+F7 | 後ろへ |
| F8 | 次 | Shift+FF8 | 前へ |
| F9 | pmenu | Shift+F9 | 出力 |
| F10 | コマンド | Shift+F10 | 左揃え |
| F11 | キー | Shift+F11 | 右揃え |
| F12 | 取り消し | Shift+F12 | ホーム |
| RMB | 縮小;サブミット | | |

**ALTERNATE**

F1 から F12 を、メインフレームの代替のキー設定にマップします。つまり、F1 - F12 を PF13 - PF24 にマップします。結果として F1 - F12 は、Shift+F1 - Shift+F12 と同じになります。右マウスボタン(RMB)は MB2 にマップされます。ファンクションキーが 10 個のみの場合、F11、F12 は使用できず、**KEYS** ウィンドウに表示されません。キーボードに 24 個ではなく 12 個のファンクションキーがある場合、F13 から F24 は F1 から F12 にマップされます。

次の表は、メインフレームの代替のキー定義を示しています。

*表 22.4 代替メインフレームキー定義*

| PC のキー | メインフレーム定義 | キー | メインフレーム定義 |
|---|---|---|---|
| F1 | ヘルプ | F7 | 後ろへ |
| F2 | ズーム | F8 | 前へ |
| F3 | 縮小;サブミット | F9 | 出力 |

| PC のキー | メインフレーム定義 | キー | メインフレーム定義 |
|---|---|---|---|
| F4 | pgm; リコール | F10 | 左揃え |
| F5 | rfind | F11 | 右揃え |
| F6 | rchange | F12 | ホーム |
| | | RMB | 縮小;サブミット |

**SAA**

F1 から F12 を IBM SAA の CUAPF1 から CUAPF12 に対する値にマップし、Shift +F1 から Shift+F12 を CUAPF13 から CUAPF24 にマップします。右マウスボタン (RMB)は MB2 にマップされます。ファンクションキーが 10 個のみの場合、F11、F12、Shift+F11 および Shift+F12 は使用できず、**KEYS** ウィンドウに表示されません。

SAA は、System Application Architecture の略語で、IBM のさまざまなシステムで使用されるアプリケーション開発の枠組みです。CUA (共通ユーザーアクセス:Common User Access)は、すべてのアプリケーションに共通である必要があるユーザーインターフェイスやコンポーネントを定義する SAS の機能です。

次の表は、IBM SAA のキー定義を示しています。

***表 22.5*** *SAA キー定義*

| PC のキー | メインフレーム定義 | キー | メインフレーム定義 |
|---|---|---|---|
| F1 | ヘルプ | Shift+F1 | 削除 |
| F2 | キー | Shift+F2 | 貼り付け |
| F3 | 縮小;サブミット | Shift+F3 | 保存 |
| F4 | ホーム | Shift+F4 | 選択 |
| F5 | pgm; リコール | Shift+F5 | 選択解除 |
| F6 | ズーム | Shift+F6 | smark |
| F7 | 後ろへ | Shift+F7 | 左揃え |
| F8 | 前へ | Shift + F8 | 右揃え |
| F9 | prevcmd | Shift+F9 | rfind |
| F10 | pmenu | Shift+F10 | rchange |
| F11 | コマンド | Shift+F11 | 取り消し |
| F12 | キャンセル | Shift+F12 | 次 |

| PC のキー | メインフレーム定義 | キー | メインフレーム定義 |
|---|---|---|---|
| RMB | 縮小;サブミット | | |

**WIN**
Windows 版 SAS のデフォルトのキー定義を使用することを指定します。WIN は、デフォルトの設定です。

## 詳細

Windows 版 SAS で使用されるデフォルトのキー定義ではなく、他のキーマッピング (z/OS 版 SAS で使用されるキーマッピングなど)を使用する場合に、PFKEY システムオプションを使用します。

前のキーマッピングの各表にあるファンクションキーの値は、Base SAS のウィンドウのみを対象としています。SAS/AF などの他の SAS プロダクトには、別のキー定義があります。

PFKEY システムオプションを指定しない場合または無効な値を指定した場合は、Windows のデフォルトのキー定義が読み込まれます。キー定義のリストを表示するには、コマンドバーに `keys` を入力し、**KEYS** ウィンドウを 開きます。

# PRIMARYPROVIDERDOMAIN=システムオプション:Windows

基本の認証プロバイダのドメイン名を 指定します。

**該当要素:** 構成ファイル、SAS 起動
**カテゴリ:** 環境制御:初期化と操作
**PROC OPTIONS GROUP=** EXECMODES
**別名:** PRIMPD=
**参照項目:** PRIMARYPROVIDERDOMAIN=システムオプション

# PRINT システムオプション:Windows

バッチモードで実行している際の SAS 出力先を指定します。

**該当要素:** 構成ファイル、SAS 起動
**カテゴリ:** 環境制御:ファイル
**PROC OPTIONS GROUP=** ENVFILES
**デフォルト:** バッチモードでは *filename*.LST となり、*filename* は SAS ジョブの名前となります。
**Windows 固有:** すべて

## 構文

**-PRINT** *destination* | -NOPRINT

### 構文記述

**PRINT** *destination*
SAS プロシジャ出力ファイルの出力先を指定します。*destination* 引数には、Windows の有効なパス名やファイル名(デバイス名を含む)、またはパス名に関連付けられた環境変数を指定できます。スペースが含まれるパス名を指定する場合は、パス名を引用符で囲む必要があります。パス名のみを指定した場合、指定されたディレクトリに *filename*.LST というデフォルトのファイル名でプロシジャ出力ファイルが作成されます。*filename* は、対応する SAS ジョブの名前です。

**NOPRINT**
SAS プロシジャ出力ファイルを作成しません。

## 詳細

PRINT システムオプションは、SAS プログラムを対話的なウィンドウ環境以外のモードで実行しているときに、SAS 出力を書き込む出力先を指定します。

PRINT システムオプションは、バッチモードのみで有効です。

SAS が対話的なモードで動作している間は、プロシジャ出力は **OUTPUT** ウィンドウに送られます。SAS がバッチモードで動作しているときは、出力は *filename*.LST というファイルに送られます。ここで *filename* は SAS ジョブ名です。PRINT システムオプションを使用して、出力先の指定を変更することができます。

# PRNGETLIST システムオプション:Windows

システムに割り当てられたプリンタを 認識するかどうかを指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | ログとプロシジャ出力制御:プロシジャ出力 |
| **PROC OPTIONS GROUP=** | LISTCONTROL |
| **デフォルト:** | PRNGETLIST |
| **Windows 固有:** | すべて |

## 構文

**PRNGETLIST** | **NOPRNGETLIST**

### 構文記述

**PRNGETLIST**
システムに割り当てられたプリンタが SAS に認識されます。

**NOPRNGETLIST**
システムに割り当てられたプリンタが SAS で認識されません。

## 詳細

PRNGETLIST オプションを指定すると、システムに割り当てられたすべてのプリンタが SAS で認識されます。NOPRNGETLIST を指定すると、システムに割り当てられたプリンタが SAS で認識されません。NOPRNGETLIST は、SAS テクニカルサポートがシステムに割り当てられたすべてのプリンタを削除するようにユーザーに示唆する方法の代替として使用します。NOPRNGETLIST は、SAS が Object Spawner を介して起動さ

れる際にプリンタやプリンタ機能を探索するというパフォーマンスペナルティを避けるためにも使用されます。

## PRTABORTDLGS システムオプション:Windows

**印刷中止**ダイアログボックスをいつ表示するかを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** BOTH

**Windows 固有:** すべて

### 構文

**-PRTABORTDLGS** BOTH | NEITHER | FILE | PRINTER

**PRTABORTDLGS=** BOTH | NEITHER | FILE | PRINTER

#### *必須引数*

**BOTH**
**印刷中止**ダイアログボックスを表示するように指定します。

**NEITHER**
**印刷中止**ダイアログボックスを表示しないよう指定します。

**FILE**
ファイルに出力しているときにのみ**印刷中止**ダイアログボックスを表示するように指定します。

**PRINTER**
プリンタに出力しているときにのみ**印刷中止**ダイアログボックスを表示するように指定します。

### 詳細

**印刷中止**ダイアログボックスは、SAS が印刷ジョブを出力先にスプーリングしている間だけ表示されます。値 NEITHER は[**印刷中止**]ダイアログボックスを表示しない場合に使用します。

### 関連項目:

“印刷ジョブの取り消し” (180 ページ)

## PRTPERSISTDEFAULT システムオプション]Windows

SAS セッション間で同じ出力先プリンタを使用するように指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** ログとプロシジャ出力制御:ODS 出力

**PROC OPTIONS GROUP=** ODSPRINT

**デフォルト:** NOPRTPERSISTDEFAULT

**Windows 固有:** すべて

## 構文

**-PRTPERSISTDEFAULT** | **-NOPRTPERSISTDEFAULT**

### *構文記述*

**PRTPERSISTDEFAULT**
SAS セッション間で同じ出力先プリンタを使用するように指定します。

**NOPRTPERSISTDEFAULT**
デフォルトのプリンタを使用するように指定します。

## 詳細

通常は、SAS 起動時に、出力先プリンタを指定する SYSPRINT システムオプションの値は Windows のデフォルトのプリンタに設定されます。PRTPERSISTDEFAULT システムオプションを指定して SAS を起動すると、SYSPRINT システムオプションの値は、最後に PRTPERSISTDEFAULT を指定して起動した SAS セッションの出力先プリンタに設定されます。

SAS セッション間で同じ出力先プリンタを使用するには、SAS を起動するたびに PRTPERSISTDEFAULT システムオプションを指定する必要があります。SYSPRINT システムオプションと PRTPERSISTDEFAULT システムオプションの両方を指定して SAS を起動した場合は、SYSPRINT システムオプションの値によって出力先プリンタが決定されます。

## 関連項目:

- “SYSPRINT システムオプション:Windows” (578 ページ)
- “印刷” (165 ページ)

# PRTSETFORMS システムオプション:Windows

**フォームを使用する**チェックボックスを**印刷設定**ダイアログボックスに含めるかどうかを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、**SAS システムオプション**ウィンドウ

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** PRTSETFORMS

**Windows 固有:** すべて

## 構文

**-PRTSETFORMS** | **-NOPRTSETFORMS**

**PRTSETFORMS** | **NOPRTSETFORMS**

### *構文記述*

**PRTSETFORMS**
**フォームを使用する**チェックボックスを**印刷設定**ダイアログボックスに表示するように指定します。

**NOPRTSETFORMS**
**フォームを使用する**チェックボックスを**印刷設定**ダイアログボックスで非表示にするように指定します。

## 詳細

NOPRTSETFORMS システムオプションは、**フォームを使用する**チェックボックスを**印刷設定**ダイアログボックスで非表示にする場合に指定します。

## 関連項目:

"SAS 印刷フォームの使用" (176 ページ)

# REALMEMSIZE システムオプション Windows

割り当てることができる実メモリの量を指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | システム管理:メモリ |
| **PROC OPTIONS GROUP=** | MEMORY |
| **デフォルト:** | 0 |
| **Windows 固有:** | 有効な値 |

## 構文

**-REALMEMSIZE** *n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

### *必須引数*

***n* | *n*K | *n*M | *n*G**
予約メモリ量をそれぞれ 1、1,024 (キロバイト)、1,048,576 (メガバイト)および 1,073,741,824 (ギガバイト)の倍数で指定できます。*n* の値には、10 進値が使用できます。たとえば、`8` はバッファ数 8、`.782k` はバッファ数 801、`3m` はバッファ数 3,145,728 を示します。32 ビットの動作環境の場合、指定可能な最大値は 4294967295 (4G-1)です。

***hex*X**
メモリ量を、16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、値 `2dx` は、メモリ量を 45 バイトに設定します。

**MIN**
値を 0 に設定します。メモリの使用量は、SAS の起動時に SAS によって決定されます。

**MAX**
メモリ容量を、指定可能な最大値に設定します。

## 詳細

一部の SAS プロシジャでは、REALMEMSIZE オプションを使用して、必要以上にページを入れ替えずに、割り当てて使用できる仮想メモリの量を指定します。一方、MEMSIZE オプションは、SAS が常時、動的に割り当てられる仮想メモリの総量に制限を設定します。仮想メモリは、実メモリとページングスペースを組み合わせてサポートされます。必要な仮想メモリの量が、利用可能な実メモリの量を超えると、動作環境でページングが開始されます。ページングを行わないようにしたり、関連するパフォーマンスの問題を回避するには、REALMEMSIZE および MEMSIZE オプションに実メモリを超えない値を設定します。

## 比較

REALMEMSIZE オプションは、SORTSIZE オプションと似ています。REALMEMSIZE オプションは、複数のプロシジャに影響を与えます。SORTSIZE オプションは、SORT プロシジャにのみ影響を与えます。

# REGISTER システムオプション:Windows

メイン SAS ウィンドウの**ツール**メニューにアプリケーションを追加します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | 環境制御:表示 |
| **PROC OPTIONS GROUP=** | ENVDISPLAY |
| **デフォルト:** | なし |
| **Windows 固有:** | すべて |

## 構文

**-REGISTER** '*menu-name*' *'command'* <'*working-directory*'>

### *構文記述*

***'menu-name'***
メニューに表示される名前を指定します。*menu-name* は、引用符で囲む必要があります。

***'command'***
実行するコマンドを指定します。*command* 引数には、.EXE、.COM、.BAT ファイルを指定することができます。また、DIR コマンドなど、動作環境のコマンドを指定することもできます。*command* は、引用符で囲む必要があります。

***'working-directory'***
アプリケーション用の作業ディレクトリを指定します。この引数は任意指定です。アプリケーションは作業ディレクトリの 指定が必要かどうかは、アプリケーションのドキュメントを参照してください。*working-directory* は、引用符で囲む必要があります。

### 詳細

メイン SAS ウィンドウの**ツール**メニューには、最大 8 個のコマンドを追加できます。メニュー名つまりコマンドに空白も特殊文字も含まれていない場合は、引用符を省略できます。リストへのコマンド追加に関する詳細な情報は“ツールメニューにアプリケーションの追加” (66 ページ)を参照してください。

## RESOURCESLOC システムオプション:Windows

SAS のリソースが格納されるファイルのディレクトリを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** !*sasroot*\core\resource

**Windows 固有:** すべて

### 構文

**-RESOURCESLOC** <(>'*directory-specification-1*' <'*directory-specification-n*')> | "."

#### *構文記述*

**'*directory-specification*'**
SAS のリソースが格納されるファイルのディレクトリを指定します。*directory-specification* にスペースが含まれる場合は、引用符で囲む必要があります。複数の *directory-specification* を指定する場合は、リストをかっこで囲みます。

**“.”**
現在の作業フォルダを、SAS のリソースが格納されるファイルのデフォルトのディレクトリに指定します。

### 詳細

SAS のリソースは、SAS で使用されるアイコン、文字列、フォントなどが格納されるダイナミックリンクライブラリです。RESOURCESLOC ディレクトリ内にあるファイルの種類は、フォントファイル(.fon, .ttf)とダイナミックリンクライブラリ(.dll)です。

RESOURCESLOC オプションを複数指定すると、検索順序を定義することができます。

## RSASUSER システムオプション:Windows

SASUSER データライブラリのメンバを更新モードで開くか、読み込み専用モードで開くかを制御します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES

**デフォルト:** NORSASUSER

**Windows 固有:** ネットワークに関する注意点

**参照項目:** “RSASUSER System Option” (*SAS System Options: Reference*)

## 構文

**-RSASUSER** | **-NORSASUSER**

### *構文記述*

**RSASUSER**
すべてのユーザーが SASUSER ライブラリを共有できる環境で、SASUSER データライブラリへのアクセスを 読み取り専用アクセスに制限します。

**NORSASUSER**
1 人のユーザーが SASUSER ライブラリ内のファイルを更新アクセスで開けるようにし、複数のユーザーで SASUSER データライブラリのメンバを共有できないようにします。SASUSER ライブラリの更新アクセスを行うには、データライブラリのメンバに対する排他的権限が必要です。NORSASUSER は、デフォルト値です。

## 詳細

RSASUSER を指定すると、メンバへの読み込み専用アクセスがすべてのユーザーに許可されるため、ユーザーのグループは SASUSER データライブラリのメンバを共有することができます。たとえば、RSASUSER が有効な場合、各ユーザーは PROFILE カタログを読み込み専用アクセスで開くことができ、複数のユーザーが同時にプロファイルカタログからデータを 読み込むことができます。ただし、プロファイルカタログに情報を書き込むことはできず、書き込もうとした場合はエラーメッセージが表示されます。

SAS セッションでの RSASUSER の指定は、そのセッションにおけるファイルアクセスだけに影響を与えます。SASUSER ライブラリのメンバをユーザーのグループで共有できるようにする場合、システム管理者は、ネットワーク版の SAS 構成ファイルに RSASUSER システムオプションを設定する必要があります。ネットワーク版の SAS 構成ファイルは、SASUSER ライブラリを共有するすべてのユーザーによって共有されます。

RSASUSER を指定し、SASUSER データライブラリにプロファイルカタログが存在しない場合には、プロファイルカタログが WORK データライブラリに作成されます。

RSASUSER システムオプションが有用かどうかは、SAS の使用方法に応じて異なります。プロファイルカタログなど、SASUSER データライブラリに保存された情報を共有する必要がある場合には、RSASUSER はたいへん有用なシステムオプションです。それに対して、SAS/ASSIST を使用する場合は、RSASUSER は有用ではありません。SAS/ASSIST では、SASUSER データライブラリへの更新アクセスが必要です。

# RTRACE システムオプション:Windows

SAS セッション時に読み込まれるリソースリストを生成します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** ログとプロシジャ出力制御:SAS ログ

**PROC OPTIONS GROUP=** LOGCONTROL

**デフォルト:** NONE

**Windows 固有:** すべて

## 構文

**-RTRACE** ALL | NONE

### *必須引数*

**ALL**
指定の SAS セッションで使用されているすべてのファイルリソースのリストを生成します。

**NONE**
ファイルリソースのリストを生成しません。

## 詳細

RTRACE および RTRACELOC システムオプションを使用すると、SAS で使用されているリソースリストのファイルを作成することができます。-RTRACE ALL を指定しても、RTRACELOC システムオプションを指定しない場合には、出力は SAS ログに書き込まれます。

## 関連項目:

- “RTRACELOC システムオプション:Windows” (555 ページ)
- “The SAS Log” (*SAS Language Reference: Concepts* 9 章)

# RTRACELOC システムオプション:Windows

SAS のひとつのセッションでリードまたはロードするリソースのリストを書き込むファイルのパス名を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES

**デフォルト:** なし

**Windows 固有:** すべて

**ヒント:** %p (PID)、%d (日付)、%t (時刻)が指定されていると、RTRACELOC ファイル名を展開できます。

## 構文

**-RTRACELOC** *filename* | *pathname**filename*

### *構文記述*

***filename* | *pathname**filename***
有効な Windows ファイル名またはパス名とファイル名を指定します。このファイルに、ファイルリソース情報が保存されます。ファイル名またはパス名にスペースが含まれる場合は、引用符で囲みます。*pathname* を指定しない場合は、ファイルリソース情報は現在のディレクトリに保存されます。

## 詳細

RTRACELOC および RTRACE システムオプションを使用すると、SAS で使用されるリソースを指定することができます。

## 関連項目:

"RTRACE システムオプション:Windows" (554 ページ)

# SASAUTOS システムオプション:Windows

自動呼び出しマクロライブラリを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** 環境制御:ファイル;マクロ:SAS マクロ

**PROC OPTIONS GROUP=** ENVFILES; MACRO

**デフォルト:** SASAUTOS

**Windows 固有:** 次に対する有効な値:*library-specification*

**参照項目:** *SAS マクロ言語: リファレンス*

## 構文

**-SASAUTOS** <(>"*library-specification-1"*…<*"library-specification-n"*)>

**SASAUTOS=**<(>*"library-specification-1"*…<*"library-specification-n"*)>

### *構文記述*

***library-specification-1… library-specification-n***
有効な Windows パス名またはパス名を割り当てた環境変数を 1 つ以上指定します。パス名は、ディレクトリまたはサブディレクトリのレベルまでであることに注意してください。OPTIONS ステートメントを使用する場合や、パス名にスペースが含まれる場合は、Windows パス名を引用符で囲む必要があります。指定するライブラリが 1 つだけの場合は、特にかっこを付ける必要はありません。*library-specification* の値には、有効な Windows パス名を指定してください。

## 詳細

SASAUTOS システムオプションでは、SAS 自動呼び出しマクロライブラリを 1 つ以上指定できます。

APPEND と INSERT システムオプションを使ってファイル指定を追加することができます。

## 関連項目:

- "SASAUTOS システムオプション" (602 ページ)
- "SAS 自動実行ファイル " (20 ページ)
- "APPEND= System Option" (*SAS System Options: Reference*)
- "INSERT= System Option" (*SAS System Options: Reference*)

- *SAS マクロ言語: リファレンス*

## SASCONTROL システムオプション:Windows

SAS アプリケーションウィンドウにてシステムと設定メニュそして最小化と最大化のボタンを含むかどうかを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** SYSTEMMENU MINMAX

**Windows 固有:** すべて

### 構文

**-SASCONTROL** SYSTEMMENU | NOSYSTEMMENU<MINMAX | NOMINMAX>

**-SASCONTROL** <SYSTEMMENU | NOSYSTEMMENU> MINMAX | NOMINMAX

**SASCONTROL=**SYSTEMMENU | NOSYSTEMMENU<MINMAX | NOMINMAX>

**SASCONTROL=**<SYSTEMMENU | NOSYSTEMMENU> MINMAX | NOMINMAX

#### *必須引数*

**SYSTEMMENU**
メイン SAS ウィンドウ内のウィンドウに、システム/コントロールメニューを表示します。

**NOSYSTEMMENU**
メイン SAS ウィンドウ内のウィンドウに**システム/コントロール**メニューを表示しません。同時に、このウィンドウのタイトルバーに最小化、最大化および閉じるボタンを表示しません。

**MINMAX**
メイン SAS ウィンドウ内のウィンドウに、最小化/最大化ボタンを表示します。

**NOMINMAX**
メイン SAS ウィンドウ内のウィンドウに、最小化/最大化ボタンを表示しません。

### 詳細

SASCONTROL システムオプションは、メイン SAS ウィンドウ内にあるウィンドウに影響を与えますが、メイン SAS ウィンドウ自体には影響を与えません。メイン SAS ウィンドウは、AWSCONTROL システムオプションによって制御します。

SASCONTROL システムオプションは、SAS/AF アプリケーションのインターフェイスをカスタマイズするための、プログラマ向けのオプションです。

### 関連項目:

“AWSCONTROL システムオプション:Windows” (492 ページ)

## SASHELP システムオプション:Windows

SASHELP カタログ内のエントリ(SAS デフォルトフォーム、デバイスリスト、ディクショナリなど)を検索するディレクトリを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES

**デフォルト:** !*sasroot**SAS product*\sashelp、!*sascfg*\sascfg

**Windows 固有:** 次に対する有効な値:*library-specification*

**参照項目:** “SASHELP= System Option” (*SAS System Options: Reference*)

### 構文

**-SASHELP** ("*library-specification-1*"…<"*library-specification-n*")>

#### *構文記述*

***“library-specification-1”… “library-specification-n”***
有効な Windows パス名またはパス名を割り当てた環境変数を 1 つ以上指定します。パス名は、ディレクトリまたはサブディレクトリのレベルまでであることに注意してください。*library-specification* の値には、有効な Windows パス名を指定してください。パス名にスペースが含まれる場合は、パス名を引用符で囲む必要があります。

### 詳細

SASHELP システムオプションは、インストール時に設定されます。通常は、インストール後に変更することはありません。

プロダクトとそれに対応するファイルは、複数のドライブやディレクトリに分かれていることがあるので注意してください。*library-specification* 引数には、Windows パス名やパス名を割り当てた環境変数を指定します。

APPEND と INSERT システムオプションを使ってファイル指定を追加することができます。

### 関連項目:

- “APPEND= System Option” (*SAS System Options: Reference*)
- “INSERT= System Option” (*SAS System Options: Reference*)

## SASINITIALFOLDER システムオプション:Windows

SAS の初期化が完了した後、ファイルを**開く**および**名前を付けて保存**ダイアログボックスの作業フォルダおよびデフォルトフォルダを、指定したフォルダに変更します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES

**デフォルト:** なし

**Windows 固有:** すべて

## 構文

**-SASINITIALFOLDER** *newfolder*

### *必須引数*

***newfolder***
現在の作業フォルダのパスと、**ファイルを開くおよび名前を付けて保存**ダイアログボックスのデフォルトフォルダを指定します。*newfolder* にスペースが含まれる場合は、引用符で囲む必要があります。

## 詳細

AUTOEXEC または INITSTMT ファイルの場所が特定された後に、SASINITIALFOLDER システムオプションが処理されます。これらのファイルの場所を確実に特定できるようにするには、これらのファイルを、SASINITIALFOLDER システムオプションに指定したフォルダ以外のフォルダに格納します。

SASINITIALFOLDER システムオプションを指定しないと、デフォルトでは現在の作業フォルダが使用されます。デフォルトで、**ファイルを開くおよび名前を付けて保存**ダイアログボックスでは SASUSER フォルダが使用されます。

次の作業フォルダは"SAS 起動時の作業フォルダの設定" (11 ページ)の情報に応じて設定されます。

# SASUSER システムオプション n:Windows

SASUSER ライブラリの名前を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES

**デフォルト:** c:\Users*userid*\Documents\My SAS Files\9.4

**Windows 固有:** 次に対する有効な値:*library-specification*;構文

## 構文

**-SASUSER** ("*library-specification-1*"…<"*library-specification-n*"> )

### *構文記述*

***"library-specification-1"… "library-specification-n"***
SAS ライブラリに使用する有効な Windows パス名またはパス名を割り当てた環境変数を 1 つ以上指定します。パス名は、ディレクトリまたはサブディレクトリのレベ

ルまでであることに注意してください。指定するライブラリが 1 つだけの場合は、特にかっこを付ける必要はありません。*library-specification* の値には、有効な Windows パス名を指定してください。

## 詳細

SASUSER システムオプションでは、ユーザーのプロファイルカタログを含む SAS ライブラリを指定できます。SASUSER システムオプションのデフォルト値は SAS 構成ファイルに定義されています。SAS 構成ファイルは、SAS インストール時に変更できます。SAS の起動時に、構成ファイルまたは SAS コマンド内で SASUSER システムオプションを使用しない場合、SASUSER ライブラリは、WORK ライブラリと同一の場所に割り当てられます。これは一時的な割り当てです。

## 関連項目:

- “Profile カタログ ” (22 ページ)
- “Sasuser データライブラリの更新” (133 ページ)

# SCROLLBARFLASH システムオプション:Windows

マウスまたはキーボードを使用してスクロールバーにフォーカスできるかどうかを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** NOSCROLLBARFLASH

**Windows 固有:** すべて

## 構文

**-SCROLLBARFLASH** | **-NOSCROLLBARFLASH**

**SCROLLBARFLASH** | **NOSCROLLBARFLASH**

### *構文記述*

**SCROLLBARFLASH**
マウスおよびキーボードを使用してスクロールバーにフォーカスできるようにします。

**NOSCROLLBARFLASH**
マウスおよびキーボードを使用してスクロールバーにフォーカスできないようにします。

## 詳細

特定の状況では、マウスまたはキーボードでスクロールバーを選択すると、カーソルが点滅することがあります。NOSCROLLBARFLASH システムオプションを使用して、カーソルの点滅を無効にすることができます。**プリファレンス**ダイアログボックスの**詳細**タブで、**スクロールバーのフォーカスを無効にする**を選択することでも、カーソルの点滅を無効にできます。

### 関連項目:

“セッションプリファレンスについて ” (58 ページ)

## SET システムオプション:Windows

SAS 環境変数を定義します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES

**デフォルト:** なし

**Windows 固有:** ファイルまたはパスを示す値は、Windows で有効である必要があります。

### 構文

**-SET** *SAS-variable* "*value*" | ("*value-1*"…<"*value-n*"> )

**SET**=*SAS-variable* "*value*" | (" *value-1*"…<"*value-n*"> )

#### *構文記述*

***SAS-variable***
定義する環境変数を指定します。

***value***
環境変数に割り当てる値または値のセットを指定します。*value* がスペースを含むパス名である場合は、*value* を引用符で囲みます。

### 詳細

SET システムオプションの指定は、Windows の SET コマンドで Windows 環境変数を定義するのと似ています。SET システムオプションによって、よく使用する外部ファイルを示す環境変数を設定することができます。たとえば、次のプログラムでは、サンプルソースライブラリを示す環境変数を定義しています。

```
-set sampsrc (!sasroot\base\sample
              !sasroot\stat\sample
              !sasroot\graph\sample)
```

SAS セッション中にライブラリ名として SAMPSRC を参照すると、リストされたディレクトリが自動的にライブラリに割り当てられます。*sasroot* も SAS 環境変数です。SAS インストールのルートディレクトリを示すこの変数は、通常は SAS 構成ファイル内で割り当てられます。

環境変数をライブラリ参照名として使用するには、SET システムオプションを SAS 起動時に指定する必要があります。OPTIONS ステートメントで指定した場合は使用できません。

SAS 起動時にコマンド行で SET を指定した場合、変数はその SAS セッションに対してのみ設定されます。環境変数を繰り返し使用するには、構成ファイルに SET システムオプションを追加するか、Windows 環境変数を作成します。

APPEND と INSERT システムオプションを使ってファイル指定を追加することができます。

*注:* AUX、CON、NUL、LPT1 - LPT9、COM1 - COM9、PRN は Windows の予約語です。CON または NUL を環境変数名として使用しないでください。

### 関連項目:

- “環境変数を使用した SAS ライブラリの割り当て” (129 ページ)
- “環境変数の使用” (150 ページ)
- “APPEND= System Option” (*SAS System Options: Reference*)
- “INSERT= System Option” (*SAS System Options: Reference*)

## SGIO システムオプション:Windows

Scatter/Gather I/O 機能を有効化します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** ファイル:SAS ファイル

**PROC OPTIONS GROUP=** SASFILES

**デフォルト:** NOSGIO

**Windows 固有:** すべて

### 構文

**-SGIO** | **-NOSGIO**

#### *構文記述*

**SGIO**
scatter-read / gather-write 機能を有効化します。scatter-read / gather-write 機能は、SAS セッションが終了するまで有効です。

**NOSGIO**
scatter-read/gather-write 機能を有効化しません。

### 詳細

SGIO システムオプションは、PC に大容量 RAM を装備している場合に、SAS の入出力ファイル(データセット、カタログ、インデックス、ユーティリティファイル、その他の入出力ファイル)の I/O パフォーマンスを大幅に向上します。scatter-read / gather-write 機能は、メモリとディスク間での中間バッファの転送を不要にします。

SGIO が有効化されている場合、SAS はディスクと RAM 間のデータ転送のために、BUFNO システムオプションで指定した数のバッファを使用します。通常は BUFNO の値が増加するにつれて、I/O パフォーマンスは向上します。BUFNO システムオプションにさまざまな値を指定し、各 SAS ジョブまたは DATA ステップを調整してください。

scatter-read / gather-write 機能は、次の条件を満たす入出力ファイルでのみ有効化されます。

- 32 ビットシステムの場合、ページサイズが 4K の倍数(4096、8192 など)であるもの
- 64 ビットシステムの場合、ページサイズが 8K の倍数(8192、16384 など)であるもの

- バージョン 6 の SAS 以外で作成されたもの
- 順次ファイル形式のもの

入出力ファイルがこれらの条件を満たさない場合、SGIO オプションが指定されていても、SGIO 機能は有効化されません。SGIO に関する詳細な情報は、次を参照してください。SGIO の使用 ドキュメント

### 関連項目:

- “BUFNO システムオプション:Windows” (496 ページ)
- “パフォーマンスを最適化する SAS の機能” (203 ページ)

## SLEEPWINDOW システムオプション:Windows

**SLEEP** ウィンドウの表示と非表示を切り替えます。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | 環境制御:表示 |
| **PROC OPTIONS GROUP=** | ENVDISPLAY |
| **デフォルト:** | SLEEPWINDOW |
| **Windows 固有:** | すべて |

### 構文

**-SLEEPWINDOW** | **-NOSLEEPWINDOW**

#### *構文記述*

**SLEEPWINDOW**
**SLEEP** ウィンドウを表示します。

**NOSLEEPWINDOW**
**SLEEP** ウィンドウを表示しません。

### 詳細

**SLEEP** ウィンドウは、SLEEP 関数または WAKEUP 関数によって DATA ステップの実行が停止されるときに表示されます。**SLEEP** ウィンドウには、DATA ステップの実行が開始されるまでの時間が表示されます。

### 関連項目:

- “SLEEP 関数:Windows” (410 ページ)
- “WAKEUP 関数:Windows” (412 ページ)

## SORTANOM システムオプション:Windows

ホストソートユーティリティのオプションを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** ソート:プロシジャオプション

**PROC OPTIONS GROUP=** SORT

**デフォルト:** なし

**Windows 固有:** すべて

## 構文

**-SORTANOM** *option(s)*

**SORTANOM=** *オプション*

### *必須引数*

***option(s)***
次のオプションを 1 つ以上指定することができます。

**b**
SyncSort を単一呼び出しモードではなく複数呼び出しモードで実行します。

**t**
SAS ログに、ソートプロセスに関する統計量を出力します。

**v**
SAS ログに、Syncsort ユーティリティに渡されたコマンドをすべて出力します。

## 関連項目:

"SyncSort へのオプションの受け渡し" (440 ページ)

# SORTCUT システムオプション:Windows

このデータサイズを上回ると内部 SAS ソートではなくホストソートを使用する上限をオブザベーション数として指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** ソート:プロシジャオプション

**PROC OPTIONS GROUP=** SORT

**デフォルト:** 0

**Windows 固有:** すべて

## 構文

**-SORTCUT** *n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

**SORTCUT=** *n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

### 必須引数

***n* | *n*K | *n*M | *n*G**
オブザベーション数をそれぞれ 1、1(*n*)、1,024 (*n*K)、1,048,576 (*n*M)および 1,073,741,824 (*n*G)の倍数で指定します。*n* を使用して K、M、G の値を指定する場合は、n に 10 進値を指定することができます。たとえば、`8` はバッファ数 8、`.782k` はバッファ数 801、`3m` はバッファ数 3,145,728 を示します。

***hex*X**
オブザベーション数を、16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、値 `2dx` はオブザベーション数 45 を示します。

**MIN**
オブザベーション数に 0 を指定します。

**MAX**
オブザベーション数に 2,147,483,647 を指定します。

## 詳細

SORTPGM=BEST を指定しており、SAS によってデータベースのソートユーティリティを使用しない決定がされた場合、SAS では SORTCUTP オプションおよび SORTCUT オプションの値を使用して、SyncSort を使用するか SAS ソートを使用するかが決定されます。ソートするデータセットの大きさが SORTCUTP システムオプションに指定したバイト(またはキロバイト、メガバイト)より大きい場合は、SAS ソートプログラムではなく SyncSort が使用されます。指定する値は、2,147,483,647 バイト以下でなければなりません。SORTCUT と SORTCUTP がどちらも定義されていない場合、または 0 に設定されている場合は、SAS ソートが使用されます。両方のオプションが設定され、どちらの条件も満たされる場合は、SyncSort が使用されます。

## 関連項目:

- “SORTCUTP システムオプション:Windows” (565 ページ)
- “SORTPGM システムオプション:Windows” (568 ページ)
- “サイズまたはオブザベーションを基準にした並べ替え” (439 ページ)

# SORTCUTP システムオプション:Windows

このデータサイズを上回ると内部 SAS ソートではなくホストソートを使用する上限をバイト数で指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ |
| **カテゴリ:** | ソート:プロシジャオプション |
| **PROC OPTIONS GROUP=** | SORT |
| **デフォルト:** | 0 |
| **Windows 固有:** | すべて |

## 構文

**-SORTCUTP** *n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

**SORTCUTP=** *n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

### *必須引数*

***n* | *n*K | *n*M | *n*G**
バイト数をそれぞれ 1、1,024 (キロバイト)、1,048,576 (メガバイト)および 1,073,741,824 (ギガバイト)の倍数で指定します。キロバイト、メガバイト、ギガバイト単位の数は、10 進値で指定することができます。たとえば、`8` はバッファ数 8、`.782k` はバッファ数 801、`3m` はバッファ数 3,145,728 を示します。

***hex*X**
バイト数を、16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、値 `2dx` は 45 バイトを指定します。

**MIN**
0 バイトを指定します。

**MAX**
2,147,483,647 バイトを指定します。

## 詳細

SORTPGM=BEST を指定しており、SAS によってデータベースのソートユーティリティを使用しない決定がされた場合、SAS では SORTCUTP オプションおよび SORTCUT オプションの値を使用して、SyncSort を使用するか SAS ソートを使用するかが決定されます。ソートするデータセットの大きさが SORTCUTP システムオプションに指定したバイト(またはキロバイト、メガバイト)より大きい場合は、SAS ソートプログラムではなく SyncSort が使用されます。指定する値は、2,147,483,647 バイト以下でなければなりません。SORTCUT と SORTCUTP がどちらも定義されていない場合、または 0 に設定されている場合は、SAS ソートが使用されます。両方のオプションが設定され、どちらの条件も満たされる場合は、SyncSort が使用されます。

次の式によって、ソートするバイト数が計算されます。

*バイト数= ((length-of-obs) + (length-of-all-keys)) * number-of-obs*

## 関連項目:

- “SORTPGM システムオプション:Windows” (568 ページ)
- “SORTCUT システムオプション:Windows” (564 ページ)
- “サイズまたはオブザベーションを基準にした並べ替え” (439 ページ)

# SORTDEV システムオプション:Windows

ホストソートユーティリティによって作成される一時ファイルのパス名を指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ |
| **カテゴリ:** | ソート:プロシジャオプション |
| **PROC OPTIONS GROUP=** | SORT |
| **デフォルト:** | 構成ファイルで設定された-WORK と同じ場所 |
| **Windows 固有:** | すべて |

## 構文

**-SORTDEV** "*pathname*"

**SORTDEV** ="*pathname*"

### *必須引数*

**"*pathname*"**
有効な Windows のパス名を指定します。

## 詳細

SORTDEV オプションでは、SyncSort ユーティリティによって作成される一時ファイルの代替パス名を指定します。パス名は、引用符で囲む必要があります。

## 関連項目:

- "WORK システムオプション:Windows" (589 ページ)
- "SyncSort へのパラメータの受け渡し" (440 ページ)

# SORTNAME System Option:Windows

ホストソートユーティリティ名を指定します。

**該当要素:** 構成ファイル、SAS 起動、SASV9_OPTIONS 環境 変数、OPTIONS ステートメント

**PROC OPTIONS GROUP=** SORT

**デフォルト:** なし

**Windows 固有:** すべて

## 構文

**SORTNAME**="*host-sort-utility-name*"

-SORTNAME *host-sort-utility-name*

## 詳細

SORTNAME=オプションには、デフォルトのホストソートユーティリティ名を指定します。syncsort を指定できます。

## 関連項目:

"SORTPGM システムオプション:Windows" (568 ページ)

# SORTPARM システムオプション:Windows

ホストソートユーティリティのパラメータを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** ソート:プロシジャオプション

**PROC OPTIONS GROUP=** SORT

**デフォルト:** なし

**Windows 固有:** すべて

## 構文

**SORTPARM**="*SyncSort-parameters*"

**-SORTPARM** "*SyncSort-parameters*"

### *必須引数*

***SyncSort-parameters***
SyncSort ユーティリティに渡すパラメータを指定します。*SyncSort-parameters* は、引用符で囲みます。

## 詳細

*SyncSort- parameters* の詳細については、SyncSort for Windows のドキュメントを参照してください。

# SORTPGM システムオプション:Windows

SAS ソートユーティリティとホストソートユーティリティのいずれを使用するかを指定します。また、使用するソートユーティリティを SAS に選択させることもできます。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** ソート:プロシジャオプション

**PROC OPTIONS GROUP=** SORT

**デフォルト:** BEST

**Windows 固有:** すべて

## 構文

**-SORTPGM** Windows 版 SAS | BEST | HOST

**SORTPGM**=Windows 版 SAS | BEST | HOST

### *必須引数*

**Windows 版 SAS**
SAS のソートルーチンを使用してソートします。

**BEST**
データのソートに最も適したソートルーチン(データベースのソート機能、SAS のソート機能または SyncSort)を、SAS によって決定します。データベースのソート機能を使用しないことを決定すると、SAS では SORTCUT と SORTCUTP の両方の値を調べます。両方のオプションの値が 0 の場合は、SAS のソート機能が使用されます。両方のオプションが設定され、どちらの条件も満たされる場合は、SyncSort ルーチンが使用されます。

**HOST**
Windows の SyncSort を使用してソートします。

### 関連項目:

- “SORT プロシジャ:Windows” (436 ページ)
- “SORTCUT システムオプション:Windows” (564 ページ)
- “SORTCUTP システムオプション:Windows” (565 ページ)

## SORTSIZE システムオプション:Windows

SORT プロシジャで使用できるメモリ量を指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** ソート:プロシジャオプション
システム管理:メモリ

**PROC OPTIONS GROUP=** MEMORY
SORT

**デフォルト:** 1G

**Windows 固有:** デフォルト値

**参照項目:** “SORTSIZE= System Option” (*SAS System Options: Reference*)

### 構文

**-SORTSIZE** *n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

**SORTSIZE**= *n* | *n*K | *n*M | *n*G | *hex*X | MIN | MAX

#### *必須引数*

***n* | *n*K | *n*M | *n*G**
メモリ量をそれぞれ 1、1,024 (キロバイト)、1,048,576 (メガバイト)および 1,073,741,824 (ギガバイト)の倍数で指定します。キロバイト、メガバイト、ギガバイト単位の数は、10 進値で指定することができます。たとえば、`8` はバッファ数 8、`782k` はバッファ数 801、`3m` はバッファ数 3,145,728 を示します。

***hex*X**
メモリ量を、16 進数で指定します。数字(0-9)で始まる値に続けて、X を指定する必要があります。たとえば、値 2dx は、メモリ量を 45 バイトに設定します。

**MIN**
使用できる最小メモリ量を指定します。

**MAX**
使用できる最大メモリ量を指定します。

### 詳細

デフォルトでは、このオプションは使用できる最大メモリ量に設定されます。SORTSIZE システムオプションによって、データセットのソートの際に行われるスワッピングの回数を減らすことができます。SORT プロシジャでは、指定したメモリ量よりも多くのメモリが必要な場合、SASWORK ディレクトリに、データを格納するための一時ユーティリティファイルが 作成されます。SORT プロシジャのアルゴリズムでは、Windows よりも効率的に 不必要なデータをスワップできます。

ソートしたい SAS データファイルを使用マシンの物理メモリに配置できる場合、SAS でのソートは非常に効率的です。SORTSIZE にはデータファイルのサイズより大きい値を指定してください。データファイルが物理メモリに収まらない場合は、SORTSIZE を 1G 以下に設定します。また、SORTSIZE には必ず MEMSIZE より少なくとも 8M 小さい値を設定してください。

### 比較

SORTSIZE オプションは、REALMEMSIZE オプションと似ています。SORTSIZE は SORT プロシジャのみに影響を与えます。REALMEMSIZE は、複数のプロシジャに影響を与えます。

### 関連項目:

- “SORT プロシジャ:Windows” (436 ページ)
- “SORT プロシジャのパフォーマンス向上” (205 ページ)

## SPLASH システムオプション:Windows

SAS の起動時にスプラッシュ画面(ロゴ画面)を表示するかどうかを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** SPLASH

**Windows 固有:** すべて

### 構文

**-SPLASH** | **-NOSPLASH**

**-SPLASH ON** | **-SPLASH OFF**

#### *構文記述*

**SPLASH または SPLASH ON**
SAS の起動時にロゴ画面を表示します。

**NOSPLASH または SPLASH OFF**
SAS の起動時にロゴ画面を表示しません。

### 詳細

SPLASH システムオプションを指定すると、SAS の起動時に SAS ロゴ画面が表示されます。

SPLASHLOC システムオプションを使用すると、カスタムのスプラッシュ画面を表示するように指定できます。

### 関連項目:

“SPLASHLOC システムオプション:Windows” (571 ページ)

## SPLASHLOC システムオプション:Windows

SAS の起動時に表示されるスプラッシュ画面のビットマップの保存場所を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** なし

**Windows 固有:** すべて

### 構文

**-SPLASHLOC** *DLL-name* <*res-number*> | *BMP-filename*

#### *構文記述*

***DLL-name***
カスタマイズしたロゴおよび著作権画面が格納されているダイナミックリンクライブラリ(DLL)を指定します。

***res-number***
ダイナミックリンクライブラリ(DLL)の名前に結び付けられたリソース番号を指定します。

***BMP-filename***
スプラッシュ画面として使用するスタンドアロンの Windows ビットマップ(BMP)ファイルのパスと名前を指定します。

### 詳細

ビットマップリソース(カスタマイズしたロゴおよび著作権画面)を作成し、それをダイナミックリンクライブラリ(DLL)に組み込むことができます。使用する DLL は、32 ビットバージョンの SAS を使用している場合は 32 ビットでなければなりません。また 64 ビットバージョンの SAS を使用している場合は 64 ビットでなければなりません。つまり、Microsoft Platform SDK のライブラリを使用して作成する必要があります。*DLL-name* を指定するときにリソース番号(*res-number*)を指定しなかった場合、デフォルトのソース番号は 1 に設定されます。

スタンドアロンの Windows ビットマップ(BMP)ファイルのパスと名前を指定して、スプラッシュ画面として使用することもできます。パスには、有効な Windows のパス名を指定します。パス名にスペースが含まれる場合は、パス名を引用符で囲む必要があります。

## STIMEFMT システムオプション:Windows

FULLSTIMER と STIMER 出力を表示する時間に使用されるフォーマットを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** ログとプロシジャ出力制御:SAS ログ

**PROC OPTIONS GROUP=** LOGCONTROL

**デフォルト:** M

**Windows 固有:** すべて

## 構文

**-STIMEFMT** *value(s)*

**STIMEFMT**=*value(s)*

### *必須引数*

***value***

STIMEFMT とともに使用するオプションを指定します。次のシステムオプションが追加されました。

**タイムスタンプオプション**

タイムスタンプオプションは次で説明されます。

| | |
|---|---|
| TS | STIMER と FULLSTIMER の一部としてのタイムスタンプを常に表示するようにします。 |
| TSFULL | FULLSTIMER の一部としてのタイムスタンプを表示するようにします。TSFULL はデフォルトの設定です。 |
| TSOFF | STIMER と FULLSTIMER のタイムスタンプを停止します。 |

**メモリ**

FULLSTIMER の一部として常に表示されます。デフォルトメモリ出力は KB にて表示されます。次のシステムオプションが追加されました。

| | |
|---|---|
| MEMFULL | STIMER の一部としてではなく FULLSTIMER の一部としてメモリ統計を記録します。 |
| MEM | FULLSTIMER および STIMER の一部としてメモリ統計を記録します。 |
| KB | キロバイト単位でメモリを記録します。 |
| MB | メガバイト単位でメモリを記録します。 |
| GB | ギガバイト単位でメモリを記録します。 |
| C | メモリ表示の数字にコンマを追加します。 |
| NC | メモリ表示の数字にコンマを追加しません。 |

**H, HOURS**

SAS ソフトウェアが STIMER 出力を `hours:minutes:seconds` の形式で表示するように指定します。

**経過時間と CPU 時間**

STIMER と FULLSTIMER の一番合うように時、分、秒のどれかにて表示するように設定できます。

| | |
|---|---|
| Z \| H \| HOURS | 時間を時:分:秒にて記録します。 |
| M \| MINUTES | 時間を分:秒にて記録します。 |
| S \| SECONDS | 時間を秒にて記録します。 |

HMS 時と分にてゼロを頭に付けないフォーマットにて記録します。

**カウンタ**
FULLSTIMER の一部として追加カウンタとして表示するよう指定します。

E | ENABLE は追加カウンタを有効にします。

D | DISABLE は追加カウンタを無効にします。

**ヘルプ**
STIMEFMT オプションのヘルプにアクセスするために使用される2つの値を提供します。

FMT は使用できるタイムスタンプフォーマットをリストします。

OPT は使用できる他のオプション値をリストアップします。

## 詳細

### *STIMEFMT ベーシック*

STIMER と FULLSTIMER システムオプションによって生成される出力のフォーマットを STIMEFMT システムを使ってカスタマイズすることができます。STIMEFMT を使って次のタスクを実行することができます。

- 使用できるフォーマットのリストアップ

```
options stimefmt = fmt;
```

- 使用できる他のオプションのリストアップ

```
options stimefmt = opt;
```

- STIMER のタイムスタンプの表示と非表示

```
options stimefmt = tson | tsoff | tsfull;
```

- 必要であればオプションの併用

```
options stimefmt = (tson YYNNDDS);
```

- メモリー値をコンマで区切ること

```
options stimefmt = c;
```

- 値を指定するときはコンマを使わないこと

```
options stimefmt = nc;
```

- メモリの単位を選択

```
options stimefmt = GB | MB | KB;
```

- STIMER と FULLSTIMER のメモリレポートの有効化

```
options stimefmt = mem;
```

- タイムスタンプでの時間表示の設定

```
options stimefmt = TOD | TIME | TIMEAMPM;
(TOD and TIME specify military time.)
```

- HOURS または MINUTES を使って CPU または実時間かの表示設定

### *タイムスタンプ表示フォーマット*

タイムスタンプのフォーマットは SAS にてサポートされている標準フォーマットにすることができます。次のフォーマットを含みます。

```
ABS.          (1970 年 1 月 1 日からの絶対秒数)
DATE.         DATE9.
DDMMYY.       DDMMYY10.     DDMMYYB.DDMMYYB10.     DDMMYYC.     DDMMYYC10.DDMMYYD.
ISO.          (ISO 標準時間)
MMDDYY.       MMDDYY10.     MMDDYY.MMDDYYB10.      MMDDYYC.     MMDDYYC10.MMDDYYD.
NLDATM.       NLDATMAP.
YYMMDD.       YYMMDD10.     YYMMDDB.YYMMDDB10.     YYMMDDC.     YYMMDDC10.YYMMDD
TOD.          (軍時間の形式で時間を書き込みます)TIME.          (軍時間の形式で時間を書き込
```

OPTIONS ステートメントの構文は次にリストしてあります。

```
options stimefmt = fmt;
```

ここで *fmt* は有効な SAS フォーマットです。

### *STIMEFMT オプションにて複数の値を使用*

STIMEFMT オプションを利用して、複数の設定を同時に行うことができます。次の例に示すように、複数の値は括弧で囲む必要があります。

```
options stimefmt = (h YYMMDD. gb c);
```

### *STIMEFMT オプションでの設定表示*

PROC OPTIONS にて STIMEFMT の現在のすべての設定値を表示します。次の例は PROC OPTIONS を実行したときのログ出力を示します。

```
proc options option=stimefmt; run;
```

***ログ 22.1*** *PROC OPTIONS からのログ出力*

```
SAS (r) Proprietary Software Release 9.3  TS1B0 STIMEFMT=(NLDATM2.HMS TIMEAMPM
KB MEMFULL TSFULL NC) Specified the output format for FULLSTIMER and STIMER.This
controls the timestamp, memory, CPU and elapsed time.
```

### *STIMEFMT をデフォルト値にリセット*

次の OPTIONS ステートメントによって STIMEFMT の設定をデフォルト値にリセットすることができます。

```
options stimefmt = normal;
```

## 関連項目:

"The SAS Log" (*SAS Language Reference: Concepts* 9 章)

---

## STIMER システムオプション Windows

システムパフォーマンス統計量の一部を SAS ログに書き出します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** ログとプロシジャ出力制御:SAS ログ

**PROC OPTIONS GROUP=** LOGCONTROL

**デフォルト:** STIMER

**Windows 固有:** レポートされる統計量

## 構文

**-STIMER** | **-NOSTIMER**

**STIMER** | **NOSTIMER**

### *構文記述*

**STIMER**
実時間と CPU 時間を SAS ログに書き出します。

**NOSTIMER**
実時間と CPU 時間を SAS ログに書き出しません。

## 詳細

STIMER システムオプションでは、DATA ステップまたはプロシジャタスクの完了までにかかった時間を、SAS ログに出力します。

次に、STIMER 出力の例を示します。

```
real time      0.96 seconds
cpu time       0.01 seconds
```

STIMER により、次の統計量が表示されます。

***表 22.6*** *STIMER 統計量の説明*

| 統計量 | 説明 |
|---|---|
| リアルタイム | SAS ジョブの処理にかかった時間。実時間は、経過時間とも呼ばれます。 |
| CPU 時間 | SAS コード、および SAS プロセスの代わりにシステムオーバーヘッドタスクを実行するためにかかった総時間。この値は、FULLSTIMER のユーザー CPU 統計量とシステム CPU 統計量を組み合わせた値です。 |

*注:* SAS 9 以降では、一部のプロシジャはマルチスレッドを使用します。複数の CPU を使用するコンピュータでは、オペレーティングシステムで同時に複数のスレッドを実行できます。したがって、CPU 時間は STIMER 出力の実時間を超える場合があります。

たとえば、SAS プロシジャでは、2 つの異なる CPU で同時に 2 つのスレッドを実行できます。CPU 時間の値は、次のように計算します。

```
CPU1 time + CPU2 time = total CPU time
1 second + 1 second = 2 seconds
```

CPU1 でスレッドを実行し、同時に、CPU2 で同一の SAS プロセスの別のスレッドを実行できるため、理論上は、実時間 1 秒あたり 2CPU 秒が費やされます。

## 比較

STIMER システムオプションは、SAS で利用できる動作環境のすべてのパフォーマンス統計量の一部を、SAS ログに書き出すことを指定します。FULLSTIMER システム

オプションでは、参照できるすべてのパフォーマンス統計量を SAS ログに書き出すかどうかを指定します。

## 関連項目:

- “FULLSTIMER システムオプション:Windows” (513 ページ)
- 12 章: “Optimizing System Performance” (*SAS Language Reference: Concepts*)

# SYSGUIFONT システムオプション:Windows

ボタンテキストとその説明テキストに使用するフォントを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** 表示設定による

**Windows 固有:** すべて

## 構文

**-SYSGUIFONT** "*font-name*" <*font-size*>

### *構文記述*

***“font-name”***
画面やダイアログボックスのテキスト要素に使用するフォント名を指定します。指定するフォント名は、システムにインストールされたフォント名と一致する、有効なフォント名("Times New Roman"や"Courier"など)である必要があります。*font-name* は、二重引用符で囲む必要があります。これは、必須引数です。

***font-size***
ウィンドウのテキストに使用するフォントのサイズを指定します。*font-size* を省略した場合、デフォルト値が使用されます。

## 詳細

SYSGUIFONT システムオプションは、画面テキストおよびダイアログボックステキスト要素に使用するテキストのフォントサイズを制御します。ウィンドウの内容のフォントを変更するには、FONT システムオプションを使用します。拡大したフォントを読みやすくするスペースを確保するために、SAS ウィンドウの最大化が必要となる場合があります。

## 関連項目:

- “FONT システムオプション:Windows” (509 ページ)
- “フォントの選択” (57 ページ)

## SYSIN システムオプション:Windows

バッチモードのソースファイルを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES

**デフォルト:** なし

**Windows 固有:** 次に対する有効な値:*file-specification*

### 構文

**-SYSIN** *file-specification* | **-NOSYSIN**

#### *構文記述*

**SYSIN *file-specification***
SAS を起動し、ファイルをバッチモードでサブミットします。*file-specification* には、Windows で有効なファイル名を指定します。

**NOSYSIN**
SAS をバッチモードで起動しますが、ファイルはサブミットしません。この指定は、SAS autoexec ファイルのテストに使用します。autoexec ファイルの処理が終了すると、SAS は終了します。

### 詳細

SYSIN システムオプションでは、SAS プログラムを含むファイルを指定します。このオプションは、非対話型モードで SAS を実行するときに使用します。このオプションは、SAS 起動時にのみ指定できます。

## SYSPARM システムオプション:Windows

SAS プログラムに渡される文字列を指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES

**デフォルト:** なし

**Windows 固有:** 次に対する有効な値および構文:*文字*

**参照項目:** *SAS マクロ言語: リファレンス*

### 構文

**-SYSPARM** <"> *characters*<">

**SYSPARM**=<"> *characters*<">

### *構文記述*

***characters***
文字列をすべて大文字で書き出します。

**"*characters*"**
文字列の大文字または小文字を保持します。

## 詳細

SYSPARM システムオプションは、SAS プログラムに渡すことのできる文字列を指定できます。

指定した文字列に SAS DATA ステップからアクセスするには、SYSPARM()関数を使用します。自動マクロ変数&SYSPARM を使用すると、SAS プログラム内の任意の場所からアクセスできます。

# SYSPRINT システムオプション:Windows

SAS 出力を印刷するための出力先プリンタを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ
**カテゴリ:** ログとプロシジャ出力制御:プロシジャ出力
**PROC OPTIONS GROUP=** LISTCONTROL
**デフォルト:** デフォルトシステムプリンタ
**Windows 固有:** すべて

## 構文

**-SYSPRINT** "*printer-name*"<"*destination*">

**SYSPRINT**="*printer-name*"<"*destination*">

### *構文記述*

**"*printer-name*"**
Windows にインストールされている名前("Charlie's HP LaserJet"など)でプリンタを指定します。インストールされているプリンタを調べるには、Windows のコントロールパネルでプリンタを選択します。*printer-name* は、二重引用符で囲む必要があります。

**"*destination*"**
プリントファイルをディスクに書き込む際のファイル名を指定します。この引数を指定すると、SAS によって生成されたプリンタ出力は、すべて指定したファイルに送られます。同じ名前の既存のファイルがある場合は、ファイルが上書きされます。出力がプリンタに直接送られない場合でも、*printer-name* に関連付けられたプリンタドライバによってフォーマットされます。*destination* は、二重引用符で囲む必要があります。

## 詳細

SYSPRINT システムオプションでは、SAS 出力を印刷するための出力先プリンタを指定できます。

別のプリンタを**印刷設定**ダイアログボックスで選択すると、SYSPRINT システムオプションの値(OPTIONS プロシジャにより表示されます)にもこの選択が反映されます。

SYSPRINT システムオプションまたは PRTPERSISTDEFAULT システムオプションを指定しないと、*printer-name* と *destination* 引数には、システムのデフォルトプリンタの値が使用されます。

SAS 起動時に PRTPERSISTDEFAULT が指定されている場合、SYSPRINT システムオプションの値は SAS セッションから SAS セッションに引き継がれます。SAS 起動時に PRTPERSISTDEFAULT システムオプションと SYSPRINT システムオプションの両方が指定されている場合、SYSPRINT システムオプションの値として、SYSPRINT システムオプションに指定したプリンタが使用されます。

***注意:***

**Windows の印刷ダイアログボックスを使用して印刷オプションを変更すると、SAS の印刷システムオプションの値が変更される可能性があります。** SYSPRINT のような SAS システムオプションを使用して印刷オプションを設定した後、Windows の印刷ダイアログボックスを使用して印刷オプションを設定した場合、SAS システムオプションは Windows の印刷ダイアログボックスで指定された値に設定されます。

## 関連項目:

- “印刷” (165 ページ)
- “PRTPERSISTDEFAULT システムオプション]Windows” (549 ページ)
- “SYSPRINTFONT システムオプション:Windows” (579 ページ)

# SYSPRINTFONT システムオプション:Windows

現在のデフォルトプリンタに、印刷時に使用するフォントを指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ |
| **カテゴリ:** | ログとプロシジャ出力制御:プロシジャ出力 |
| **PROC OPTIONS GROUP=** | LISTCONTROL |
| **デフォルト:** | なし |
| **Windows 固有:** | すべて |
| **参照項目:** | “SYSPRINTFONT= System Option” (*SAS System Options: Reference*) |

## 構文

**-SYSPRINTFONT** ("*font-name*" <BOLD | NORMAL> <REGULAR | ITALIC> <*character-set*> <*point-size*> <<NAMED"*printer-name*" | UPRINT="*printer-name*" | DEFAULT | ALL>> )

**SYSPRINTFONT**="*font-name*" <BOLD | NORMAL> <REGULAR | ITALIC> <*character-set*> <*point-size*> <NAMED"*printer-name*" | UPRINT="*printer-name*" | DEFAULT | ALL>

### 構文記述

**"*font-name*"**
印刷に使用するフォント名を指定します。指定するフォント名は、システムにインストールされたフォント名と一致する、有効なフォント名("SAS Monospace"や"Courier"など)である必要があります。*font-name* は、二重引用符で囲む必要があります。これは、必須引数です。

**BOLD | NORMAL**
フォントの太さを指定します。デフォルトの設定は NORMAL です。

**REGULAR | ITALIC**
フォントのスタイルを指定します。デフォルトの設定は REGULAR です。

***character-set***
印刷に使用する文字セットを指定します。デフォルトの設定は"Windows"です。有効な値は、Western、Central European、Cyrillic、Greek、Turkish、Arabic、Baltic、Thai です。指定したフォントでサポートされない文字セットを指定すると、デフォルトの文字セットが使用されます。デフォルトの文字セットがそのフォントでサポートされていない場合、そのフォントにおけるデフォルトの文字セットが使用されます。

***point-size***
印刷に使用するサイズをポイントで指定します。1 から 7200 の整数で指定する必要があります。この引数を省略すると、10 ポイントに設定されます。

**NAMED "*printer-name*"**
Sasuser.Profile2 カタログ内の名前付きプリンタに対して、フォント情報を更新します。プリンタ名には、**印刷設定**ダイアログボックスに表示される名前と正確に一致する名前を指定する必要があります。ただし、大文字と小文字の区別はありません。*printer-name* は、二重引用符で囲む必要があります。このキーワードは任意指定です。

**UPRINT="*printer-name*"**
設定を適用するユニバーサルプリンタを指定します。UPRINT は、SAS レジストリに表示されているプリンタにのみ使用できます。*プリンタ名*には、**印刷設定**ダイアログボックスに表示される名前と正確に一致する名前を指定する必要があります。ただし、大文字と小文字の区別はありません。Universal Prints が複数の単語で構成される場合、*printer-name* は、一重引用符または二重引用符で囲む必要があります。引用符は、printer-name と一緒に保存されます。

**DEFAULT**
Sasuser.Profile2 カタログ内の-SYSPRINT システムオプションに指定されているプリンタに対して、フォント情報を更新します。

**ALL**
Sasuser.Profile2 カタログ内の、インストールされているすべてのプリンタに対して、フォント情報を更新します。このキーワードは任意指定です。

## 詳細

SYSPRINTFONT システムオプションでは、現在のデフォルトプリンタ(SYSPRINT システムオプションにより指定される)または任意指定のキーワード NAMED または ALL で示されるプリンタに、印刷時に使用するフォントを設定できます。この情報は、SASUSER.PROFILE2 カタログに格納されます。

SYSPRINTFONT システムオプションを構成ファイル、コマンド行または **SAS システムオプション**ウィンドウで指定する場合は、オプションの引数をかっこで囲みます。OPTIONS ステートメントで SYSPRINTFONT システムオプションを指定している場合、かっこは不要です。

キーワードに DEFAULT または何も指定せずに SYSPRINTFONT を使用し、その後で SYSPRINT システムオプションまたは**印刷設定**ダイアログボックスで現在のデフォルトプリンタを変更すると、現在のデフォルトプリンタで使用するフォントは次のように決定されます。

1. Sasuser.Profile2 内で、現在のデフォルトプリンタにフォントが指定されている場合は、そのフォントが使用されます。
2. SYSPRINTFONT に指定されているフォントがプリンタにインストールされている場合は、そのフォントが使用されます。
3. Sasuser.Profile2 でプリンタに対してフォントを定義しておらず、SYSPRINTFONT でもプリンタに対して有効なフォントが指定されていない場合、現在の表示フォントがスケーラブルであれば、その表示フォントが印刷に使用されます。
4. 現在の表示フォントがスケーラブルでない場合は、10 ポイントの SAS Monospace が使用されます。
5. SAS Monospace フォントを使用できない場合は、プリンタのデフォルトフォントが印刷に使用されます。

*注:* 標準形式の文字を使用して、行と列の区切りや枠付きの表形式のレポートを読みやすく印刷するには、次のフォントとプリンタを使用します。

- SAS Monospace フォントまたは SAS Monospace Bold フォント
- TrueType フォントをサポートするプリンタ

## 例

### *例 1: デフォルトプリンタのフォントを指定*

次の例では、デフォルトプリンタで 12 ポイントの SAS Monospace フォントを使用することを指定します。

```
-sysprintfont ("SAS Monospace" 12)
```

### *例 2: 名前指定したプリンタのフォントを指定*

この例では、LPT1:に割り当てられた HP LaserJet IIIsi PostScript という名前のプリンタで、10 ポイントの Courier New を使用することを指定します。プリンタを指定する際、SAS の**印刷設定**ダイアログボックスの表示名を使用することに注意してください。

```
-sysprintfont ("Courier New" named
    "HP LaserJet IIIsi Postscript on LPT1:")
```

### *例 3: SAS コマンド行でのユニバーサルプリンタへのフォントの 指定*

次の例では、PDF ユニバーサルプリンタに Albany AMT フォントを指定します。

```
sysprintfont=('courier' 11 uprint='PDF')
```

## 関連項目:

"SYSPRINT システムオプション:Windows" (578 ページ)

# TOOLDEF システムオプション:Windows

ツールボックスの表示位置を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** TOP RIGHT

**Windows 固有:** すべて

## 構文

**-TOOLDEF** TOP | CENTER | BOTTOM <LEFT | CENTER | RIGHT>

### *構文記述*

**TOP | CENTER | BOTTOM**
ツールボックスの垂直方向の位置を指定します。デフォルト値は TOP です。

**LEFT | CENTER | RIGHT**
ツールボックスの水平方向の位置を指定します。デフォルト値は RIGHT です。

## 詳細

TOOLDEF システムオプションでは、ツールボックスを表示するときの、ディスプレイ内での表示位置を指定できます。

最初に垂直方向の位置を指定する必要があります。水平方向の位置の指定は任意です。省略すると、RIGHT が使用されます。

*注:* ツールボックスは、メイン SAS ウィンドウを基準にしてではなく、ディスプレイ全体を基準に位置が決定されます。ツールボックスではなくツールバーを使用する場合は、このオプションの効果はありません。

## 関連項目:

- “ツールバーのカスタマイズ” (68 ページ)
- “ツールバーを使用してコマンドの発行” (42 ページ)

# TOOLSMENU システムオプション

SAS ウィンドウに**ツール**メニューが含まれるかどうかを指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

## 構文

**TOOLSMENU** | **NOTOOLSMENU**

### *構文記述*

**TOOLSMENU**
**ツール**メニューを SAS ウィンドウに含めるように指定します。

**NOTOOLSMENU**
**ツール**メニューを SAS ウィンドウに含めないように指定します。

## EMAILDLG システムオプション

ユニバーサル印刷のメニューを有効化するかどうかと、印刷の初期設定を設定するかどうかを指定します。

**該当要素:** 構成ファイル、SAS の起動
**カテゴリ:** ログおよびプロシジャ出力制御:ODS 印刷
**PROC OPTIONS GROUP=** ODSPRINT

### 構文

**UNIVERSALPRINT** | **NOUNIVERSALPRINT**

#### *構文記述*

**UNIVERSALPRINT**
ユニバーサル印刷サービスを使用してすべての印刷を送ります。

**別名** UPRINT

**NOUNIVERSALPRINT**
ユニバーサル印刷サービスを使用して印刷を無効にします。

**別名** NOUPRINT

### 詳細

ユニバーサル印刷サービスにより、SAS アプリケーションとプロシジャは双方向かつ一括処理の印刷が可能になります。ODS PRINTER の割り当て先は UNIVERSALPRINT オプションあるいは NOUNIVERSALPRINT オプションのどちらが設定されていても、必要に応じてユニバーサル印刷サービスを使用します。

UNIVERSALPRINT が指定されている場合、SAS は以下の環境を設定します。

- **ユニバーサル印刷**のメニュー項目とダイアログボックスはウィンドウ環境で有効化されています。
- SAS が起動すると、PRINTERPATH=システムオプションは初期設定の null ではない値、通常はポストスクリプトレベル 1 に設定されます。
- SYSPRINT= system オプションは無視されます。PRINTERPATH=システムオプションの値は出力先を指定します。
- PRINTERPATH=""を設定することで PRINTERPATH の値を初期設定値にします。

SAS で Windows 印刷環境を使用するには、NOUNIVERSALPRINT に設定します。SAS は以下の環境を設定します。

- PRINTERPATH=システムオプションの初期値は null になり、SAS は Windows の印刷メニューとダイアログボックスを有効化します。

- PRINTERPATH=システムオプションを null ではない値に設定すると、SAS はユニバーサル印刷を使用します。SAS は Windows の印刷ダイアログボックスを使用しますが、ダイアログボックス内で参照されるプリンターは使用されません。
- PRINTERPATH=""のとき、SAS は Windows 印刷ダイアログボックスを使用し、SYSPRINT=システムオプションは出力先を指定します。

### 関連項目:

- “Universal Printing” (*SAS Language Reference: Concepts* 15 章)

#### システムオプション

- “PRINTERPATH= System Option” (*SAS System Options: Reference*)
- “SYSPRINT システムオプション:Windows” (578 ページ)
- “UPRINTCOMPRESSION System Option” (*SAS System Options: Reference*)

## UPRINTMENUSWITCH システムオプション:Windows

**ファイル**メニューでユニバーサル印刷コマンドを有効化します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | ログとプロシジャ出力制御:ODS 出力 |
| **PROC OPTIONS GROUP=** | ODSPRINT |
| **デフォルト:** | NOUPRINTMENUSWITCH |
| **Windows 固有:** | すべて |

### 構文

**-UPRINTMENUSWITCH** | **-NOUPRINTMENUSWITCH**

#### *構文記述*

**UPRINTMENUSWITCH**
**ファイル**メニューで印刷コマンドを指定すると、**ユニバーサル印刷**ダイアログボックスが表示されます。

**NOUPRINTMENUSWITCH**
**ファイル**メニューで印刷コマンドを指定すると、Windows のダイアログボックスが表示されます。

### 詳細

ユニバーサル印刷メニューおよびダイアログボックスを使用できるようにするには、SAS 起動時に UNIVERSALPRINT システムオプションと UPRINTMENUSWITCH システムオプションを両方指定する必要があります。UNIVERSALPRINT オプションを指定せずに UPRINTMENUSWITCH オプションを指定した場合、ユニバーサル印刷メニューおよびダイアログボックスは表示されません。

### 関連項目:

“Windows 環境での SAS の印刷について” (165 ページ)

## USER システムオプション:Windows

デフォルトの永久 SAS データライブラリ名を指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES

**デフォルト:** なし

**Windows 固有:** 次に対する有効な値:*library-specification*

**参照項目:** “USER= System Option” (*SAS System Options: Reference*)

### 構文

**-USER** "*library-specification*"

**USER=**"*library-specification*"

#### *必須引数*

***library-specification***
SAS セッション中に作成されるデータセットを格納するための、デフォルトのライブラリ参照名、環境変数または Windows パス名を指定します。パス名は、ディレクトリまたはサブディレクトリのレベルまでであることに注意してください。*library-specification* の値には、有効な Windows パス名を指定してください。

### 詳細

USER システムオプションを指定すると、1 レベル名で作成したデータセットはすべて、指定されたライブラリに永久に保存されます。一時的なデータセットを作成する場合は、最初の部分に WORK を使用した 2 レベル名(たとえば `work.tempdata` など)を指定します。

## USERICON システムオプション:Windows

ユーザー定義のアイコンに対応したリソースファイルのパス名を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** なし

**Windows 固有:** すべて

## 構文

**-USERICON** *icon-resource-filename number-of-icons*

### *必須引数*

***icon-resource-filename***
ユーザー定義のアイコンに対応したリソースファイルの、Windows 完全パス名を指定します。パス名にスペースが含まれる場合は、パス名を引用符で囲む必要があります。

***number-of-icons***
指定したリソースファイルに格納するアイコンの最大数を指定します。

## 詳細

USERICON システムオプションでは、アイコンに対応したリソースファイルの完全パス名および指定したリソースファイルに格納するアイコンの最大数を指定します。

アイコンのファイルは、Win32 Software Development Kit (SDK)を使用してコンパイルする必要があります。詳細は、SDK のドキュメントを参照してください。ユーザー定義のアイコンは、SAS/AF または SAS/EIS で作成したアプリケーションに組み込むことができます。

## 例

次の USERICON システムオプションでは、C:\MYSTUFF\MYICONS.DLL に格納する 10 個のアイコンを指定します。

```
-usericon c:\mystuff\myicons.dll 10
```

# VERBOSE システムオプション:Windows

構成ファイルに指定されているすべてのシステムオプションの設定を、端末とバッチログのどこに書き出すかを制御します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | ログとプロシジャ出力制御:SAS ログ |
| **PROC OPTIONS GROUP=** | LOGCONTROL |
| **デフォルト:** | NOVERBOSE |
| **Windows 固有:** | レポートされる情報の量 |

## 構文

**-VERBOSE** | **-NOVERBOSE**

### *構文記述*

**VERBOSE**
システムオプションの設定をログに書き出します。

**NOVERBOSE**
システムオプションの設定をログに書き出さないように指定します。NOVERBOSE はデフォルトの設定です。

### 詳細

VERBOSE システムオプションでは、SAS 起動時にコマンドラインまたは構成ファイルで設定された SAS システムオプションの設定を書き出すことができます。ターミナルから SAS を起動すると、設定はターミナルに表示されます。バッチジョブの一部として SAS を起動すると、設定はバッチログに書き出されます。VERBOSE システムオプションでは、SAS システムオプションの設定は変更できません。

VERBOSE システムオプションは、エラー診断ツールとして使用できます。SAS システムの起動時にエラーメッセージが表示された場合は、このオプションを使用して、システムオプションの指定にエラーがないかどうかを確認できます。

### 関連項目:

"The SAS Log" (*SAS Language Reference: Concepts* 9 章)

## VIEWMENU システムオプション

SAS ウィンドウに**表示**メニューが含まれるかどうかを指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | 環境制御:表示 |
| **PROC OPTIONS GROUP=** | ENVDISPLAY |

### 構文

**VIEWMENU** | **NOVIEWMENU**

#### *構文記述*

**VIEWMENU**
SAS ウィンドウに**表示**メニューが含まれるように指定します。

**NOVIEWMENU**
SAS ウィンドウに**表示**メニューが含まれないように指定します。

## WEBUI システムオプション:Windows

Web 拡張機能を有効にします。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動 |
| **カテゴリ:** | 入力制御:データ処理 |
| **PROC OPTIONS GROUP=** | INPUTCONTROL |
| **デフォルト:** | NOWEBUI |
| **Windows 固有:** | すべて |

## 構文

**-WEBUI** | **-NOWEBUI**

### *構文記述*

**WEBUI**
Web 拡張機能を有効にします。

**NOWEBUI**
specifies to disable web enhancements.

## 詳細

Microsoft Internet Explorer を インストールし、WEBUI システムオプションを指定すると、SAS の**エクスプローラ**ウィンドウなどの特定のウィンドウが IE の Web ページのように機能します。たとえば、マウスでオブジェクトをポイントすると、そのオブジェクトが選択され、マウスをクリックするとオブジェクトのデフォルトの操作が呼び出されます。

オブジェクトを範囲選択するには、**Shift** キーを押しながら、グループの先頭と末尾のオブジェクトをクリックします。

複数の項目を選択するには、Ctrl キーを押しながら、グループ内の各項目をクリックします。

# WINDOWSMENU システムオプション:Windows

メニューを表示するウィンドウに、**ウィンドウ**メニューを表示するかどうかを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** NOWINDOWSMENU

**Windows 固有:** すべて

## 構文

**-WINDOWSMENU** | **-NOWINDOWSMENU**

**WINDOWSMENU** | **NOWINDOWSMENU**

### *構文記述*

**WINDOWSMENU**
NOAWSMENUMERGE システムオプションを指定した場合、メインメニューに**ウィンドウ**メニューが表示されます。

**NOWINDOWSMENU**
NOAWSMENUMERGE システムオプションを指定した場合、メインメニューに**ウィンドウ**メニューが表示されません。

## 詳細

WINDOWSMENU システムオプションは、NOAWSMENUMERGE システムオプションを指定した場合にのみ有効です。

## 関連項目:

“AWSMENUMERGE システムオプション:Windows” (495 ページ)

# WORK システムオプション:Windows

Work ライブラリの場所を指定します。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:ファイル

**PROC OPTIONS GROUP=** ENVFILES

**デフォルト:** !TEMP\SAS Temporary Files

**Windows 固有:** すべて

**参照項目:** “WORK= System Option” (*SAS System Options: Reference*)

## 構文

**-WORK** *filename* | *ディレクトリ*

### *必須引数*

***filename***
ディレクトリやオプションキーワードのリストを含むファイルを指定します。SAS ではディレクトリをファイルに含まれているリストから選択します。このディレクトリは、現在の SAS セッションの Work ライブラリの場所です。

***ディレクトリ***
現在の SAS セッションの Work ライブラリの場所であるディレクトリを指定します。

## 詳細

filename オプションを使用すると、指定したファイルが開かれ、Work ライブラリの場所として使用するディレクトリの 1 つが選択されます。ディレクトリは、ランダムに選択されるか、または利用可能な容量に基づいて選択されます。選択を行うには、METHOD キーワードを使用します。

ディレクトリオプションを使用すると、指定したディレクトリを Work ライブラリの場所として使用して、初期化が続行されます。

filename オプションには、Work ライブラリ用に使用できるディレクトリのリストを含めます。個々の SAS Work ライブラリは引き続き単一のディレクトリに存在します。METHOD=RANDOM を使用すると、Work ライブラリ用のディレクトリがディレクトリのリストからランダムに選択されます。Work ライブラリの場所として、1 セッションに 1 つのディレクトリが選択されます。この選択により、複数のハードウェアシステム間で I/O 負荷のバランスを調整できます。METHOD=SPACE を使用すると、利用可能な容量が最も大きいディレクトリが指定されます。METHOD キーワードを指定していない場合は、デフォルトでディレクトリはランダムに選択されます。

### 関連項目:

"Work データライブラリ " (23 ページ)

## XCMD システムオプション Windows

現在の SAS セッションで X コマンドが有効になるようにします。

**該当要素:** 構成ファイル、SAS 起動

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** XCMD

**Windows 固有:** すべて

### 構文

**-XCMD** | **-NOXCMD**

**-XCMD ON** | **-XCMD OFF**

#### *構文記述*

**XCMD or XCMD ON**
現在の SAS セッションで X コマンドを有効にします。

**NOXCMD or XCMD OFF**
現在の SAS セッションで X コマンドを無効にします。

### 詳細

XCMD を指定すると、現在の SAS セッションで X コマンドを有効にすることができます。

NOXCMD を指定すると、次の要素が無効化されます。

- FILENAME ステートメントの PIPE デバイスタイプと NAMEPIPE デバイスタイプ
- CALL SYSTEM ルーチン
- X コマンド
- 動的データ交換(DDE:Dynamic Data Exchange)
- %SYSEXEC マクロ
- SYSTASK ステートメント
- FILENAME 関数

### 関連項目:

"X コマンド:Windows" (374 ページ)

## XMIN システムオプション:Windows

X コマンドに指定したアプリケーションを最小化したウィンドウの状態で開くか、またはアクティブウィンドウの状態(デフォルト)で開くかを指定します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** NOXMIN

**Windows 固有:** すべて

### 構文

**-XMIN** | **-NOXMIN**

**XMIN** | **NOXMIN**

#### *構文記述*

**XMIN**
X コマンドに指定したアプリケーションを最小化したウィンドウの状態で起動します。

**NOXMIN**
X コマンドに指定したアプリケーションをアクティブウィンドウの状態(デフォルト)で起動します。

### 詳細

XMIN システムオプションを使用すると、X コマンドに指定したアプリケーションを最小化したウィンドウまたはアクティブウィンドウの状態(デフォルト)で開くことができます。

## XSYNC システムオプション:Windows

X コマンドまたは X ステートメントの実行を同期とするか非同期とするかを制御します。

**該当要素:** 構成ファイル、SAS 起動、OPTIONS ステートメント、[**SAS システムオプション**]ウィンドウ

**カテゴリ:** 環境制御:表示

**PROC OPTIONS GROUP=** ENVDISPLAY

**デフォルト:** XSYNC

**Windows 固有:** すべて

### 構文

**-XSYNC** | **-NOXSYNC**

**XSYNC** | **NOXSYNC**

### *構文記述*

**XSYNC**
オペレーティングシステムコマンドの実行を SAS セッションと同期とする場合に指定します。つまり、コマンドが完了するまでは、SAS に制御が戻りません。X コマンドまたは X ステートメントによって開始したプロセスが終了するまで、SAS セッションに戻ることはできません。XSYNC はデフォルトの設定です。

**NOXSYNC**
オペレーティングシステムコマンドの実行を SAS セッションと非同期とする場合に指定します。つまり、制御はただちに SAS に戻され、コマンドの実行中も SAS セッションは中断されません。NOXSYNC を有効にすると、X コマンドまたは X ステートメントを実行し、その結果として開始されたプロセスの終了を待たずに SAS セッションに戻ることができます。

## 詳細

XSYNC システムオプションの値は、次の要素の実行に影響します。

- X ステートメント
- X コマンド
- CALL SYSTEM ルーチン
- %SYSEXEC ステートメント

## 関連項目:

- “SAS から Windows または MS-DOS のコマンドを実行する ” (28 ページ)
- “XWAIT システムオプション:Windows” (592 ページ)
- “X ステートメント:Windows” (467 ページ)
- “X コマンド:Windows” (374 ページ)
- “CALL SYSTEM ルーチン:Windows” (391 ページ)
- “マクロステートメント” (601 ページ)

# XWAIT システムオプション:Windows

DOS シェルを閉じる前に、DOS プロンプトに対して EXIT を入力するかどうかを指定します。

| | |
|---|---|
| **該当要素:** | 構成ファイル、SAS 起動、OPTIONS ステートメント |
| **カテゴリ:** | 環境制御:表示 |
| **PROC OPTIONS GROUP=** | ENVDISPLAY |
| **デフォルト:** | XWAIT |
| **Windows 固有:** | すべて |

## 構文

**-XWAIT** | **-NOXWAIT**

**XWAIT** | **NOXWAIT**

### 構文記述

**XWAIT**
SAS セッションに戻るために、EXIT の入力を必要とするのを指定します。XWAIT はデフォルトの設定です。

**NOXWAIT**
指定したコマンドの実行後に、コマンドプロセッサによって自動的に SAS セッションに戻るのを指定します。EXIT を入力する必要はありません。

## 詳細

XWAIT システムオプションは、Excel などの Windows アプリケーションには影響を与えません。**コマンドプロンプト** ウィンドウで実行するアプリケーションのみに適用されます。

XWAIT システムオプションは、次の要素から起動した**コマンドプロンプト**ウィンドウに影響します。

- X ステートメント
- X コマンド
- CALL SYSTEM ルーチン
- %SYSEXEC ステートメント

## 関連項目:

- “SAS から Windows または MS-DOS のコマンドを実行する ” (28 ページ)
- “XSYNC システムオプション:Windows” (591 ページ)
- “X ステートメント:Windows” (467 ページ)
- “X コマンド:Windows” (374 ページ)
- “CALL SYSTEM ルーチン:Windows” (391 ページ)
- “マクロステートメント” (601 ページ)

*23 章*

# Windows 版での変数の長さと精度

**Windows 版での変数の長さと精度** . . . . . 595
**数値変数** . . . . . 595
**文字変数** . . . . . 596

## Windows 版での変数の長さと精度

SAS の数値と文字データ表現の保存方法についての詳細については、*SAS 言語リファレンス: 解説編*を参照してください。データ表現のトピックは、動作環境によって異なります。ここでは、Windows 版 SAS でのデータ表現法について説明します。

- “数値変数” (595 ページ)
- “文字変数” (596 ページ)

## 数値変数

SAS データセットにおけるデフォルトの数値変数の長さは 8 バイトです。SAS の数値変数の長さは、DATA ステップの LENGTH ステートメントで制御することができます。Windows 版 SAS では、8 バイト長の数値の Windows データタイプは LONG REAL です。浮動小数点値の精度は、ほぼ 15 桁までです。数値によっては、精度は 16 桁になることがあります。Windows データタイプ LONG REAL の表記に関する詳細については、Intel 社の開発者向け Web サイトを参照してください。表 23.1 (595 ページ) SAS 変数値に保存できる有効桁と最大整数値を指定します。

***表 23.1*** *Windows 版 SAS 変数の長さ別有効桁および最大整数値*

| 長さ(バイト) | 正確に表示される整数の最大値 | 指数表記 | 保存される有効桁 |
|---|---|---|---|
| 3 | 8,192 | $2^{13}$ | 3 |
| 4 | 2,097,152 | $2^{21}$ | 6 |

| 長さ(バイト) | 正確に表示される整数の最大値 | 指数表記 | 保存される有効桁 |
|---|---|---|---|
| 5 | 536,870,912 | $2^{29}$ | 8 |
| 6 | 137,438,953,472 | $2^{37}$ | 11 |
| 7 | 35,184,372,088,832 | $2^{45}$ | 13 |
| 8 | 9,007,199,254,740,992 | $2^{53}$ | 15 |

たとえば常に 0 から 100 までの値をとることがわかっている数値変数の場合、保存される数値の長さを 3 バイトにすることで、データセット内のスペースを節約することができます。次に、例を示します。

```
data mydata;
   length num 3;
   more data lines
run;
```

*注:* ダミー変数(0 または 1 のみを保存することを目的とする変数)は、3 バイト長の変数に格納できます。

***注意:***
**3 バイトに制限された変数は、その値が小さな整数である場合にだけ使用してください。整数ではなく実数を保存している場合、できるだけ確実に精度を高めるため長さを 8 バイトに指定してください。** 変数長を 8 バイト未満にした場合、変数値が大きくなると、または有効桁が多くなると、算術演算の結果を保存するときに精度が失われる可能性があります。

変数の最大数は、最初に検出された項目により制限されます。

- オブザベーション長
- 名前、ラベル、メタデータに使用可能な保存域の合計
- データセットが保存されるマシンの使用可能メモリ量

32 ビットプラットフォームでは 2 GB まで、64 ビットプラットフォームでは約 2*46 バイトまでのオブザベーション長を持つデータセットを定義できます。オブザベーション長は、BUFSIZE オプションの値を超えることはできません。

文字セットが 1 バイトで、各変数の名前やラベルなどのデータに最大で 352 バイトを使用する場合、変数の最大数は約 1,350,000 となります。名前、ラベル、出力形式名の合計がそれより短い場合は、66,666,666 を超える変数を保存できます。変数名やその他のメタデータ(データセットラベル、圧縮名などのデータ)をすべて保存するためには最大で 1GB が必要です。最大が 352 バイトとなるのは、ラベルの値の 256 バイトに加えて、入力形式、出力形式、変数名について 32 バイトずつ加算されるからです。

これらの制限を超えない場合、変数の最大可能数は 64 ビットホストで 2GB です。

## 文字変数

Windows 版 SAS では、文字値は ASCII 照合順序を用いてソートされます。先に述べた数値のダミー変数のかわりとして、同じ目的で機能する長さ 1 バイトの文字変数を選択できます。

変数の最大数は、最初に検出された項目により制限されます。

- オブザベーション長
- 名前、ラベル、メタデータに使用可能な保存域の合計
- データセットが保存されるマシンの使用可能メモリ量

32 ビットプラットフォームでは 2 GB まで、64 ビットプラットフォームでは約 2*46 バイトまでのオブザベーション長を持つデータセットを定義できます。オブザベーション長は、BUFSIZE オプションの値を超えることはできません。

文字セットが 1 バイトで、各変数の名前やラベルなどのデータに最大で 352 バイトを使用する場合、変数の最大数は約 4,050,000 となります。名前、ラベル、出力形式名の合計がそれより短い場合は、200,000,000 を超える変数を保存できます。変数名やその他のメタデータ(データセットラベル、圧縮名などのデータ)をすべて保存するためには最大で 1GB が必要です。最大が 352 バイトとなるのは、ラベルの値の 256 バイトに加えて、入力形式、出力形式、変数名について 32 バイトずつ加算されるからです。

これらの制限を超えない場合、変数の最大可能数は 64 ビットホストで 2GB です。

*24 章*

# Windows 版に固有の SAS マクロ機能

**Windows 版に固有の SAS マクロ機能** . . . . . **599**
**自動マクロ変数** . . . . . **599**
**マクロステートメント** . . . . . **601**
**マクロ関数** . . . . . **602**
**自動呼び出しライブラリ** . . . . . **602**
自動呼び出しライブラリの概要 . . . . . 602
SASAUTOS システムオプション . . . . . 602

## Windows 版に固有の SAS マクロ機能

通例では、SAS マクロ言語は複数のオペレーティングシステム間で移植可能です。このセクションでは、システムに依存するマクロ機能のコンポーネントについて説明します。詳細については、*SAS マクロ言語: リファレンス*を参照してください。

次に示すマクロ機能は Windows 版に特有のものです。

- “自動マクロ変数” (599 ページ)
- “マクロステートメント” (601 ページ)
- “マクロ関数” (602 ページ)
- “自動呼び出しライブラリ” (602 ページ)

*注:* CON、NUL、PRN、COM1 から COM9 まで、そして LPT1 から LPT9 までは Windows 版での予約語です。これらの予約語を、マクロ変数の名前として使用することはできません。

## 自動マクロ変数

次に示す自動マクロ変数は Windows 版に特有の値を含みます。

SYSCC
SAS の終了時に、SAS が Windows に戻す現在の SAS 条件コードを含みます。SAS の終了時に、SAS はこの条件コードを動作環境で解釈できる戻り値に変換します。

*注:* ERRORCHECK=NORMAL の場合、LIBNAME または FILENAME ステートメントまたは SAS/SHARE の LOCK ステートメントにエラーが存在する場合でも、戻り値は 0 になります。また、ファイルが存在しないため%INCLUDE ステートメントが失敗した場合でも、SAS ジョブまたはセッションは終了しません。詳細については、次の ERRORCHECK=システムオプションを参照してください。*SAS システムオプション: リファレンス*

SYSDEVIC
現在のグラフィックデバイスの名前を付与します。現在のグラフィックデバイスは、DEVICE システムオプションによって決定されます。サイトで使用可能なグラフィックデバイスを確認するには、各サイトの SAS 担当者に問い合わせてください。DEVICE システムオプションの詳細については、"DEVICE システムオプション:Windows" (501 ページ) および *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。

SYSENV
Windows では、**FORE** または **BACK** の各値が含まれます。Windows で EG を実行している場合、SYSENV の値は **BACK** です。

SYSJOBID
Windows 上の SAS タスクを特定する数字を返します。

SYSMAXLONG
Windows 上で許容される最大長の整数値を返します。その値は 2,147,483,647 です。

SYSRC
SAS セッション中に実行される Windows コマンドでの Windows のステータスを保持します。この変数には、文字列として Windows コマンドステータスの 10 進値が格納されます。

たとえば、次のステートメントを考えます。

```
options noxwait;
x 'dirf'; /* Invalid Windows command */
%put This Windows status is &sysrc; x 'dir'; /* Valid Windows command  */
%put The corrected Windows status is &sysrc;
```

次の行が SAS ログに書き込まれます。`This Windows status is 1 The corrected Windows status is 1`

この OPTIONS ステートメントでは XWAIT オプションが無効化されているため、Windows のコマンドプロンプトウィンドウが自動的に終了します。SAS セッションに戻るために、**exit** を入力する必要はありません。"This Windows status is"の値は 1、また"The corrected Windows status is"の値も 1 になります。XWAIT オプションを使用してこの例を実行する場合は、SAS でコードを実行する前に **exit** を入力する必要があります。**exit** を入力した後に、ステートメント"This Windows status is"では値 9009 が返されます。また、"The corrected Windows status is"の値は 0 です。NOXSYNC システムオプションを使用した場合、SYSRC の値は自動的に 0 になります。

SYSSCP
動作環境の省略形 WIN を返します。

SYSSCPL
使用中の特定の Windows 環境の名前を返します。次のような値が戻されます。

W32_DSRV08
Microsoft Windows 2008 Datacenter Server

W32_ESRV08
Microsoft Windows 2008 Enterprise Server

W32_SRV08
Microsoft Windows Server 2008

W32_VSPRO
Microsoft Windows 7

X64_8PRO
Windows 8 Professional または Enterprise

X64_8HOME
Windows 8

X64_DSRV12
Windows Server 2012 DataCenter

X64_SRV12
Windows Server 2012 Standard

X64_VSPRO
Microsoft Windows 7 x64

X64_DSRV08
Microsoft Windows Server 2008 Datacenter 64-bit edition for x64-based computers

X64_ESRV08
Microsoft Windows Server 2008 Enterprise 64-bit edition for x64-based computers

X64_SRV08
Microsoft Windows Server 2008 Standard 64-bit edition for x64-based computers

## マクロステートメント

次のマクロステートメントは Windows 版固有の動作です。

%SYSEXEC
動作環境のコマンドを即時実行し、リターンコードを SYSRC 自動マクロ変数に格納します。%SYSEXEC ステートメントは、“SAS からの Windows コマンドまたは MS-DOS コマンド実行の概要” (28 ページ) で説明する X ステートメントと同様です。%SYSEXEC ステートメントは、マクロ内部またはオープンコードで使用できます。%SYSEXEC ステートメントの構文は次のとおりです。

%SYSEXEC <*command*>;

*コマンド*引数には、動作環境の任意のコマンドまたは動作環境のコマンドを生成する一連のマクロ操作を指定できます。*コマンド*引数を使用して、メモ帳などの Windows アプリケーションを起動することもできます。

*コマンド*引数を省略すると、対話型のコマンドプロンプトサブプロセスが起動します。SAS セッションに戻るには、コマンドプロンプトに EXIT と入力して Enter キーを押します。SYSEXEC ステートメントで*コマンド*引数を省略した場合、SYSRC 自動変数には 0 が設定されます。

次に、簡単な%SYSEXEC の例を示します。`%sysexec time;`

このステートメントは次に示す行を表示するコマンドプロンプトセッションを起動します。`The current time is:16:32:45.16 Enter new time:`

*注:* %SYSEXEC ステートメントでは、X ステートメントおよび X コマンドと同じように、XSYNC および XWAIT システムオプションが使用されます。これらのシステムオプションに関する詳細については、“XSYNC システムオプション:Windows” (591 ページ) および “XWAIT システムオプション:Windows” (592 ページ) を参照してください。

## マクロ関数

%SYSGET マクロ関数の動作は、Windows 版に固有の機能です。

%SYSGET
引数として渡される Windows 環境変数の値(文字列)を返します。Windows 環境変数と SAS 環境変数は、どちらも%SYSGET 関数を使用して変換できます。環境変数が存在しない場合は、警告メッセージが表示されます。次に、%SYSGET 関数の構文を示します。

%SYSGET(*環境変数名*);

%SYSGET 機能を使用するための例を示します。

```
%let var1=%sysget(comspec);
%put The COMSPEC environment variable is &var1;
```

次の行が SAS ログに書き込まれます。COMSPEC 環境変数は C:\Windows\system32\cmd.exe です。

## 自動呼び出しライブラリ

### *自動呼び出しライブラリの概要*

このセクションでは、自動呼び出しライブラリの使用方法のうちでシステムに固有な方法について説明します。概略については、*SAS マクロ言語: リファレンス*を参照してください。

自動呼び出しライブラリには、SAS マクロを定義するファイルが格納されています。SAS 提供の自動呼び出しマクロもあります。自動呼び出し機能を使用するには、SAS システムオプションの MAUTOSOURCE を設定します。SAS のインストール時に、SAS 構成ファイルに SASAUTOS システムオプションが使用されている場合、SAS 提供のデフォルトマクロの検索場所が指定されます。独自の自動呼び出しマクロを定義して、Windows ディレクトリに格納することもできます。

自動呼び出しマクロを Windows ディレクトリに格納する場合は、ファイル拡張子を SAS にする必要があります。ディレクトリ内の各マクロファイルには、ファイル名と同じマクロ名が付いたマクロ定義が存在する必要があります。たとえば、ディレクトリ内に格納された PRTDATA.SAS という名前のファイルには、PRTDATA という名前のマクロを定義する必要があります。

### *SASAUTOS システムオプション*

SAS プログラムにユーザー独自の自動呼び出しマクロを使用するには、SASAUTOS システムオプションを使用して、この独自の自動呼び出しマクロの検索場所を指定する必要があります。SASAUTOS オプションの構文は “SASAUTOS システムオプション:Windows” (556 ページ) で得られます。

SASAUTOS システムオプションは、SAS の起動時に設定したり、SAS セッション中に OPTIONS ステートメントで使用したりすることもできます。また、次の例のように SAS 構成ファイルを編集すると、独自の自動呼び出しライブラリを SAS 提供のライブラリに追加できます。

```
-sasautos ("c:\mymacros")
          "!sasroot\core\sasmacro"
          "!sasroot\base\sasmacro"
          "!sasroot\stat\sasmacro"
          more library specifications
          )
```

自動呼び出しライブラリは、ユーザーが指定した順序で検索されます。前の例の SASAUTOS オプション設定を使用して PRTDATA という名前のマクロを呼び出した場合、ディレクトリ C:\MYMACROS でマクロが検索されてから、各!SASROOT ライブラリが検索されます。

# 4 部

# 付録

*付録 1*
***Windows 版で OLE オブジェクトを自動化するための SCL メソッド*** . . . . . *607*

*付録 2*
***Windows 版に固有の SAS エラーメッセージ*** . . . . . *617*

*付録 3*
***Windows 版でのグラフィックに関する注意点*** . . . . . *625*

*付録 4*
***インタラクティブ SAS セッションにおける Windows 版でのデフォルトキー設定*** . . . . . *627*

*付録 5*
***Cleanwork ユーティリティ*** . . . . . *637*

*付録 6*
***Sasiotest ユーティリティ*** . . . . . *641*

*付録 7*
***ASCII システムでの EBCDIC データの使用*** . . . . . *647*

*付録 1*

# Windows 版で OLE オブジェクトを自動化するための SCL メソッド

**OLE クラスメソッドの要約** . . . . . **607**

**ディクショナリ** . . . . . **608**
_COMPUTE_ . . . . . 608
_DISABLE_DEFAULT_ACTION_ . . . . . 609
_DO_ . . . . . 609
_ENABLE_DEFAULT_ACTION_ . . . . . 610
_EXECUTE_ . . . . . 610
_GET_EVENT_ . . . . . 611
_GET_PROPERTY_ . . . . . 611
_GET_REFERENCE_ID_ . . . . . 612
_GET_TYPE_ . . . . . 613
_IN_ERROR_ . . . . . 613
_NEW_ . . . . . 614
_SET_PROPERTY_ . . . . . 614
_UPDATE_ . . . . . 615

## OLE クラスメソッドの要約

表 A1.1 (607 ページ) (OLE)がリンクし、かつ埋め込まれているオブジェクトを用いてユーザーが使用可能な SCL メソッドのリストを含みまた、SCL メソッドが適用される OLE クラスがどれかを表示します。

***表 A1.1*** *OLE および OLE オートメーションに有効な SCL メソッド*

| メソッド | SAS OLE クラス | SAS OLE オートメーションクラス |
|---|---|---|
| _COMPUTE_ | Yes | Yes |
| _DISABLE_DEFAULT_ACTION_ | Yes | No |
| _DO_ | Yes | Yes |
| _ENABLE_DEFAULT_ACTION_ | Yes | No |
| _EXECUTE_ | Yes | No |

| メソッド | SAS OLE クラス | SAS OLE オートメーションクラス |
|---|---|---|
| _GET_EVENT_ | Yes | No |
| _GET_PROPERTY_ | Yes | Yes |
| _GET_REFERENCE_ID_ | Yes | Yes |
| _GET_TYPE_ | Yes | No |
| _IN_ERROR_ | Yes | Yes |
| _NEW_ | No | Yes |
| _SET_PROPERTY_ | Yes | Yes |
| _UPDATE_ | Yes | No |

*注:* _NEW_メソッドはあらゆるクラスに使用できますが、OLE オートメーションクラスは特殊な要件のためにこのメソッドを上書きします。

これらのメソッドの参照情報については、残りのセクションを参照してください。

# ディクショナリ

## _COMPUTE_

OLE オートメーションオブジェクト上の 1 つのメソッドを呼び出し、値を 1 つ返します。

### 構文

**CALL NOTIFY**(*OLE-object-name*,'_COMPUTE_',*in-OLE-method*<,*in-parm*…,*in-parm*> ,*out-value*);

**CALL SEND**(*OLE-object-id*,'_COMPUTE_',*in-OLE-method*<,*in-parm*…,*in-parm*> ,*out-value*);

### 詳細

***表 A1.2*** *OLE 引数*

| 引数 | 文字(C)または数字(N) | 説明 |
|---|---|---|
| *in-OLE-method* | C | OLE メソッド名を指定します。 |
| *in-parm* | C または N | OLE メソッドへパラメータを与えます。 |
| *out-value* | C または N | OLE メソッドから返される値を格納します。 |

_COMPUTE_メソッドは、OLE オートメーションサーバーで公開されたメソッドを、パラメータを使用して呼び出します。必要なパラメータ(*in-parm* 引数)の数は、オブジェクトやメソッドによって異なります。戻り値を持つメソッドだけが_COMPUTE_メソッドで使用できます。戻り値を持たないメソッドは、_DO_メソッドで使用します。

### 例

次の例では、OLE コントロールのポジション 2 にあるアイテムの内容を、変数 *item2obj* に格納します。

```
length item2obj $ 200;
call notify('oleobj', '_COMPUTE_',
            'GetItem', 2, item2obj);
```

次の例では、スプレッドシートオブジェクトのセルメソッドを使用して、行 2 と列 5 のセルの位置を計算し、そのセルの値を 100 に設定します。

```
call send(oleobj, '_COMPUTE_', 'Cells',
          2, 5, cellobj1);
call send(cellobj1, '_SET_PROPERTY_',
          'Value', 100);
```

## _DISABLE_DEFAULT_ACTION_

OLE オブジェクトのデフォルト動作を無効化します。

### 構文

**CALL NOTIFY**(*OLE-object-name*,'_DISABLE_DEFAULT_ACTION_');

### 詳細

このメソッドにより、OLE オブジェクトのためのデフォルト Verb がオブジェクトをダブルクリックしても実行されないようにします。デフォルトでは、デフォルト操作が有効になります。

## _DO_

OLE オートメーションオブジェクト上のメソッドを戻し値なしで呼び出します。

### 構文

**CALL NOTIFY**(*OLE-object-name*,'_DO_',*in-OLE-method*<,*in-parm*…,*in-parm*> );

**CALL SEND**(*OLE-object-id*,'_DO_',*in-OLE-method*<, *in-parm*…,*in-parm*> );

### 詳細

*表 A1.3 OLE 引数*

| 引数 | 文字(C)または数字(N) | 説明 |
|---|---|---|
| *in-OLE-method* | C | OLE メソッド名を指定します。 |
| *in-parm* | C または N | OLE メソッドへパラメータを与えます。 |

_DO_メソッドは、OLE オートメーションサーバーで公開されたメソッドを、そのパラメータを使用して呼び出します。必要なパラメータ(*in-parm* 引数)の数は、OLE オブジェクトやメソッドによって異なります。値を戻さないメソッドだけが_DO_メソッドで使用できます。値を戻すメソッドは、_COMPUTE_メソッドで使用します。

### 例

次の例では AboutBox メソッドを OLE コントロールに送信し、コントロールのバージョン情報ボックスを表示します。

```
call notify('oleobj', '_DO_', 'AboutBox');
```

## _ENABLE_DEFAULT_ACTION_

OLE オブジェクトのデフォルト動作を有効化します。

### 構文

**CALL NOTIFY**(*OLE-object-name*,'_ENABLE_DEFAULT_ACTION_');

### 詳細

このメソッドを使用すると、OLE オブジェクトをダブルクリックしたときに、オブジェクトのデフォルト Verb が実行されます。デフォルトでは、デフォルト操作が有効になります。

## _EXECUTE_

オブジェクトの OLE Verb を実行します。

### 構文

**CALL NOTIFY**(*OLE-object-name*,'_EXECUTE_',*in-verb*<,*in-verb*…*in-verb*> );

### 詳細

*表 A1.4 OLE 引数*

| 引数 | 文字(C)または数字(N) | 説明 |
|---|---|---|
| *in-verb* | C | 実行する OLE Verb を指定します。 |

OLE オブジェクトが作成されていれば、このオブジェクトで使用できる Verb の一覧は、OLE オブジェクトの**関連する Verb** ウィンドウに表示されます。一度に複数の OLE Verb を指定することができます。

オブジェクトに無効な Verb を実行しようとすると、SCL プログラムが中断され、Verb が存在しないというメッセージが表示されます。

## _GET_EVENT_

最後に受信した OLE コントロールイベントの名前を返します。

### 構文

**CALL NOTIFY**(*OLE-object-name*,'_GET_EVENT_',*out-event*);

### 詳細

*表 A1.5 OLE 引数*

| 引数 | 文字(C)または数字(N) | 説明 |
|---|---|---|
| *out-event* | C | 最後に受信した OLE コントロールイベントの戻された名前を格納します。 |

このメソッドは、指定した OLE コントロールから最後に送信されたイベントの名前を返します。

## _GET_PROPERTY_

オートメーションオブジェクトのプロパティの値を返します。

### 構文

**CALL NOTIFY**(*OLE-object-name*,'_GET_PROPERTY_',*in-OLE-property*,*out-property-value*);

**CALL SEND**(*OLE-object-id*,'_GET_PROPERTY_',*in-OLE-property*,*out-property-value*);

### 詳細

*表 A1.6 OLE 引数*

| 引数 | 文字(C)または数字(N) | 説明 |
|---|---|---|
| *in-OLE-property* | C | OLE プロパティの名前を指定します。 |
| *out-property-value* | C または N | 返されたプロパティの値を格納します。 |

_GET_PROPERTY_メソッドは、オートメーションオブジェクトのプロパティの値を取得するために使用します。

## _GET_REFERENCE_ID_

オートメーションオブジェクトメソッドで使用する参照識別子を返します。オートメーションオブジェクトメソッドでは、パラメータの 1 つとしてオートメーションオブジェクトが必要です。

### 構文

**CALL NOTIFY**(*OLE-object-name*,'_GET_REFERENCE_ID_',*out-refid*);

**CALL SEND**(*OLE-object-id*,'_GET_REFERENCE_ID_',*out-refid*);

### 詳細

*表 A1.7 OLE 引数*

| 引数 | 文字(C)または数字(N) | 説明 |
|---|---|---|
| *out-refid* | C | 返された参照識別子を格納します。 |

GET_REFERENCE_ID_メソッドは、オートメーションオブジェクトの識別子を取得するために使用します。返される値は、後続の_DO_呼び出しまたは_COMPUTE_呼び出しで使用されます。このオブジェクトメソッドはパラメータの 1 つとしてオートメーションオブジェクトを必要とします。この値はオブジェクトパラメータで使用されます。

### 例

次の例では、オートメーションオブジェクトの参照識別子が返されます。この識別子は、オブジェクト識別子を必要とするオートメーションメソッドにパラメータ値として送信されます。

```
call notify('oleobj1', '_GET_REFERENCE_ID_',
            refid);
call notify('oleobj2', '_DO_', 'NewAppl',
            refid, p1, p2);
```

## _GET_TYPE_

オブジェクトの種類を返します。

### 構文

**CALL NOTIFY**(*OLE-object-name*,'_GET_TYPE_',*out-type*);

### 詳細

***表 A1.8*** *OLE 引数*

| 引数 | 文字(C)または数字(N) | 説明 |
|---|---|---|
| *out-type* | C | 返されたオブジェクトの種類を格納します。 |

_GET_TYPE メソッドは、オブジェクトの種類を取得するために使用します。種類には、Embedded、Linked、Bitmap、Device Independent Bitmap、Picture があります。

## _IN_ERROR_

オブジェクトのエラーステータスを返します。

### 構文

**CALL NOTIFY**(*OLE-object-name*,'_IN_ERROR_',*error-status*<,*error-msg*> );

**CALL SEND**(*OLE-object-id*,'_IN_ERROR_',*error-status*<,*error-msg*> );

### 詳細

***表 A1.9*** *OLE 引数*

| 引数 | 文字(C)または数字(N) | 説明 |
|---|---|---|
| *error-status* | N | オブジェクトに対してオートメーションエラーが発生しているかどうかを示す値を返します。 |
| *error-msg* | C | オートメーションエラーメッセージを返します。 |

オートメーション呼び出し中に発生したエラーは、_IN_ERROR_を使用すると検出できます。_IN_ERROR_メソッドは、最後のオートメーション呼び出しのステータスを返します。また、後続のどのオートメーション呼び出しよりも先に呼出されます。

### 例

次の例では、前述の_GET_PROPERTY_呼び出し中に発生したエラーを検出します。

```
length errmsg $ 200;
call send(objid,'_GET_PROPERTY_',
          'ActiveObject', actobj);
call send(objid,'_IN_ERROR_',inerror, errmsg);
if inerror then
   link handle_err;
```

## _NEW_

OLE オートメーションサーバーの新しいインスタンスを作成します。

### 構文

**CALL SEND**(*OLE-instance*,'_NEW_',*new-OLE-id*,*init-arg*,*OLE-auto-app*);

### 詳細

OLE オートメーションサーバーを参照する SCL コードを使用する前に、まず、OLE オートメーションクラスのインスタンスを作成する必要があります。

_INIT_メソッドの詳細については、SAS/AF のオンラインドキュメントにあるオブジェクトクラスの説明を参照してください。

### 例

次の例では、OLE オートメーションサーバーの新しいインスタンスを作成し、新しいオブジェクトに SCL 識別子 **exclauto** を割り当てます。次の例の **Excel.Application.8** は、システムレジストリ内での Microsoft Excel の識別子です。

```
hostcl=loadclass('sashelp.fsp.hauto');
call send (hostcl, '_NEW_', exclauto, 0,
           'Excel.Application.8');
```

## _SET_PROPERTY_

オートメーションオブジェクトのプロパティに値を割り当てます。

### 構文

**CALL NOTIFY**(*OLE-object-name*,'_SET_PROPERTY_',*in-OLE-property*,*in-value*);

**CALL SEND**(*OLE-object-id*,'_SET_PROPERTY_',*in-OLE-property*,*in-value*);

## 詳細

***表 A1.10*** *OLE 引数*

| 引数 | 文字(C)または数字(N) | 説明 |
|---|---|---|
| *in-OLE-property* | C | OLE プロパティ名を指定します。 |
| *in-value* | C または N | OLE プロパティに割り当てる値を格納します。 |

_SET_PROPERTY_メソッドでは、オートメーションオブジェクトのプロパティに値を割り当てます。

# _UPDATE_

オブジェクトの現在の内容、または異なる HSERVICE エントリの内容に基づいて、オブジェクトを更新します。

## 構文

**CALL NOTIFY**(*OLE-object-name*,'_UPDATE_'<,*in-hservice*> );

## 詳細

***表 A1.11*** *OLE 引数*

| 引数 | 文字(C)または数字(N) | 説明 |
|---|---|---|
| *in-hservice* | C | オブジェクトの更新に使用する HSERVICE エントリの名前を指定します。 |

_UPDATE_メソッドは、オブジェクトを再作成し、現在の属性に基づいてオブジェクトの内容を更新します。*in-hservice* パラメータは OLE オブジェクトと一緒のみ使用され、HSERVICE カタログエントリの名前が指定されます。*in-hservice* パラメータを指定すると、*OLE-object* に指定されたオブジェクトが、*in-hservice* パラメータが参照する HSERVICE エントリに保存されたオブジェクトに変更されます。

*in-hservice* を指定せずに_UPDATE_メソッドを使用すると、オブジェクトの内容は、現在の OLE オブジェクトのソースを使用して更新されます。これは、リンクされたオブジェクトを手動で更新するのに便利です。

## 例

この例では、OBJ1 に保存されたオブジェクトを、Sasuser.Examples.Sound1.Hservice オブジェクトに置き換えます。

```
length refid $ 30;
call notify('obj1','_update_',
            'sasuser.examples.sound1.hservice');
```

*付録 2*

# Windows 版に固有の SAS エラーメッセージ

**SAS エラーメッセージの概要** . . . . . 617
**リターンコードと完了ステータス** . . . . . 617
**ファイルへのアクセス** . . . . . 618
**SAS 機能の使用** . . . . . 619
**OLE の使用** . . . . . 620
**ネットワークの使用** . . . . . 621
**内部エラーの解決** . . . . . 622
**オペレーティングシステムと Windows のエラーメッセージの解決** . . . . . 623
**初期化と終了のエラーメッセージ** . . . . . 623

## SAS エラーメッセージの概要

このセクションでは、ユーザーに役立つ完了コードとエラーメッセージについて説明します。このエラーメッセージリストでは、エラーメッセージに固定スペース文字を使用します。エラーメッセージ内のイタリック体の単語は、ファイル名や番号など可変的な項目を表します。メッセージごとには、メッセージの原因やその意味、また、その解方法が説明されています。

## リターンコードと完了ステータス

SAS ジョブ終了時に、リターンコードが、Windows バッチ変数の ERRORLEVEL に返されます。値が 0 の場合は、正常に終了したことを示します。ABORT ステートメントを使用して、ERRORLEVEL の値を指定できます。ABORT ステートメントに、オプション引数 n を整数で指定します。ABORT ステートメントには、RETURN 引数または ABEND 引数を指定することもできます。n を指定しないでステートメントを発行すると、ERRORLEVEL 変数に次の値が設定されます。

abort;
　ERRORLEVEL 変数は、3 に設定されます。

abort return;
　ERRORLEVEL 変数は、4 に設定されます。

abort abend;
ERRORLEVEL 変数は、5 に設定されます。

n 引数には、1 から 65,535 までの範囲の値を指定できます。ERRORLEVEL 変数は、Windows バッチファイルで IF コマンドの条件として使用されます。ERRORLEVEL 変数の詳細については、Windows のユーザーズガイドを参照してください。次の表に、ERRORLEVEL 変数の値をまとめます。

*注:* ERRORLEVEL 変数を調べるには、START/ WAIT コマンドを使用して SAS を起動します。**DOS** ウィンドウから、次のように入力します。

```
start/wait "title" "C:\Program Files\SASHome\SASFoundation\9.4\sas.exe"
```

***表 A2.1*** *ERRORLEVEL 変数の値*

| 状態 | 重大度 | リターンコードの値 |
|---|---|---|
| すべてのステップが正常に終了しました。 | SUCCESS | 0 |
| 警告が発行されました。 | WARNING | 1 |
| エラーが発生しました。 | ERROR | 2 |
| ユーザーが ABORT ステートメントを発行しました。 | INFORMATIONAL | 3 |
| ユーザーが ABORT RETURN ステートメントを発行しました。 | FATAL | 4 |
| ユーザーが ABORT ABEND ステートメントを発行しました。 | FATAL | 5 |
| SAS 内部エラー | INFORMATIONAL | 6 |

## ファイルへのアクセス

このセクションでは、SAS を使用してファイル(外部ファイルまたは SAS ファイル)にアクセスする際に表示されるエラーについて説明します。ファイルにアクセスするときに問題が発生した場合、FILENAME ステートメントと LIBNAME ステートメント、関数の妥当性を必ずチェックし、正しいファイルを指しているかどうかを確認してください。また、有効なファイル参照名やライブラリ参照名を指定しているかを確認してください。

**CORE カタログを初期化できません。システムの 日付/時間を確認してください。**
SAS の CORE カタログの日付またはタイムスタンプを修正してください。お使いのマシンの日時を正しく設定してください。このメッセージは 内部エラー 602 と連動して発行されます。“内部エラーの解決” (622 ページ) を参照してください。

**エラー: 日付/時刻を修正してください。**
アクセスしようとしているファイルの日付またはタイムスタンプは将来日付です。お使いのマシンの日時を正しく設定してください。

**エラー：*ファイル名*ファイルは使用されています。**
アクセスしようとしているファイルは、別の Windows アプリケーションなどの他の Windows プロセスで使用されています。

**エラー：*ファイル名-1* のロード時に、ファイルが見つかりませんでした。エラーの原因ファイル：*ファイル名-2*。**
要求されたファイルのロード時に、DLL 従属ファイルが見つかりませんでした。SAS では、SAS 構成ファイルの PATHDLL システムオプションで指定される!SASROOT\CORE\SASEXE ファイルが使用できなくなることがあります。ネットワークエラーや、ドライブ障害が原因である場合があります。PATHDLL に!SASROOT\CORE\SASDLL の場所が指定されていることを確認してください。

**エラー：メンバーまたはライブラリの *ファイル名* は使用できません。**
ファイル *ファイル名* は、別の Windows アプリケーションで使用されています。

**エラー：モジュール *モジュール名* が、検索パスにありません。**
このエラーの原因として、次が考えられます。

- SAS 構成ファイルの PATH システムオプションが正しくない
- 起動しようとした製品がインストールされていない
- 従属イメージがインストールされていない

**エラー：ライブラリ *ライブラリ名* は使用されているため、クリアまたは再割り当てできません。**
使用中のライブラリに、ライブラリ参照名を再割り当てしようとしています。

**エラー：*ファイル名* にアクセスする際に、オペレーティングシステムエラー *n* 番が発生しました。**
予期せぬリターンコードがオペレーティングシステムから戻されました。詳細については “オペレーティングシステムと Windows のエラーメッセージの解決” (623 ページ) を参照してください。

**エラー：物理ファイルは *ファイル名* に存在しません。**
アクセスしようとしているファイルが存在しません。正しいドライブとディレクトリを指定しているか確認してください。このエラーは、書き込み保護されたストレージデバイスに書き出そうとした場合にも発生します。

**エラー：メンバ (`membername`) への書き込みアクセスは拒否されました。**
書き込み保護されたストレージデバイス上のファイルか、または読み取り専用のファイルを更新しようとしています。

## SAS 機能の使用

このセクションでは、Windows 環境で SAS 言語や SAS プロシジャの機能を使用する際に発生する可能性があるエラーについて説明します。お使いのステートメントやプロシジャの構文は必ず確認してください。また、予期した結果が出ない場合、外部ファイルや SAS ファイルの内容が正しいかどうかを確認してください。

**中断されました。**
[**中断**]をクリックしたため、印刷ジョブがキャンセルされました。

**エラー：スプールのためのディスクスペースが不足しています。**
現在印刷ジョブのスプールに使用できるディスク容量が不足しています。このメッセージは、ディスク容量を確保できないことを示します。

**エラー : メモリ不足のためスプールできません。**
メモリ不足のためスプールできません。ただし、このメッセージは、ある時点でより多くの領域を使用できるようになることを示唆します。

**警告 : SAS オプション*オプション名*は、SAS システムの起動時または SAS 内の環境の起動時のみに有効です。オプションを無視しました。**
一部の SAS システムオプションは、SAS 構成ファイルまたは SAS コマンドでのみ設定できます。詳細については、"Windows 版に固有の SAS システムオプション" (473 ページ) を参照してください。

**エラー : 構成オプション*オプション名*が不明です。**
SAS コマンドまたは SAS 構成ファイルで、無効な SAS システムオプションを確認してください。Windows のシステムオプションに関する詳細については、"Windows 版に固有の SAS システムオプション" (473 ページ) を参照してください。

**エラー : プリンタマネージャによりジョブが終了しました。**
プリンタマネージャにより印刷ジョブが終了(削除)されました。

**ツールボックスのカタログ : *カタログ名*が無効です。**
TOOLLOAD コマンドを使用して、存在しないツールボックスを SAS カタログからロードしようとしています。コマンドのツールボックス名とカタログ名のスペルを確認してください。**SAS のエクスプローラ**ウィンドウで、SAS カタログの一覧を参照できます。

**ツールが定義されていません。**
ロードしようとしたツールボックスには、ツールが定義されていません。ツールボックスには、1 つ以上のツールを定義する必要があります。

**指定したプリンタドライバにアクセスできません。**
SAS からプリンタドライバにアクセスできません。SYSPRINT システムオプションまたは**印刷設定** ダイアログボックスに、正しいプリンタドライバが指定されているかどうかを確認してください。

**プリンタ名が見つかりません。**
[**プリンタ設定** ]ダイアログボックスで指定したプリンタが見つかりません。

## OLE の使用

このセクションでは、SAS アプリケーションで Windows の OLE (Object Linking and Embedding)機能を使用する際に表示されるエラーについて説明します。

**OLE Error:*nnnnnnnn<error message text>***
このセクションでは説明していない OLE エラーが発生しました。このエラーメッセージが表示され、エラーメッセージからは原因を判断できない場合、エラーの原因を判断できる SAS のインストールの責任者に問い合わせてください。責任者から、SAS Institute テクニカルサポートに問い合わせが行われることもあります。

**OLE : 形式を選択して貼り付けは使用できません。サポートしていないフォーマットがクリップボードにあります。**
Windows クリップボードに、SAS/AF ソフトウェアが FRAME エントリで使用できるフォーマットがありません。

**. . . リンクダイアログボックスを起動します。よろしいですか。**
リンクのデータソースが見つかりませんでした。別のデータソースにリンクをリダイレクトする場合は、[**リンク** ]ダイアログボックスを呼び出すことができます。

**OLE:静的オブジェクトでの操作はできません。**
静的オブジェクトでは、この操作をサポートしていません。静的オブジェクトとは、基本的にはオブジェクトのイメージで、他のデータを含むことも、データにリンクされることもありません。

**静的オブジェクトを変換できません。**
静的オブジェクトを他の種類のオブジェクトに変換することはできません。

**OLE:Verb はオブジェクトには無効です。**
オブジェクトに渡された OLE Verb が、無効であるか、または送信できませんでした。SCL コードをチェックし、OLE Verb にスペルミスがないかどうかを確認してください。

**OLE:サーバーアプリケーションを起動できませんでした。**
オブジェクトのサーバーアプリケーションを起動できませんでした。実行ファイルが欠損しているか、おそらく、ネットワーク接続が切断されています。

**OLE:メンバーまたは名付けられたパラメータの 1 つが不明です。**
自動化している OLE オブジェクトに無効なメンバー(プロパティまたはメソッド)が指定されています。サポートされるメンバについては、OLE サーバーアプリケーションのマニュアルを参照してください。

**OLE:多次元配列へのアクセスはサポートされていません。**
SAS では、多次元配列はサポートされていません。サーバーアプリケーションから多次元データにアクセスするには、まずサーバーアプリケーションを使用し、データを一次元形式に変換する必要があります。

**OLE:オブジェクトは自動化できません。**
このオブジェクトは自動化をサポートしていません。

**OLE:パラメータの 1 つが誤ったタイプです。OLE:パラメータの 1 つが無効なタイプです。**
**OLE:パラメータの値が範囲を越えています。**
OLE オートメーションサーバーのメソッドに渡されたパラメータが誤ったタイプであるため、解釈できません。

**OLE:アプリケーションは使用中です。**
OLE サーバーアプリケーションはタスクで使用中のため、この要求を受けることができません。

**OLE:コントロールのライセンスがありません。**
この種類のコントロールを使用するライセンスがありません。詳細については、制御のドキュメントを参照してください。

**OLE:次のネットワークデバイスに接続できません。*UNC-drive-name***
リンクされたオブジェクトは、現在接続されていないネットワークドライブに存在します。ドライブの UNC 名がメッセージの後に表示されます。

**変換するオブジェクトが選択されていません。**
[**変換** ]ダイアログボックスは起動しましたが、変換するオブジェクトが選択されていません。

**このウィンドウには、リンクされたオブジェクトがありません。**
[**リンク** ]ダイアログボックスは起動しましたが、リンクされた OLE オブジェクトがフレームにありません。

## ネットワークの使用

このセクションでは、SAS をネットワーク上で使用しているときに発生するエラーについて説明します。

次のエラーは、いずれも適切なアクセス権限がない場合にネットワーク上で発生する可能性があります。

- **エラー：旧メンバー *ファイル名* の削除に失敗しました。**
- **エラー：*ファイル名* のファイル削除に失敗しました。**
- **エラー：*ファイル名* のテンポラリメンバー名の変更に失敗しました。**
- **エラー：ファイル *ファイル名* に対する権限がありません。**

ネットワークソフトウェアでは、ネットワーク監視システムを使用して、ネットワークファイルへのアクセスを制御します。ファイルの読み取りは可能で、更新は不可能というアクセス権を設定できます。前述の最後のエラーメッセージは、ファイルにアクセスした場合と同様に、ディレクトリにアクセスした場合にも表示されます。

## 内部エラーの解決

内部エラーは、SAS の起動を妨げる致命的なエラーです。これらのエラーメッセージが表示されると、原因を簡単な調査で解決できる場合もあります。最も一般的なエラーメッセージを次に示します。

**ホスト内部エラー：10**
SAS の初期化時に Ctrl キーと Break キーが同時に押下されました。そのため、SAS が終了しました。

**ホスト内部エラー：11**
メモリーの空き容量が必要です。問題を修正するには、スワップ領域をより空き領域があるディスクへと交換してください。また、現在のスワップディスクからファイルを削除して、少なくとも 20MB のメモリーを解放することもできます。

**ホスト内部エラー：12**
指定した SAS 構成にエラーがあると判断されました。エラーのあるシステムオプションについての説明文が表示されます。

**ホスト内部エラー：13**
SAS の一部をロードできません。ロードできない部分が説明文で示されます。ロードできない部分を適切なディレクトリに復元してください。

**ホスト内部エラー：24**
ウィンドウ環境を初期化できませんでした。適切な処置を説明するメッセージが表示されます。

**エラー 208：プロファイルカタログを開けません。**
Sasuser.Profile カタログにアクセスしている必要があります。プロファイルカタログが存在しない場合は作成してください。ディスクに十分な空き容量があるか、または、ネットワーク上で SAS を実行している場合は正しいファイルにアクセスしているかを確認してください。Sasuser.Profile カタログは SASUSER システムオプションで指定されたディレクトリで開きます。詳細は “SASUSER システムオプション:Windows” (559 ページ) を参照してください。関連項目 “Profile カタログ ” (22 ページ) 。

**WORK ライブラリが定義されていません。SASUSER ライブラリが定義されていません。SASHELP ライブラリが定義されていません。**
これらのライブラリに適切なディレクトリパスが指定されているかどうかを、SAS コマンドまたは SAS 構成ファイルで確認してください。

**エラー 302：Sasmsg ライブラリが定義されていません。**
このライブラリに適切なディレクトリパスが指定されているかどうかを、SAS コマンドまたは SAS 構成ファイルで確認してください。

**Work ライブラリを初期化できません。SASUSER ライブラリを初期化できません。**
WORK ライブラリと SASUSER ライブラリに適切なディレクトリが指定されているかどうかを、SAS コマンドまたは SAS 構成ファイルで確認してください。ネットワーク上で SAS を実行している場合は、必要なファイルにアクセスできる必要があります。

**エラー 401:メッセージサブシステムを初期化できません。**
SAS ライブラリすべてに適切なディレクトリパスが指定されているかどうかを、SAS コマンドまたは SAS 構成ファイルで確認してください。SAS 構成ファイルが適切な場所にあるかどうかも確認してください。ディレクトリパスに誤りがない場合、SAS コンサルタントに問い合わせてください。SAS メッセージファイルが破損しているか、不用意に削除された可能性があります。

**エラー 601:SETINIT 情報が無効です。**
SETINIT 情報を確認してください。詳細については、Windows 版 SAS 9.4 のインストール手順を参照してください。

**エラー 602:オプションを初期化するために CORE カタログにアクセスできませんでした。**
CORE カタログにアクセスできません。使用している PC の日付の設定が適切でない可能性があります。Windows の DATE コマンドを使用し、日付を検証して設定してください。

## オペレーティングシステムと Windows のエラーメッセージの解決

Windows から予期しない戻り値が返ってくる場合、Windows エラー番号が SAS ログに書き込まれます。Windows プログラミングマニュアルや Windows ユーザーマニュアルでエラー番号を参照し、エラーの原因を究明することができます。または、SAS のインストール担当者にエラー番号を報告してください。SAS のインストール担当者が、Windows エラー番号について IT スタッフに確認したり、Microsoft 技術サポートに連絡することが必要になる場合があります。

## 初期化と終了のエラーメッセージ

SAS の初期化時または終了時に SAS からエラーメッセージが発行される場合、SAS ログにエラーの内容を説明するエラーメッセージが書き込まれる場合があります。SAS ログが初期化される前か、閉じられた後で発行されたエラーメッセージは、利用可能であれば、**MSG** ウィンドウへと書き込まれます。また、エラーメッセージは Windows ファイルである SAS コンソールログへと書き込まれることがあります。Windows では、SAS コンソールログは通常、**c:\Users***user ID***\AppData**.SAS コンソールログの格納場所とファイル名は、アプリケーションイベントログから取得できます。アプリケーションイベントログを開くには、**ファイル名を指定して実行**ダイアログボックスから、**eventvwr** をサブミットし、**アプリケーション**をクリックします。

*付録 3*

# Windows 版でのグラフィックに関する注意点

TrueType フォントは WIN、WINPRTx、WMF、EMF、および GIF デバイスドライバ で使用できます。デフォルト以外の TrueType フォントを使用して印刷するには、初めにフォント名を確認する必要があります。TrueType フォントを確認するには、**スタート** ⇨ **設定** ⇨ **コントロールパネル** を選択して、フォントアイコンをダブルクリックします。Truetype フォントは、'T' が 2 つ重なったアイコンか、またはフォントの横に TrueType が表示されます。

SAS/GRAPH プログラム上で、フォント名を使用して TrueType フォントに FONT=もしくは F= オプションを指定できます。たとえば、次のように指定できます。

```
title2 font="arial"
  'This is TrueType font arial';
```

*注:* Truetype システム フォント名は引用符で囲む必要があります。

ハードコピーデバイスで別のフォントを使用する方法や、代替フォントをデフォルトのフォントに設定する方法など、FONT=オプションの詳細については、*SAS/GRAPH: Reference* を参照してください。

*付録 4*

# インタラクティブ SAS セッションにおける Windows 版でのデフォルトキー設定

**Windows 版でのデフォルトキー定義** . . . . . 627
**メイン SAS ウィンドウでのショートカットキー** . . . . . 629
**拡張エディタでのショートカットキー** . . . . . 631
**印刷プレビュー内でのショートカットキー** . . . . . 635

## Windows 版でのデフォルトキー定義

次の表に拡張エディタを除く、プライマリ SAS アプリケーション ウィンドウ(例:プログラムエディタ、ログ、出力)のデフォルトキー定義を示します。“拡張エディタでのショートカットキー” (631 ページ) 拡張エディタのデフォルトキー定義を示します。この表に示されていないキーの組み合わせは、Windows によって予約済みであるか、SAS 内で変更できないように定義されているものです。“メイン SAS ウィンドウでのショートカットキー” (629 ページ) を参照してください。

この表に掲載されているキー定義を参照または変更するには、**ツール** ⇨ **オプション** ⇨ **キー**を選択するか、KEYS コマンドを発行します。

***表 A4.1*** *Windows 版 SAS のデフォルトのキー定義*

| キー | デフォルト設定 | キー | デフォルト設定 |
|---|---|---|---|
| F1 | ヘルプ | Alt + F1 | |
| F2 | 再表示 | Alt + F2 | |
| F3 | 終了 | Alt + F3 | |
| F4 | 記録 | Alt + F11 | |
| F5 | wpgm | Alt + F12 | |
| F6 | ログ | Ctrl + B | LIBNAME |
| F7 | 出力 | Ctrl + D | ディレクトリ |

| キー | デフォルト設定 | キー | デフォルト設定 |
|---|---|---|---|
| F8 | 縮小;サブミット | Ctrl + E | 消去 |
| F9 | キー | Ctrl + G | |
| F11 | コマンドフォーカス | Ctrl + H | ヘルプ |
| F12 | | Ctrl + I | オプション |
| Shift + F1 | subtop | Ctrl + J | |
| Shift + F2 | | Ctrl + K | 切り取り (プログラムエディタのみ) |
| Shift + F6 | | Ctrl + L | ログ |
| Shift + F7 | 左揃え | Ctrl + M | 選択 |
| Shift + F8 | 右揃え | Ctrl + Q | filename |
| Shift + F9 | | Ctrl + R | rfind |
| Shift + F10 | wpopup | Ctrl + T | タイトル |
| Shift + F11 | | Ctrl + U | 選択解除 |
| Ctrl + F1 | | Ctrl + Y | |
| Ctrl + F2 | | RMB | wpopup |
| Ctrl + F3 | | Shift + 右マウスボタン | |
| Ctrl + F11 | | Ctrl + 右マウスボタン | |
| Ctrl + F12 | | MMB | |
| | | Shift + 中央マウスボタン | |
| | | Ctrl + 中央マウスボタン | |

*注:*

1. RMB は右マウスボタンです。
2. MMB は中央マウスボタンです。(中央ボタンのないマウスデバイスもあります。)

# メイン SAS ウィンドウでのショートカットキー

ここに示されているキーは **KEYS** ウィンドウにはありません。これらのショートカットキーを使用すると、編集およびその他の作業が簡単になります。

***表 A4.2*** *メイン SAS ウィンドウのキー設定*

| キーの組み合わせ | 操作 |
|---|---|
| ダイアログ ボックスと入力フィールド | |
| Tab キー | 次のフィールドへ移動 |
| Shift + Tab キー | 前のフィールドへ移動 |
| テキスト周辺に移動 | |
| Ctrl + -> (右矢印キー) | 次の単語へ移動 |
| Ctrl + <- (左矢印キー) | 前の単語へ移動 |
| ホーム | 行頭へ移動 |
| 終了 | 行末へ移動 |
| Ctrl + Home | 上端へ移動 |
| Ctrl + End | 終端へ移動 |
| Page Up | page up |
| Page Down | page down |
| Ctrl + Page Up | 上端へ移動 |
| Ctrl + Page Down | 終端へ移動 |
| Ctrl + Tab | 開いている次の SAS ウィンドウへと移動 (NEXTWIND コマンド) |
| Ctrl + Shift + Tab キー | 開いている前の SAS ウィンドウへ移動 (PREVWIND コマンド) |
| テキストを選択 | |
| Shift + -> (右矢印)キー | 右に移動しながら選択 |
| Shift + <- (左矢印キー) | 左に移動しながら選択 |

| キーの組み合わせ | 操作 |
|---|---|
| Shift + Home | 行の先頭を選択 |
| Shift + End | 行の最後を選択 |
| Shift + Ctrl + Home | 上端を選択 |
| Shift + Ctrl + End | 終端を選択 |
| Shift + Page Up | 上へ進む |
| Shift + Page Down | 下へ進む |
| Shift + Ctrl + Page 上へ | 上端を選択 |
| Shift + Ctrl + Page Down | 終端を選択 |
| Shift + マウスボタン 1 | クリックした位置まで、現在選択されているテキストの選択範囲を拡張 |
| 切り取り、コピー、および貼り付け | |
| Delete | 次の文字(または選択されたテキスト)を消去 |
| Ctrl + Delete | カーソルキーから現在の単語の終わりまでを消去 |
| Ctrl + Backspace | カーソルキーから現在の単語の始まりまでを消去 |
| Ctrl + マウスボタン 1 | 行(クリックされた行)全体を選択 |
| Ctrl + Z | 直前の操作を取り消し |
| Ctrl + X | 選択されたテキストを切り取り |
| Ctrl + C | 選択されたテキストを貼り付けバッファにコピー |
| Ctrl + V | テキストを貼り付け |
| ウィンドウ 制御 | |
| Alt キー | メインメニューバーから、またはメインメニューバーへフォーカスを切り替え |
| Shift + F5 | ウィンドウをカスケード表示 |
| Shift + F4 | ウィンドウを垂直に並べます |
| Shift + F3 | ウィンドウを水平に並べます |
| Ctrl + F6 | 次のウィンドウへ |

| キーの組み合わせ | 操作 |
| --- | --- |
| Alt + F4 | SAS を終了します |
| Ctrl + F4 | アクティブなウィンドウを閉じます |
| Ctrl + W | access 新しい SAS エクスプローラウィンドウに アクセスします。 |
| Shift + F10 | ポップアップメニューを開きます |
| ドッキングビューのサイズ変更 | |
| Alt + W + S | ドッキングビューのサイズ調整を開始します |
| -> (右矢印キー) | 分割バーを少し右に移動 |
| <- (左矢印キー) | 分割バーを少し右に移動 |
| Ctrl + -> (右矢印キー) | 分割バーを右に移動 |
| Ctrl + <- (左矢印キー) | 分割バーを左に移動 |
| ホーム | 分割バーを左端に移動 |
| 終了 | 分割バーを右端に移動 |
| Return | ドッキングビューの現在のサイズを受け入れ、ドッキングビューのサイズ調整を終了します |
| Esc | サイズ調整をせずにドッキングビューのサイズ調整を終了します |
| その他 | |
| Alt + Enter | 選択したオブジェクトの**プロパティ**ダイアログボックスを開きます。<br>このコマンドは、ツリービューまたはリストビューでのみ有効 |
| Esc + 文字 (または数字) | NOT**EPAD** ウィンドウ内の色またはハイライト属性 |

## 拡張エディタでのショートカットキー

ここで示されるショートカットキーはデフォルトのショートカットキーです。

***表 A4.3*** *拡張エディタのデフォルトショートカットキー*

| カテゴリ | コマンド | ショートカットキー |
|---|---|---|
| 省略形 | 新しい省略形を追加 | Ctrl + Shift + A |
| | ワンポイントアドバイスを表示 | Alt+F1+何も選択しない |
| | 現在のワンポイントアドバイスを非表示 | Esc |
| コードの折りたたみと展開 | 展開縮小が可能なすべてのブロックを縮小 | Alt+Ctrl+テンキーの- |
| | 現在の行を縮小 | Alt+テンキーの- |
| | 展開縮小が可能なすべてのブロックを展開 | Alt+ Ctrl+テンキーの+ |
| | 現在の行を展開 | Alt+テンキーの+ |
| | 現在の行の展開と縮小を切り替えます | Alt+テンキーの* |
| コマンド/マクロサポート | マクロを追加または変更 | Ctrl + Shift + M |
| | 最後に記録されたマクロを実行 | Ctrl + F1 |
| | コマンドとマクロを実行 | Alt + F8 |
| | マクロの開始と完了 | Alt + Shift + R |
| 編集 | コピー選択 | Ctrl + C<br>Ctrl + Insert |
| | 切り取り選択 | Ctrl + X<br>Shift+Delete |
| | 現在の文字を削除<br>*注:* SHIFT+END を使用して行の最後まで選択を拡張し、それから delete キーまたは backspace キーを押下することで行全体を削除できます。 | Delete |
| | 前の文字を削除 | Backspace または Shift +Backspace |
| | 次の単語の直前まで削除 | Ctrl + Delete |
| | 前の単語の直前まで削除 | Ctrl + Backspace |

| カテゴリ | コマンド | ショートカットキー |
|---|---|---|
| | キャリッジリターンを挿入 | Enter |
| | クリップボードから貼り付け | Ctrl + V<br>Shift+Insert |
| | やり直し | Ctrl + Y<br>Alt + Shift + Backspace |
| | 取り消し | Ctrl + Z<br>Alt + Backspace |
| ヘルプ | SAS の手順のヘルプを参照 | プロシジャ名にカーソルを置き、F1 キーを押します |
| | コンテクストヘルプ | F1 |
| 行を選択 | 次の選択された行に移動 | F2 |
| | 前の選択された行に移動 | Shift + F2 |
| | 現在の行の選択箇所を切り替えます | Ctrl + F2 |
| 移動 | 任意の行に移動(双方向) | Ctrl + G |
| | カーソルをダイルの上部に移動 | Ctrl + Page Up<br>Ctrl + Home |
| | カーソルをダイルの下部に移動 | Ctrl + Page Down<br>Ctrl + End |
| | カーソルを下へ移動 | Down |
| | カーソルをページの下へ移動 | Page Down |
| | カーソルを左へ移動 | 左 |
| | カーソルを右へ移動 | 右 |
| | カーソルを行の最初に移動 | ホーム |
| | カーソルを行の最後に移動 | 終了 |
| | カーソルを対応するかっこに移動 | Ctrl + [<br>Ctrl + ] |

| カテゴリ | コマンド | ショートカットキー |
|---|---|---|
| | カーソルをペアの DO/END キーワードに移動 | Alt + [<br>Alt + ] |
| | カーソルを次の大文字と小文字が変わる位置に移動 | Alt + 右矢印キー |
| | カーソルを次の単語の先頭に移動 | Ctrl + 右矢印キー |
| | カーソルを前の大文字と小文字が変わる位置に移動 | Alt + 左矢印キー |
| | 前の単語の先頭までカーソルを移動 | Ctrl + 左矢印キー |
| | カーソルを上方向に移動 | 上矢印キー |
| | カーソルを上方向に 1 ページ分移動 | Page Up |
| | 表示されている最初の行までカーソルを移動 | Alt + 上矢印キー |
| | 表示されている最後の行までカーソルを移動 | Alt + 下矢印キー |
| | 表示画面を下方向にスクロール | Ctrl + 上矢印キー |
| | 表示画面を上方向にスクロール | Ctrl + 下矢印キー |
| オプションの設定 | 挿入モードと上書きモードを切り替え | Insert |
| 選択 | 左方向の文字に選択範囲を拡大 | Shift + 左矢印キー |
| | 右方向の文字に選択範囲を拡大 | Shift + 右矢印キー |
| | 下方向に選択範囲を拡大 | Shift + 下矢印キー |
| | 選択範囲を下方向に 1 ページ分拡大 | Shift + Page Down |
| | 選択範囲を文書の最初まで拡大 | Ctrl + Shift + Home<br>Ctrl + Shift + Page Up |
| | 選択範囲を行の最初まで拡大 | Shift + Home |

| カテゴリ | コマンド | ショートカットキー |
|---|---|---|
| | 選択範囲を文書の最後まで拡大 | Ctrl + Shift + End<br>Ctrl + Shift + Page Down |
| | 選択範囲を行の最後まで拡大 | Shift + End |
| | 選択範囲を次に大文字と小文字が変わる位置まで拡大 | Alt + Shift + 右矢印キー |
| | 選択範囲を前の大文字と小文字が変わる位置まで拡大 | Alt + Shift + 左矢印キー |
| | 上方向に選択範囲を拡大 | Shift + 上矢印キー |
| | 選択範囲を上方向に 1 ページ分拡大 | Shift + Page Up |
| | 選択範囲を前の単語の先頭まで拡大 | Ctrl + Shift + 左矢印キー |
| | 次の単語の先頭直前まで選択範囲を拡大 | Ctrl + Shift + 右矢印キー |
| | すべて選択 | Ctrl + A |
| 選択操作 | 空白を除去 | Ctrl + Shift + W |
| | 行コメントで選択範囲にコメントします | Ctrl + / |
| | 選択されたテキストを小文字に変換しま | Ctrl + Shift + L |
| | 選択されたテキストを大文字に変換します | Ctrl + Shift + U |
| | Tab 範囲選択 | Tab+範囲選択 |
| | コメントの解除 | Ctrl + Shift + / |
| | 左 Tab 範囲選択 | Shift+Tab+範囲選択 |

## 印刷プレビュー内でのショートカットキー

**印刷プレビュー**ウィンドウ内の次の表でショートカットキーを使用することができます。

***表 A4.4*** *[印刷プレビュー]ウィンドウでのショートカットキー*

| 操作 | ショートカットキー |
|---|---|
| 次のページ | Alt + N |
| 前のページ | Alt + P |
| 拡大 | Alt + Z |
| ヘルプ | Alt + H |
| 印刷 | Alt + R |
| ウィンドウを閉じます | Alt + C or<br>Alt + F4 |

*付録 5*

# Cleanwork ユーティリティ

**Cleanwork ユーティリティ** . . . . . **637**

**Microsoft タスクスケジューラを使用した Cleanwork ユーティリティのスケジュール設定** . . . . . **638**

**ディクショナリ** . . . . . **638**

Cleanwork . . . . . 638

## Cleanwork ユーティリティ

Cleanwork ユーティリティは、一時 SAS ファイルと SAS ユーティリティディレクトリを検索してクリアします。Cleanwork ユーティリティ cleanwork.exe は、コンソールベースのアプリケーションであり、Base SAS の一部として、SASROOT ディレクトリにインストールされます。Cleanwork ユーティリティは、SAS Disk Cleanup Handler ユーティリティに代わるものです。

Cleanwork は、次の形式を使用して、SAS 作業ディレクトリの指定されたディレクトリまたはボリュームを検索します。プロセス ID が現在使用されておらず、nodename がディレクトリを作成したマシンのホスト名と一致した場合は、cleanwork はエントリを削除します。ディレクトリの内容は削除されます。

#TD<pid>, _TD<pid>, _TD<pid>_<nodename>_

ここで

- *#TD* は SAS V6.X プレフィックスです。
- *_TD* は SAS V7+プレフィックスです。
- *<pid>*はプロセス ID です。
- *nodename* はそのディレクトリを生成したマシンのホスト名です。

Cleanwork ユーティリティは、次の形式の SAS ユーティリティディレクトリも削除します。

SAS_util*NNNNPPPPPPPP_nodename*

ここで

- *NNNN* は、重複しない乱数です。
- *PPPPPPPP* は、プロセス ID の 16 進数形式での表現です。
- *nodename* はそのディレクトリを生成したマシンのホスト名です。

# Microsoft タスクスケジューラを使用した Cleanwork ユーティリティのスケジュール設定

引数セクションで指定したオプションで、指定した時刻に Cleanwork を自動的に実行するようにスケジュール設定できます。次の手順に従い、Microsoft タスクスケジューラから Cleanwork をスケジュール設定します。

1. 必要なオプションを使用して、Cleanwork の実行をスケジュール設定します。オプションの説明については、引数セクションを参照してください。
   a. [ **スタート** ]を選択し、*タスクスケジューラ*と入力します。
   b. [**操作**]ウィドウから、[**基本タスクの作成**]を選択します。
   c. 名前を入力し、[**次へ**]を選択します。
   d. 頻度を設定し、[**次へ**]を選択します。
   e. 開始日と時刻を選択し、[**次へ**]を選択します。
   f. [**プログラムの開始**]を選択し、**次へ**]を選択します。
   g. cleanwork.exe の場所を入力します。
   h. [ **引数の追加** ]フィールドに、オプション(/v volume など)を入力します。
   i. [**次へ**]を選択し、[**完了**]を選択します。
   j. 新しいタスクを右クリックし、[**プロパティ**]を選択します。
      a. [**セキュリティオプション**]ボックスの[**ユーザーまたはグループの変更**]を選択します。
      b. [**選択するオブジェクト名を入力してください**]ウィンドウに *SYSTEM* と入力し、[**名前の確認**]を選択します。SYSTEM のユーザーアカウントを設定していない場合は、[**taskeng.exe**]ウィンドウが表示されます。SYSTEM が設定されている場合は、[taskeng.exe]ウィンドウは表示されません。
2. 同じブログラムの別の構成を作成するには、ステップ 1 を繰り返します。たとえば、ボリュームのスキャンを週に一度行い、ログを毎日クリアする場合、あるいはその逆の場合などが考えられます。この手順は、オプションです。
3. 後日に出力を確認する必要がある場合は、cleanwork.exe パスと output–redirection オプションを使用してバッチスクリプトを作成します。この操作は、Windows タスクスケジューラで必要になります。たとえば、コマンド `cleanwork.exe /v c:/verbose >c:\temp\cleanwork.log` で cleanwork.bat スクリプトを作成します。この手順はオプションです。

# ディクショナリ

## Cleanwork

一時 SAS ファイルを検索してクリアします。

**注意: ユーザーが保存する必要がある AppData\Roaming\SAS\LOGS ディレクトリ内の <module name>.PID.DATE.log ファイルは、別のディレクトリに置く必要があります。**

**注意: Cleanwork は_TD####ディレクトリを検索して削除します。ディレクトリに対して同じテンプレートを使用するサードパーティのアプリケーションがないことを確認してから cleanwork ユーティリティを実行する必要があります。**

## 構文

**cleanwork** [/userlog] | [/alllogs] [/v volume(s)] [/d dir(s)] [/list] [/verbose]

### *引数*

**/userlog**
現在のユーザーの AppData\Roaming\SAS\LOGS ディレクトリの SAS ログファイルを消去します。

**/alllogs**
すべてのユーザーの AppData\Roaming\SAS\LOGS ディレクトリを消去します。

***注意:***
**このオプションでは、管理者権限が必要です。**

**/v volume**
SAS ディレクトリを削除するために検索するボリュームを指定します。

*注:* 複数のボリュームを指定する場合は、スペースで区切ります。

**/d dir(s)**
指定されたディレクトリのすべての SAS ディレクトリをクリアします。dir(s)は、1 つ以上のディレクトリへの Windows または UNC のパスです。

*注:* パスは、次のようにスペースで区切る必要があります。/d path1 path2...

**/list**
削除するファイルを一覧表示します。このオプションでは、ファイルまたはフォルダは削除されません。

**/verbose**
プログラムのアクティビティの追跡を表示します。

## 例

### *例 1*

```
cleanwork /d c:\Temp
```

```
Cleans all SAS Work and Utility directories located in the c:\Temp directory
hierarchy.
```

### *例 2*

```
cleanwork /d c:\Temp /userlog
```

```
Cleans all SAS Work and Utility directories located in the c:\Temp directory
hierarchy and all log files in the \AppData\Roaming\SAS\LOGS directory.
```

### 例 3

```
cleanwork /v c: u:
```

```
Cleans all SAS Work and Utility directories located on the c: and u: volumes.
```

### 例 4

```
cleanwork /alllogs
```

```
Scans all users for a SAS\LOGS directory and cleans all log files found. This
option will require administrator privileges to remove other users' files.
```

### 例 5

```
cleanwork /d c:\Temp /list
```

```
List all SAS Work and Utility directories located in the c:\Temp directory
hierarchy to be deleted.
```

### 例 6

```
cleanwork /v c:>cleanwork.log
```

```
Cleans all SAS Work and Utility directories located on the c: volume and dumps
the output to the file cleanwork.log instead of printing it out to the screen.
```

*注:* Windows タスクスケジューラでは、このオプションを使用しないでください。その代わりに、このコマンドを使用してスクリプトを記述し、実行するタスクスケジューラを設定します。

*付録 6*

# Sasiotest ユーティリティ

**Sasiotest ユーティリティ** . . . . . **641**
**ベストプラクティスと考慮事項** . . . . . **642**
**許容可能な SAS I/O パフォーマンス** . . . . . **643**
**ディクショナリ** . . . . . **643**
SASIOTEST コマンド . . . . . 643

## Sasiotest ユーティリティ

パフォーマンスの問題をトラブルシューティングする場合は、SAS が使用するファイルシステムのスループットレートを知る必要があります。Sasiotest ユーティリティは、次の操作を実行します。

- 指定された負荷でのシステムの I/O 動作を測定する
- SAS I/O と同様にファイルを開いてデータの読み込みおよび書き込みを行う
- ファイルを作成してテスト中のファイルシステムに書き込み、そのファイルを読み込むことで、読み書きのパフォーマンスをテストする
- 経過した実際の時間と I/O レート(MB/s)を取り込む出力ファイルを書き込む
- ランダムなページを読み込む
- 複数の I/O テストを同時に実行してファイルシステムを過負荷にし、パフォーマンスを特定する

次に、Sasiotest ユーティリティの Windows での動作について説明します。

- 32 ビットまたは 64 ビット Windows システムで実行する
- SAS に付属し、SASHOME ディレクトリの **`\SASHome\SASFoundation\9.4\sasiotest.exe` にインストールされる**
- スタンドアロンプログラムであり、コマンドプロンプトウィンドウから起動できる
- Windows API Writefile()、Readfile()、および Closefile()を使用する
- ファイルキャッシュ外部の Scatter-Gather I/O テスト用に Readfilescatter()および Writefilescatter() API をサポートする
- システムに SAS がインストールされているかどうかに関係なく、コピーしてスタンドアロンの実行可能ファイルとして使用できる
- 以前の SAS リリースがあるマシンにコピーして使用できる

SAS の I/O の実行方法と SAS ファイルシステムの I/O に関する最小推奨事項については、SAS ドキュメント(http://support.sas.com/kb/42/197.html)を参照してください。このドキュメントは、SAS 導入をサポートするファイルシステムの推奨 I/O メトリックについて説明し、I/O 特性を微調整して SAS パフォーマンスを最適化するために役立つガイドラインを示しています。既存のパフォーマンスの問題を解決するためにファイルシステムのスループットをテストしている場合は、特に次のドキュメントを参照してください。SAS ログをホストの監視と組み合わせることで、パフォーマンスの問題を解決する方法について説明しています。

- *SAS のパフォーマンスの問題の解決ホストベースのツールの採用*
- *SAS でパフォーマンスの問題を解決するための具体的なアプローチ*
- *SAS パフォーマンスの監視 - 詳細な説明*

## ベストプラクティスと考慮事項

次の概念について考慮してから、I/O スループットテストを開始してください。

- SAS は、Windows ファイルキャッシュを使用して読み書き操作を実行します。読み書きテストのファイルサイズは、ホストシステムのファイルキャッシュよりも大きくする必要があります。ファイルサイズは、ホストシステムの RAM よりも大きくなる可能性があります。たとえば、1GB のファイルサイズでテストを実行し、システムのキャッシュが 5GB である場合は、テストプロセスで読み込みテストのディスク I/O は限定されたものになります。書き込まれたファイルは、システムファイルキャッシュに残り、後続の読み込み操作では、ディスクではなくキャッシュから読み込まれます。読み込みテストがディスクではなくキャッシュでパスすると、読み込みパフォーマンスのメトリックが正しくなくなり、一貫した信頼性できる結果が得られません。
- このいずれかのツールの 1 インスタンス(1 回の読み書きテスト)の呼び出しによって、その時点で 1 つの SAS ジョブがファイルシステムでどのように実行されるかが示されます。システムが他のユーザーの作業負荷によって既に極度にビジーである場合は、この操作が適切なこともあります。
- このテストを複数インスタンス呼び出すと、同時 SAS 作業負荷がある状態を示されます。これは、他の同時ジョブがないか、または非常に負荷が少ないシステムで推奨されます。
- これらのテストセットを 1 日のさまざまな時間、およびビジーな数日間にわたり実施し、さまざまな作業負荷を実行するシステムで I/O 特性の包括的なプロファイルを取得します。
- 同じファイルシステムまたはシステムで同じ時刻にテストを実行します。
- テストするファイルシステムで作成するファイルを指定します。バッチモードを使用している場合は、リダイレクト演算子(>)を使用して、出力を別のファイルシステムに送信します。例: sasiotest.exe を含む.bat ファイル […引数…] > test.txt。
- このツールからの出力は、1 秒で書き込まれる MB と 1 秒で読み込まれる MB です。これらのメトリックを使用すると、SAS をサポートするファイルシステムの I/O を分析して微調整できます。

# 許容可能な SAS I/O パフォーマンス

Sasiotest ユーティリティから実行されるテスト出力は、スループットの結果を 1 秒あたりの MB で示します。SAS ファイルシステムでは、各コアについて 1 秒あたり 100MB 以上が推奨されています。

# ディクショナリ

## SASIOTEST コマンド

Windows でのファイル I/O スループットを測定するために、SAS ファイルに似たファイルを読み書きします。

**要件:** -sgio オプションを使用する場合は、-pagesize は x86 システムでは 4K の倍数、x64 および IPF システムでは 8K の倍数にする必要があります。

-sgio オプションを指定する場合は、-numpages が必要です。

*filename* は最初の引数です。

### 構文

**SASIOTEST** *filename <options>*

#### *必須引数*

***filename***

読み書きするファイル名を指定します。ファイルシステムパスを含めることができます。

#### *オプション引数*

***<options>***

**-w または-r**

-w は指定したファイルを作成して書き込みます。

-r は指定したファイルを読み込みします。

**デフォルト** -w

**-seq または-ran**

*-seq* 指定したファイルから順次に読み込みます。

*-ran* 指定したファイルからランダムに読み込みます。

**デフォルト** -seq

**-filesize *n***

読み書きするファイルのサイズを指定します。たとえば、*n* では 800K、512K、5MB、2GB を指定できます。

**デフォルト** -filesize を指定しない場合は、デフォルトは 1MB です。

**-pagesize <*n*>**
I/O 操作のページサイズを指定します。たとえば、*n* では 8K、64K、128K を指定できます。

**デフォルト** -pagesize を指定しない場合は、デフォルトは 8K です。

**-sgio**
I/O 操作で Scatter Gather (SGIO)読み込み API と SGIO 書き込み API を使用するかどうかを指定します。次のオプションは、SGIO に固有です。

**-numpages <*n*>**
各 SGIO 操作で読み書きを行うページ数を指定します。

**-unbuffered**
書き込みページでファイルキャッシュを介さないように指示します。

**-writethru**
中間キャッシュをラストスルーし、ディスクに直接書き込むように指示します。

## 例

### *例 1*

```
c:\m904_win>sasiotest test.out -w SAS I/O Test Utility: v1.0

WARNING: Filesize option not specified. Defaulting to 1 MB filesize.
WARNING: Pagesize option not specified. Defaulting to 8K pagesize.
Sequentially writing 1048576 bytes to file: test.out with a 8192 pagesize.
Write Throughput: 71.100308 MB/Sec.
```

### *例 2*

```
c:\m904_win>sasiotest test.out -w -pagesize 64k SAS I/O Test Utility: v1.0

WARNING: Filesize option not specified. Defaulting to 1 MB filesize.
Sequentially writing 1048576 bytes to file: test.out with a 65536 pagesize.
Write Throughput: 72.561777 MB/Sec.
```

### *例 3*

```
c:\m904_win>sasiotest test.out -w -pagesize 4k -filesize 2M SAS I/O Test Utility: v1.0

Sequentially writing 2097152 bytes to file: test.out with a 4096 pagesize.
Write Throughput: 45.795315 MB/Sec.
```

### *例 4*

```
c:\m904_win>sasiotest test.out -r SAS I/O Test Utility: v1.0

WARNING: Filesize option not specified. Defaulting to 1 MB filesize.
WARNING: Pagesize option not specified. Defaulting to 8K pagesize.
Sequentially reading 1048576 bytes from file: test.out with a 8192 pagesize.
Read Throughput: 258.301703 MB/Sec.
```

*注:* 直前に書き込んだファイルを読み込んでいる場合でも、PAGESIZE および FILESIZE 引数を指定しないと、sasiotest ユーティリティでは 8K および 1MB のデフォルト値が使用されます。例 3 で作成したファイルを正しく読み込む方法については、例 5 を参照してください。

### *例 5*

```
c:\m904_win>sasiotest test.out -r -pagesize 4k -filesize 2M -numpages 2 SAS I/O Test Uti
```

```
Sequentially reading 2097152 bytes from file: test.out with a 4096 pagesize.
Read Throughput: 237.732948 MB/Sec.
```

### *例 6*

```
c:\m904_win>sasiotest test.out -w -sgio -pagesize 4k -filesize 2G -numpages 256 SAS I/O
```

```
Gather Writing 256 pages - 2 iterations to file: test.out with a 4096 pagesize.
Write Throughput: 18.363657 MB/Sec.
```

### *例 7*

```
c:\m904_win>sasiotest test.out -r -sgio -pagesize 4k -filesize 2G -numpages 256 SAS I/O
```

```
Scatter Reading 256 pages - 2 iterations from file: test.out with a 4096
pagesize.
Read Throughput: 33.205119 MB/Sec.
```

*付録 7*

# ASCII システムでの EBCDIC データの使用

**About EBCDIC and ASCII Data** . . . . . **647**
Overview of EBCDIC and ASCII Data Representation . . . . . 647
EBCDIC File Structures . . . . . 648
ASCII File Structure . . . . . 648
Numeric Values . . . . . 649

**Moving Data from EBCDIC to ASCII Systems** . . . . . **649**
Overview of Accessing EBCDIC Data on ASCII Systems . . . . . 649
Example of Incorrect Conversion of Packed-Decimal Numeric Data . . . . . 650
Convert EBCDIC Files with Fixed-Length Records . . . . . 651
Convert EBCDIC Files with Variable-Length Records . . . . . 652
Read EBCDIC Data from Structured COBOL Files . . . . . 654

## About EBCDIC and ASCII Data

### *Overview of EBCDIC and ASCII Data Representation*

*Extended Binary Coded Decimal Interchange Code (EBCDIC)* is an 8-bit character encoding method for IBM mainframe machines. *American Standard Code for Information Interchange (ASCII)* is a 7-bit character encoding method for most other machines, including Windows, UNIX, and Macintosh machines.

Hexadecimal digits are used to represent one byte or eight bits of data. In a binary system, each bit can have the value 0 or 1. An aggregation of four bits can therefore take on 16 ($2^4$) possible values. This means that two hexadecimal digits can be used to represent one byte of data. In the EBCDIC and ASCII encoding methods, each character is represented by two hexadecimal digits. (This pertains primarily to Western language, single-byte encoding methods. There are other encoding methods that store a single character in two bytes of storage, such as encoding methods that are used for Japanese or Korean data.)

Each encoding method represents the same data differently, as shown in the following examples:

- On an EBCDIC system, the digit 4 is represented by the hexadecimal value 'F4'x. On an ASCII system, the digit 4 is represented by the hexadecimal value '34'x.
- On an EBCDIC system, the hexadecimal value '50'x represents the symbol &. On an ASCII system, the same hexadecimal value represents the letter P.

When SAS reads a file, it expects the data in the file to be in the encoding that matches the ENCODING= option for the SAS session. For example, on a Windows machine, the default encoding for a single-byte SAS session with a US English locale is LATIN1. SAS expects the data in a file on that Windows machine to use a LATIN1 encoding. However, if a file originates on an EBCDIC machine and it is stored on a Windows machine, then SAS would misinterpret the data from this file if no other encoding information is provided. For this reason, specific steps must be performed to convert data that originates on an EBCDIC system before it can be used on an ASCII system (for example, the Windows machine). Here are the two main methods to make EBCDIC data available on an ASCII system:

- On the ASCII system, read the data directly from the EBCDIC system.
- Use an FTP program to move the data, with or without any conversion of the data.

## EBCDIC File Structures

When you decide how to move data from an EBCDIC system to an ASCII system, consider the structure of the EBCDIC source file. On EBCDIC systems, you might have files with fixed-length records or files with variable-length records. Either type of file contains a header with information about the file. The header includes a Record Format attribute that indicates whether the records are fixed length or variable length. The header for a file with fixed-length records includes a Logical Record Length attribute that indicates the length of each record in bytes.

In SAS, the Record Format attribute corresponds to the RECFM= option in a FILENAME statement. To access a file with fixed-length records, specify RECFM=F. To access a file with variable-length records, specify RECFM=V. Similarly, the Logical Record Length attribute corresponds to the LRECL= option.

The Logical Record Length attribute in the header for a file with variable-length records indicates the maximum record length. Each record in a file with variable-length records begins with a *record descriptor word (RDW)*. The RDW is a 4-byte binary integer field. The first two bytes of the RDW indicate the length of the current record. The last two bytes of the RDW contain information that is used by the operating system. The length of the record includes the four bytes of the RDW at the beginning of the record. Because the length of each record is specified in an EBCDIC file (either in the header or in the RDW), there are no end-of-record indicators in EBCDIC files.

A file with variable-length records also contains *block descriptor words (BDWs)*. Like the RDW, the BDW is a 4-byte, binary integer field. The first two bytes indicate the block size, and the last two bytes are used by the operating system. Each block can contain multiple records. If the block size is not specified when the file is created, the default block size is the logical record length plus 4. Otherwise, the size of a block is the number of bytes that are contained in the block. This value is the sum of the record lengths in the block (obtained from the RDWs) plus 4 (the length of the BDW).

## ASCII File Structure

On ASCII systems, a file does not contain a header with information about the file, such as record format or lengths. The RECFM attribute for ASCII files is variable (RECFM=V), and the record length (LRECL) is unlimited. Instead of defining record lengths like EBCDIC files do, ASCII files use end-of-record indicators to flag the end of a record. On a Windows machine, the end-of-record indicators are the carriage return (CR) and line feed (LF) characters. On a UNIX machine, an LF indicates the end of a record. On a Macintosh machine, a CR indicates the end of a record. Other types of machines use different combinations of characters to identify the end of record. For all

ASCII machines, the hexadecimal value for CR is '0D'x, and the hexadecimal value for LF is '0A'x.

When SAS reads a file from disk on an ASCII machine, default values for some file attributes must be used because these attributes are not defined. The default RECFM value is V (variable-length record), and the default LRECL value is 32767. This means that SAS scans the input from an ASCII file, parses the data into variable values based on the INPUT statement, and looks for an end-of-record indicator. If the end of a record is not found within the specified number of characters (based on LRECL), then SAS truncates the record and prints a message in the log. For example, suppose LRECL is set to 256, and there is a record that is 300 characters. SAS reads the first 256 characters based on the INPUT statement, and then discards the last 44 characters. A message in the log states that "One or more lines have been truncated." You can override the current LRECL value using the LRECL= option in the INFILE statement.

### Numeric Values

When stored as character data, the decimal digits 0 through 9 each occupy one byte of storage. One 8-bit byte includes two 4-bit *nibbles*. Each nibble can have 16 ($2^4$) possible values. The first nibble is the *high-order nibble*, and the second is the *low-order nibble*. In EBCDIC and ASCII systems, the high-order nibble has a standard value. Decimal digits are represented in EBCDIC with a high-order nibble of F. Decimal digits are represented in ASCII with a high-order nibble of 3. This means that in an EBCDIC system, the digits 0 through 9 are represented by the hexadecimal values 'F0'x through 'F9'x. In an ASCII system, the digits 0 through 9 are represented by the hexadecimal values '30'x through '39'x. This encoding method treats decimal digits as characters.

As an alternative to storing decimal digits as characters, there are other encoding methods that can be used on an EBCDIC system. For example, a packed-decimal encoding method represents two decimal digits in one byte of storage. A zoned-decimal encoding method represents one decimal digit in one byte of storage, and the sign of the entire value is included within one byte of storage. (The byte that stores the decimal digit and the sign of the entire value can be either the first byte or the last byte, depending on the type of machine.)

You must know the numeric encoding that is used on the source EBCDIC system so that the source data is interpreted correctly on the ASCII system. For SAS, this means that you must specify the correct informats to use for numeric data.

## Moving Data from EBCDIC to ASCII Systems

### Overview of Accessing EBCDIC Data on ASCII Systems

There are several ways to access EBCDIC data on an ASCII system. For example, some ASCII machines have peripheral devices that can read 3480 or 3490 cartridge tapes that are created on an EBCDIC system. These devices can read the data directly from a tape into an application on an ASCII machine. Alternatively, these devices can copy data from a tape and store it on the ASCII machine's hard drive.

A more common method of moving and converting data is to use an FTP program to transfer the data. By default, most FTP programs convert EBCDIC data into ASCII when transferring data. If the source data contains only character data (including digits that are encoded as characters), this is the recommended method. During the conversion process, the FTP program creates the appropriate end-of-record indicators for the ASCII

system. After conversion, you can use an INFILE statement to access the newly created file on the ASCII system. Use an INPUT statement to specify the correct informat values to use when reading the data in the file.

*Note:* Even when all of the EBCDIC source data is encoded as character data, there might be some characters that are not interpreted correctly during conversion. The correct interpretation of these characters depends on the encoding method that is used on the EBCDIC machine. As a best practice, verify that your data was converted correctly by viewing the data that SAS reads from a converted file.

When an EBCDIC file contains numeric data that is not encoded as character data, such as when a packed-decimal or zoned-decimal encoding method is used, the default FTP conversion does not work correctly. Some numeric data can resemble standard character data. In this case, FTP conversion incorrectly assigns ASCII characters to EBCDIC numeric data. For more information, see “Example of Incorrect Conversion of Packed-Decimal Numeric Data”.

*Note:* There is no way to correctly convert packed-decimal encoded data from EBCDIC into ASCII. Other methods to convert the data must be used if a packed-decimal, zoned-decimal, or other numeric encoding method is used on the EBCDIC system. For more information, see “Convert EBCDIC Files with Variable-Length Records” on page 652.

In some instances, a byte of EBCDIC data might be interpreted in ASCII as an end-of-line flag or end-of-file flag. If SAS is reading a file with variable-length records when one of these hexadecimal values is encountered, then you might observe unintended results. Depending on the expected data values based on specified informats, you might observe anything from invalid data errors to unexpected termination of the DATA step.

## Example of Incorrect Conversion of Packed-Decimal Numeric Data

This example demonstrates the problems that can result when you convert packed-decimal numeric data as if it were encoded as character data. Suppose an EBCDIC data file contains the numeric value 505, stored as a packed-decimal value ('505C'x). If you looked at the file with an EBCDIC file browser or editor, you would see the characters &*. This is because '50'x corresponds to & and '5C'x corresponds to *. The FTP program interprets the & character and converts it to the ASCII value '26'x. The FTP program converts the * character to the ASCII value '2A'x, and the resulting converted value is '262A'x. The correct packed-decimal value in ASCII should be '000505'x. Because the input data does not conform to the expected packed-decimal informat, SAS outputs an error to the log that states that the data is invalid. Each time invalid data is encountered, SAS writes an error to the log, and outputs the contents of the input buffer and the corresponding DATA step variables.

***Table A7.1*** *Incorrect Conversion of Packed-Decimal Numeric Data*

| Step | Action | Value |
|---|---|---|
| 1 | FTP program reads the EBCDIC packed-decimal numeric value ‘505’. | '505C'x |
| 2 | FTP program interprets the value as standard EBCDIC characters. | &* |
| 3 | FTP program converts to standard ASCII hexadecimal characters. | '262A'x |

| Step | Action | Value |
|---|---|---|
| 4 | SAS flags the data as invalid because packed-decimal numeric data is expected (based on the specified informat value). | ??? |

## Convert EBCDIC Files with Fixed-Length Records

### FTP the File in Binary

When you convert an EBCDIC file with fixed-length records, use FTP to transfer the file in binary. Then, with a FILENAME or INFILE statement, specify RECFM=F, and assign the same value to LRECL that the file has in the EBCDIC system. Use the formatted input style with the following informats:

- $EBCDIC*w*. for character input data
- S370F*xxxw.d* for numeric input data

  *Note:* There are many S370F*xxxw.d* informats. Select those informats that match the type of data that you have. For more information, see *SAS Formats and Informats: Reference* for SAS 9.3 and higher, or see *SAS Language Reference: Dictionary* for earlier versions of SAS.

Because you are transferring the source file in binary, there is no processing to add end-of-record indicators. For this reason, you must specify the exact number of bytes that are specified for the source file in the EBCDIC system. If there are bytes in the source file that would be interpreted as end-of-record indicators or end-of-file indicators in an ASCII context, SAS treats those bytes simply as data.

### Example: Convert an EBCDIC File with Fixed-Length Records into an ASCII File

The following code reads a file, fixed.txt, that was previously transferred via FTP in binary from an EBCDIC system to an ASCII system. The source file has fixed-length records that are 60 bytes long. Based on the informat in this example, the last three bytes in each record contain numeric data that was stored using the packed-decimal encoding method.

```
filename test1 'c:\fixed.txt' recfm=f lrecl=60;
data one;
infile test1;
input @1  name  $ebcdic20.
      @21 addr  $ebcdic20.
      @41 city  $ebcdic15.
      @56 state $ebcdic2.
      @58 zip   $s370fpd3.;
run;
```

## Convert EBCDIC Files with Variable-Length Records

### Overview of Converting EBCDIC Files with Variable-Length Records

When you convert an EBCDIC file with variable-length records, you can use an FTP program. The FTP program removes BDWs and RDWs and adds end-of-record indicators that are expected by the ASCII system. The data in the file is converted from EBCDIC to ASCII. If all of the data in the EBCDIC file is encoded as characters, then this process typically works correctly.

*Note:* Even when all of the EBCDIC source data is encoded as character data, there might be some characters that are not interpreted correctly during conversion. The correct interpretation of these characters depends on the encoding method that is used on the EBCDIC machine. As a best practice, verify that your data was converted correctly by viewing the data that SAS reads from a converted file.

When an EBCDIC file contains numeric data that is not encoded as character data, such as when a packed-decimal or zoned-decimal encoding method is used, the default FTP conversion does not work correctly. For more information, see "Overview of Accessing EBCDIC Data on ASCII Systems" on page 649. To prevent misinterpretation of data during conversion, transfer the file in binary via FTP without converting the data to an ASCII encoding. When the data is transferred in binary and is not converted, be aware that the BDW and RDW information is removed automatically. This removes information that SAS needs to read the data successfully.

### Read Files Directly from the EBCDIC System

If you have direct access between the ASCII machine and the EBCDIC machine, then the best practice is to read the file directly. Direct access is enabled via a peripheral device on the ASCII machine that can read an EBCDIC tape. You can access the file via the FTP access method in a FILENAME statement. There are several advantages to this method of accessing EBCDIC data:

- file preprocessing is not required
- copying the source file is not required
- FTP access method works for fixed-length and variable-length records
- DATA step processing works as expected

The main disadvantage is that this method requires more time for processing because you are accessing the data remotely.

This method of accessing EBCDIC data applies if you have a 3480 or 3490 cartridge tape reader attached to your ASCII machine. In this case, you do not need to preprocess the file on an EBCDIC machine. You can read it directly from the tape by setting RECFM=S370VB and using the $EBCDIC*w*. and S370F*xxxw.d* informats.

In a FILENAME statement, specify the FTP access method and the source file name, and provide values for the HOST=, USER=, and PASS= options. The HOST= option specifies the name of the EBCDIC machine, USER= specifies the user account that you use to log on, and PASS= specifies the password that you use to log on. The FTP access method uses an FTP program on the ASCII machine to open a connection between the ASCII machine and the EBCDIC machine. The SAS system connects to and logs on to the mainframe machine with the specified user account and password. The FTP program transfers the file.

*Note:* If you specify the PASS= option, the password is saved as text in your SAS program. The password is not visible in the SAS log. As an alternative to the PASS=

option, you can specify the PROMPT option and provide a password at the prompt when you execute the SAS program.

For EBCDIC files with variable-length records, you must also specify the S370V and RCMD= options. The S370V option indicates that the records in the source file have variable lengths. For the RCMD= option, specify RCMD="SITE RDW" to indicate that the FTP process should keep the RDW information during the file transfer.

If you experience connection problems to the EBCDIC machine, you can add the DEBUG option to see the informational messages that are sent to and from the FTP server.

### *Example: Read an EBCDIC Source File Directly with the FTP Access Method*

This example shows how to read an EBCDIC file with variable-length records directly from an EBCDIC machine using the FTP access method. The user is prompted for her MVS logon password. The ZIP code is input as a 5-digit EBCDIC number, represented by one digit per byte. The comments section is varying in length up to 200 characters. After the data is read, it is printed to verify the contents of the data set.

```
filename test1 ftp "'SASEBCDIC.VB.TEST1'" host='MVS' user='SASEBCDIC' PROMPT
          s370v rcmd='site rdw';
data one;
infile test1;
input @1  name     $ebcdic20.
      @21 addr     $ebcdic20.
      @41 city     $ebcdic10.
      @51 state    $ebcdic2.
      @54 zip      s370ff5.
      @60 comments :$ebcdic200.;
run;

proc print;
run;
```

### *Reformat an EBCDIC File with Variable-Length Records with IEBGENER*

Suppose that you do not have direct access between the ASCII machine and the EBCDIC machine. That is, you do not have a peripheral device that reads EBCDIC data on the ASCII machine. In this situation, you can convert the data by reformatting the file on the mainframe machine. By changing the format of the file, you prevent the FTP program from removing the RDW information that SAS requires to read the data correctly. After you reformat the file, you can transfer the file in binary to the ASCII machine.

To reformat the source file, use the IEBGENER program on the EBCDIC machine. Use this program to make an exact copy of the file with altered header information. Specifically, use IEBGENER to change the RECFM value from V (variable-length records in blocks) to U (undefined record length and unblocked). After making this change, the FTP program no longer removes the RDW information during the file transfer.

When you run the IEBGENER program, in addition to the required arguments, specify the following overrides:

```
SYSUT1 DCB=(RECFM=U,BLKSIZE=32760)
SYSUT2 DCB=(RECFM=U,BLKSIZE=32760) DISP=(NEW,CATLG)
```

*Note:* Do not use the original values of RECFM and BLKSIZE for SYSUT1.

Transfer the new version of the file in binary using an FTP program on the ASCII machine. In SAS, use a FILENAME or INFILE statement to read the transferred file. Set the options appropriately.

- Set the RECFM= option to S370V if the record format for the original file was variable (RECFM=V). Set the RECFM= option to S370VB if the record format for the original file was variable and blocked (RECFM=VB). By specifying the RECFM= option as S370V or S370VB, you tell SAS to process the RDW information for each record and input the correct number of bytes for each record.
- Specify the same value for the LRECL= option that is in the original file. If you do not specify a value for the LRECL= option, SAS uses the default LRECL value (32767). Using the default value could cause SAS to truncate data records if they are longer than the default LRECL value.

Use the formatted input style with the informats that are described in "FTP the File in Binary" on page 651.

### Example: Read a File with Modified Header Data

This example reads a file that was generated from an EBCDIC file with a header that was modified to change the file format. The modified file was transferred to an ASCII machine for SAS processing. For more information, see "Reformat an EBCDIC File with Variable-Length Records with IEBGENER" on page 653.

The TRUNCOVER option is included in the INFILE statement because the Comment variable can be up to 60 characters (but it is likely shorter). Without the TRUNCOVER option, the INPUT statement could attempt to read past the end of the record. Data from the next record would continue to be assigned to the Comment variable until the variable was full. The LRECL= option is not specified because the default value is sufficient to handle the longest record in the file. After the data is read, it is printed to output for verification.

```
filename test1 'c:\vbtest.xfr' recfm=s370vb;
data one;
infile test1 truncover;
input @1  name    $ebcdic14.
      @15 addr    $ebcdic18.
      @33 zip     s370ff5.
      @38 comment $ebcdic60.;
run;

proc print;
run;
```

## Read EBCDIC Data from Structured COBOL Files

### About Structured COBOL Files

A structured COBOL file is generated using an OCCURS DEPENDING ON clause. This type of file has variable-length records. And, when the file is transferred via FTP in binary, there is no BDW or RDW information. Each record is divided into three parts: a record header (a fixed-length portion of the record), an index variable, and one or more data segments. The documentation for the file provides the length of the record header, the index variable, and a data segment. The record header is the same length for each record. It contains information that pertains to all of the data segments that follow. The

index variable provides the number of data segments for the current record. The remainder of the record contains the data segments.

Because of the structure of the records, SAS is able to read the data in these files. The length of a record is the sum of the header length, the index length, and the product of the index value and the size of each data segment. For each data segment, SAS reads the segment, and then outputs a copy of the header and the current data segment to a new observation in a SAS data set.

When you read a structured COBOL file, specify RECFM=N in your FILENAME statement. This tells SAS that you are reading a stream of data that does not conform to a typical file structure. Any restrictions to record length are ignored when SAS reads a data stream because SAS does not attempt to buffer the input. SAS writes a statement to the SAS log to notify you that SAS reads a data stream as unbuffered when RECFM=N.

SAS reads an entire structured COBOL file as a single, long record. Therefore, if you need to skip some data or move past a space, you must use relative column pointers in your INPUT statement. Line holders are ignored because the contents of the file are treated as a single input record. The *@column* pointers do not work for these files.

***CAUTION:***

**Do not use @*column* pointers when you specify RECFM=N.** Using *@column* pointers initiates an infinite loop in which SAS reads and outputs the same data repeatedly until you halt the program or until no more disk space is available.

### *Example: Read Data from a Structured COBOL File*

In this example, an EBCDIC file was transferred via FTP in binary without first processing the file using IEBGENER. The record header (fixed-length) portion of each record is 59 bytes in length and contains a combination of character and numeric data. The index variable is two bytes. There is another space (one byte) to separate the index variable from the remainder of the record. The data segment portion of the record consists of one or more repeats of 13 bytes in length. Each repeat contains a combination of character and numeric data.

```
filename test1 'c:\VB.TEST' recfm=n;

data one;
infile test1;
input name $ebcdic20. addr $ebcdic20. city $ebcdic10. st $ebcdic2. +1
      zip s370ff5. +1 idx s370ff2. +1;
do i = 1 to idx;
   input cars $ebcdic10. +1 years s370ff2. ;
   output;
   if i lt idx then input +1 ;
end;
run;
```

# 推奨資料

以下の参考資料もご覧ください。

- *Base SAS プロシジャガイド*
- *SAS ファイルの移動とアクセス*
- *SAS コンポーネントオブジェクトリファレンス*
- *SAS/CONNECT ユーザーガイド*
- *SAS データセットオプション:リファレンス*
- *SAS 出力形式と入力形式:リファレンス*
- *SAS 関数と CALL ルーチンリファレンス*
- *SAS 言語リファレンス:SAS 言語リファレンス:解説編*
- *SAS 各国語サポート(NLS):リファレンスガイド*
- *SAS Output Delivery System:ユーザーガイド*
- *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 ユーティリティリファレンス*

SAS 刊行物の総一覧については、sas.com/store/books 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 Books
SAS Campus Drive
Cary, NC 27513-2414
電話: 1-800-727-0025
ファクシミリ: 1-919-677-4444
メール: sasbook@sas.com
Web アドレス: sas.com/store/books

# 用語集

**American Standard Code for Information Interchange**
基本的な 128 文字の組み合わせを割り当てる 7 ビットの標準エンコードです。様々なコンピュータシステムをサポートします。ASCII は英語のアルファベットの大文字、小文字、句読点、0 から 9 までの数字、制御文字をエンコードします。この 128 文字セットは、他の多くのエンコーディングにも含まれます。短縮形:ASCII

**ASCII**
関連項目:American Standard Code for Information Interchange

**ASCII 照合順序**
特定の ASCII エンコーディングにより、テキストデータの並べ替えに使用されるルール。並べ替え順序は、ASCII エンコーディングのコードページの各コードポイントの場所により決定されます。Windows Latin1 コードページの並べ替えの優先順序は、句読点、数字、大文字、小文字の順です。大文字 A (コードポイント 41)は、小文字 g (コードポイント 67)より優先されるため、A は g の前に並べ替えられます。

**AUTOEXEC.SAS**
SAS の起動時に自動的に実行される SAS ステートメントを含むファイル。自動実行ファイルは、SAS システムオプションの指定、よく使用するフォルダやディレクトリへのライブラリ参照名やファイル参照名の割り当てに使用されます。

**AWS**
関連項目:アプリケーションワークスペース

**CONFIG.SYS**
デバイスドライバ、ファイル操作要素、メモリ管理オプションなどのオペレーティングシステムのプロパティを指定する DOS 構成コマンドを含むシステムファイル。

**CPU**
関連項目:中央演算処理装置

**CPU 時間**
コンピュータシステムの CPU が、要求した演算や他の操作の実行に要する時間。

**DCOM**
関連項目:分散 COM

**DDE**
関連項目:動的データ交換

**DLL**
関連項目:動的リンクライブラリ

**DOS**
パーソナルコンピュータのディスクオペレーティングシステム。SAS のドキュメントでは、DOS は、Microsoft が IBM 用に開発した MS-DOS (Microsoft Disk Operating System)を指します。

**EBCDIC**
関連項目:Extended Binary Coded Decimal Interchange Code

**Extended Binary Coded Decimal Interchange Code**
IBM メインフレームおよびミッドレンジコンピュータ上でのデータ表示のための、シングルバイトとマルチバイトのエンコーディングファミリ。EBCDIC は英語のアルファベットの大文字、小文字、句読点、0 から 9 までの数字、制御文字の拡張セットをエンコードします。短縮形:EBCDIC

**filename**
ファイルに使用される識別子。ファイル名は、例えば、PROFILE.SC2 のようにファイル拡張子を含みます。

**GUI**
関連項目:グラフィカルユーザーインターフェイス

**HTML**
関連項目:HyperText Markup Language

**HyperText Markup Language**
コードを使用し、テキストファイル内のテキストのレイアウトやスタイルを示すコーディングシステム。イメージ、サウンド、ビデオストリーム、アプレット(小さなソフトウェアアプリケーション)を HTML ドキュメントに埋め込む際に使用する HTML コードもあります。すべての Web ブラウザで、HTML ドキュメントの処理が可能です。短縮形:HTML

**libref**
SAS ライブラリの場所に関連づけられた SAS 名。例えば、MYLIB.MYFILE という名前では、MYLIB はライブラリ参照であり、MYFILE は SAS ライブラリ内のファイルです。

**NTFS**
関連項目:NT ファイルシステム

**NT ファイルシステム**
ディレクトリやファイルを編成する高度なシステム。NTFS では、長いファイル名、フルセキュリティアクセスコントロール、ファイルシステムのリカバリ、大容量ストレージメディアをサポートします。短縮形:NTFS

**Object Linking and Embedding**
クライアント/サーバーアーキテクチャを利用した、Windows でサポートされるプロセス間通信の方法。OLE を使用すると、あるアプリケーションで作成されたオブジェクトを別のアプリケーションに埋め込んだり、リンクしたりできます。短縮形:OLE

**ODBC**
参照項目:Open Database Connectivity

**ODBC ドライバ**
ロード可能なライブラリモジュール。データが特定のベンダ製のデータ管理ソフトウェアで作成、メンテナンスされる場合に、このデータのアクセス、操作、更新を可能にする標準化されたインターフェイスを提供します。たとえば、ODBC 用 SAS(c)ドライバを使用すると、ODBC 規格に準拠しているアプリケーションから SAS データソースにアクセス、操作、更新することができます。

**ODS**
関連項目:Output Delivery System

**OLE**
関連項目:Object Linking and Embedding

**Open Database Connectivity**
データのアクセスを可能にするために、共通アプリケーションプログラミングインターフェイス(API)を提供するインターフェイスの規格。Windows 動作環境で実行する多くのソフトウェア製品は、この規格を順守しているため、他のソフトウェア製品を使用して作成されたデータへのアクセスが可能になります。短縮形:ODBC

**Output Delivery System**
マークアップ言語(HTML、XML)、PDF、リスト、RTF、ポストスクリプト、SAS データセットなどのさまざまな形式の出力を作成できる SAS コンポーネント。短縮形:ODS

**pathname**
ディレクトリの階層。通常はファイル名で終わり、アプリケーションまたはユーザーがファイル検索のために移動する場所。パス名は絶対的な(システム内での完全なアドレス)ものであるか、相対的な(システムの他の部分からの位置)もののいずれかになります。

**PCL**
関連項目:Printer Command Language

**PID**
関連項目:プロセス ID

**Printer Command Language**
Hewlett-Packard により開発された、Hewlett-Packard プリンタの制御コマンド言語。各 PCL コマンドは、Esc キーと一連のコード番号から構成されます。PCL は、Hewlett-Packard プリンタの各種モデルまたはタイプで使用できるように各種バージョンが開発されました。短縮形:PCL

**Profile カタログ**
関連項目:Sasuser.Profile カタログ

**SAS AWS**
関連項目:アプリケーションワークスペース

**sasroot**
サイトまたはコンピュータ上での、SAS がインストールされているディレクトリやフォルダの名前の表示。

**SASUSER.PROFILE カタログ**
特定のユーザーまたはサイトの SAS ウィンドウ環境の属性に関する情報を格納する SAS カタログ。このカタログには、ファンクションキーの定義、グラフィックアプリケーションのフォント、ウィンドウ属性、対話型 SAS プロシジャで使用されるその他の情報が含まれます。

**SAS システムオプション**
SAS セッション中に任意の数の操作に適用される SAS 言語要素の タイプ。システムオプションは、SAS セッションの初期化、ハードウェア/ソフトウェアとの SAS 相互通信、SAS ファイルの入力/出力処理を制御できます。

**SAS 名**
SAS 変数や SAS データセットのようなアイテムに関連づけられる名前。ほとんどの SAS 名は、文字かアンダーラインで始まる必要があります。それに続く文字は複数の文字、数字、あるいはアンダーラインが使えます。空白と特殊文字(アンダーラインを除く)は使用できません。ただし、VALIDVARNAME=システムオプションにより、どのルールが SAS の変名に適用されるかを指定できます。SAS 名の最大長は、割り当てられた言語要素によって決まります。

**SMP**
関連項目:対称型マルチプロセッシング

**TCP/IP**
2 種類のネットワーキングプロトコル名の省略形。TCP (Transmission Control Protocol)は、イーサネット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際の標準的なプロトコルです。TCP を利用すると、プロセス間の情報を適切な順序で転送できます。IP (Internet Protocol)は、動作環境間の接続を管理するプロトコルです。IP は、ネットワーク経由で情報を特定の動作環境にルーティングします。また、転送時に情報は細分化され、組み立てられます。

**アプリケーションワークスペース**
他のウィンドウ(子ウィンドウ)を含むウィンドウや、他のウィンドウの起動が可能なウィンドウ。ただし、同一のソフトウェアアプリケーションの一部である親ウィンドウ内には含まれません。短縮形:AWS

**ウィンドウバー**
現在の SAS セッションに開かれている SAS ウィンドウを表すボタンが表示される、SAS メインウィンドウの下部にあるバー。いずれかのボタンを選択すると、選択したボタンに関連付けられたウィンドウがアクティブウィンドウになり、他のウィンドウよりも手前に表示されます。ボタンを右クリックすると、関連付けられたウィンドウの移動、サイズ変更、最小化、最大化、終了が行えるメニューが表示されるか、そのウィンドウに特有の他のメニューが表示されます。

**エラーメッセージ**
SAS でプログラムの処理を続けられないことを示すメッセージ。SAS ログやメッセージウィンドウに表示されます。

**オブジェクト**
プログラミング言語のコマンドによって操作されるエンティティ。オブジェクト指向のプログラミングでは、オブジェクトは属性(オブジェクト要素)とエンティティを記述する動作(メソッド)を編集したものになります。単一の情報(例:int=10)の構成物である単純なデータタイプと違い、オブジェクトは複雑であり、構成される必要があります。

**外部ファイル**
ホストオペレーティングシステム、または他のベンダのソフトウェアアプリケーションを使用して作成、メンテナンスされるファイル。外部ファイルはデータと保存された SAS のステートメントの両方を読み込むことが可能です。

**拡張エディタ**
SAS ユーザーが SAS プログラムの作成時やデバッグ時に利用できる、色分け機能やコードセクション機能などの機能を備えた ASCII テキストエディタ。拡張エディタには、SAS プログラムエディタの機能も備わっています。

**拡張サーバーメモリ**
32 ビットのオペレーティングシステムで実行しているサーバーでは、4GB の基本メモリを超えるメインメモリを表します。

**拡張メモリ**
関連項目:拡張サーバーメモリ

**各国語文字**
ある国または国家グループで記述される言語に固有の文字。たとえば、“n”にティルデ(ñ)が付いた文字は、スペイン語の文字です。

**環境変数**
ある文字列を他の文字列と同等に扱いまた、特定の環境で使用できる 1 つの変数。たとえば、!SASROOT 環境変数は、!SASROOT という名前を SAS がインストールされているパスと結びつけ、SAS 内のみで使用でき、オペレーティングシステムに認識されることはありません

**基本メモリ**
32 ビットのオペレーティングシステムで実行しているサーバーでは、メインメモリの最初の 4 GB となります。64 ビットのオペレーティングシステムで実行しているサーバーでは、メインメモリすべてが基本メモリです。

**キャッシュ**
最近アクセスしたデータを保存する小容量で高速なメモリ領域。キャッシュは、同じデータへのアクセスを高速化するために使用されます。

**クラス**
オブジェクトのテンプレート。クラスには、オブジェクトの特性(属性またはインスタンス変数など)を説明するデータと、オブジェクトが実行する操作(メソッド)が含まれます。

**グラフィカルユーザーインターフェイス**
ウィンドウ、メニュー、アイコン、ボタン、チェックボックスなどのグラフィカルオブジェクトを使用してソフトウェアアプリケーションの機能を表し、このオブジェクトを使用してアプリケーションを操作するシステム。一方、コマンドラインインターフェイスは、テキストを入力してソフトウェアアプリケーションを操作します。多くのグラフィカルユーザーインターフェイスでは、ファイルキャビネット、フォルダ、定規、はさみなどの実際に存在する品物をイメージさせるグラフィックが使用されています。短縮形:GUI

**クリップボード**
アプリケーション間でデータを受け渡す際に、一時的にデータを格納する場所。たとえば、Windows 動作環境では、クリップボードを使用して、Excel と SAS セッション間で情報を受け渡すことができます。

**現在のフォルダ**
アプリケーションの実行時にコマンドや操作が適用されるフォルダ。

**構成ファイル**
SAS を実行環境を定義する SAS システムオプションを含む外部ファイル。システムオプションは、SAS の起動時に必ず有効化されます。

**コマンドプロンプト**
オペレーティングシステムコマンド入力位置を示す記号。

**コントロールパネル**
マウストラッキングスピードやタイトルバーの色など、Windows セッションの特性を指定できる Windows アプリケーション。

**コンポーネント**
オブジェクト指向のプログラミング環境で、他のコンポーネントに特定のサービスを提供する自己完結型の再利用可能なプログラミングオブジェクト。

**サーバー**
要請してくるクライアントに対して、リソースやサービスをネットワークを介して供給するソフトウェア。

**サブクラス化**
既存のクラスから新しいクラスへと導く作業。新しいクラスは、その親の特性(属性やインスタンス変数)や操作 (メソッド)を継承します。新しいクラスはまた、ユーザーが設定した属性とメソッドを持ちます。

**システムオプション**
関連項目:SAS システムオプション

**自動マクロ変数**
ユーザーではなく SAS により定義され、SAS セッションの情報が設定されるマクロ変数。たとえば、SYSPROCESSID 自動マクロ変数には、現在の SAS プロセスのプロセス ID が含まれます。

**自動呼び出しマクロ**
コンパイルされていないソースコードとテキストが自動呼び出しマクロライブラリに格納されているマクロ。コンパイル済みマクロとは異なり、自動呼び出しマクロを初めて呼び出すと、実行される前にコンパイルされます。

**署名行**
拡張エディタの場合、ステップキーワード(DATA、PROC、MACRO)が表示される SAS コードの行を表します。

**シリアルポート**
一度に 1 ビットずつデータが伝送される I/O ポート(通常、RS-232 インターフェイスが使用される)。大部分のプロッタや、一部のレーザープリンタは、シリアルポートを使用してホストコンピュータに接続されます。

**スレッド**
オペレーティングシステムによってスケジュール設定可能な 処理の最小単位。

**スワッピング**
関連項目:スワップ

**スワップ**
データやプログラムコードをコンピュータシステムのメインメモリとハードディスクなどのストレージデバイスの間で移動すること。

**対称型マルチプロセッシング**
入出力や処理の高速化が可能なハードウェアおよびソフトウェアアーキテクチャ。SMP マシンには、複数の CPU と、スレッドに対応したオペレーティングシステムが

組み込まれます。通常、SMP マシンは複数のコントローラから構成されます。1 つのコントローラに対して複数のディスクドライバが用意されます。短縮形:SMP

**タイトルバー**
Windows で、ウィンドウのタイトルを表示するウィンドウの要素。タイトルバーはウィンドウの上部にあり、ウィンドウがアクティブのときに強調表示されます。

**タスクバー**
Windows デスクトップの下部にある、アクティブなアプリケーションを表示するバー。タスクバーを使用すると、アプリケーションを簡単に切り替えたり、アプリケーションを元のサイズに戻したり、移動、サイズ変更、最小化、最大化、終了が行えます。

**致命的エラー**
プログラムの異常終了の原因や、プログラムの起動の妨げとなるエラー。

**中央演算処理装置**
コンピュータの主要なハードウェア要素。CPU では、プログラム命令を実行し、コンピュータの他の部分の操作を制御します。短縮形:CPU

**ツールチップ**
ツールバー上のツールアイコンのような、グラフィカルユーザーインターフェースの特定の要素上にカーソルを置いたときに表示される記述テキスト。

**ツールバー**
Windows で、SAS コマンドまたは SAS マクロに関連付けられたアイコンを表示する SAS ウィンドウ環境の要素。アイコンを選択すると、関連付けられたコマンドやコマンド行が実行されます。ツールバーは、メイン SAS ウィンドウのメニューバーの下にあります。

**ツールボックス**
SAS コマンドまたは SAS マクロに関連付けられたアイコンが収められた SAS ウィンドウ環境の要素。アイコンを選択すると、関連付けられたコマンドやコマンド行が実行されます。

**テキスト文字列**
関連項目:文字列

**デバイスドライバ**
コンピュータと、プリンタやディスクドライブなどの外部デバイス間の操作を制御するプログラム。

**動的データ交換**
アプリケーション間でデータを共有するための、PC 環境の標準メカニズム。短縮形:DDE

**動的リンクライブラリ**
実行時に、必要に応じて読み込まれる実行可能なプログラムモジュールの集まり。短縮形:DLL

**ドッキングビュー**
[エクスプローラ]ウィンドウや[結果]ウィンドウなどを左側に組み合わせた、メイン SAS ウィンドウのビュー。

**名前付きパイプ**
クライアントとサーバー間、サーバーとクライアント間、または関連付けられていないプロセス間の双方向通信を可能にする、名前付きオブジェクト。名前付きパイプを使用することで、複数の SAS セッションを含む Windows アプリケーション間の通信が可能になります。

**名前の付いていないパイプ**
UNIX オペレーティングシステムと、その派生 OS 上で、あるコマンドを他のコマンドにリンクして、一方の標準出力が他方の標準入力になる機能。

**ネットワーク**
相互接続された複数のコンピュータ。

**バイナリ**
2 進法を表す用語。2 進数は 2 つの値のうち 1 つを持ちます。0 もしくは 1 です。2 進数はビットと呼ばれ、値が 0 の場合は off、値が 1 の場合は on になります。

**バイナリファイル**
バイナリ形式で保存されるファイル。テキストエディタで編集することはできません。バイナリファイルは通常は実行可能ですが、データのみを含む場合があります。

**パイプ**
関連項目:名前の付いていないパイプ

**バッチファイル**
ファイル実行時に連続して処理されるオペレーティングシステムコマンドを含むファイル。

**バッチモード**
実行中の SAS プログラムの非対話型メソッド 、ここでファイル(あらゆる必要なオペレーティングシステムのコマンドとをともなう SAS ステートメントを含む)は動作環境のバッチキューへとサブミットされて処理されます。

**標準出力**
コマンドから導かれるデータの主な出力先。標準出力は、ファイルにリダイレクトされたり、別のコマンドにパイピングされたりしない限り画面に表示されます。

**標準入力**
コマンドに取り込まれるデータの主なソース。標準入力は、ファイルからリダイレクトされたり、別のコマンドからパイピングされたりしない限り、キーボードから入力されます。

**ファイル拡張子**
ファイルに保存されている情報の種類を識別できる、ディレクトリのファイルの分類。たとえば、.sas7bcat は UNIX のファイル拡張子、.pdf は Adobe Acrobat のファイル拡張子です。

**フォント**
特定の文字の形、間隔、重み、サイズを持つ書体。フォント内の文字は数字、記号、英数字です。

**プロセス ID**
オペレーティングシステムによって、プロセスごとに割り当てられる重複しない番号。短縮形:PID

**分散 COM**
ネットワーク上の別のコンピュータにあるコンポーネント間でのサービスの要求を可能にする、コンポーネントオブジェクトモデル(COM)の拡張仕様。短縮形:DCOM。

**ポータブル**
作成された動作環境以外の動作環境でプログラムを実行できる属性。

**ホストオプション**
SAS ステートメントでは、特定の動作環境に特有のオプションを表します。

**メモリ**
中央演算処理装置(CPU)がプログラムの操作に割り当てる必要がある作業領域のサイズ。

**メモリベースのライブラリ**
SAS セッションまたは SAS ジョブの実行時に、データストレージデバイスではなく、基本メモリまたは拡張サーバーメモリに格納される SAS ライブラリ。

**メンバ名**
SAS ライブラリの SAS ファイルに割り当てられる名前。

**文字値**
英字、0 から 9 の数字、その他の特殊文字を含む値。

**文字セット**
1 つの言語や言語グループで使用される文字のセット。1 つの文字セットには各国文字、特殊文字、0 から 9 までの数字、制御文字が含まれます。

**文字定数**
SAS ステートメントの引用符で囲まれた文字列。変数名ではなく、固定値を示します。最大文字数は、32,767 です。文字定数は、文字リテラルと呼ばれることもあります。

**文字列**
英数字または他のキーボード文字、あるいはその両方。

**文字列**
関連項目:文字列

**ユニバーサル印刷**
SAS 出力を PDF、Postscript、GIF、PNG、SVG、PCL ファイル、プリンタに送信する SAS の機能。ユニバーサル印刷システムでは、出力をカスタマイズする多数のオプションが提供されます。また、ユニバーサル印刷システムは、SAS がサポートするすべての動作環境で利用できます。

**ライブラリ参照**
関連項目:libref

**ライブラリメンバー**
SAS ライブラリの SAS ファイル形式のいずれか。ライブラリメンバーには、データセット、ビュー、カタログ、保存されたプログラム、アクセス記述子が含まれます。

**論理名**
SAS ライブラリまたは動作環境によって表示方法が変化する動作環境リソースへの参照。例は Work という名の SAS ライブラリとディスクまたはテープファイルのような出力デバイスです。

# キーワード


_
_ALL_オプション
  SYSTASK ステートメント 463
_ANY_オプション
  WAITFOR ステートメント 466
_COMPUTE_メソッド 608
_DISABLE_DEFAULT_ACTION_メソッド 609
_DO_メソッド 609
_ENABLE_DEFAULT_ACTION_メソッド 610
_EXECUTE_メソッド 610
_GET_EVENT_メソッド 611
_GET_PROPERTY_メソッド 611
_GET_REFERENCE_ID_メソッド 612
_GET_TYPE_メソッド 613
_IN_ERROR_メソッド 613
_NEW_メソッド 614
_SET_PROPERTY_メソッド 614
_UPDATE_メソッド 615

!
!DDE_FLUSH 文字列 275

.
.INI ファイル 223
  ASCII エディタを使用して作成する 226
  SSCU を使用して作成する 223

$
$BYVALw.出力形式
  DLL 304
$CSTRw.出力形式
  DLL 304
$HEXw.出力形式 379
$HEXw.入力形式 418

%
%INCLUDE ステートメント 455
  外部ファイルの読み込み 159
%SYSEXEC ステートメント 601
%SYSGET 関数 602

1
16 進値
  固定小数点表現のバイナリ値への変換 417
  バイナリ実数値の変換 378
  浮動小数点表現のバイナリ値への変換 417
  文字値の変換 379
  文字データへの変換 418

A
ABEND オプション
  ABORT ステートメント 443
ABORT ステートメント 443
ACCESSIBILITY システムオプション 486
ACCESS メンバタイプ 122
ALIGNSASIOFILES システムオプション 486
ALTLOG システムオプション 487
ALTPRINT システムオプション 488
APPEND システムオプション 489
ARG ステートメント
  DLL 295
ASCII
  代替文字 57
ASCII and EBCDIC
  conversion issues 647
ASCII file structures 648
ASCII numeric encodings 649
ASCII systems 647
ASCII 照合順序
  1 文字返す 390
  文字位置 410
  文字列を返す 393
ATTRIB ステートメント 444
AUTHPROVIDERDOMAIN システムオプション 490

AUTHSERVER システムオプション 490
AutoComplete 43
AUTOEXEC システムオプション 491
autoexec ファイル 20
  作成に使用するテキストエディタ 20
  使用 21
  代替ファイル 491
  デフォルト 20
  名前が変更されたファイルの検索 21
  非表示 21
AutoScroll 55
AUTOSCROLL コマンド 327
AWSCONTROL システムオプション 492
AWSDEF システムオプション 493
AWSMAXIMIZE コマンド 328
AWSMENUMERGE システムオプション 495
AWSMENU システムオプション 494
AWSMINIMIZE コマンド 328
AWSRESTORE コマンド 329
AWSTITLE システムオプション 495

B
BMDP Engine 141
BMDP ファイル
  SAS データセットへの変換 428
  読み取り 141
BUFNO システムオプション 496
BUFSIZE システムオプション 497
BYTE 関数 390

C
CALL RECONNECT ルーチン 289
CALL ルーチン 389
CAPS コマンド 329
CATALOG プロシジャ 425
CATCACHE システムオプション 498
  パフォーマンス 203
CGM ドライバ 194
CGM ファイル
  SAS/GRAPH での作成 193
  他のアプリケーションにエクスポート 193
CIMPORT プロシジャ 426
  SAS 9.4 のファイルを以前のリリースで使用する 140
CLEANUP システムオプション 499
cleanwork ユーティリティ 638
COBOL
  DLL 形式 303
COLLATE 関数 393
COLOR コマンド 330
COMDEF システムオプション 500
COMMAND オプション
  SYSTASK ステートメント 463
COMMAND コマンド 330
COMPRESS データセットオプション
  パフォーマンス 204
CONFIG システムオプション 501
CONTENTS プロシジャ 427
CONVERT プロシジャ 428
CPORT プロシジャ 431
  SAS 9.4 のファイルを以前のリリースで使用する 140
CUT コマンド 332
C 言語
  DLL 形式 301

D
data store 647
DATASETS プロシジャ 432
DATA ステップ
  開始時間のスケジュール 412
  出力の表示 178
  中断 410
  電子メールの送信 48, 51
DDE (動的データ交換) 267
  !DDE_FLUSH 文字列 275
  DDE トリプレット 269
  Excel からの読み込み 271
  Excel への書き出し 271
  HOTLINK オプション 274
  NOTAB オプション 273
  Word からの読み込み 272
  Word への書き出し 271
  アプリケーションコマンドの起動 272
  アプリケーションの制御 270
  外部ファイルの参照 269
  欠損値 269, 276
  構文 268
  タブ文字 273
  動的データ転送 275
  例 270
DDE サーバー
  X コマンドを使用して開く 270
DDE トリプレット 269
DEVICE システムオプション 501
DINFO 関数 394
DLGABOUT コマンド 332
DLGCDIR コマンド 333
DLGCOLUMNSIZE コマンド 333
DLGCOLUMNSORT コマンド 333
DLGCONVERT コマンド 334
DLGENDR コマンド 334
DLGFIND コマンド 334
DLGFONT コマンド 335
DLGLIB コマンド 335
DLGLINKS コマンド 336
DLGOPEN コマンド 336

DLGPAGESETUP コマンド 338
DLGPREF コマンド 338
DLGPRTPREVIEW コマンド 340
DLGPRTSETUP コマンド 340
DLGPRT コマンド 339
DLGREPLACE コマンド 341
DLGRUN コマンド 341
DLGSAVE コマンド 342
DLGSMAIL コマンド 344
DLL 291
  IML プロシジャからルーチンを起動する 310
  MODULE 引数の出力形式 301
  MODULE 引数の入力形式 301
  MODULE 関数 407
  MODULE の引数として使用する式 300
  MODULE の引数として使用する定数 300
  MODULE ログメッセージ 305
  PEEK 関数 297
  SASCBTBL 属性テーブル 292
  SAS からアクセスする 318
  引数を値として渡す 308
  構造 309
  効率的にアクセスする 298
  変数を構造引数としてグループ化する 299
  文字列引数 297
  文字列引数を更新する 307
  呼び出し 407
  戻されたポインタにアクセスする 308
DOPEN 関数 395
DOPTNAME 関数 395
DOPTNUM 関数 396
DOS シェル
  EXIT コマンドを使用可能にする 592
  終了する 30

## E

EBCDIC and ASCII
  conversion issues 647
EBCDIC file structures 648
EBCDIC numeric encodings 649
EBCDIC systems 647
ECHO システムオプション 502
EMAILDLG システムオプション 503
EMAILSYS システムオプション 504
ENCODING=オプション
  %INCLUDE ステートメント 455
  FILENAME ステートメント 448
  FILE ステートメント 445
  INFILE ステートメント 457
ENGINE システムオプション 505
ENHANCEDEDITOR システムオプション 506
ERRORLEVEL バッチ変数 617
Excel アプリケーション
  DDE を使用した読み込み 271
Excel スプレッドシート
  OLE 248

## F

FDELETE 関数 397
FEXIST 関数 397
file structures
  ASCII 648
  EBCDIC 648
FILEEXIST 関数 398
FILELOCKWAITMAX= システムオプション 507
FILELOCKWAIT システムオプション 506
FILENAME 関数 398
FILENAME ステートメント 448
  DDE 構文 268
  電子メールの送信 48
  名前付きパイプ 282
  名前の付いていないパイプ 280
FILEOPEN コマンド 346
FILEREF 関数 399
FILE コマンド 160, 344
FILE ステートメント 445
  外部ファイルへの書き込み 158
FILL コマンド 346
FILTERLIST システムオプション 508
FINFO 関数 400
FONTALIAS システムオプション 510
FONTSLOC システムオプション 511
FONT システムオプション 509
FOOTNOTE ステートメント 454
FOPTNAME 関数 402
FOPTNUM 関数 403
FORMCHAR システムオプション 512
FORM ウィンドウ 177
FORTRAN
  DLL 形式 302
FRAME エントリ
  OLE オブジェクトの挿入 234
  OLE オブジェクトの編集 238
FULLSTIMER システムオプション 513

## G

GSUBMIT 347
GSUBMIT コマンド 161, 347
GUI 5

## H

HELPHOST システムオプション 77, 515
HELPINDEX システムオプション 515
HELPLOC システムオプション 516
HELPPORT システムオプション 77
HELPREGISTER システムオプション 517
HELPTOC システムオプション 519
HEXw.出力形式 378
HEXw.入力形式 417
HOME コマンド 347
HOSTINFOLONG システムオプション 520
HOSTPRINT システムオプション 521
HOTLINK オプション
  DDE 274
HSERVICE エントリ
  OLE オブジェクトを読み込む 236
HTML 出力 61
  ODS 181
  プリファレンス 180
HTML ヘルプ 73

## I

I/O のディスクペース 198
IBw.d 出力形式 380
IBw.d 入力形式 419
ICON コマンド 348
ICON システムオプション 521
INCLUDE コマンド 161, 348
INFILE ステートメント 457
  外部ファイルの読み込み 159
INITSTMT システムオプション 522
INSERT システムオプション 523
IntelliMouse 55

## J

Java Runtime Environment (JRE)オプション 524
JREOPTIONS システムオプション 524

## K

KEYS ウィンドウ
  カタログエントリにコンテンツを保存する 351
  マウスボタン 539
KILL オプション
  SYSTASK ステートメント 463

## L

LENGTH ステートメント 459
LIBNAME ウィンドウ
  ライブラリ参照名のクリア 131
LIBNAME 関数 404
  ライブラリ参照名のクリア 132
LIBNAME ステートメント 460
  ライブラリ参照名のクリア 131
  ライブラリ参照名の割り当て 126
LINESIZE システムオプション 525
LIST オプション
  SYSTASK ステートメント 463
LOADMEMSIZE システムオプション 526
LOGEVENT.EXE ユーティリティ 216
LOG システムオプション 526

## M

MAPS システムオプション 528
MAXMEMQUERY システムオプション 529
MCIPISLP 関数 405
MCIPISTR 関数 406
member-name 構文 151
  解釈 153
MEMBLKSZ システムオプション 529
MEMCACHE システムオプション 531
  64 ビット Windows 環境 200
MEMLIB オプション
  LIBNAME ステートメント 460
MEMLIB システムオプション 532
  64 ビット Windows 環境 200
MEMMAXSZ システムオプション 532
MEMSIZE システムオプション 533
Microsoft HTML ヘルプ 73, 516
Microsoft IntelliMouse 55
MNAME=オプション
  SYSTASK ステートメント 463
MODULE 関数 407
  引数として使用する式 300
  引数として使用する定数 300
  出力形式 301
  入力形式 301
  ログメッセージ 305
MS-DOS コマンド
  SAS から実行する 28
MSGCASE システムオプション 535
MSG システムオプション 534
MSYMTABMAX システムオプション 536
MVARSIZE システムオプション 537

## N

NEWS システムオプション 538
nibble 649
NOBLOCK オプション
  名前付きパイプ 287

NOTAB オプション
DDE 273
Notepad ウィンドウ
カタログエントリにコンテンツを保存する 351
NOTEPAD テキストエディタ
コードをサブミットする 25
numeric values
ASCII and EBCDIC 649
NUMKEYS システムオプション 538
NUMMOUSEKEYS システムオプション 539

## O

OBS システムオプション 540
OCX
*参照項目*: OLE カスタムコントロール (OCX)
ODBC ドライバ 145
ODS HTML 出力 181
ODS 出力
リモートブラウジング 76
OLE 234
FRAME エントリ内でのオブジェクトの編集 238
OLE Verb の起動 239
SAS/AF アプリケーションでの OCX 250
SAS/AF カタログの互換性 234
エラーメッセージ 620
オブジェクトとアプリケーションの自動化 243
オブジェクトの変換 241
オブジェクトを FRAME エントリに挿入する 234
リンクされたオブジェクト 240
OLE Verb
起動 239
OLE オブジェクト
FRAME エントリ内での編集 238
FRAME エントリへの挿入 234, 237
HSERVICE エントリからの読み込み 236
オートメーションメソッドとプロパティ 262
クリップボードから貼り付ける 236
自動化 243
変換 241
OLE カスタムコントロール(OCX) 250
FRAME エントリへの挿入 251
SCL コードをイベントに割り当てる 253
SCL メソッド 253
イベントから引数値を取り込む 254
イベント処理 253, 255
イベントマップダイアログボックス 253
コンボボックスにアイテムを追加する 256
コンボボックスのアイテムを検索する 256
サブクラス化 255
テキスト値を取り込む 256
登録 251
プロパティ 251
OLE クラスメソッド 607
OLE 自動化 243, 259
Excel スプレッドシートを取り込む 248
SAS セッションからのフィードバック 260
SAS のインスタンスの作成 260
Visual Basic のコードと対応する SCL 249
値のプロパティ 245
オブジェクトのメソッドとプロパティ 262
外部インスタンスの作成 246
サーバーが返した配列値 244
サーバーメソッドのオプションパラメータ 246
例 261
OPLIST システムオプション 540
OPTIONS ステートメント
システムオプション 477
OPTIONS プロシジャ 433
OSIRIS Engine 142
OSIRIS ファイル
データセットへの変換 428
読み取り 141

## P

PAGENO システムオプション 541
PAGESIZE システムオプション 542
PAPERTYPE システムオプション 543
PATH システムオプション 543
PCLEAR コマンド 327
PDw.d 出力形式 382
PDw.d 入力形式 420
PEEKLONG 関数 409
文字列引数にアクセスする 297
PFKEY システムオプション 544
PIBw.d 出力形式 383
PIBw.d 入力形式 421
PL/I
DLL 形式 302
PLIST コマンド 327
PMENU コマンド 350
PMENU プロシジャ 434
PRIMARYPROVIDERDOMAIN= システムオプション 547
PRINTTO プロシジャ 435
プロシジャ出力の送信 183
PRINT システムオプション 547

PRNGETLIST システムオプション 548
PROC SQL クエリ 220
Profile カタログ 22
  削除 22
  デフォルト 22
  場所の変更 22
PRTABORTDLGS システムオプション 549
PRTPERSISTDEFAULT システムオプション 549
PRTSETFORMS システムオプション 550

## Q

QueryWindow メソッド
  メソッド 263
Quit メソッド
  メソッド 264

## R

RANK 関数 410
RBw.d 出力形式 384
RBw.d 入力形式 422
RBwd 384
REALMEMSIZE システムオプション 551
REGISTER システムオプション 552
Remote Browser Server 75
  インストール 76
  ポート番号 77
RESOURCESLOC システムオプション 553
RETURN オプション
  ABORT ステートメント 443
ROUTINE ステートメント
  DLL 293
RSASUSER システムオプション 553
RTF (リッチテキスト形式)
  ウィンドウのコンテンツを保存する 369
RTF(リッチテキストフォーマット) 114
RTRACELOC システムオプション 555
RTRACE システムオプション 554

## S

SAS 9.4 ファイル
  以前のリリース 140
SAS Institute の Web サイト 75
SAS NOTEPAD テキストエディタ
  コードをサブミットする 25
SAS Web ページ 75
SAS_EXECFILENAME 環境変数 94
SAS_EXECFILEPATH 環境変数 94
SAS, 起動 5
  SAS が起動しない場合 14
  SAS ファイル 9
  Windows のサービスとして 222
  コマンドプロンプトから 9
  実行ダイアログボックス 9
  ショートカットから 8
  スタートメニューから 8
  代替構成ファイル 17
  対話モード 5
  バッチモード 5
  プログラムアイテムから 8
SAS/AF アプリケーション
  OCX 250
  アイコン 319
  自動起動 319
  ロゴ 319
SAS/AF カタログ
  OLE の互換性 234
SAS/GRAPH
  CGM ファイルを作成 193
  ネイティブドライバ 188, 189
  汎用ドライバ 187
SASAUTOS システムオプション 556, 602
SASCBTBL 属性テーブル 292
  ARG ステートメント 295
  ROUTINE ステートメント 293
  構文 293
  重要性 296
SASCONTROL システムオプション 557
Sashelp カタログ
  エントリの検索 558
SASHELP システムオプション 558
SASINITIALFOLDER システムオプション 558
sasiotest コマンド 643
SASUSER システムオプション 559
Sasuser データライブラリ 133
  エンジンの変更 133
Sasuser ライブラリ
  名前の指定 559
  読み込み/書き込みアクセス制御 553
Sasuser ライブラリ参照名
  フォルダ 133
SAS オートメーションオブジェクト 261, 265
SAS コマンド 9
  システムオプションの追加 8
SAS サービス
  インストール 229
  起動 231
  構成ユーティリティ(SSCU) 223
  削除 231
SAS システムダイアログボックスについて 75, 332
SAS セッション 39
  ウィンドウタイトルの設定 262

カスタマイズ 57
最小化 65
サンプル 11
終了 33, 262
終了時に設定を保存する 59
終了の確認 59
セッションプリファレンス 58
タイトル 65
中断 28
ビジーステータスの確認 261
ビットマップの貼り付け 56
表示/非表示の切り替え 262
フォントの選択 57
SAS セッションプリファレンス
Web 61
詳細 61
全般 59
表示 60
編集 60
SAS セッションを中断する 28
SAS 入門ガイド 74
SAS の起動
*参照項目*: SAS, 起動
SAS の反応性 199
SAS ファイル 14
*関連項目*: ファイル
autoexec ファイル 20
Profile カタログ 22
SAS の起動 9
Work データライブラリ 23
構成ファイル 15
レジストリファイル 23
SAS プログラミングの学習 74
SAS プロセス
終了 32
SAS ヘルプとドキュメント 74
SAS リソース
ディレクトリの場所 553
SAS ログ機能 185
SAVE コマンド 351
Scatter-read/Gather-write 204, 562
データセットのアクティブ化 323
SCL
電子メールの送信 48, 52
SCL メソッド
OLE 607
OLE コントロール 253
SCROLLBARFLASH システムオプション 560
SCROLLBAR コマンド 327
SET Windows コマンド 130
SET コマンド
外部ファイルのファイル参照名 151
SET システムオプション 561
外部ファイルのファイル参照名 151
SGIO システムオプション 562
SGIO データセットオプション 323
SHELL=オプション
SYSTASK ステートメント 463
SLEEPWINDOW システムオプション 563
SLEEP ウィンドウ
有効化/無効化 563
SLEEP 関数 410
SMARK コマンド 327
SMP (対称型マルチプロセッシング) 199
SMTP
電子メールの送信 52
SORTANOM システムオプション 563
SORTCUTP システムオプション 565
SORTCUT システムオプション 564
SORTDEV システムオプション 566
SORTNAME システムオプション 567
SORTPARM システムオプション 567
SORTPGM システムオプション 568
SORTSEQ=オプション
PROC SORT ステートメント 440
SORTSIZE=システムオプション
パフォーマンス 205
SORTSIZE システムオプション 569
SORT プロシジャ 436
SyncSort 439
ソートユーティリティ 568
パフォーマンス 205, 219
メモリの制限 569
SOUND Call ルーチン 390
SPLASHLOC システムオプション 571
SPLASH システムオプション 570
SPSS Engine 143
SPSS ファイル
再フォーマット 144
データセットへの変換 428
読み取り 141
SSCU (SAS Service Configuration Utility) 223
STATE オプション
SYSTASK ステートメント 463
STATUS=オプション
SYSTASK ステートメント 463
STATVAR オプション
SYSTASK ステートメント 463
STIMEFMT システムオプション 571
STIMER システムオプション 574
STORE コマンド 351
Submit メソッド
メソッド 264
SUBTOP コマンド 93, 352
SyncSort 439
アクティブにする 564, 565
一時ファイルの場所 440
一時ファイルのパス名 566
オプションの指定 563

オプションを渡す 440
サイズまたはオブザベーションを基準にして並べ替える 439
ソートアルゴリズムとして設定する 439
パラメータの指定 567
パラメータを渡す 440
SYSCC 自動マクロ変数 599
SYSDEVIC 自動マクロ変数 600
SYSENV 自動マクロ変数 600
SYSGUIFONT システムオプション 576
SYSIN システムオプション 577
SYSJOBID 自動マクロ変数 600
SYSMAXLONG 自動マクロ変数 600
SYSPARM システムオプション 577
SYSPRINTFONT システムオプション 579
SYSPRINT システムオプション 578
SYSRC 自動マクロ変数 600
SYSSCPL 自動マクロ変数 600
SYSSCP 自動マクロ変数 600
SYSTASK ステートメント 463
SYSTEM Call ルーチン 391

**T**

TAGSORT オプション
PROC SORT ステートメント 438
パフォーマンス 205
TASKNAME=オプション
SYSTASK ステートメント 463
TIMEOUT=オプション
WAITFOR ステートメント 466
TITLE ステートメント 466
TOOLCLOSE コマンド 353
TOOLDEF システムオプション 581
TOOLEDIT コマンド 353
TOOLLARGE コマンド 353
TOOLLOAD コマンド 354
TOOLSMENU システムオプション 582
TOOLSWITCH コマンド 355
TOOLTIPS コマンド 355
Top メソッド
メソッド 264
TRANSLATE 関数 411
TrueType フォント 625

**U**

UNIVERSALPRINT システムオプション 165, 583
UPRINTMENUSWITCH システムオプション 165, 584
USERICON システムオプション 585
USER システムオプション 585

**V**

VERBOSE システムオプション 586
VIEWMENU システムオプション 587
Visual Basic のコード
対応する SCL 249

**W**

WAITFOR ステートメント 466
WAIT オプション
SYSTASK ステートメント 463
WAKEUP 関数 412
WATTACH コマンド 356
WATTENTION コマンド 357
WAUTOSAVE コマンド 357
WBROWSE 358
WBROWSE コマンド 358
WCOPY コマンド 358
WCUT コマンド 359
WDOCKVIEWMINIMIZE コマンド 360
WDOCKVIEWRESIZE コマンド 360, 361
WDOCKVIEWRESTORE コマンド 361
WDOCKVIEW コマンド 359
WDRAG コマンド 327
WEBUI システムオプション 587
Web 拡張 587
使用可能にする 67
Web サイト
ヘルプ 75
Web ブラウザ
開始ページ 61
起動 358
次を選択します。 61
プロシジャ出力を送る 180
Web プリファレンス 61
WEDIT コマンド 361
WEMAILFMT コマンド 362
WEXITSAVE コマンド 362
WFILE コマンド 363
WGROW コマンド 327
WHIDECURSOR コマンド 363
WHSBAR コマンド 364
Windows 4
エラーメッセージ 623
エンタープライズ環境 4
サポートされるエディション 4
システムツール 213
メモリのブロックサイズ 529
メモリベースのライブラリ 532
Windows Server 2003
サポートされるエディション 4
Windows Vista
サポートされるエディション 4
Windows XP
サポートされるエディション 4

WINDOWSMENU システムオプション 588
Windows アプリケーション
  SAS セッションから実行する 467
  サブミット 391
Windows イベントビューア
  ログメッセージの書き込み 185
Windows のコマンド
  SAS から実行する 28
  XWAIT システムオプション 30
  XWAIT システムオプションと XSYNC システムオプション 31
  X ステートメントまたはコマンドを使用して実行する 29
  条件付きで発行する 30
  同期実行と非同期実行 31
Windows のサービス
  SAS の起動 222
Windows 版 SAS
  SAS プロダクトのヘルプ 74
  バッチモード 26
Windows フォント 510
Windows プリンタマネージャ 521
WINPxxx ドライバ 187, 189
WINSERT コマンド 365
WMENUPOP コマンド 365
WMF ファイル
  グラフィックのエクスポート 195
WMOVE コマンド 327
WMRU コマンド 366
WNAVKEYUNMARK コマンド 367
WNEWTITLE コマンド 367
WNEXTEDIT コマンド 368
Word
  DDE を使用した書き込み 271
  DDE を使用した読み込み 272
WORK システムオプション 589
Work データライブラリ 23, 133
  pathname 589
  位置 23
  一時サブフォルダ 23
  エンジンの変更 134
  デフォルトのフォルダ 23
  フォルダの削除 23
  変数を使用して指定する 134
  メモリベースのライブラリ 532
WORK フォルダ 61
WPASTE コマンド 368
WPGM コマンド 368
WPOPUP コマンド 369
WRTFSAVE コマンド 369
WSCREENTIPS コマンド 370
WSHRINK コマンド 327
WSTATUSLN コマンド 371
WUNDO コマンド 372
WVSBAR コマンド 372
WWINDOWBAR コマンド 373

## X

XCMD システムオプション 590
XMIN システムオプション 591
XSYNC システムオプション 591
  XWAIT システムオプション 31
  同期コマンド実行と非同期コマンド実行 31
XWAIT システムオプション 592
  DOS シェルを終了する 30
  XSYNC システムオプション 31
X コマンド 374
  DDE サーバーを開く 270
  Windows のコマンドを実行する 29
  現在の SAS セッションで有効 590
  最小化したアプリケーションを開く 591
  同期と非同期 591
X ステートメント 467
  Windows のコマンドを実行する 29
  同期と非同期 591

## Z

ZDw.d 出力形式 386
ZDw.d 入力形式 423
ZOOM コマンド 374

## あ

アイコン
  SAS/AF アプリケーション 319
  ユーザー定義 67, 585
アウトプットウィンドウ
  カタログエントリにコンテンツを保存する 351
  行サイズ 525
  スクロール頻度 327
値のプロパティ 245
アプリケーション
  最小化 591
  ツールバー 71
  ツールメニューに追加する 66, 552
アプリケーションログ
  メッセージ, LOGEVENT.EXE 216
  メッセージ, ユーザー定義関数 214
安全なサーバーログイン
  検索 490
以前のリリースからの移行 136
一般プリファレンス 59
  ツールバーのカスタマイズ 68
イベントビューア
  SAS イベントの表示 214
  アプリケーションログ 214
  ログメッセージの書き込み 185

イベントマップダイアログボックス 253
イメージ使用状況の統計量 513
色
  ウィンドウの色の変更 63
  カラー印刷 174
  選択したウィンドウ部分 330
  テキストの色属性 56
印刷 165
  DATA ステップ出力 178
  FORM ウィンドウ 177
  GRAPH ウィンドウから 189
  GRAPH ウィンドウでのプレビュー 189
  SAS/GRAPH ネイティブドライバ 188, 189
  SAS/GRAPH 汎用ドライバ 187
  Windows プリンタマネージャ 521
  WINPxxx ドライバ 187, 189
  印刷可能なウィンドウ 168
  印刷形式 176
  印刷ダイアログボックス 168
  ウィンドウ 168
  ウィンドウ内 166
  ウィンドウのプレビュー 175
  オプション 166
  カラー印刷 174
  行サイズ 173
  行番号 174
  グラフィック 186
  コマンド 177
  システムに割り当てられたプリンタの認識 548
  出力先プリンタ 549, 578
  ジョブの取り消し 180
  中止 549
  デフォルトプリンタ 179
  バッチモード 179
  バッチモードのプロシジャ出力ファイル 547
  ビットマップ 174
  ファイル 178
  フォーム 550
  フォント 579
  フォントの指定 169
  部数 174
  プリンタ設定 340
  プリンタの指定 169
  プリンタの変更 168
  ページサイズ 173
  ページ設定 170
  ページ範囲 174
  ページ番号 174
  問題 14
  用紙の種類 543
印刷オプション 166
印刷形式 176
印刷するページ範囲 174
印刷設定ダイアログボックス
  開く 340
  フォームを使用するチェックボックス 550
印刷ダイアログボックス 168, 339
印刷中止 549
印刷中止ダイアログボックス 549
印刷プレビューウィンドウ 175
  起動 340
  ショートカットキー 175, 635
インターフェイス 36
インターフェイスデータファイル 121
インターフェイスライブラリエンジン 124
インデックス 121
インデックスファイル
  SAS ヘルプとドキュメント 515
インデント
  自動 91
引用符で囲まれた Windows ファイル名 157
ウィンドウ
  RTF ファイルに保存する 369
  アクティブウィンドウを最後に編集したウィンドウに変更する 368
  位置 62
  色 56, 330
  印刷 168
  印刷する 166
  外部ファイルの行をコピーする 348
  外部ファイルへの保存 53
  切り取ってクリップボードに移動する 359
  クリア 54, 367
  クリップボードから貼り付ける 368
  クリップボードにコピーする 351, 358
  現在のウィンドウを添付ファイルとしてメールで送信する 59
  コンテンツを電子メールで送信する 47
  コンテンツをファイルに保存する 344, 363
  最小化 348
  最大化 374
  ツールバー 71
  テキストの強調表示 56
  テキストを切り取る 332
  電子メールにコンテンツを添付する 356
  特殊文字の属性 56
  フォント 509, 576
ウィンドウ環境 36, 37
  色 63
  コマンドを使用してカスタマイズする 62
  システムオプションを使用してカスタマイズする 64

メイン SAS ウィンドウを制御するコマンド 315
ウィンドウの最小化 348
ウィンドウバー 37, 40, 60
切り替え 373
ウィンドウプロシジャ
バッチジョブ 10
ウィンドウメニュー 588
埋め込み文字 346
上書きモード 60
エクスプローラウィンドウ
ライブラリ参照名のクリア 131
エディタ
拡張エディタ 79
プログラムエディタ 107
エラーチェック
拡張エディタ 90
エラーメッセージ 617
OLE 620
SAS 機能 619
大文字への変換 535
解釈 623
完了ステータス 617
終了 623
初期化 623
内部エラー 622
ネットワーク 621
ファイルへのアクセス 618
ライブラリ 534
リターンコード 617
エンジン 123
BMDP 141
OSIRIS 142
SPSS 143
決定規則 125
デフォルトのエンジン 124
ビューエンジン 123
複数の割り当て 128
変更, Sasuser データライブラリ 133
変更, Work データライブラリ 134
変数への割り当て 129
ライブラリエンジン 123
割り当て 127, 128
大文字
警告とエラーメッセージ 535
変換 329
オブザベーション
処理の終点 540
オンラインドキュメント 73
オンラインヘルプ 73

**か**

カーソル
スクロールバーの点滅 560
非表示 62
非表示にする 363
ホームポジション 347
改行
プログラムエディタ 110
外部 DLL
*参照項目:* DLL
外部ファイル 147
%INCLUDE ステートメント 159
%INCLUDE ステートメントを使用して読み込む 159
FILE コマンド 160
FILE ステートメント 158
FILE ステートメントを使用して書き出す 158
GSUBMIT コマンド 161
I/O の操作法 162
INCLUDE コマンド 161
INFILE ステートメント 159
INFILE ステートメントを使用して読み込む 159
UNC パス 154
引用符で囲まれた Windows ファイル名 157
ウィンドウのコンテンツを書き出す 160
ウィンドウのコンテンツを保存する 344
ウィンドウの保存 53
ウィンドウへの行のコピー 348
各国語文字 164
コマンドを使用してアクセスする 160
作業ディレクトリ 157
削除 397
参照 148, 454, 459
情報アイテムの値 400
情報アイテムの名前 402
ステートメントを使用してアクセスする 157
追加 163
テキストエディタウィンドウへのコピー 161
デフォルトのファイル拡張子 151
長いファイル名 154
ファイル参照名 149
ファイル参照名による検証 397
物理名による検証 398
レコード形式の変更 162
連結 150
連結ファイル 156
ワイルドカード 150
外部ファイルの存在 398
カウンタ 217, 218
拡張エディタ 79
ウィンドウ間の切り替え 368
キーワードの作成 100
機能 79
コードをサブミットする 25
自動保存ファイル 83

ショートカットキー 631
全般オプション 102
デザインオプション 104
配色 105
開くダイアログボックス 346
ファイルタイプへのファイル拡張子の割り当て 101
ファイルを開く 346
プリファレンスの設定 60
プログラムエディタに切り替える 107
無効化する 10
有効/無効 107
有効化/無効化 361, 506
拡張エディタウィンドウ
概要 80
カタログエントリにコンテンツを保存する 351
キーボードマクロ 96
行のブックマーキング 91
行番号コマンド 84
コーディングエラー 90
コードセクションの折りたたみ 99
サブミットするプログラムまたはカタログエントリのパス 93
サブミットするプログラムまたはカタログエントリのファイル名 93
自動インデント 91
ショートカットキー 95
省略形 91
スクロールコマンド 84
挿入位置 85
タブ 91
テキストの検索と置換 88
テキストの選択 86
テキストのドラッグ 88
テキストの編集 86
同一ファイルの複数ビュー 83
開く 361
ファイルの保存 82
ファイルを開く 81
プログラムのサブミット 92
ワードチップ 92
拡張エディタオプションウィンドウ 102
格納されたプログラムファイル 122
仮想メモリ 551
カタログ 121, 139
9.4 形式への変換 136
9.4 でのバージョン 6 の変換 138
Profile カタログ 22
以前のリリースからの移行 136
バージョン 7 と 8 138
リリース 6.03 および 6.04 から 9.4 への変換 140
開いておける数 498
カタログエントリ
ウィンドウのコンテンツを保存する 351
カタログの変換 136, 138, 140
リリース 6.08 から 9.4 へ 139
カタログメンバタイプ 121
各国語文字
外部ファイル 164
環境変数
外部ファイルのファイル参照名 150
拡張エディタ 93
定義 561
関数 389
完了ステータス 617
キー定義 54, 627
キーボードマクロ 96
キーワード
ユーザー定義 100
起動ファイル
システムオプション 475
キャリッジコントロール文字 184
行サイズ 173
強調表示 56
行のブックマーキング 91
行番号 174
拡張エディタ 84
行番号の使用
プログラムエディタ 109
グラフィカルインターフェイス 5
グラフィック
CGM ドライバ 194
CGM ファイルのエクスポート 193
SAS/GRAPH からのインポート 190
SAS/GRAPH へのエクスポート 192
WMF ファイルとしてエクスポートする 195
印刷 186
クリップボードから貼り付ける 190, 193
作成 195
他のアプリケーションからのインポート 190
他のアプリケーションで使用するためのエクスポート 192
ディスプレイ上での作成 185
ファイル形式のインポート 190
グラフィックのインポート 190
グラフィックのエクスポート 192
グラフ出力
デバイスドライバ 501
クリップボード
OLE オブジェクトの貼り付け 236
WPASTE コマンドを使用した貼り付け 368
ウィンドウのコンテンツのコピー 351, 358, 359
グラフィックの貼り付け 190, 193
コードをサブミットする 26, 56, 347
ステートメントをサブミットする 161
テキストの選択とコピー 55

ビットマップを SAS セッションに貼り付ける 56
非テキストウィンドウでの選択とコピー 56
警告
大文字への変換 535
結果タブ 60
結果ビューアウィンドウ 181
欠損値
DDE 269, 276
現在のフォルダ 11, 41
初期パスの設定 66
ステートメントを使用して変更する 42
対話型の選択 41
変更 41
検索ダイアログボックス 334
検索と置換
拡張エディタ 88
プログラムエディタ 111
検索パス
実行可能ファイル 543
コーディングエラー 90
コード, サブミットする 25
SAS NOTEPAD テキストエディタ 25
拡張エディタ 25
クリップボード 26, 56, 347
登録されたファイルタイプ 28
ドラッグアンドドロップ 27
バッチモード 26
プログラムエディタ 25
コードセクションの折りたたみ 99
コードをサブミットする
*参照項目*: コード, サブミットする
構成ファイル 8, 15
カスタム 16
検索順序 17
システムオプション 475
システムオプションの指定 15
処理オプション 20
処理順序 17
代替 17, 501
デフォルト 15
編集 16
命名規則 16
用途 15
互換性 4, 136
固定小数点表現のデータ
書き込み 380
読み取り 419, 421
固定小数点表現のバイナリ値
16 進値の変換 417
このウィンドウの使い方 74
コマンド
AutoComplete 機能 43
Windows 環境 326
Windows 版ではサポート対象外 327
印刷 177
ウィンドウ環境のカスタマイズ 62
外部ファイルにアクセスする 160
拡張エディタウィンドウ 84
コマンド行から発行する 44
コマンドバーから発行する 43
ツールバーから発行する 42
発行 42
メニューから発行する 42
コマンドウィンドウ
表示場所 500
コマンド行 5
切り替え 350
コマンドの発行 44
コマンドバー 37
オプションの設定 330
コマンドの発行 43
ヘルプ 73
コマンドプロンプト
SAS の起動 9
コマンドメソッド
メソッド 263
コンボボックス
アイテムの検索 256
アイテムの追加 256


## さ


サーバーログイン 490
最近使用したファイルのリスト 59, 366
サウンドの生成 390
サブクラス化 255
時間の表示 571
シグネチャ行 99
システムオプション 473
OPTIONS ステートメント 477
SAS コマンドに追加する 8
ウィンドウ環境のカスタマイズ 64
起動ファイル 475
構成ファイル 475
構成ファイルでの指定 15
制限 474
設定の表示 474
設定の変更 475
ターミナルに書き出す 586
電子メールの送信 44
パス名の挿入 523
パス名の追加 489
複数の場所に設定 477
メイン SAS ウィンドウの制御 314
ライブラリの連結 476
リモートブラウジング 77
ログに書き出す 586
システム管理者
ファイアウォールとリモートブラウジング 76

システムパフォーマンスの統計量 574
実行可能ファイル
  検索パス 543
実行ダイアログボックス
  SAS の起動 9
  開く 341
自動インデント 91
自動的に保存する 357
自動保存 60, 114, 357
自動保存ファイル 83
自動マクロ変数 599
自動呼び出しライブラリ 602
  SASAUTOS システムオプション 602
  指定 556
集計構文 151
終了ダイアログボックス 334
終了のエラーメッセージ 623
終了プリファレンス 59
出力
  プレビュー 175
  ページサイズ 542
  文字のフォーマッティング 512
  リモートブラウザで表示する 75
出力形式 377
  バイナリデータの書き込み 377
出力のプレビュー 175
ショートカット
  SAS の起動 8
ショートカットキー
  印刷プレビューウィンドウ 635
  拡張エディタ 95, 631
  キーボードマクロ 96
  行のブックマーキング 91
  削除 106
  テキストの選択と編集 86, 87
  メイン SAS ウィンドウ 629
  リセット 107
  割り当て 106
照合順序
  作成 438
詳細プリファレンス 61
省略形 91
初期化のエラーメッセージ 623
新規ライブラリダイアログボックス 126
数値変数
  長さと精度 595
スクリーンチップ 38
  切り替え 370
スクロール
  拡張エディタ 84
  ドラッグ時のスクロール 113
  プリファレンス 62
スクロールバー
  オン/オフの切り替え 364, 372
  カーソルの点滅 560
  フォーカスを無効にする 62
  横方向 364
スタートメニュー 8
ステータスウィンドウ 10
ステータス行 37, 38, 60
  オン/オフの切り替え 371
  領域の割合 371
ステートメント 443
  外部ファイルにアクセスする 157
  クリップボードからサブミットする 161
  実行命令 522
スプラッシュ画面 66
  起動時に表示する 570
  ビットマップ位置 571
制限オプション 474
正のバイナリ整数データ
  書き込み 383
  読み取り 421
セクション 508 486
セッションプリファレンス 58
ソースファイル
  バッチモード 577
ソートアルゴリズム
  SyncSort 439
ソートユーティリティ 568
  ホストソートユーティリティの名前 567
ゾーン 10 進データ
  書き込み 386
  読み取り 423
挿入位置
  拡張エディタ 85
  プログラムエディタ 109
挿入モード
  オン/オフの切り替え 365
速度 198

## た

ダイアログボックス
  ヘルプ 74
対称型マルチプロセッシング(SMP) 199
タイトルバー 495
ダイナミックリンクライブラリ(DLL)
  *参照項目*: DLL
対話型処理 209
対話型セッション 5
  ファイルのドラッグ 27
タスクマネージャウィンドウ 357
タブ
  拡張エディタ 91
  プログラムエディタ 110
タブ文字
  DDE 273
ダミー変数
  長さと精度 596
置換ダイアログボックス 341
ツールチップ

オン/オフの切り替え 355
ツールのカスタマイズダイアログボックス 68
開く 353
ユーザー補助機能 486
ツールバー 37, 38
カスタマイズ 67
カスタムコントロール 318
コマンドの発行 42
全般的なプリファレンスの設定 68
ツールの削除 71
ツールの作成例 72
ツールの追加 70
デフォルトに戻す 72
閉じる 353
保存 71
ボタンサイズ 353
ツールボックス
ツール切り替え 355
閉じる 353
表示場所 581
ボタンサイズ 353
ロード 354
ツールメニュー
アプリケーションの追加 66, 552
データ
他のアプリケーションからアクセスする 145
データセット 120
9.4 形式への変換 136
Scatter-read/Gather-write 323
以前のリリースからの読み込み/書き込み 4
サイズの計算 207
作成, リリース 6.08 から 6.12 まで 140
バッファ数 496
バッファサイズ 497
パフォーマンス 207
リリース 6.03 と 6.04 138
リリース 6.08 から 6.12 までの変換 138
リリース 6.08 から 8.2 まで 137
データセットオプション 323
データセットの変換
リリース 6.08 から 6.12 まで 138
データディレクティブ
電子メール 50
データの追加
外部ファイル 163
データビュー 121
データファイル 121
インターフェイス 121
サイズの上限を超える 135
ネイティブ 121
データベースファイル
SAS/ACCESS を使用してアクセスする 144
データライブラリ 125
LIBNAME ステートメントを使用して割り当てる 126
OLE を使用して割り当てる 262
Sasuser 133
Work 23, 133
新規ライブラリダイアログボックス 126
属性のリスト表示 462
デフォルトの永久ライブラリ 585
マルチエンジンアーキテクチャ 122
ライブラリ参照名の指定 125
連結 132
テーブルメンバタイプ 120
ディスクコントローラ 198
ディスクスペース 198
ディスクリプタファイル 122
ディレクトリ
ID 値 395
空の場合は削除する 397
情報 394
情報アイテムの数 396
情報アイテムの名前 395
存在 398
ディレクトリの存在 398
ディレクトリリスト 392
テキスト
マーク解除 367
テキストエディタ
コードをサブミットする 25
テキストエディタウィンドウ 107
上書きモード(デフォルト) 60
自動保存の間隔 60
テキストの削除 111
プリファレンス 60
テキストのコピー 55
テキストの選択
拡張エディタ 86
プログラムエディタ 110
テキストの選択とコピー 55
テキストのマーク解除 60, 367
デバイスドライバ 501
デフォルトのエンジン 124
電子メール
SAS を使用した送信 44
ウィンドウのコンテンツの送信 47
ウィンドウのコンテンツを添付する 356
現在のウィンドウを添付ファイルとしてメールで送信する 59
サポートされるインターフェイス 44
初期化 44
送信に使用する DATA ステップ 48, 51
送信に使用する FILENAME ステートメント 48
送信に使用する SCL 48, 52

送信に使用する SMTP 52
送信に使用する電子メールソフトウェア 45
データディレクティブ 50
添付の形式 362
メールの送信ダイアログボックス 46, 344
電子メールインターフェイス 504
電子メールダイアログボックス 503
添付の形式 362
統計量
システムパフォーマンス 574
動的データ交換
*参照項目:* DDE (動的データ交換)
登録されたファイルタイプ 28
ドキュメント 73
インデックスファイル 515
ヘルプファイルの場所 516
目次ファイル 519
特殊文字の属性 56
ドッキングビュー 39, 60
ウィンドウのドッキングとドッキング解除 39
最小化および復元 39
サイズ変更 39
有効化/無効化 40
ドッキングビューウィンドウ
オン/オフの切り替え 359
最小化 360
タスクバーから復元する 361
分割バーのサイズ変更モード 361
ドメインサフィックス
認証プロバイダを割り当てる 490
ドメイン認証サーバー
安全なサーバーログインの検索 490
ドラッグアンドドロップ
拡張エディタウィンドウ 88
コードをサブミットする 27
デフォルトアクションより優先する 113
デフォルトでない操作 238
プログラムエディタ 112
ドラッグアンドドロップ(デフォルト以外) 113
ドラッグ時のスクロール 113
ドラッグ修飾子 237
取り消し 372

## な

内部エラー 622
長いファイル名 154
名前付きパイプ 282
1 クライアント, 1 サーバー 284
1 サーバー, 複数クライアント 286
CALL RECONNECT ルーチン 284
NOBLOCK オプション 287
SCL 284
構文 282
次のクライアントに接続する 284, 289
データを待つ 287
定義 279
例 284
名前の付いていないパイプ 280
構文 280
定義 279
リダイレクトの順序 281
例 281
名前を付けて保存ダイアログボックス
開く 342
ファイルフィルタ 508
フォルダ 558
並べ替え
SyncSort 439
入力形式 415
9.1 への変換 416
バイナリデータの読み取り 416
認証プロバイダ
ドメインサフィックスを割り当てる 490
ネイティブデータファイル 121
ネイティブライブラリエンジン 123
ネットワーク
エラーメッセージ 621
ネットワークのパフォーマンス 204

## は

バージョン 6 のカタログ 138
バージョン 6 のファイル 136
バージョン 7 と 8 のカタログ 138
バージョン 7 と 8 のファイル 136
ハードウェア 197
I/O に必要なディスクスペース 198
プロセッサの速度 198
メモリ 198
ハードディスクドライブ 198
構成 198
マッピング 163
配色 105
バイナリ実数データ
16 進数への変換 378
書き込み 384
読み取り 422
バイナリ整数データ
書き込み 380
読み取り 419
バイナリデータ
書き込み 377
読み取り 416
パイプ 279
名前が付いていない 280
名前付き 282
ファイル属性 402

パス名
システムオプション値に挿入する 523
システムオプションに追加する 489
バスの速度 198
パック 10 進データ
書き込み 382
読み取り 420
バッチジョブ
ウィンドウプロシジャ 10
サブミット 10
ステータスウィンドウ 10
取り消し 10
バッチモード 5, 26
印刷 179
ソースファイル 577
プロシジャ出力ファイル 547
ログファイル 526
バッファ
データセットの処理に使用するバッファ数 496
ページサイズ 497
パフォーマンス
DLL 298
SAS の機能 203
SORT プロシジャ 205, 219
対話型処理 209
調整メソッド 205
データセットのサイズ, 計算 207
ネットワーク 204
ハードウェア 197
反応性 199
複数プロセッサに対応した I/O 拡張 199
メモリベースのライブラリ 200
パフォーマンスオブジェクト 217
パフォーマンスカウンタ 217
パフォーマンスモニタ 217
PROC SQL クエリ 220
起動 217
構成する 219
反応性 199
ビットマップ
SAS セッションへの貼り付け 56
印刷 174
ビューエンジン 123
ビューメンバタイプ 120
表示プリファレンス 60
開くダイアログボックス
拡張エディタで開く 346
指定されたパス 66
デフォルトエディタで開く 336
ファイルフィルタ 508
フォルダ 558
ファイアウォール
リモートブラウジング 76
ファイル 118
SAS 9.4 でのリモートホストの SAS ファイル 141
SAS 9.4 のファイルを以前のリリースで使用する 140
移行 136
印刷 178
開始時にサブミットする 59
拡張エディタでの保存 82
拡張エディタで開く 81
共有 135
互換性 136
情報アイテムの数 403
タイプ 118
転送 144
ファイル拡張子 118
複数の SAS セッション 135
プログラムエディタでの保存 114
プログラムエディタで開く 108
リモートホスト 141
ロックされたファイル 507
ファイルアクセス
エラーメッセージ 618
ファイル拡張子 118
変更 119
ファイルキャッシュ
メモリベースのライブラリ 202, 531
ファイル形式
リリース間の互換性 136
ファイル参照名 149
FILENAME ステートメント 149
FILENAME ステートメントを使用した割り当て 398
SET コマンド 151
SET システムオプション 151
環境変数 150
クリア 155
検証 399
ディレクトリへの割り当て 151
ファイルショートカットの割り当て 149
リスト 155
連結ディレクトリへの割り当て 153
連結ファイルへの割り当て 154
割り当ての取り消し 398
ファイルタイプ
ファイル拡張子の割り当て 101
ファイル名
クリア 54
ファイルの回帰
SAS 9.4 のファイルを以前のリリースで使用する 140
ファイルの拡張子
ファイルタイプへの割り当て 101
ファイルのショートカット 149
ファイルフィルタ 508
ファイルメニュー
最近使用したファイルのリスト 366

ユニバーサル印刷のコマンド 584
ファイルリソース
リスト 554
ファイルリソーストラッキングシステム 555
ファンクションキー
数 538
マッピング 544
ファンクションキーのマッピング 544
フォルダ
HTML 出力ファイル 61
Sasuser ライブラリ参照名 133
WORK フォルダ 61
現在のフォルダ 41
現在のフォルダの設定 11
作業フォルダへのライブラリ参照名の割り当て 127
デフォルトの構造 24
ライブラリ参照名の割り当て 127
フォルダの変更ダイアログボックス 333
フォント 57
TrueType 625
Windows フォントを SAS フォントに割り当てる 510
印刷 579
ウィンドウ 509
ウィンドウのコンテンツ 576
ディレクトリの場所 511
変更 169
フォントダイアログボックス 335
複数プロセッサに対応した I/O 拡張 199
浮動小数点表現のデータ
16 進数への変換 378
書き込み 384
読み取り 422
浮動小数点表現のバイナリ値
16 進値の変換 417
ブラウザ
開始ページ 61
起動 358
選択 61
フラットファイル 203
プリファレンス
HTML 出力 180
Web 61
結果タブ 60
終了時に保存する 362
詳細 61
セッションプリファレンス 58
全般 59
表示を選択します。 60
編集 60
プリファレンスダイアログボックス 58, 338
プリントマネージャ 521
プログラム, サブミット 352
プログラムアイテム
SAS の起動 8
プログラムエディタ 107
RTF テキスト 114
SAS 起動時に開始する 10
改行 110
拡張エディタに切り替える 107
機能 115
行番号の使用 109
検索と置換 111
コードをサブミットする 25
自動保存 114
挿入位置 109
タブ 110
テキストの選択 110
ドラッグ/ドロップアクションより優先する 113
ドラッグアンドドロップ 112
ドラッグ時のスクロール 113
ファイルの保存 114
ファイルを開く 108
プログラムエディタウィンドウ
カタログエントリにコンテンツを保存する 351
プログラムのサブミット 352
拡張エディタ 92
プログラムメンバタイプ 122
プロシジャ 425
最大メモリ量 529
プロシジャ出力
PRINTTO プロシジャを使用した送信 183
Web ブラウザに表示する 180
キャリッジコントロール文字 184
システムオプションを使用した送信 184
代替ファイル 488
名前を付けて保存ダイアログボックスを使用して送る 183
バッチモード 547
ファイルに送る 183
プロセスの終了 32, 357
プロセッサ
複数に対応した I/O 拡張 199
プロセッサの速度 198
ページサイズ 173, 542
ページ設定 170
ページ設定ダイアログボックス 338
ページ番号 174
リセット 541
ヘルプ
Microsoft HTML ヘルプ 73
SAS プロダクト 74
Web サイト 75
インデックスファイル 515
オンライン 73

コマンドバー 73
ダイアログボックス 74
ヘルプメニュー 74
ヘルプメニューに追加する 65
メイン SAS ウィンドウ 38
目次ファイル 519
リモートブラウザで表示する 75
ヘルプファイル
Microsoft HTML ヘルプの場所 516
登録 517
メイン SAS ウィンドウのヘルプメニューに表示する 517
ヘルプメニュー 74
ヘルプの追加 65
変換ダイアログボックス 334
編集プリファレンス 60
変数
SET Windows コマンドを使用した定義 130
SET システムオプションを使用した定義 129
Work データライブラリ 134
構造引数としてグループ化する 299
長さと精度 595
ライブラリ参照名の割り当て 129
変数の精度 595
変数の長さ 595
ポート番号
Remote Browser Server 77
ポインタ
返す 308
構造に渡す 300
ホストコマンド
SAS セッションからサブミットする 374
SAS セッションから実行する 467
サブミット 391
条件付きで実行する 392
ホストコンピュータ
名前の指定 77, 515
ホストソートユーティリティ
名前 567
保存ダイアログボックス
指定されたパス 66
ポップアップメニュー
オン/オフの切り替え 365
表示 369


## ま


マウス
IntelliMouse 55
マウスボタン 539
マクロ, キーボード 96
マクロ関数 602
マクロ機能 599
自動マクロ変数 599
自動呼び出しライブラリ 602
マクロ関数 602
マクロステートメント 601
マクロステートメント 601
マクロ変数
メモリ内変数のサイズ 537
マクロ変数シンボルテーブル 536
マップデータセット
ライブラリの名前 528
マルチエンジンアーキテクチャ 122, 123
マルチメディア機器
MCI 文字列コマンドをサブミットする 406
アクティブになるまで待機する 405
メールの送信ダイアログボックス 46, 344
命名規則
構成ファイル 16
メイン SAS ウィンドウ 36
位置 493
ウィンドウ環境コマンド 315
ウィンドウの制御 557
ウィンドウバー 40
ウィンドウを外部ファイルに保存する 53
カスタマイズ 492
画面のヒント 38
コンポーネント 36
最小化 328, 521
サイズと配置 65
最大化 328
ショートカットキー 629
ステータス行 38
寸法 493
制御 314
制御のためのシステムオプション 314
タイトル 495
ツールメニューにアプリケーションを追加する 552
ドッキングビュー 39
表示画面と機能 313
ヘルプ 38
ヘルプファイル 517
メニュー 41
メニュー項目を組み込む 495
メニューバー 494
元の状態に戻す 329
メッセージ
Windows イベントビューアへのログメッセージの書き込み 185
メニュー 41
コマンドの発行 42
システム/コントロールメニュー 557
メニュー項目の説明を表示する 38
メニュー項目を組み込む 495
メニューバー 37
表示 494

メモ
  大文字への変換 535
メモリ
  SAS がロードする実行プログラム 526
  SORT プロシジャの上限 569
  仮想メモリ 551
  使用状況の統計量 513
  総量の上限 533
  パフォーマンス 198
  プロシジャでの最大量 529
  マクロ変数シンボルテーブル 536
  メモリ内マクロ変数 537
  メモリベースのライブラリのブロックサイズ 529
メモリ内マクロ変数
  サイズ 537
メモリベースのライブラリ 200
  SAS ライブラリの処理 201
  Work ライブラリ 532
  ファイルキャッシュ 202, 531
  メモリの最大量 532
メンバ 118
メンバタイプ 118
  アクセス 122
  カタログ 121
  テーブル 120
  ビュー 120
  プログラム 122
目次ファイル 519
文字値
  16 進数への変換 379
文字式
  特定の文字を変換する 411
文字データ
  16 進データの変換 418
文字のフォーマッティング
  デフォルト 512
文字変数
  長さと精度 596
文字列
  プログラムに渡す 577

や
ユーザー定義アイコン 67, 585
ユーザー定義のキーワード 100
ユーザー補助機能 486
ユーザーライブラリ参照名 134
ユニバーサル印刷 165, 166
  コマンドを使用可能にする 584
  使用可能にする 583
用紙の種類 543

ら
ライブラリ
  システムオプションで連結する 476
  デフォルトのアクセス方式 505
  メモリベース 200, 201
ライブラリエンジン 123
ライブラリ参照名 122, 125
  関連付ける 462
  起動時に使用可能にする 128
  クリア 131, 404
  ユーザー 134
  リスト 130
ライブラリ参照名, 割り当て
  1 つのディレクトリへ 127
  GUI 126
  LIBNAME 関数 404
  作業フォルダへ 127
  複数のフォルダへ 127
  複数を 1 つのディレクトリへ 128
  変数 129
ライブラリ参照名のクリア 131
ライブラリダイアログボックス 335
リソースディレクトリ 553
リソース不足 499
リターンコード 617
リッチテキスト形式(RTF)
  ウィンドウのコンテンツを保存する 369
リッチテキストフォーマット(RTF) 114
リモートブラウジング 75
  ODS 出力 76
  コンピュータ名 515
  システムオプション 77
  出力とヘルプの表示 75
  ファイアウォール 76
リリース 6.03 と 6.04 のデータセット 138
リリース 6.08 から 8.2 までのデータセット 137
リリース 6.08 のカタログ 139
リンクされた OLE オブジェクト 240
  自動更新する 241
  リンクダイアログボックスを使用して更新する 240
リンクダイアログボックス
  開く 336
  リンクされた OLE オブジェクトを更新する 240
レジストリファイル 23
列の設定ダイアログボックス 333
列の並べ替えダイアログボックス 333
連結データライブラリ 132
  同じ名前 133
  更新アクセス権 132
  出力アクセス権 132
  入力アクセス権 132
連結ディレクトリ
  アクセス 156
  ファイル参照名の割り当て 153
連結ファイル

アクセス 156
ファイル参照名の割り当て 154
ログ
PRINTTO プロシジャを使用した送信 183
Windows イベントビューアへのメッセージの書き込み 185
システムオプションを使用した送信 184
代替ログ 487
名前を付けて保存ダイアログボックスを使用して送る 183
バッチモード 526
ファイルに送る 183
メッセージの表示 502
メッセージ用のニュースファイル 538
ログイベント
Windows イベントビューアへの書き込み 185
ログウィンドウ
カタログエントリにコンテンツを保存する 351
行サイズ 525
スクロール頻度 327
ログメッセージの書き込み
Windows イベントビューア 185
ロゴ
SAS/AF アプリケーション 319
ロゴ画面 66
起動時に表示する 570
ビットマップ位置 571
ロックされたファイル 507

## わ

ワードチップ 92
ワイルドカード 150