



THE
POWER
TO KNOW.

SAS[®] Windows版ガイド 9.3

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2011. *SAS®Windows 版ガイド* 9.3. Cary, NC: SAS Institute Inc.

SAS®Windows 版ガイド 9.3

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

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

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

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

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

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

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

Electronic book 2, 2012 年 8 月

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

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

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

目次

このドキュメントについて.....	vii
Windows 版 SAS 9.3 の新機能.....	xi
Windows 版 SAS のユーザー補助機能.....	xiii
推奨資料.....	xxi

1 部 Windows 版 SAS の実行 1

1 章・入門ガイド: Windows 版 SAS	3
SAS: Windows の機能を活用する.....	4
SAS の起動.....	6
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 SAS の操作	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.....	59
オンラインヘルプとドキュメントの利用.....	75
3 章・Windows 環境での SAS エディタの使用	81
拡張エディタの使用.....	81
プログラムエディタの使用.....	110
4 章・Windows 版での SAS ファイルの使用	119
SAS ファイルについて.....	120
マルチエンジンアーキテクチャ.....	124
データライブラリの使用.....	127
複数の SAS セッションから SAS ファイルにアクセスする.....	137
Windows 版 SAS 9.3 で他のバージョンのファイルを使用する.....	138
SAS 9.3 のファイルを以前のリリースで使用する.....	142
SAS 9.3 でリモートホストの SAS ファイルを使用する.....	142
BMDP ファイル、OSIRIS ファイル、SPSS ファイルの読み取り.....	143
SAS ファイルを動作環境間で転送する.....	146
SAS/ACCESS を使用し、データベースファイルにアクセスする.....	146
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.....	147
5 章・Windows 版での外部ファイルの使用	149
外部ファイルについて.....	149
外部ファイルの参照.....	150
SAS ステートメントを使用し、外部ファイルにアクセス.....	161
SAS コマンドを使用し、外部ファイルにアクセスする.....	163
高度な外部 I/O の操作法.....	165

通信ポートからデータを読み込む	168
6 章・Windows 版での SAS 出力の管理	171
印刷	171
プロシジャ出力の Web ブラウザへの表示	186
プロシジャ出力と SAS ログをファイルに送る	188
SAS ログ機能を使用したログメッセージの書き込み	190
グラフィックの作成	191
7 章・Windows 版でのパフォーマンスに関する注意点	203
ハードウェアに関する注意点	203
パフォーマンスを最適化する Windows の機能	206
パフォーマンスを最適化する SAS の機能	213
ネットワークのパフォーマンスに関する注意点	214
高度なパフォーマンス調整メソッド	214
2 部 SAS を他の Windows アプリケーションとともに使用する	219
8 章・Windows 版での Lotus Notes を使用した SAS データの配布	221
SAS での Lotus Notes の使用について	221
DATA ステップと SCL コードを使用し、Lotus Notes データベースに入力する	222
9 章・Windows 版 SAS で Windows システムツールの使用	229
SAS での Windows システムツールの使用について	229
イベントビューアアプリケーションログ	230
パフォーマンスツール	234
SAS を Windows のサービスとして起動する	240
10 章・Windows 版 SAS/AF ソフトウェアでの OLE の使用	251
OLE について	252
SAS/AF カタログの互換性	252
OLE オブジェクトを FRAME エントリに挿入する	252
FRAME エントリ内での OLE オブジェクトの編集	256
OLE Verb の起動	257
リンクされた OLE オブジェクトを使用する	258
OLE オブジェクトの変換	259
OLE オブジェクトとアプリケーションの自動化	261
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	268
11 章・Windows 版で OLE を使用し、他のアプリケーションから SAS を制御する	277
SAS の自動化について	277
SAS のインスタンスの作成	278
SAS セッションからのフィードバックの取得	278
OLE を用いた SAS の自動化の例	279
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	280
ディクショナリ	281
12 章・Windows 版で動的データ交換を使用する	285
動的データ交換(DDE)の概要	285
SAS での DDE 構文	286
DDE 外部ファイルの参照	287
DDE の例	288

13 章・Windows 版で名前付きパイプと名前の付いていないパイプを使用する	297
パイプの概要	297
名前の付いていないパイプの使用	298
名前付きパイプの使用	300
14 章・Windows 版 SAS から外部 DLL にアクセスする	309
SAS のダイナミックリンクライブラリ(DLL)の概要	309
SASCBTBL 属性テーブル	310
外部 DLL を使用する場合に特に注意すべき点	315
例	325
15 章・Windows 版 SAS/AF のプログラミングで特に注意すべき点	331
SAS の表示画面と機能の制御	331
メイン SAS ウィンドウの制御	332
SAS から外部 DLL にアクセスする	336
カスタムツールバーコントロールの設計、保存、ロード	336
SAS/AF アプリケーションの自動起動	337
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	337
3 部 Windows 版に固有の SAS 言語の機能 339	
16 章・Windows 版に固有のデータセットオプション	341
Windows 版に固有の SAS データセットオプション	341
ディクショナリ	341
17 章・Windows 版に固有の SAS コマンド	343
Windows 版に固有の SAS コマンド	344
ディクショナリ	345
18 章・Windows 版に固有の SAS 出力形式	395
Windows 版に固有の SAS 出力形式	395
バイナリデータの書き込み	395
以前のリリースのユーザー定義入力形式を SAS 9.3 で利用する	396
ディクショナリ	397
19 章・Windows 版に固有の SAS 関数と CALL ルーチン	407
Windows 版に固有の SAS 関数と CALL ルーチン	407
ディクショナリ	408
20 章・Windows 版に固有の SAS 入力形式	435
Windows 版に固有の SAS 入力形式	435
バイナリデータの読み取り	436
以前のリリースのユーザー定義入力形式を SAS 9.3 用に変換する	437
ディクショナリ	438
21 章・Windows 版に固有の SAS プロシジャ	445
Windows 版に固有の SAS プロシジャ	445
ディクショナリ	445
22 章・Windows 版に固有の SAS ステートメント	465
Windows 版に固有の SAS ステートメント	465
ディクショナリ	465
23 章・Windows 版に固有の SAS システムオプション	491

Windows 版に固有の SAS システムオプション	493
SAS システムオプション設定の表示	494
SAS システムオプション設定の変更	495
複数の場所に設定されたシステムオプションの処理	497
Windows のシステムオプション概要	497
ディクショナリ	515
24 章 • Windows 版での変数の長さ精度	621
Windows 版での変数の長さ精度	621
数値変数	621
文字変数	622
25 章 • Windows 版に固有の SAS マクロ機能	625
Windows 版に固有の SAS マクロ機能	625
自動マクロ変数	625
マクロステートメント	628
マクロ関数	628
自動呼び出しライブラリ	629
4 部 付録 631	
26 章 • Windows 版で OLE オブジェクトを自動化するための SCL メソッド	633
OLE クラスメソッドの要約	633
ディクショナリ	634
27 章 • Windows 版に固有の SAS エラーメッセージ	643
SAS エラーメッセージの概要	643
リターンコードと完了ステータス	643
ファイルへのアクセス	644
SAS 機能の使用	645
OLE の使用	646
ネットワークの使用	647
内部エラーの解決	648
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	649
初期化と終了のエラーメッセージ	649
28 章 • Windows 版でのグラフィックに関する注意点	651
SAS/GRAPH での TrueType フォントの使用	651
29 章 • Windows 版での対話型 SAS セッションのデフォルトキー設定	653
Windows 版でのデフォルトキー定義	653
メイン SAS ウィンドウでのショートカットキー	655
拡張エディタでのショートカットキー	657
印刷プレビュー内でのショートカットキー	661
30 章 • SAS Disk Cleanup Handler	663
The SAS Disk Cleanup Handler Utility	663
Disk Cleanup Handler の使用	664
31 章 • Java Runtime Environment エラーのトラブルシューティング	667
Java Runtime Environment エラーのトラブルシューティング	667
用語集	671
キーワード	683

このドキュメントについて

SAS 言語のための構文規則

SAS 言語のための構文規則概要

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

構文要素

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

キーワード

プログラムの作成ときに使用する 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。

次の SAS 構文の例では、構文の最初の単語がキーワードです。

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

次の例では、CALL ルーチンの最初の 2 つの単語がキーワードです。

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

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

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

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

DUPLEX | NODUPLEX

引数

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

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

CHAR (*string*, *position*)

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

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

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

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

スタイル規則

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

大文字太字

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

```
ERROR<message>;
```

大文字

リテラルの引数を示します。

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

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

斜体

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

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

```
LINK label;
```

- 引数に割り当てられる非リテラル値。

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

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

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

特殊文字

SAS 言語要素の構文には、次の特殊文字も使用されます。

=

等号は、一部の言語要素(システムオプションなど)のリテラル値を示します。

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

MAPS = *location-of-maps*

<>

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

次の CAT 関数の例では、少なくとも項目が 1 つ必要です。

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

|

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

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

CMPMODEL = BOTH | CATALOG | XML

...

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

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

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

'value' or "value"

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

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

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

;

セミコロンは、ステートメントまたは CALL ルーチンの終わりを示します。

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

SAS ライブラリおよび外部ファイルへの参照

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

x このドキュメントについて

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

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

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

Windows 版 SAS 9.3 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 9.3 では実装されています。

- プログラム出力結果のデフォルトは HTML になりました。リスト形式の出力を表示するには、リストをプリファレンスダイアログボックスの結果タブで選択します。
- ユーザープロファイルパスは Server 2008 R2 を含みます。
- INT64 の戻り値のタイプは ROUTINE ステートメントに対して RETURNS=属性に含まれます。詳細については、[NT64 タイプ \(312 ページ\)](#)を参照してください。
- ハードドライブの推奨事項に、特定のパフォーマンス情報が含まれるようになりました。詳細については、[ハードドライブ \(205 ページ\)](#)を参照してください。
- バッチモードでの SAS の実行についての情報が追加されました。詳細については、[バッチモード \(6 ページ\)](#)を参照してください。
- SAS をインストールするのに SASHOME ディレクトリを選択できるようになりました。詳細については、[デフォルトの構成 \(15 ページ\)](#)を参照してください。
- SAS Service Configuration Utility を起動する手順が変更されました。詳細については、[ユーティリティセクション \(240 ページ\)](#)を参照してください。
- 付録として、JRE(Java Runtime Environment)のトラブルシューティングが追加されました。詳細については、[トラブルシューティング \(667 ページ\)](#)を参照してください。
- SAS 9.3 のメンテナンスリリース 2 では、付録の JRE に関するエラーのトラブルシューティングが修正されました。JRE 1.6.0_23 は JRE 1.6.0_24 に置き換えられました。オペレーティングシステムが 64bit の場合でも、32bit 版の JRE をインストールする必要があることが新しい記述では明記されています。
- Windows Server 2008 および Server 2008 R2 がサポート対象になりました。
- Windows 2008 EPIC (Itanium)はサポート対象外となりました。

システムオプション

以下のシステムオプションが新規に追加されました。

UNIVERSALPRINT (p. 609)

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。

次のシステムオプションが拡張されました。

MEMSIZE (p. 560)

新しい初期設定値は 2G です。

SORTSIZE (p. 596)

新しい初期設定値は 256M です。

STIMEFMT (p. 598)

FULLSTIMER および STIMER 出力にかかる時間を表示するのに使用する形式を指定します。

APPEND (p. 518)

APPEND は CMPLIB および AUTOEXEC をサポートするようになりました。

INSERT (p. 550)

INSERT は CMPLIB および AUTOEXEC をサポートするようになりました。

Windows 版 SAS のユーザー補助機能

Windows 版 SAS のユーザー補助機能について

Windows 版 SAS には、障害を持つ SAS ユーザーの利便性を向上するため、次のユーザー補助機能と互換性機能が用意されています。これらの機能は、米国のリハビリテーション法第 508 条(1973 年改正)で制定されている電子情報技術のアクセシビリティ基準を遵守しています。

- “ユーザー補助機能が備わったウィンドウおよびダイアログボックス” (xiii ページ) ユーザー補助機能が備わった SAS ウィンドウおよびダイアログボックス を表示します。
- “ACCESSIBILITY システムオプション” (xv ページ) 標準のユーザーインターフェースおよび完全なユーザー補助機能が備わったインターフェースの使用方法を説明します。
- SAS ウィンドウの可読性の拡張については、次を参照してください。“フォントの拡大” (xvii ページ) および “アイコンの拡大” (xviii ページ)
- キーボード、メニュー、コマンドを使用して、ドッキングビューや SAS エクスプローラの一覧ビューへのアクセスや変更が行えます。次を参照してください。
 - “メイン SAS ウィンドウのドッキングビューのサイズ変更” (xviii ページ)
 - “指定した列を基準にした、ウィンドウの一覧ビューの並べ替え” (xviii ページ)
 - “一覧ビューの個々の列のサイズ調整” (xix ページ)
- ユーザー補助機能でメニューを読むのが困難な場合は、次を参照してください。“メニューへのアクセスの改善” (xix ページ)

ユーザー補助機能が備わったウィンドウおよびダイアログボックス

米国リハビリテーション法第 508 条(1973 改訂)を遵守した SAS ウィンドウや SAS ダイアログボックス(メイン SAS ウィンドウを含む)については、次の表を参照してください。

ユーザー補助機能が備わった SAS ウィンドウおよびダイアログボックス

ウィンドウまたはダイアログボックス	関連するウィンドウまたはダイアログボックス
SAS システムについて	著作権情報 サイト情報
フォルダの変更	なし
ツールのカスタマイズ、タブのカスタマイズ	ビットマップブラウザ ツールを開く ツールの保存
クリップボードの DDE トリプレット	なし
拡張エディタ	検索 置換 キーボードマクロの実行 キーボードマクロ キーの割り当て キーボードマクロの作成 キーボードマクロの編集 キーボードマクロのインポート キーボードマクロのエクスポート 省略キー 拡張エディタオプション ユーザー定義のキーワード SAS 拡張子 拡張エディタキー キーの割り当て 注: Alt + [と Alt +] を打鍵してもマクロでは機能しません。
エクスプローラー	検索 ライブラリの作成 カタログの作成 エクスプローラオプション プロパティ カタログ
エクスポートウィザード	表示されるすべてのダイアログボックス
検索	なし
フォント	なし

ウィンドウまたはダイアログボックス	関連するウィンドウまたはダイアログボックス
インポートウィザード	表示されるすべてのダイアログボックス
ライブラリ	ライブラリの作成 ライブラリの変更
ログ	ログオプション
お気に入りフォルダ	なし
ノートパッド	なし
開く	なし
出力	出力オプション
プリファレンス	なし
印刷 (ウィンドウおよび[ユニバーサル印刷]ダイア ログボックス)	ページ設定 印刷の中止 印刷プレビュー 印刷設定 (ウィンドウおよび[ユニバーサル印刷]ダイア ログボックス)
プログラムエディタ	なし
置換	なし
結果	結果のプロパティ 名前の変更 オブジェクトとして保存 テンプレートと結果のダイアログボックス
結果ビューア - SAS 出力	なし
実行	なし
SAS システムオプション	なし
保存	なし
メールの送信	なし
ウィンドウ	なし

ACCESSIBILITY システムオプション

標準ユーザーインターフェイスまたはユーザー補助機能を完備したインターフェイスへのアクセス

ACCESSIBILITY システムオプションを使用すると、標準ユーザーインターフェイスまたはユーザー補助機能を完備したインターフェイスのどちらを使用するのかを指定することができます。標準ユーザーインターフェイスでは、前のセクションのリストにある大部分のウィンドウやダイアログボックスのコンポーネントを読み上げるユーザー補助機能が有効化されます。ユーザー補助機能を完備したインターフェイスでは、前のセクションで説明したすべてのウィンドウのコンポーネントを読み上げるユーザー補助機能が有効化されます。

ユーザー補助機能を完備したインターフェイスでは、キーボードからすべてのコマンドやタブ付きページを操作できるように、次のダイアログボックスにボタンが追加されます。

- [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
- SAS の[プロパティ]ダイアログボックスの一部

ACCESSIBILITY システムオプションの指定は、SAS 起動時に構成ファイルまたはコマンドプロンプトで行います。ACCESSIBILITY には、次の値を指定できます。

標準

推奨されるユーザーインターフェイスの使用を指定します。すべてのユーザー補助機能は有効化されません。これはデフォルトの設定です。

拡張

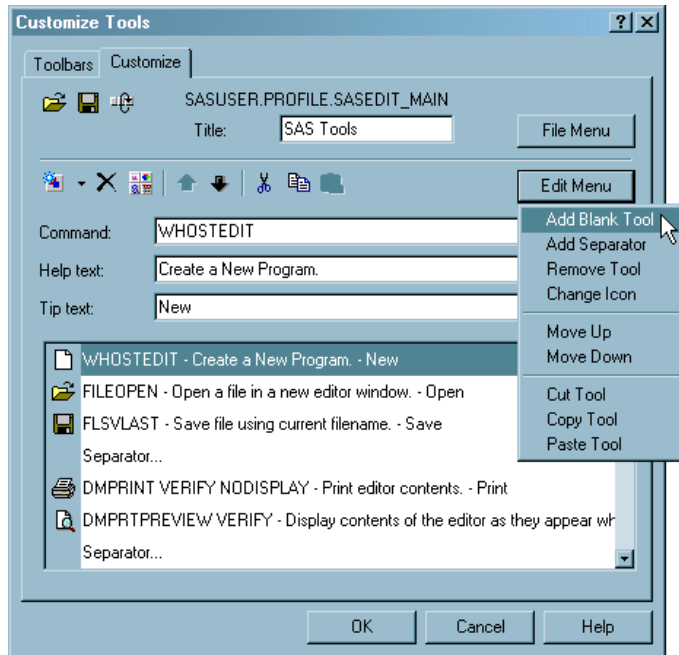
ユーザー補助機能を完備したインターフェイスの使用を指定します。

詳細については、次を参照してください。[“ACCESSIBILITY システムオプション: Windows” \(515 ページ\)](#)

ユーザー補助機能が備わった[ツールのユーザー設定]ダイアログボックスの使用

[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]タブには、[ツールバーを開く]や[ツールの追加]などのファイルコマンドや編集コマンドのコマンドボタンが提供されています。ACCESSIBILITY システムオプションをにして SAS を起動する場合は拡張された場合、SAS はを追加します。ファイルメニューボタンと[ユーザー設定]タブへのエディットメニューボタンこれらのボタンを使用すると、キーボードを使用して、[ファイル]メニューや[編集]メニューから利用できるコマンドを発行することができます。

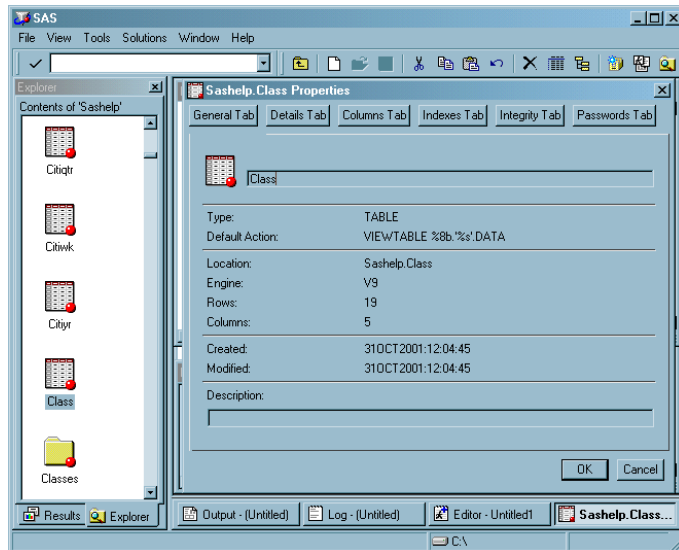
ユーザー補助機能が備わった[ツールのユーザー設定]ダイアログボックス



ユーザー補助機能が備わった[プロパティ]ダイアログボックスの使用

ACCESSIBILITY システムオプションをに設定する場合は拡張された場合、SAS プロパティダイアログボックスのタブページはボタンとして使用できます。[プロパティ]ダイアログボックス内を移動する場合、Ctrl キーと Page Up キーまたは Ctrl キーと Page Down キーを押します。

ユーザー補助機能が備わった[プロパティ]ダイアログボックス



フォントの拡大

テキストを読みやすくするには、[フォント]ダイアログボックスや SYSGUIFONT システムオプションを使用してフォントを拡大します。

フォントダイアログボックスを開くには、コマンドバーで次を入力します。dlgfont 代わりに、次を選択できます。ツール ⇒ オプション ⇒ フォントのフォントサイズを指定します。ボックスの大きさを変更してからをクリック OK

ワードのようなボタンテキストと記述テキストのフォントを拡大するには SAS エクスプローラーウィンドウ内の SAS 環境のコンテンツでは、SAS の起動時に設定ファイル内で、あるいは、コマンドプロンプトで SYSGUIFONT システムオプションを使います。次の SAS コマンドを実行すると、フォントサイズ 16 の Times New Roman フォントが使用されます。

```
sas -sysguifont "times new roman" 16
```

注: 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。

アイコンの拡大

アイコンを拡大するには、次の操作を実行します。

1. 選択ツール ⇒ ユーザー設定
2. においてツールバータブ、選択大きいアイコン

メイン SAS ウィンドウのドッキングビューのサイズ変更

キーボードを使ってドッキングビューのサイズ変更するには、Alt + W + S を押下するかまたは次を選択します。ウィンドウ ⇒ ドッキングビューのサイズを指定もしくは、を入力します。コマンドバーで wdockviewresize を入力します。

ドッキングビューのサイズを変更するには、キーボードの次のキーを使用します。

- 右向き矢印(->)キーまたは左向き矢印(<-)キーを押すと、分割バーが右または左に若干移動します。
- Ctrl キーと右向き矢印(->)キーまたは Ctrl キーと左向き矢印(<-)キーを押すと、分割バーが右または左に大幅に移動します。
- Home キーを押すと、分割バーが左端に移動します。
- End キーを押すと、分割バーが右端に移動します。
- Enter キーを押すと、ドッキングビューのサイズが確定されます。
- Esc キーを押すと、ドッキングビューのすべてのサイズ変更がキャンセルされます。

詳細については、次を参照してください。“WDockViewResize コマンド: Windows” (378 ページ)

指定した列を基準にした、ウィンドウの一覧ビューの並べ替え

SAS エクスプローラのように、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から、一覧に関連付けられた詳細を基準に一覧を並べ替えることができます。

1. ウィンドウがまだリストではない場合、次を選択します。**ビュー** ⇨ **リスト**
2. 入力コマンドバーで `dlgcolumnsort` を入力します。
3. 並べ替えの基準とする列を選択します。
4. クリック**並べ替え**
5. クリック**閉じる**

詳細については、次を参照してください。“[DLGCOLUMNSORT コマンド: Windows](#)” (351 ページ)

一覧ビューの個々の列のサイズ調整

[列の設定]ダイアログボックスを使用して、一覧ビューの個々の列をサイズ調整できます。この操作を行うには、詳細を表示する必要があります。

1. ウィンドウがまだリストではない場合、次を選択します。**ビュー** ⇨ **リスト**
2. 全体がまだ表示されていない場合、次を選択します。**ビュー** ⇨ **詳細**
3. 入力コマンドバーで `dlgcolumnsize` を入力します。
4. サイズを調整する列を選択します。
5. Tab キーを押して、列の幅をピクセル単位で入力します。
6. 次のいずれかの操作を実行します。
 - クリック**ダイアログボックスを閉じずに変更を表示するには、適用**をクリックします。変更の終了後、Enter キーを押してダイアログボックスを閉じます。
 - クリック**変更を適用してダイアログボックスを閉じるには、OK**をクリックします。

SAS システムオプションウィンドウの列は、サイズを調整できません。

詳細については、次を参照してください。“[DLGCOLUMNSIZE コマンド: Windows](#)” (351 ページ)

メニューへのアクセスの改善

メニュー項目にカスタムアイコンを割り当てるアプリケーションでは、ユーザー補助機能によるメニューの可読性が妨げられる可能性があります。NOMENUICONS システムオプションを指定すると、SAS のすべてのウィンドウからメニューアイコンが取り除かれます。メニューアイコンを取り除いた場合、標準的な Windows メニュー構造が使用されます。

NOMENUICONS システムオプションは、構成ファイルに指定するか、SAS 起動時に指定するか、OPTIONS ステートメントを使用して指定します。

-MENUICONS | -NOMENUICONS

MENUICONS | NOMENUICONS

MENUICONS

メニューにアイコンを表示します。

NOMENUICONS

メニューにアイコンを表示しません。

例を次に示します。`sas -nomenuicons` はメニューアイコンを除外して SAS を起動します。OPTIONS ステートメントでは、次のように指定します。

```
options nomenuicons;
```


推奨資料

次に、参考文献を示します。

- 『SAS プロシジャ』
- 『SAS ファイルの移行とアクセス』
- SAS コンポーネント オブジェクト:リファレンス
- 『SAS/CONNECT 使用の手引き』
- 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 機能と CALL ルーチン:リファレンス
- SAS 言語リファレンス:概念
- 『SAS 各国語サポート(NLS): 使用の手引き』
- SAS 出力送出システム:使用の手引き
- 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
- SAS ユーティリティ:リファレンス

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

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

1 部

Windows 版 SAS の実行

1 章	
入門ガイド: Windows 版 SAS	3
2 章	
Windows 版 SAS の操作	35
3 章	
Windows 環境での SAS エディタの使用	81
4 章	
Windows 版での SAS ファイルの使用	119
5 章	
Windows 版での外部ファイルの使用	149
6 章	
Windows 版での SAS 出力の管理	171
7 章	
Windows 版でのパフォーマンスに関する注意点	203

1 章

入門ガイド: Windows 版 SAS

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	5
Java の起動	5
SAS の起動	6
SAS を対話モードまたはバッチモードで使用	6
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	9
SAS ファイルからの起動	10
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	24
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 テムコマンドを条件付きで発行	30

XWAIT 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。SAS 9.3 では、次の Windows 環境がサポートされます。

表 1.1 SAS 9.3 でサポートされる Windows エディション

チップアーキテクチャ	オペレーティングシステム	エディション
x86, x86-64 (x64)	Windows XP	Professional
	Windows Server 2003	Standard Edition
	Windows Vista	Business
	Windows Vista	Enterprise
	Windows Vista	Ultimate
	Windows 7	Professional, Ultimate
	Windows Server 2008	Standard, Enterprise, Datacenter
x86-64 (x64)	Windows Server 2003	Enterprise エディション
	Windows Server 2003	DataCenter Server
	Windows Server 2008 R2	Standard, Enterprise, Datacenter

注: その他のバージョンの Windows はサポートされていません。

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 が提供する最も強力なツールと技術を使用した情報共有機能として、次の機能がサポートされています。

- OLE
- DDE (Dynamic Data Exchange)
- ODBC (Open Database Connectivity)
- 電子メールシステム

- Lotus Notes
- 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。

しかし、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 9.3 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。

SAS9.3 では、Java に依存する SAS/STAT プロシジャおよび SAS/GRAPH プロシジャは独立した SAS 製品としてではなく、Base SAS の一部として供給されます。その結果として、SAS 9.3 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 9.3 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 ウィンドウ環境で生成されるデフォルトの HTML 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。

SGPLOT、SGPANNEL、SGSCATTER、SGRENDER のような Graph Template Language および、すべての SAS/GRAPH プロシジャは SAS/GRAPH プロシジャから Base SAS へと移動するため、Java は SAS 9.3 で Base SAS の一部として実装されます。また、ODS GRAPHICS はデフォルトで有効となり、SAS ウィンドウ環境のみへの統計グラフをレンダリングできる、ほとんどのプロシジャの画像をレンダリングします。ODS グラフでは、ODS GRAPHICS ステートメントを使用して、グラフィック環境を制御する必要があります。ODS GRAPHICS ON ステートメントを指定して、ODS グラフィックを有効にする必要はありません。詳細については、“[Java Runtime Environment エラーのトラブルシューティング](#)” (667 ページ)を参照してください。

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 95、98、NT、ME、2000 のオペレーティングシステム下での 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、エクスプローラーを開いて、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。.LOG および.LST ファイルは、デフォルトではプログラムと同じフォルダに置かれます。

.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。

エクスプローラー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 Batch Submit 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 エクスプローラー**を開き、**ビュー** ⇒ **オプション**と選択します。
2. それから、**ファイル形式** ⇒ **SAS システムプログラム**と選択します。
3. **編集**を選択します。
4. **バッチサブミット操作**を範囲指定し、**デフォルトに設定** ⇒ **OK**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**と選択します。
2. 次に示すものと似たコマンドを入力します。“c:\Program Files\SAS Institute\SAS\V8\Sas.exe” -sysin c:\test.sas プログラムの格納場所を指定します。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\SAS Institute\SAS\V8\Sas.exe" -sysin c:\test.sas`
3. .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\SAS Institute\SAS\V8\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\SAS Institute\SAS\V8\Sas.exe" -sysin c:\test1.sas
"c:\Program Files\SAS Institute\SAS\V8\Sas.exe" -sysin c:\test2.sas
"c:\Program Files\SAS Institute\SAS\V8\Sas.exe" -sysin c:\test3.sas
"c:\Program Files\SAS Institute\SAS\V8\Sas.exe" -sysin c:\test4.sas
"c:\Program Files\SAS Institute\SAS\V8\Sas.exe" -sysin c:\test5.sas
```

- この例では、Windows 2000 と Windows XP 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\SAS Institute\SAS\V8\Sas.exe"
-sysin c:\test1.sas
Start "Job2" "c:\Program Files\SAS Institute\SAS\V8\Sas.exe"
-sysin c:\test2.sas
Start "Job3" "c:\Program Files\SAS Institute\SAS\V8\Sas.exe"
-sysin c:\test3.sas
Start "Job4" "c:\Program Files\SAS Institute\SAS\V8\Sas.exe"
-sysin c:\test4.sas
Start "Job5" "c:\Program Files\SAS Institute\SAS\V8\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 2000 および Windows XP 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\SAS Institute\SAS\V8\Sas.exe"
-sysin c:\test1.sas
"C:\Program Files\SAS Institute\SAS\V8\Sas.exe"
-sysin c:\test2.sas
"C:\Program Files\SAS Institute\SAS\V8\Sas.exe" -sysin c:\test3.sas
"C:\Program Files\SAS Institute\SAS\V8\Sas.exe"
-sysin c:\test4.sas
"C:\Program Files\SAS Institute\SAS\V8\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せん。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せん。スケジュールタスクプログラムは Windows NT 4 ではマイコンピュータ直下にあります。Windows 2000、Windows 98、Windows ME 環境下では、コントロールパネル内にあります。Windows 95 にはスケジューラーはありません。
 1. スケジュール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呼び出します。
 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選択します。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]をクリックします。

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. すべてのプログラムを選択します。
3. SAS を選択します。
4. SAS 9.3(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。例を次に示します。

```
"c:\program files\SAS\SASFoundation\9.3\sas.exe" -config "c:\mydir\sasv9.cfg"
```

[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プロンプトウィンドウ)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できます。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を使用して対話型セッションを開始するには、次の操作を実行します。

1. **スタート** ⇨ **ファイル名を指定して実行**と選択します。
2. **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\SAS\SASFoundation\9.3` に SAS がインストールされる場合、`c:\program files\SAS\SASFoundation\9.3\sas.exe` および、指定したいオプションを入力します。

注: Vista および Windows 7 では、**プログラムとファイルの検索**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\SAS\SASFoundation\9.3\sas.exe -ls 80 -ps 60
-config c:\program files\SAS\SASFoundation\9.3\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\SAS\SASFoundation\9.3\sas.exe
-sysin c:\mysas\programs\progl.sas
-config c:\program files\SAS\SASFoundation\9.3\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。
- .sas または .sasv7bpgm のファイル拡張子を持つ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。
- ステータスウィンドウで**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の編集タブで、拡張エディタを無効化します。

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” (110 ページ)、“編集プリファレンス” (62 ページ) および “ENHANCEDEDITOR システムオプション: Windows” (534 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” (585 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。

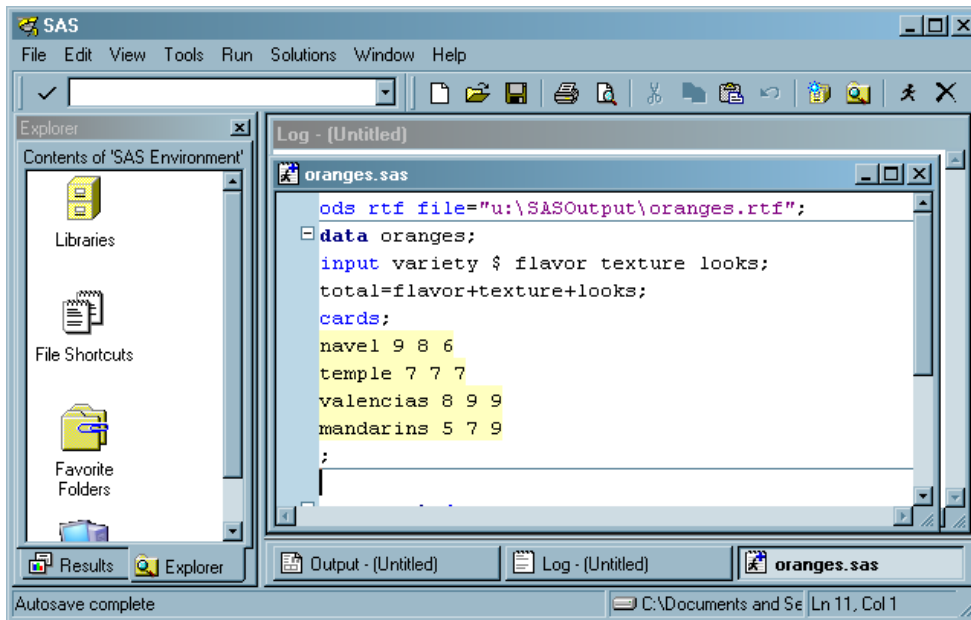
- スタートメニューからの SAS の起動
- サンプル SAS プログラムのサブミット
- プログラム出力の検証

- 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、**すべてのプログラム** ⇒ SAS ⇒ 9.3 と選択します。

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。

画面 1.1 サンプル SAS プログラムのサブミット

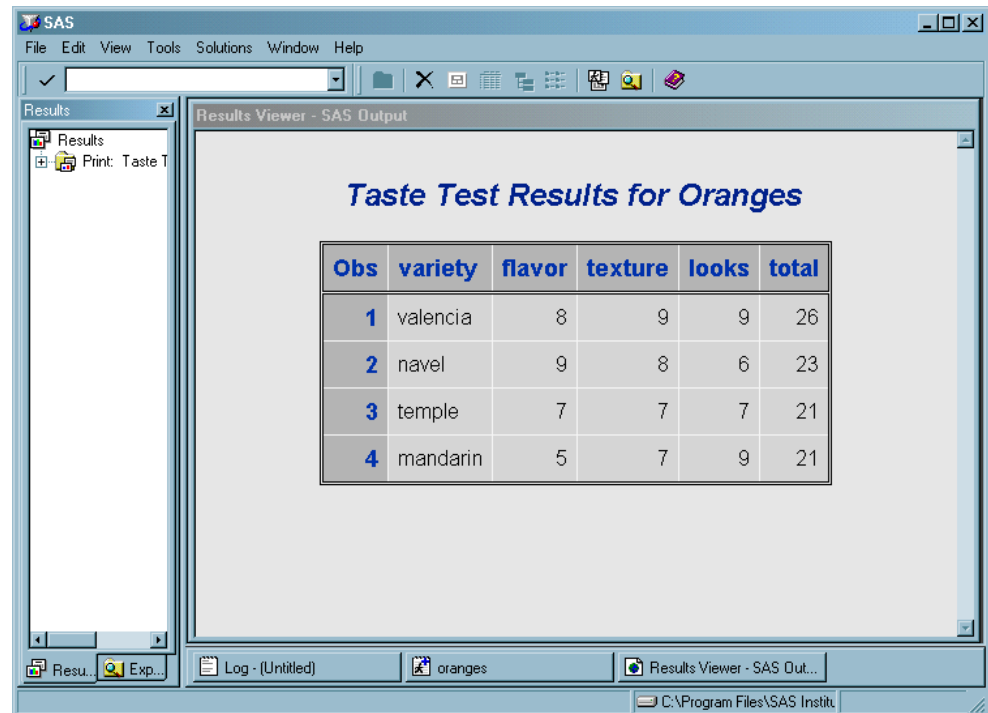


拡張エディタウィンドウに、次の SAS コードが表示されます。

```
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;
```

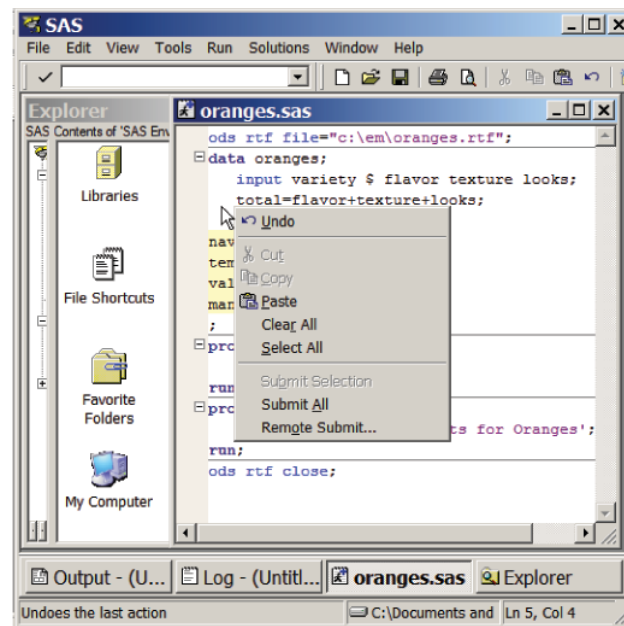
プログラムをサブミットすると、次のように**結果ビューア**ウィンドウに出力が表示されま

画面 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。

画面 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が無効化されている場合は、SAS セッションを終了するかどうかを確認せずにセッションが終了されます。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の設定](#)” (60 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 Vista、Windows 7、Windows Server 2008、Windows Server 2008 R2 環境下では、SAS コンソールログは通常 `c:\Users\ユーザーID\AppData` 内に置かれます。その他すべての Windows 動作環境では、SAS コンソールログは通常 `c:\Documents and Settings\ユーザーID\Application Data` 内に置かれます。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、**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。

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**とアクセスすることでプリンターを削除します。
2. プリンタの Web サイトから新しいプリンタドライバをダウンロードして、現在のドライバを別のドライバに置き換えます。
3. 次のパスで、`profile2.SAS7bcat` を `profile2.old` に、`profile.SAS7bcat` を `profile.old` に名前を変更します。`c:\Documents and Settings\ユーザーID\My Documents\My SAS Files\9.3\`
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、`SASVx.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。SAS 構成ファイルで使用するシステムオプションのリストについては、“[Windows のシステムオプション概要](#)” (497 ページ) を参照してください。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` フォルダに格納されていました。`!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 9.3 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/9.3` にあります。デフォルトのフォルダ `!SASHOME` は `c:\program files\SASHome` にあります。デフォルト構成の場所は `c:\program files\SASHome\SASFoundation\9.3\nls\en\sasv9.cfg` です。

`!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 がデフォルトのフォルダにインストールされ、デフォルトの言語が英語である場合、`!SASROOT` フォルダ内の `SASV9.CFG` ファイル

は
`-config "c:\program files\SASHOME\SASFoundation\9.3\nls\en\sasv9.cfg"`
 を含みます。

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。

`!SASHOME` フォルダについての詳細については、“[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](#)” (495 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を使用し、SAS 環境変数の SASROOT、SASEXT0、SASFOLDER、MYSASFILES、SASCFG、SASAUTOS、SAMPSIO、SAMPSRC、EISIMAGE、INSTALL が定義されます。Setup アプリケーションでは、この警告の下に SASUSER、WORK、HELPLC、DMSEXP、APPLETLOC、TEXTURELOC、RESOURCESLOC、JREOPTIONS、SASSCRIPT、SASHELP、MSG、PATH の各システムオプションが追加されます。

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 Vista、Windows 7、Windows Server 2008、Windows Server 2008 R2 では、Windows ユーザープロファイルフォルダへのパスは `c:\Users\ユーザーID\Documents\My SAS Files\9.3` です。その他すべての環境では、ユーザープロファイルフォルダは `c:\Documents and Settings\ユーザーID\My Documents\My SAS Files\9.3` 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、次のように指定します。

```
"c:\program files\SAS\SASFoundation\9.3\sas.exe -config c:\mysas\mysasconfig.CFG"
```

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” (529 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 .SASVx.CFG (x は SAS のバージョン番号) ファイルが検索されます。Windows Vista、Windows 7、Windows Server 2008、Windows Server 2008 R2 では、Windows ユーザープロファイルフォルダは `c:\Users\ユーザーID\Documents\` です。その他すべての環境では、ユーザープロファイルフォルダは `c:\Documents and Settings\ユーザーID\My 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 .SASVx.CFG (x は SAS のバージョン番号) ファイルが検

索されます。Windows Vista、Windows 7、Windows Server 2008、Windows Server 2008 R2 では、Windows ユーザープロファイルフォルダは `c:\Users\ユーザーID\Documents\My SAS Files\9.3` です。その他すべての環境では、ユーザープロファイルフォルダは `c:\Documents and Settings\ユーザーID\My Documents\My SAS Files\9.3` 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、次の環境変数はを含んでいる場合があります。-obs 2m -sasinitialfolder c:\myfolder -linesize max

SAS 起動コマンド行

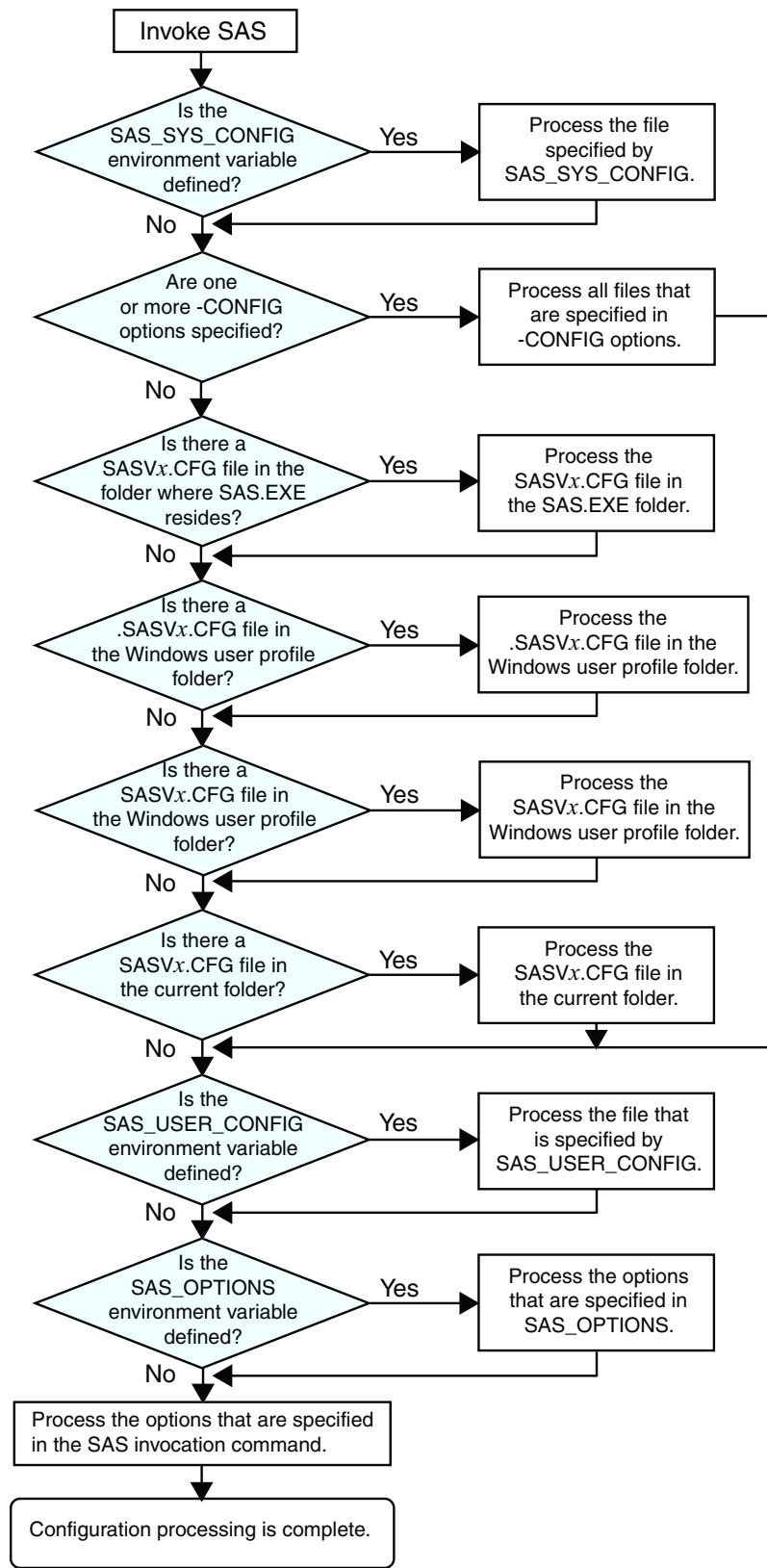
SAS 起動時に使用するコマンドに、追加システムオプションを指定することができます。このシステムオプションは、常に構成ファイルで設定されたオプション値より優先されます。

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\SASVx.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 1.1 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構成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 1.1 (19 ページ) では、SAS 構成ファイル処理の流れを説明しています。

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” (529 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” (520 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。他の名前にする場合は、AUTOEXEC システムオプションを使用して、SAS に自動実行ファイルの保存場所を指定する必要があります。たとえば、SAS Windows ショートカットのリンク先フィールドの SAS.EXE ファイルへのパス指定の後に、次のオプションを指定することができます。

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

指定した自動実行ファイルが見つからない場合、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\SAS\SASFoundation\9.3\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](#)” (493 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\SAS\SASFoundation\9.3\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。既存のプロファイルカタログにアクセスせずに SAS を起動した場合は、SAS ではデフォルトのキー定義とウィンドウ構成を含むプロファイルカタログが作成されます。

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

Windows Vista、Windows 7、 Windows Server 8	Windows XP および Windows Server 2003
<code>-sasuser "c:\Users\user-ID \Documents\My SAS Files \9.3"</code>	<code>-sasuser "c:\Documents and Settings\user-ID\My Documents\My SAS Files\9.3"</code>

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、デフォルト(\My SAS Files\9.3 という名前のフォルダ)の代わりにプロファイルカタログの場所を指定します。このオプションは、マシンを他のユーザーと共有する際に SAS セッションをカスタマイズする場合や、ネットワーク経由で SAS にアクセスする場合に使用します。

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: Windows” (586 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マシ、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](#)” (136 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。たとえば、次のように指定されます。`!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](#)” (135 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、重複しないように、`_TDnnnnnnnnnn` 形式を使用して作成されます。この TD は一時フォルダを意味し、`nnnnnnnnnn` は各 SAS セッションのプロセス ID 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と選択します。

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\9.3 になります。このドキュメントの例では、!SASHOME フォルダは c:\Program Files\SASHOME\SASFoundation\9.3 と呼ばれると仮定します。

x64 システム上の 32 ビットのコンテンツは c:\program files\SASHome \x86\SASFoundation\9.3 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 9.3 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 \SharedFilesI になります。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お問い合わせください。

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ダードな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” (370 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、**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**ウィンドウを最小化**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`、`.SASV7BPGM` 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 9.3 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**ウィンドウ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](#)” (365 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、ファイルをタスクバー上の SAS アイコンにドラッグし、そこでマウスボタンを押したままにしておくと、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](#)” (484 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せん。

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 です。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](#)” (409 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 routine](#) のセクションを参照してください。

XWAIT 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場合は、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 1.3 \(31 ページ\)](#) で要約されています。

表 1.3 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	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が、アプリケーションが終了するまでは SAS に戻れません。
XSYNC	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 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 と切り替えられません。
NOXWAIT 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 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) 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


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

 をサブミットします。*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。
- ステータスウィンドウで**キャンセル**を選択。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。
- SAS コマンド行から BYE コマンドまたは ENDSAS コマンドを発行。
- ENDSAS ステートメントをサブミット。
- タスクリストから SAS セッションを終了。セッションプロセス(プロセス名は SAS の起動方法によって異なります)を選択し、**タスクの終了**を選択。
- メイン SAS ウィンドウのメニューバーで、ファイルメニューから **SAS の終了**を選択。
- ファイルポップアップメニューから**終了**を選択。
- Alt と F4 を同時に押下。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 Vista と、それ以降の Windows 動作環境下では、Windows SAS コンソールログは通常は `c:\Users\ユーザーID\AppData` に置かれます。Vista 以前の Windows 動作環境下では、SAS コンソールログは通常 `c:\Documents and Settings\ユーザーID\Application Data` 内に置かれます。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サブミットします。Windows Vista と、それ以降の動作環境下では、**Windows ログ**カテゴリーを展開して、**アプリケーション**をクリックします。Vista 以前の Windows 動作環境下では、**アプリケーション**をクリックします。

2 章

Windows 版 SAS の操作

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	55
ウィンドウとファイル名のクリア	55
キーの定義	56
Microsoft IntelliMouse を利用した操作	56
クリップボードの使用	57
テキストの強調表示と特殊文字の作成	58
SAS セッションのカスタマイズ	59
フォントの選択	59
セッションプリファレンスの設定	60
コマンドを使用した SAS ウィンドウ環境のカスタマイズ	64
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	66
メイン SAS ウィンドウ	67
ツールバーのカスタマイズ	69
オンラインヘルプとドキュメントの利用	75
Microsoft HTML ヘルプの使用	75
コマンドバーからのヘルプの利用	75
ダイアログボックスのヘルプの利用	76
SAS プロダクトのヘルプの利用	76
ヘルプメニューからのヘルプの利用	76
SAS リモートブラウザでの出力とヘルプの表示	77

SAS インターフェイスの概要

SAS ウィンドウ環境

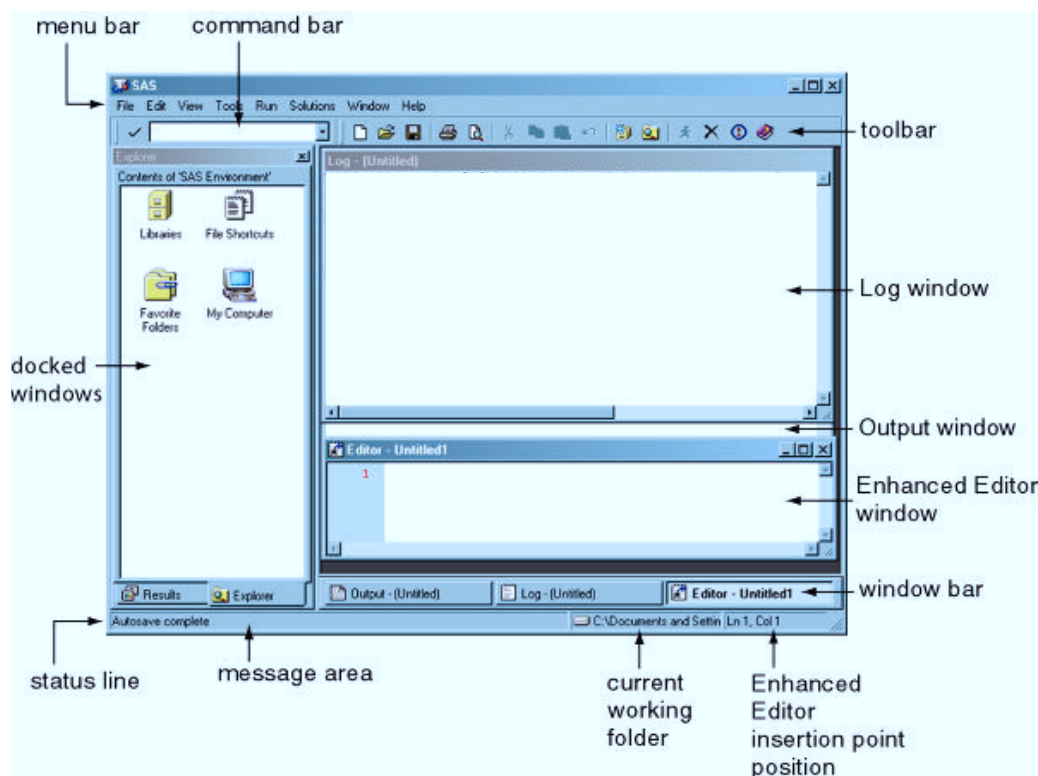
SAS ウィンドウ環境とは、メイン SAS ウィンドウ内に開かれるウィンドウを指します。PC またはターミナルエミュレータから SAS を起動すると、メイン SAS ウィンドウが開きます。

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することもできます。

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、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。ナビゲーションに使用されるウィンドウなどの特定のウィンドウは、**プリファレンス**ダイアログボックスで**ドッキングビュー**が有効化されていれば、メイン SAS ウィンドウの左側にドッキングさせることができます。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。

ウィンドウバーの有効、無効を切り替えるには、**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を使用します。

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せん。

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。

ステータスバーの有効、無効を切り替えるには、[プリファレンスダイアログボックス](#)を使用します。メッセージ行、現在のフォルダ、カーソル位置の表示を切り替えるには、[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を使用します。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](#)” (69 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の有効、無効を切り替えるには、[ツールバーのユーザー設定ダイアログボックス](#)のツールバーにスクリーンチップを表示するオプションを使用するか、または TOOLTIP コマンドをコマンドバーに入力します。

その他のすべてのスクリーンヒントの表示を切り替えるには、[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 コマンドを使用します。スクリーンチップの表示切替に関する情報については“[セッションプリファレンスの設定](#)” (60 ページ)、[“ツールバーのカスタマイズ”](#) (69 ページ)、[“WSCREENTIPS コマンド: Windows”](#) (388 ページ)、と“[TOOLTIPS コマンド: Windows](#)” (373 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. **ドッキングメニュー項目を切り替えるには、Window ⇨ Docked と選択します。**

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](#)” (61 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](#)” (377 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、“[メイン SAS ウィンドウのドッキングビューのサイズ変更](#)” (xviii ページ)、と“[WDOCKVIEWRESIZE コマンド: Windows](#)” (378 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。

- ウィンドウの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](#)” (378 ページ)、と “[WDOCKVIEWRESTORE コマンド: Windows](#)” (379 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。
- この設定は、プリファレンスダイアログボックスの表示タブで切り替えることもできます。
 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。プリファレンスダイアログボックスを使用するか、コマンドバーで `wcreentips` スクリーンチップを有効にできます。スクリーンヒントの有効に関する詳細については、“[表示プリファレンス](#)” (61 ページ)、と “[WSCREENTIPS コマンド: Windows](#)” (388 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。

ファイルをウィンドウバーのボタン上にドロップしようとする、エラーメッセージが表示されます。

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。現在のフォルダに別のデフォルトフォルダを指定するには **Start In** フィールドの値を編集します。このフィールドには、SAS プログラムショートカットのプロパティタブからまたは SAS 起動中に SASINITIALFOLDER システムオプションを指定することでアクセスできます。詳細については、[“SASINITIALFOLDER システムオプション: Windows” \(585 ページ\)](#) を参照してください。

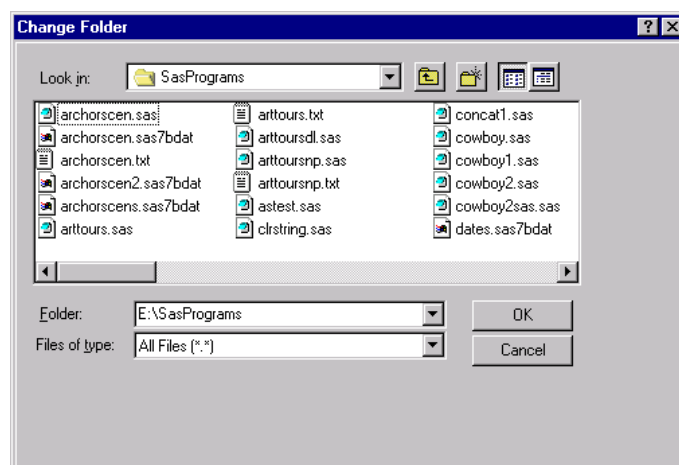
新しい現在のフォルダの対話型の選択

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. ツール ⇒ オプション ⇒ **現在のフォルダの変更** を選択してください。
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。次に、**フォルダの変更** ダイアログボックス(次の画面に表示)を使用して、新しい現在のフォルダを選択します。

画面 2.1 フォルダの変更ダイアログボックス



プロジェクトごとに異なるフォルダを使用するようにファイルを編成している場合、**フォルダの変更**ダイアログボックスでプロジェクトをすばやく切り替えることができます。異なるプロジェクトを選択するたびに、ダイアログボックスには、**フォルダ**リストボックスで選択したディレクトリが保持されます。

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の)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、**サブミット**を選択すると、コマンドバーに submit と入力した場合と同じ結果を得られます。

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](#)” (69 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あらゆる 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も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みます。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、上矢印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. **最後に使用したコマンド順に並べる**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、次の操作を行います。

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を使用する**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](#)” (348 ページ) を参照してください。ツールバーのツールのカスタマイズに関する詳細な情報は“[全般的なツールバーのプリファレンスの設定](#)” (70 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

ウィンドウそ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。**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の設定](#)” (60 ページ) 、と“[COMMAND コマンド: Windows](#)” (348 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、Lotus Notes などのメール API)
- VIM (Lotus Notes などの 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せん。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“FILENAME ステートメント, EMAIL (SMTP)アクセス方式” (*SAS ステートメント: リファレンス*) “EMAILHOST=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*)、と“EMAILPORT 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。

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 動作中にこれらの電子メールオプションをオプションステートメントにて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'>)
```

-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唯一のプロトコルは LOGIN 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。

たとえば、ログイン ID が J.B. Smith で、パスワードが rosebud の場合、SAS の起動はこのようになります。

```
c:\sas\sas.exe -emailsys vim
-emailid "J.B. Smith"
-emailpw rosebud
-emailpw{SAS002}EE4ED9392F3339441E2CD77C44A7355D
```

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。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、次のどちらかの操作を行います。

- **ファイル** ⇨ **メール送信**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、次に示す SAS の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っ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。

画面 2.2 メール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、次のフィールドが含まれます。

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。本文フィールドに入力したテキストは、右側で自動的に折り返されます。

テキスト量が多い場合、次のようにテキストファイルを添付します。

添付ファイル

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。送信者と受信者は、このアイコンをダブルクリックすることで、添付されたファイルを開くことができます。ただし、そのファイルの拡張子が、ファイルマネージャによって Windows アプリケーションに割り当てられている必要があります。たとえば、.TXT 拡張子は、メモ帳に割り当てられています。

添付ファイルを選択するファイル選択ダイアログボックスを表示するには、**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、**添付ファイル**フィールドのファイルのアイコンを選択して、**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。

宛先フィールドと Cc フィールドに指定したアドレスが有効かどうかを確認するには、**名前のチェック**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。ゲートウェイを使用して LAN (local-area-network) 外のユーザーに送信する場合など、**名前のチェック**でアドレスを確認できないような相手にメールを送ることも可能です。

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。しかし、**名前のチェック**をクリックすると、このアドレスは不明であると見なされます。この最終的な宛先アドレスが、ローカルのアドレス帳を使用して確認することができないからです。しかし、**送信**をクリックすれば、エラーが発生することなく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 から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別に、SAS で作成されるファイルの種類を次に示します。

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 出力 拡張エディタ プログラムエディタ

ファイルの種類	アクティブウィンドウ
.bmp	エクスプローラー グラフィック 結果
.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 参照ファイル EMAIL 'アドレス' <電子メールオプション>;

ここで

参照ファイル

は有効な参照ファイルです。

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。アドレス指定を変更するディレクティブ

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 (Lotus Notes などの 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 版では、デフォルト値は 256 です。レコード長には、1 から 1,073,741,823 (1GB)の値を指定できます。

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 ステートメント, EMAIL (SMTP)アクセス方式” (SAS ステートメント: リファレンス) を参照してください。

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! 複数アドレス

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! 複数アドレス

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! 複数アドレス

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!' 件名

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";
data _null_;
file mymail;
put 'Jim, ';
put 'This is my SAS configuration file.';
put 'I think you might like the';
put 'new options I added.';
run;
```

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 */
/* 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;

```

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 name dept;
/* Assign the TO attribute */
put '!EM_TO!' name;
/* Assign the SUBJECT attribute */
put '!EM_SUBJECT! Report for ' dept;
put name ',';
put 'Here is the latest report for ' dept '.';
if dept='marketing' then
put '!EM_ATTACH! c:\mktrept.txt';
else /* ATTACH the appropriate report */
put '!EM_ATTACH! c:\devrept.txt';

/* Send the message. */
put '!EM_SEND!';
/* Clear the message attributes.*/
put '!EM_NEWMSG!';
/* Abort the message before the */
/* RUN statement causes it to */
/* be sent again. */
put '!EM_ABORT!';
cards;
Susan marketing
Jim marketing
Rita development
Herb development
;
run;

```

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。ディレクティブは DATA ステップの最後にてメッセージが自動的に送られないようにします。

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 was successfully set up,
open the file to write to */
if rc = 0 then do;
fid = fopen('mailit','o');
if fid > 0 then do;
/* fput statements are used to
implement writing the mail and
the components such as subject,
who to mail to, etc. */
fputrc1= fput(fid,line1);
rc = fwrite(fid);
fputrc2= fput(fid,'!EM_TO! '||mailto);
rc = fwrite(fid);

```

```

fputcrc3= fputc(fid,'!EM_CC! '||copyto);
rc = fwrite(fid);
fputcrc4= fputc(fid,'!EM_ATTACH! '||attach);
rc = fwrite(fid);
fputcrc5= fputc(fid,'!EM_SUBJECT! '||subject);
rc = fwrite(fid);
closerc= fclose(fid);
end;
end;
return;
cancel:
call execcmd('end');
return;

```

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が表示されます。

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 `d1gsave` 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**メニュー項目を選択すると、その名前のファイルが現在のエディタの内容で**上書き**されるか、現在のエディタの内容が**追加**されます。

2. ウィンドウの内容を保存するファイルの名前を選択するか、または名前を付けます。**ファイルの種類**にて**保存**リストから、ファイルの種類を選択することもできます。ほとんどの種類のファイルはプレーンテキストとして保存され、選択された種類に基づいてファイル拡張子が割り当てられます。例外として、RTF ファイルは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。
- 編集メニューから**すべてクリア**を選択します。
- 新規作成ボタン(空白のページ)をクリックします。
- ファイルメニューで**新規作成**または**プログラムの新規作成**を選択します。
- コマンドバーに CLEAR と入力して、ENTER 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”(653 ページ)、と“拡張エディタでのショートカットキー”(657 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呼び出す操作を、Ctrl + H キーに割り当てるには、次のコマンドを CTL H の横にある定義列に指定します。

```
zoom; recall
```

SAS では、使用されるまで(キーを押されるまで)コマンドの構文は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 処理を *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 SAS GRAPH グラフィックを取り込むウィンドウ詳細については、“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” (195 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。構文は次のとおりです。

```
gsubmit buf=default
```

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**” (69 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 \(58 ページ\)](#)、と [表 2.3 \(59 ページ\)](#) にリストアップされています。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 コマンドを使用するか、ツール ⇒ オプション ⇒ フォントを選択して、フォントダイアログボックスを開きます。

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参照するか、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、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。

フォント項目では、SAS/GRAPH フォントを選択できません。

注意:

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。モニターによっては、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。

セッションプリファレンスの設定

セッションの環境設定について

SAS セッションは、操作しやすいように構成できます。例を次に示します。

- 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](#)” (60 ページ)
- “[一般プリファレンス](#)” (61 ページ)
- “[表示プリファレンス](#)” (61 ページ)
- “[編集プリファレンス](#)” (62 ページ)
- “[結果プリファレンス](#)” (62 ページ)
- “[Web プリファレンス](#)” (63 ページ)
- “[詳細プリファレンス](#)” (63 ページ)

プリファレンスダイアログボックスの使用

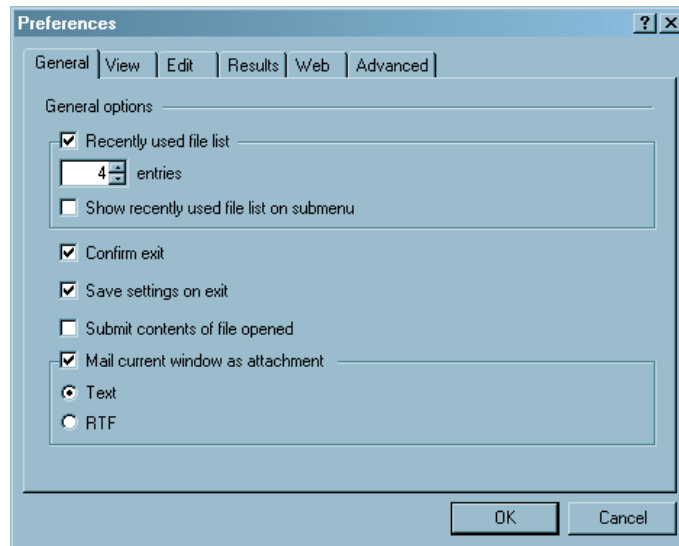
SAS セッションをカスタマイズするには、次のいずれかの方法でプリファレンスダイアログボックスを開きます。

- コマンドバーに `d1gpref` 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。変更を終了したら **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。

画面 2.3 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。1721 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.\$SAS としてオペ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 c:\Users\ユーザID\AppData です。他のすべての Windows 環境では、アプリケーションデータフォルダのパス名 c:\Documents and Settings\ユーザID\Application Data 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。結果タブで設定できるオプションは、次のとおりです。

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: リスティングフォーマットを使って出力を表示したいときは、プリファレンスウィンドウにてリスティングオプションを選択してください。

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

WORK フォルダに HTML 出力ファイルを保存する場合に指定します。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、“**TEMPLATE プロシジャ: スタイルテンプレートの作成 プロシジャ**” (*SAS Output Delivery System: ユーザーガイド*) 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

内部ブラウザ、インター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**” (376 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> 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した場合、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行は表示されません。

最大スクロールを選択した場合、プロシジャが完了するまで、ウィンドウに出力は書き込まれません。

行スクロールを選択した場合、アウトプットウィンドウ全体がいっぱいになると、アウトプットウィンドウは、**行スクロール**ボックスにて指定した行数だけスクロールさ

れます。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](#)” (345 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

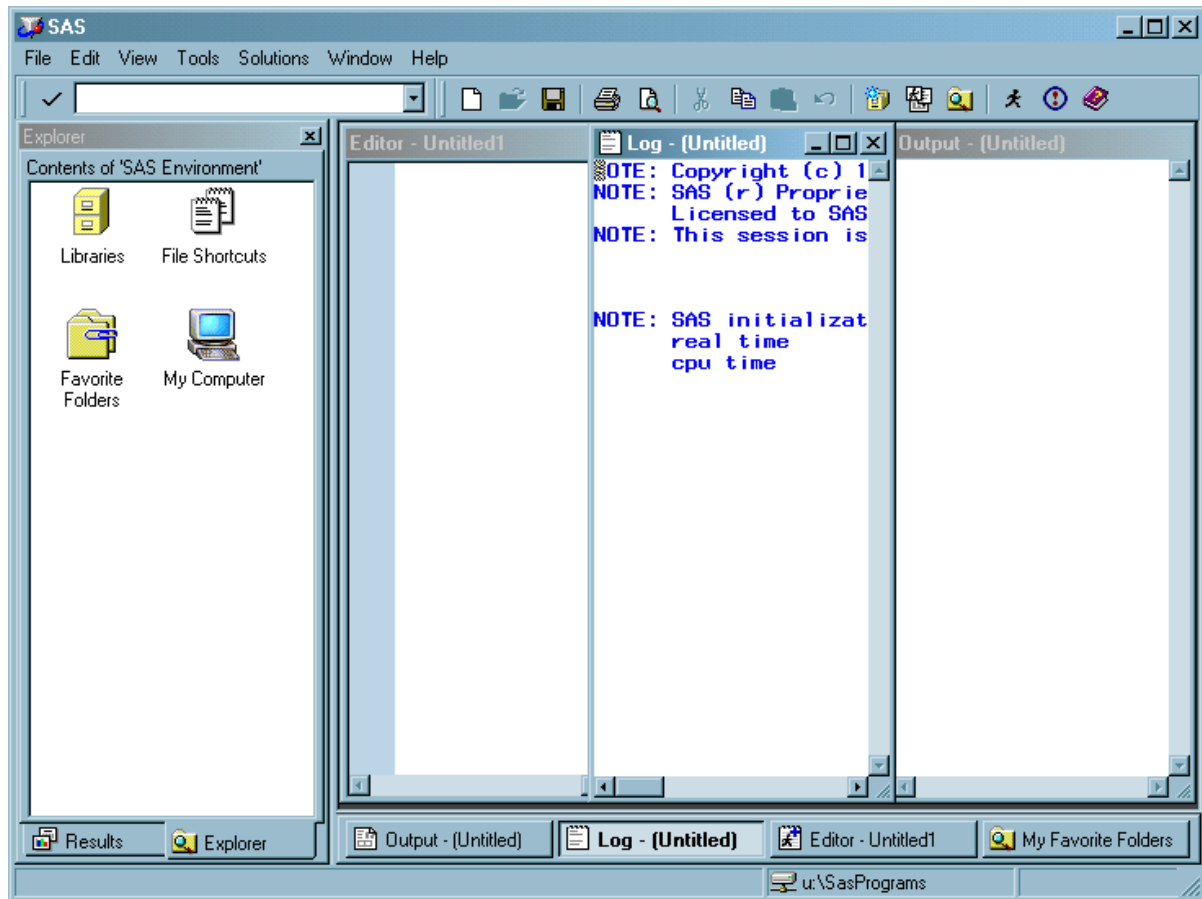
デフォルトの構成でも効率的に SAS を使用できますが、簡単にアクセスできるように別のウィンドウも開いて、メイン SAS ウィンドウ内に再配置できます。たとえば、**お気に入りフォルダ**ウィンドウを開いて最小化にして、すべてのウィンドウを一度に表示できるようにモザイク状に配置できます。

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. **ウィンドウ** ⇨ **上下に並べる**を選択します。

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。

画面 2.4 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” (333 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 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。“[メイン SAS ウィンドウのサイズと配置の変更](#)” (67 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](#)” (67 ページ) を参照してください。

HELPPREGISTER

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。“[HELPPREGISTER システムオプション: Windows](#)” (545 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](#)” (67 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](#)” (68 ページ) を参照してください。

SASINITIALFOLDER

SAS 起動時に開くと名前を付けて保存のダイアログボックスで表示する作業フォルダやデフォルトフォルダのパス名を指定します。“[SASINITIALFOLDER システムオプション: Windows](#)” (585 ページ) を参照してください。

SPLASHLOC と NOSPLASH

SAS セッションの開始時に表示されるロゴスクリーンのパス名または DLL (ダイナミックリンクライブラリ)名を指定します。指定しないと、ロゴスクリーンは表示されません。“[SAS 起動時のカスタムロゴ画面の表示](#)” (68 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](#)” (69 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](#)” (69 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](#)” (493 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](#)” (332 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” (522 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 が含まれます。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” (524 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

HELPPREGISTER 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。HELPPREGISTER 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、Windows Vista には含まれていません。Microsoft HTML ヘルプ (.chm) の使用をお勧めしません。

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のいずれかで、複数の HELPPREGISTER 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、“HELPPREGISTER システムオプション: Windows” (545 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” (549 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"
```

メニューに Excel が追加されます。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](#)” (579 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](#)” (585 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](#)” (597 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](#)” (612 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](#)” (70 ページ)
- “[全般的なツールバーのプリファレンスの設定](#)” (70 ページ)
- “[ツールバーのカスタマイズ](#)” (70 ページ)
- “[ツールバーへのツールの追加](#)” (72 ページ)
- “[ツールバーからのツールの削除](#)” (73 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ツールバーをカスタマイズし保存する](#)” (73 ページ)

- “ツールをデフォルト設定にリセットする” (74 ページ)
- “作成可能な便利なツール例” (74 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 TOOLEEDIT を入力します。
- ツール ⇨ カスタマイズを選択します。

一般的なツールバーの設定はツールバータブ、ツールバーのツールの定義はユーザー設定タブを使用します。

全般的なツールバーのプリファレンスの設定

ツールバータブでは、ツールバーの操作と表示の制御を設定します。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。

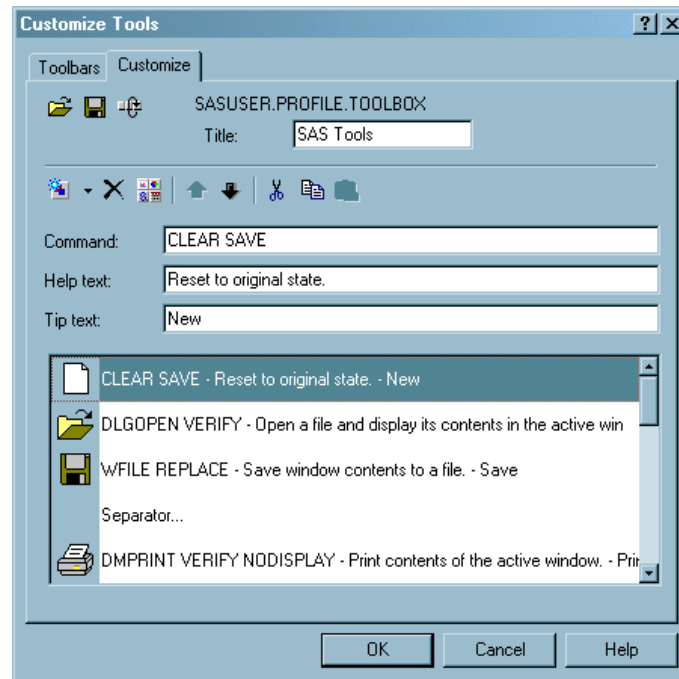
- **オートコンプリートを使用する**を選択した場合、前に入力したコマンドが保存され、コマンドの入力を開始するとコマンドが表示されます。
- コマンドバーリストのコマンドを最近入力した順に並べ替えるには、**最後に使用したコマンド順に並べる**を選択します。この設定が選択されていない場合、ドロップダウンリストのコマンドは、使用頻度の順に並べられます。
- **コマンドの保存数**で、コマンドバーリストに保存するコマンド数を入力します。有効な値の範囲は、0-50 です。デフォルトは 15 です。

ツールバータブを構成したら、**ユーザー設定**をクリックしてカスタマイズを行うか、**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

次のユーザー設定タブでは、ツールバーのコマンドを追加、削除、変更できます。

画面 2.5 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**と**区切り**のどちらかの追加を選択できます。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**アクション**も含まれます。

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。コマンドボックスで、コマンドを変更することができます。

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で、ヘルプテキストを変更することができます。

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

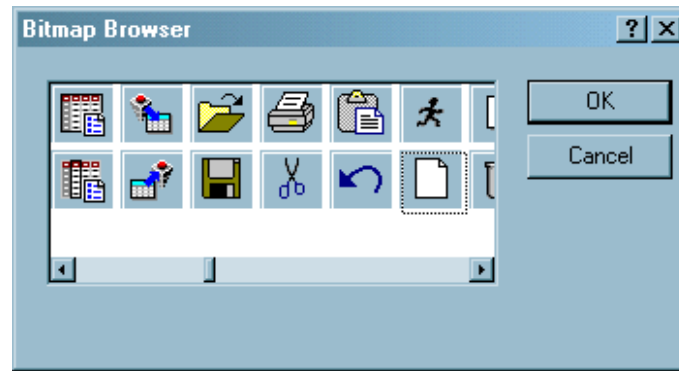
ツールバーに定義されたボタン、コマンド、ヘルプテキスト、区切りが示されます。

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の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アクションを選択します。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. ツールのアイコンを選択する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にスクリーンヒントテキストを入力します。
5. ツールを上へ移動ボタンやツールを下へ移動ボタンをクリックして、ツールバーリストのツールを並べ替えます。
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。

画面 2.6 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

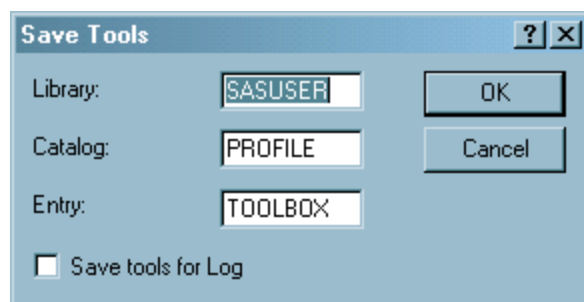
ツールバーにコマンドを追加する前に、そのコマンドがメニューから使用できることを確認します。常に有効な印刷コマンドとコピーコマンドを除き、メニューから使用できるコマンドにのみボタンを使用できます。

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 [画面 2.7 \(74 ページ\)](#) のように表示されます。
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。

画面 2.7 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](#)” (372 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。ツールの設定をデフォルトに戻すかどうかを確認するメッセージが表示されます。はいをクリックすると、ツールは、SAS がインストールされた時点の設定に戻ります。

SAS アプリケーションでそのウィンドウのデフォルトのツールバーを定義している場合、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元に戻ります。

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. **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。
チップ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**ボタンや**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を並べ替えます。
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に、その項目の定義が表示されます。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 Institute の 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。

リリース情報

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。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デフォルトの URL が表示されます。デフォルトの URL は、プリファレンスダイアログボックスの Web タブで指定できます。詳細については、“[WBROWSE コマンド: Windows](#)” (376 ページ) を参照してください。

SAS Institute のホームページなど、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せん。

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。HELPHOST のデフォルト値として、X11

DISPLAY 環境変数に指定されたホスト名が設定されます。また、HELPPORT のデフォルト値として、Remote Browser Server の標準ポートが設定されます。

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=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)と “HTTPSERVERPORTMAX=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)。

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、または結果ウイ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。結果ウインドウで 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、ログイ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、ウイ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](#)” (542 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*内の 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 環境での SAS エディタの
使用

拡張エディタの使用	81
拡張エディタの機能	81
拡張エディタウィンドウの使用	82
独自のキーワードの作成	102
ファイルタイプへのファイル拡張子の割り当て	103
拡張エディタオプションの設定	104
拡張エディタキーボードのカスタマイズ	108
拡張エディタの有効と無効の切り替え	109
プログラムエディタの使用	110
プログラムエディタの使用の概要	110
拡張エディタとプログラムエディタの切り替え	110
ファイルを開く	110
行番号の使用	111
挿入位置の移動	112
タブの使用	112
改行について	112
テキストの選択	112
テキストの削除	113
テキストの検索と置換	113
テキストのドラッグアンドドロップ	114
ドラッグ時のスクロール	116
リッチテキストフォーマットのテキストの使用	116
ファイルの保存	117
自動保存によるプログラムエディタファイルの保存	117
SAS テキストエディタ独自の機能について	117

拡張エディタの使用
拡張エディタの機能

拡張エディタには、従来のプログラムエディタのいくつかの機能のほかに、次の機能が追加されています。

- HTML および XML ドキュメントの要素のほかに、SAS プログラム要素および SCL プログラム要素を識別するために、テキストを色分けして表示できます。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。

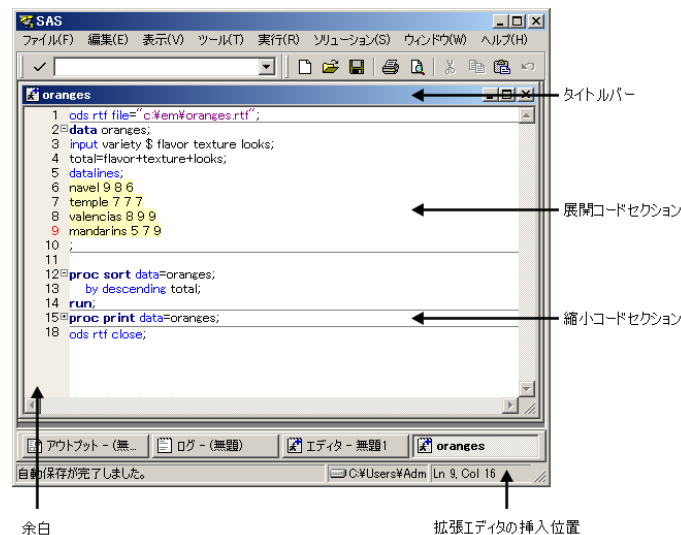
- Enter キーを押すと、次の行が自動的にインデントされます。
- SAS プロシジャ、DATA ステップ、マクロのセクションを展開または折りたたむことにより、SAS プログラムの高レベルのフローを表示したり、各ステートメントの詳細を表示します。
- キーボードマクロの記録機能を使用して、プログラム編集コマンドを記録および再生するマクロを作成できます。
- 省略形を使用して、テキストを入力するためのショートカットを作成できます。
- コード行にブックマークを付けることにより、プログラムまたはドキュメントの異なるセクションに簡単にアクセスできます。
- ほとんどの拡張エディタコマンドのショー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。

画面 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。ファイルが新規に作成された場合、ファイル名は **Editor Untitled x** になります。 x は、ウィンドウ番号です。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次の方法で移動できます。

- **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

操作	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。 <ul style="list-style-type: none"> • コマンドバーに <code>wedit</code> と入力します。 • 次のメニューパスを選択します。表示 ⇨ 拡張エディタ • 新規 ツールバーボタンをクリックします。 • 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。 <ul style="list-style-type: none"> • 開く ツールバーボタンをクリックします。 • 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 • コマンドバーに <code>fileopen</code> と入力します。* <p>ファイルを選択します。</p> <p>開く をクリックします。</p> <p><small>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。</small></p>

操作	実行手順
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。	<ol style="list-style-type: none"> 1. Ctrl キーを押したままにします。 2. ファイルを選択します。 3. 開くをクリックします。
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。	<p>次のうちいずれかを実行します。</p> <ul style="list-style-type: none"> • コマンドバーに <code>wedit "ファイル名"</code> を入力します。 • コマンドバーに <code>fileopen "ファイル名"</code> 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。	<ol style="list-style-type: none"> 1. ファイルをアクティブウィンドウにします。 2. ウィンドウ → 新しいウィンドウを選択します。 <p>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。</p>
開かれたファイルにファイルを追加します。	<ol style="list-style-type: none"> 1. 追加するファイルをアクティブウィンドウにします。 2. ファイル → 追加を選択します。 3. 追加するフォルダおよびファイルを選択します。 4. 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: 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](#)” (585 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、次の操作を実行します。

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” (585 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、オペレーティングシステムの 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 Vista、Windows 7 および Windows Server 2008 の場合、Application Data フォルダのパス名は c:\Users\ユーザーID\AppData\Roaming です。他の Windows オペレーティングシステムの場合、Application Data フォルダのパス名は c:\Documents and Settings\ユーザー名\Application Data\SAS\EnhancedEditor 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:\Documents and Settings\myuserid\Application Data\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、次の操作を実行します。

1. スタート ⇨ コンピューター ⇨ 整理 ⇨ フォルダーを選択します。
2. 検索オプション ⇨ 表示を選択します。
3. すべてのファイルとフォルダを表示を選択します。
4. 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。
5. 適用および OK を選択します。

自動保存機能の設定の詳細については、“編集プリファレンス” (62 ページ) および “WAUTOSAVE コマンド: Windows” (375 ページ) を参照してください。

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、次の操作を実行します。

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。たとえば、myfile.sas:1 および myfile.sas:2 のようになります。

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	デフォルト値(<i>n</i>)	許容可能な最大値	例
:In	現在の行の後に、 <i>n</i> 行を追加します。	1	9999	:I4 現在の行の後に、4 行を挿入します。
:IA <i>n</i>	現在の行の後に、 <i>n</i> 行を追加します。	1	9999	:IA4 現在の行の後に、4 行を挿入します。
:IB <i>n</i>	現在の行の前に、 <i>n</i> 行を追加します。	1	9999	:IB2 現在の行の前に、2 行を挿入します。
:D <i>n</i>	現在の行から始まる <i>n</i> 行を削除します。	1	9999	:D3 現在の行から始まる 3 行を削除します。

コマンド	説明	デフォルト値(n)	許容可能な最大値	例
:Rnm	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 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 挿入位置移動の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 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 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、次の操作を実行します。

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。	<ol style="list-style-type: none"> 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。 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。 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。	<ol style="list-style-type: none"> 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。 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。 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	<ol style="list-style-type: none"> 1. 選択したテキスト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。 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。 4. マウスの左ボタンを離します。
コードまたはデータの列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](#)” (657 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、次の操作を実行します。

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、次の操作を実行します。
 - 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、次の操作を実行します。
 - 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、次の操作を実行します。

1. 次のメニューパスを選択します。ツール ⇒ オプション ⇒ **拡張エディタ** ⇒ **全般**
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、次の操作を実行します。

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string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、次の操作を実行します。

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。

正規表現を使用する

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string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、次の操作を実行します。

- 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	<p>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。</p> <p>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。</p>
不当な引用文字列	<p>プログラムの行の中から、同じ色のものを検索します。</p> <p>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。</p>
不当なコメント	<p>プログラムの行の中から、同じ色のものを検索します。</p> <p>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。</p>
対応する DO-END ペア単位	<p>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。</p> <p>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。</p> <p>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。</p>
対応するかっこまたは角かっこ	<p>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。</p> <p>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。</p>
セミコロン(;)の欠落	<p>標準のテキストで表示されるキーワードを検索します。</p>

カラーコーディングできるコンポーネント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。コンポーネントは**ファイ**

ルの要素ボックスに表示されます。プログラムコンポーネントのカラー定義の詳細については、“[デザインタブの設定](#)” (107 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、次の操作を実行します。

1. 次のメニューパスを選択します。ツール ⇒ オプション ⇒ 拡張エディタ ⇒ 全般
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ものではありません。特定の拡張エディタが開いている間のみ使用可能です。

表 3.8 行のブックマーキングのショートカットキー

操作	ショートカットキー
行にブックマークを付ける	Ctrl+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、myv6sasfiles という省略形を定義した場合、この省略形は'c:\winnt\profiles\myid\personal\mysasfiles\v6';に展開されます。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、次の操作を実行します。

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、次の操作を実行します。

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. 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。

省略形を削除するには、次の操作を実行します。

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ます。
ツール ⇒ **キーボードマクロ** ⇒ **マクロ**
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、次の操作を実行します。

- 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	ファイルを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。 <ul style="list-style-type: none"> サブミットツールバーボタンをクリックします。 F3 キーまたは F8 キーを押します。 次のメニューパスを選択します。実行 ⇨ サブミット コマンドバーに <code>submit</code> 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。 <ul style="list-style-type: none"> コマンドバーに <code>subtop</code> と入力します。 次のメニューパスを選択します。実行 ⇨ 先頭行をサブミット
最初の行から数えて、任意の行数をサブミットする。	<ul style="list-style-type: none"> コマンドバーに <code>subtop n</code> と入力します。<code>n</code> は、サブミットする行数です。 次のメニューパスを選択します。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” (370 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](#)” (104 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。

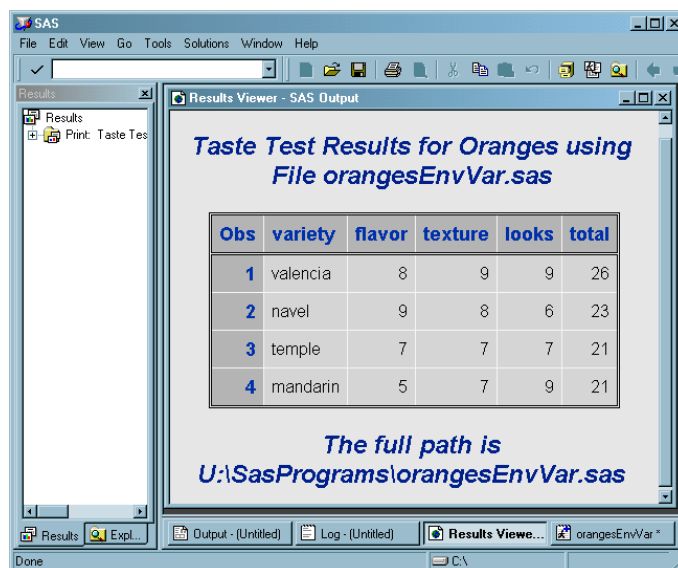
```
data oranges;
input variety $ flavor texture looks;
total=flavor+texture+looks;
cards;
navel 9 8 6
temple 7 7 7
valencias 8 9 9
mandarins 5 7 9
;

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。

画面 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MIT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MITするときや、バッチセッションでコードをサブMIT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MIT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MIT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MIT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MITされる場合、環境変数は最後にサブMIT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MIT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” (657 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+

ショートカットキー	タスク
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
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、“[拡張エディタキーボードのカスタマイズ](#)” (108 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、“[拡張エディタキーボードのカスタマイズ](#)” (108 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、記録を終了します。**ツール ⇒ キーボードマクロ ⇒ 記録終了**
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、“[拡張エディタキーボードのカスタマイズ](#)” (108 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、次の操作を実行します。

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. キーボードマクロの作成ダイアログボックスを開くには、**作成**をクリックします。
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、次の操作を実行します。

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. キーボード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、次の操作を実行します。

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、次の操作を実行します。

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ます。**ツール** ⇨ **キーボードマクロ** ⇨ **マクロ**
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空白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、次の操作を実行します。
 - 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

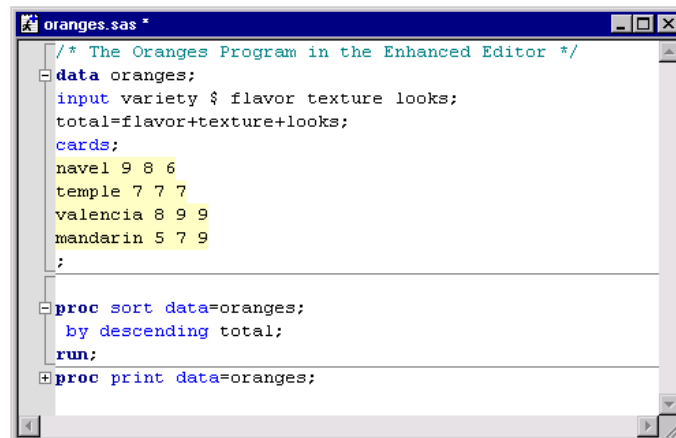
コードセクションの折りたたみでは、コードの1つ以上のセクションを展開または折りたたみできます。セクションは、ステップキーワード、コメント、またはセクションワードやコメント上のスペースから始まります。セクションは、次のステップキーワード、コメント、または次のセクションワードやコメント上のスペースで終了します。ステップキーワードには、DATA ステートメント、PROC ステートメント、%マクロステートメントがあります。シグネチャ行は、ステップキーワードが表示される行です。

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。

画面 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



```

oranges.sas *
/* The Oranges Program in the Enhanced Editor */
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 9
;

proc sort data=oranges;
by descending total;
run;

proc print data=oranges;

```

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、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 ⇒ 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せん。
- 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 プロシー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. キーワード名に空白は使用できません。

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、次の操作を実行します。

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、次の操作を実行します。

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種類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、次の操作を実行します。

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、次の操作を実行します。

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

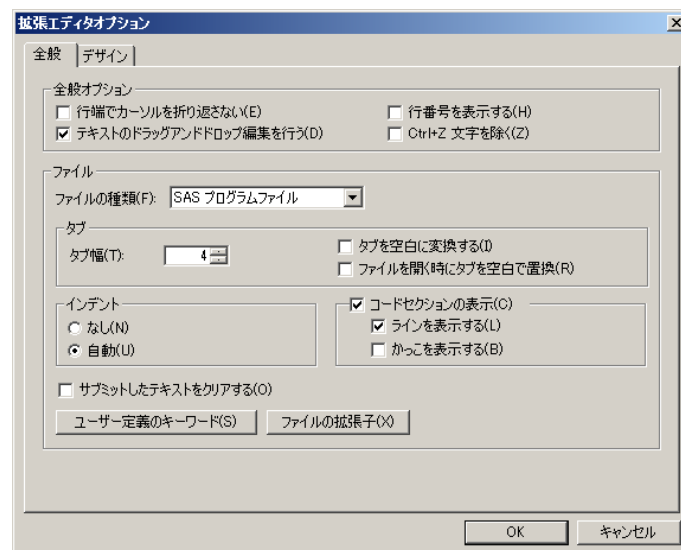
拡張エディタオプションウィンドウを開く

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。

画面 3.4 拡張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前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。

デザインタブ

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、“[デザインタブの設定](#)” (107 ページ) および “[配色の使用](#)” (107 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。

デザインタブの設定

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が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、次の操作を実行します。

1. 次のメニューパスを選択して、エディタ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**” (107 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、次の操作を実行します。
 - 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、次の操作を実行します。

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、次の操作を実行します。

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せん。
5. **OK**をクリックします。

配色を削除するには、次の操作を実行します。

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。

拡張エディタキーボードのカスタマイズ

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、次の操作を実行します。

1. 次のメニューパスを選択します。**ツール** ⇨ **オプション** ⇨ **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、次の操作を実行します。

1. 次のメニューパスを選択します。ツール ⇒ オプション ⇒ 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のリセット

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](#)” (657 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、次の操作を実行します。

1. 次のメニューパスを選択します。ツール ⇒ オプション ⇒ 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](#)” (534 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](#)” (379 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、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、あらかじめ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については、“[ENHANCEDEDITOR システムオプション: Windows](#)” (534 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、次の操作を実行します。

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 `d1gopen` 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” (585 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、次の操作を実行します。

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 の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。エディタオプションダイアログボックスを開くには、次の操作を実行します。

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全般タブでコマンド行を選択します。

タブの使用

使用するテキストエディタによって、タブ文字がそのまま使用されるものと、空白文字に変換されるものがあります。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)。

```

Program Editor - (Untitled)
OPTIONS LS=132 PS=60;
DATA CHILDREN;
  INPUT SEX $ AGEM HEIGHT WEIGHT;
  AGE=INT(AGEM/12);
  IF AGE<=16;
CARDS;
F 143 56.3 85.0
F 155 62.3 105.0
F 153 63.3 108.0
F 161 59.0 92.0
F 191 62.5 112.5
F 171 62.5 112.0
F 185 59.0 104.0
F 142 56.5 69.0
F 160 62.0 94.5
F 140 53.8 68.5
F 139 61.5 104.0
F 178 61.5 103.5
F 157 64.5 123.5
F 149 58.3 93.0

```

- 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](#)」(57 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、次の操作を実行します。

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. 次の情報が必要になります。

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、次の操作を実行します。

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. 次の情報が必要になります。

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、次の操作を実行します。

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ヘドラッグします。
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。

SAS のターゲットウィンドウで利用できるドラッグアンドドロップアクションを、[表 3.14 \(115 ページ\)](#) に示します。

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

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	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)。

この動作は、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(ディスクのアイコン)。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ます。このダイアログボックスで、ファイル名を指定します。

新しいファイル名を付けてファイルを保存するには、次の操作を実行します。

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](#)” (585 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、次の操作を実行します。

1. 次のメニューパスを選択します。**ツール** ⇨ **オプション** ⇨ **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については、“[編集プリファレンス](#)” (62 ページ)、[“WAUTOSAVE コマンド: Windows”](#) (375 ページ) および “[AUTOSAVELOC=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。

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なります。

- 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空白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	120
SAS ファイルについて	120
SAS ファイルのファイル拡張子	120
SAS データセット(メンバタイプ: Data または View)	122
SAS カタログ(メンバタイプ: Catalog)	123
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)	123
アクセスディスクリプタファイル(メンバタイプ: Access)	124
マルチエンジンアーキテクチャ	124
SAS ライブラリ	124
SAS エンジン	124
データライブラリの使用	127
データライブラリ	127
ライブラリ参照名の指定	127
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 ..	128
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	128
環境変数を使用した SAS ライブラリの割り当て	130
ライブラリ参照名の割り当てのリストの表示	132
ライブラリ参照名のクリア	132
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	134
Sasuser データライブラリの更新	135
WORK データライブラリの使用	135
Windows (NTFS)における大きいデータセットの使用	137
複数の SAS セッションから SAS ファイルにアクセスする	137
Windows 版 SAS 9.3 で他のバージョンのファイルを使用する	138
Windows 版 SAS 9.3 での他のバージョンのファイルの使用について	138
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	139
リリース 6.03 とリリース 6.04 の SAS データセットの使用	140
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	140
SAS 9.3 でバージョン 7 と 8 の SAS カタログを使用する	140
SAS 9.3 でバージョン 6 の SAS カタログを変換する	140
リリース 6.08 の SAS カタログを SAS 9.3 に変換する	141
リリース 6.03 およびリリース 6.04 の SAS カタログを SAS 9.3 に変換する	142
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	142
SAS 9.3 のファイルを以前のリリースで使用する	142
SAS 9.3 でリモートホストの SAS ファイルを使用する	142
BMDP ファイル、OSIRIS ファイル、SPSS ファイルの読み取り	143
概要: BMDP ファイル、OSIRIS ファイル、SPSS ファイルの読み取り	143

BMDP Engine	143
OSIRIS Engine	144
SPSS Engine	145
SAS ファイルを動作環境間で転送する	146
SAS/ACCESS を使用し、データベースファイルにアクセスする	146
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	147

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 9.3 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](#)” (124 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 9.3 で他のバージョンのファイルを使用する](#)” (138 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	解説
.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 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 ファイルの拡張子を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 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 システムオプション” (SAS システムオプション: リファレンス) および “WORKTERM 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。

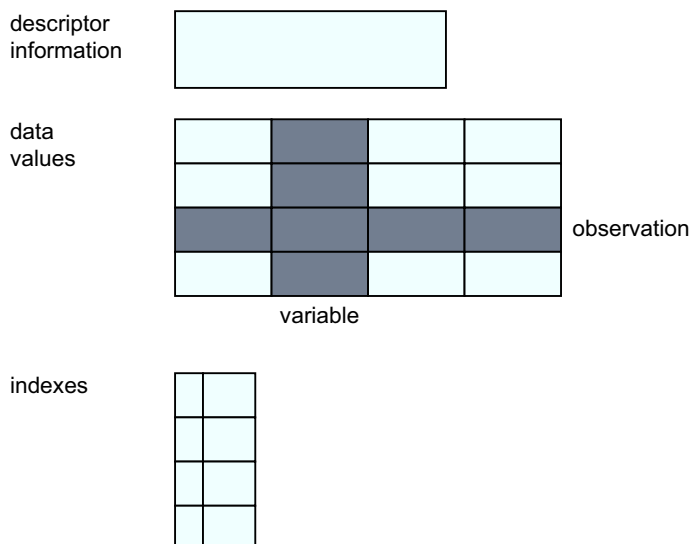
SAS データセット(メンバタイプ: Data または View)

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](#)” (447 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



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 ファイルなどがあります。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*” (143 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、.sas7bview 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。あるいは環境変数、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**”(127 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。使用するエンジンの指定が必要なファイルの種類もありますが、自動的に適切なエンジンが選択されるファイルの種類もあります。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](#)” (127 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	解説
デフォルト	V9, BASE	SAS System 9、9.1、9.2、SAS 9.3 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。
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 9.3 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 9.3 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 ⇒ システム選択すると、システムオプションウィンドウが開きます。その後、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	解説
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。
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](#)” (143 ページ) および “[ENGINE システムオプション: Windows](#)” (533 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。このフォルダは混合モードライブラリと言います。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 SAS で使用されるデフォルトのエンジンを指定することができます。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” (533 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。ライブラリの作成ダイアログボックスを使用して対話的に作成されたライブラリ参照名は、**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**ライブラリの作成**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” (128 ページ) に記述されている LIBNAME ステートメントまたは LIBNAME 関数を使用します。
- “環境変数を使用した SAS ライブラリの割り当て” (130 ページ) に記述されている環境変数を定義します。

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、ライブラリの作成ツールバーボタン 、LIBASSIGN コマンドまたはエクスプローラを使用して、**ライブラリの作成**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 ウィンドウが表示されたら、**新規作成**ツール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てメニューから**新規作成**を選択します。

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ときには、**ライブラリフォルダ**を右クリックし、**新規作成**を選択しても、**ライブラリの作成**ダイアログボックスを開くことができます。

ライブラリの作成ダイアログボックスとエクスプローラの詳細については、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引数の説明は、*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 9.3 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 を割り当てられます。

```
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 9.3 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](#)” (134 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](#)” (496 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 9.3 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 9.3 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 9.3 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 9.3 データセットを読み込むことはできません。SAS 9.3 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> の移行に関する記述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 9.3 で他のバージョンのファイルを使用する](#)” (138 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](#)” (143 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記述を *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。**ライブラリの作成ダイアログボックスで、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、環境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。環境変数によって、Windows 環境内で、ある文字列と別の文字列が結び付けられます。SAS では、次の 2 つの環境変数が認識されません。

- SAS 環境変数
- 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**ライブラリの作成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(前回の割り当てを優先)
2. SAS 環境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:\WINNT\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ます。

```
libname temp c:\public
```

このライブラリ参照名は、c:\public を示すと解決されます。

次に、環境変数を SAS に定義する 2 つの方法を示します。

- SET システムオプションを使用します。このオプションで、SAS (内部)環境変数を定義します。
- Windows の SET コマンドを発行します。このコマンドで、Windows (外部)環境変数を定義します。Windows 環境では、この他にシステムのプロパティダイアログボックスを使用して環境変数を定義することもできます。システムのプロパティダイアログボックスは、コントロールパネルからアクセスすることも、マイコンピュータを右クリックしプロパティを選択してアクセスすることもできます。

注意:

エンジンを環境変数に割り当てることはできません。環境変数をライブラリ参照名として使用する場合、デフォルトエンジンを使用する必要があります。

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。

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 環境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、環境変数 TEST をこの SAS ライブラリに割り当てられます。

```
-set test c:\sas\mysasdata
```

環境変数を割り当てる際、SAS では環境変数名を実際に使用するまで環境の参照が有効であるかどうかを確認しません。たとえば、環境変数 TEST が SAS 構成ファイルで定義されている場合、環境変数 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 ステートメントで環境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](#)” (587 ページ) を参照してください。

環境変数名を 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、使用する前に環境変数として定義する必要があります。たとえば、環境変数 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、環境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 ステートメントに環境変数を使用する場合、環境変数の前に感嘆符(!)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 を起動する前に環境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 の環境変数を定義することはできません。

環境変数が 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、環境変数が認識されます。Windows の起動時に(グローバル変数の作成)実行する Windows の AUTOEXEC.BAT ファイルに環境変数を定義するか、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環境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、環境変数は認識されません。

SET コマンドを使用して定義した環境変数は、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、環境変数 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。ライブラリ参照名として定義した環境変数を SAS プログラム内でまだ使用していない場合、その環境変数は、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” (133 ページ)
- “LIBNAME ステートメント” (133 ページ)
- “LIBNAME 関数: Windows” (425 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: 環境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 Protokol

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 Protokol 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 Protokol 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、置換されません。

```
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 Protokol 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 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。

表 4.4 SASUSER ライブラリ参照名のフォルダ

Windows Vista、Windows 7、 Windows Server 2008、Windows Server 2008 R2	C:\Users\user ID\Documents\My SAS Files\9.3
Windows XP Windows Server 2003	C:\Documents and Settings\user ID\My Documents\My SAS Files\9.3

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、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 9.3 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](#)” (586 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 TDnnnnnnnnnn 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 という名前の一時的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アイコンをダブルクリックし、次に WORK フォルダアイコンを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 9.3 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環境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環境変数を設定します。

```
-set myvar c:\ tempdir
-work !myvar
```

SET オプションは、MYVAR 環境変数を 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](#)” (611 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せん。

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**のどちらかを選択します。

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。MEMBER モードを選択すると、自分だけがそのデータセットにアクセスできます。RECORD モードを選択すると、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、ある 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。

1. コマンドバーに、FSEDIT または FSVIEW と入力します。
2. **メンバの選択**ダイアログボックスで、ライブラリ名を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” (580 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 9.3 で他のバージョンのファイルを使用する

Windows 版 SAS 9.3 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 9.3 形式に変換する必要はありません。

SAS 9.3 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 4.5 (138 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 9.3 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 9.3 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 9.3 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 4.5 SAS 9.3 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 ビット環境の場合は、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。 64 ビット環境の場合は、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 ビット環境の場合は、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。 64 ビット環境の場合は、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。
リリース 6.08 - 6.12	V6 Engine が自動的に検出されます。32 ビット環境の場合は、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。 64 ビット環境の場合は、SAS から、V6 データセットを読み込みますが、書き込みません。	32 ビット環境の場合は、SAS から、バージョン 6 のカタログを読み込みますが、書き込みません。 64 ビット環境の場合は、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。

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 6.03 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、リリース 6.04 および 6.03 のデータセットを除いて、バージョン 6 (32 ビット環境)、バージョン 7 および 8 のデータセットを SAS 9.3 で読み込み、更新し、書き込むために、SAS 9.3 のデータセットに変換する必要はありません。64 ビット環境では、Cross-Environment Data Access (CEDA) 機能による制限があります。

32 ビット環境では、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、読み込みのみ可能です。バージョン 6 のカタログを更新するには、SAS 9.3 カタログに変換する必要があります。

<http://support.sas.com/migration> の移行フォーカスエリアでは、リリース 6.08 からバージョン 8 までに作成された SAS ファイルの使用または変換方法について詳細に説明されています。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する場合に、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](#)” (140 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 9.3 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。(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 9.3 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 9.3 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 9.3 形式に変換することを検討してください。SAS 9.3 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 9.3 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、移行フォーカスエリア(<http://support.sas.com/migration>)から情報にアクセスできます。

SAS 9.3 でバージョン 7 と 8 の SAS カタログを使用する

SAS 9.3 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 9.3 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 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 9.3 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 9.3 形式に変換しないと、Windows 版 SAS 9.3 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クなどを使用して、SAS 9.3 セッションで読み込み可能な場所に転送します。
3. 9.3 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 プロシジャおよび CIMPORT プロシジャの詳細については、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 9.3 に変換する

リリース 6.08 から SAS 9.3 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 9.3 の CIMPORT プロシジャを使用して SAS 9.3 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 9.3 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 9.3 では使用できません。

Windows 版 SAS 6.08 から SAS 9.3 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 9.3 で読み込み可能な場所に転送します。
3. 9.3 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 V9.3 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 9.3 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 9.3 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 プロシジャの詳細については、*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 9.3 に変換する

リリース 6.04 の SAS カタログを SAS 9.3 SAS カタログに変換する場合、*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 9.3 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> の移行フォーカスエリアで "Using a SAS 9 File Under a Previous Version of SAS" を参照してください。

SAS 9.3 でリモートホストの SAS ファイルを使用する

SAS 9.3 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 9.3 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 9.3 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> の移行フォーカスエリア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 9.3 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 PDF 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](#)” (449 ページ) を参照してください。

- “BMDP Engine” (143 ページ)
- “OSIRIS Engine” (144 ページ)
- “SPSS Engine” (145 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みます。

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](#)” (449 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* 引数は、環境変数名またはファイル参照名にすることもできます。環境変数名またはファイル参照名にする場合は、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。

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。

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 デクシオ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 データデクシオ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 データデクシオナリを受け入れるように設計されているためです。OSIRIS デクシオ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 デクシオ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、デクシオナリファイルは 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。¹SPSS 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。たとえば、-1 から.A および-2 から.B。

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。

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 sascode temp;
data _null_; set new; file sascode;
if formatd > formatl then do;
revision=cat(format,formatl,'.2');
put 'format' +1 name +1 revision ';' ;
end;
run;
data temp; set abc._all_;
%inc sascode/source2;
run;
```

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アを使用できません。

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 ODBCSESV)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	149
外部ファイルの参照	150
外部ファイルにアクセスする	150
ファイル参照名の使用	151
引用符で囲まれた Windows ファイル名の使用	159
作業ディレクトリでのファイルの使用	160
SAS ステートメントを使用し、外部ファイルにアクセス	161
概要: 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	161
FILE ステートメントの使用	161
INFILE ステートメントの使用	162
%INCLUDE ステートメントの使用	163
SAS コマンドを使用し、外部ファイルにアクセスする	163
概要: SAS コマンドを使用し、外部ファイルにアクセスする	163
FILE コマンドの使用	164
INCLUDE コマンドの使用	164
GSUBMIT コマンドの使用	165
高度な外部 I/O の操作法	165
概要: 高度な外部 I/O の操作法	165
レコード形式の変更	165
外部ファイルへのデータの追加	166
ドライブマッピングの設定	166
各国語文字を含む外部ファイルの読み込み	167
通信ポートからデータを読み込む	168
概要: 通信ポートからデータを読み込む	168
通信ポートのタイムアウト	168
通信ポートのタイムアウト関連のオプション	169

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せんが、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。

移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の**情報**については、SAS のカスタマーサポートセンターウェブページを参照してください。<http://support.sas.com>

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](#)”(465 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 ステートメントセクション](#)で記述されています。

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、外部ファイル参照は Windows の標準的なファイル指定であるとみなされません。Windows の標準的なファイル指定でない場合、ファイル参照が FILENAME ステ

ートメント、FILENAME 関数、環境変数で定義されたファイル参照名であるかどうかを確認します。ファイル参照がこれらのファイル参照名でない場合、ファイル参照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いずれにも該当しない場合は、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を表示します。

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**を選択して、これ以降のすべての SAS セッションにショートカットを再割り当てすることもできます。

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. ファイル上にカーソルを置き、右マウスボタンをクリックして**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名を入力し、Enter を押すか、または **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。これらのトピックは“[連結ディレクトリへのファイル参照名の割り当て](#)” (155 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](#)” (156 ページ) に記述されています。

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、現在の作業ディレクトリにある、wild で始まり、.dat で終わるすべての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は“[高度な外部 I/O の操作法](#)” (165 ページ) で説明されています。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](#)” (469 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](#)” (130 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。コントロール パネルの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 オプションを使用できます。

```
-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、上記の例と同じである Windows SET コマンドは

```
SET MYVAR=C:\MYDATA\TEST.BAT
```

です。また、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を使用することで、SET コマンドを定義することもできます。

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。MSDOS ウィンドウで環境変数を定義した後で、[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 月の地域別売り上げデータのすべてがディレクトリ C:\SAS\MYDATA に格納されている場合、次の FILENAME ステートメントを実行して、ファイル参照名 JAN をこのディレクトリに割り当てます。

```
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、次のプログラムがファイル `C:\PROGRAMS\TESTPGM.SAS` を SAS にサブミットします。

```
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:

```
%include test ("pgm1");
```

を読み込んでサブミットします。

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 (154 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では作業ディレクトリ\PROGRAMS\PGM1.SAS:

```
%include programs (pgm1);
```

に格納されたプログラムがサブミットされます。

次の例では FILE コマンドを使用して、アクティブなウィンドウの内容を作業ディレクトリ\PROGRAMS\TESTPGM.DAT:

```
file programs (testpgm);
```

へと保存します。

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する必要があります。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” (158 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 SAS コマンドでの外部ファイル参照が、次のとおりであるとします。

```
progs (member1)
```

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 \(154 ページ\)](#) 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する必要があります。使用されるプロトコルは、読み込み、書き込み、更新のど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](#)” (158 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](#)” (155 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
```

説明

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 の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名を表示するには、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、編集 ⇨ **削除**を選択します。
3. メッセージボックスで **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

概要: 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";
```

オペレーティングシステムの予約済み物理名の使用

予約済みの名前を、引用符で囲まれた物理ファイル名として使用できます。オペレーティングシステムの予約済み物理名を使用すると、通信ポート(COM1 など)から直接データを読み込むなど、さまざまな作業を行えます。次の表に、物理名とそのデバイスタイプキーワードを示します。

表 5.2 Windows の予約済み物理名

物理名	デバイスタイプ	使用
COM1-COM9	COMMPORT	通信ポートからの読み込み/書き込み。
NUL	DUMMY	データの破棄。この名前は、テストを行うときに便利です。

オペレーティングシステムの物理名を指定する際、コロンの使用は任意です。たとえば、COM1:または COM1 と指定できます。詳細については、Windows のドキュメントを参照してください。

次に、シリアル(RS-232C ポート)を介してデータを転送する外部デバイスまたはアプリケーションのデータを取得する方法を示します。

```
options noxwait xsync;
data _null_;
if symget("sysscpl") = "XP_PRO" then
rc = system("mode COM1:9600,n,8,1,xon=on");
stop;
run;
```

```
filename commdata commport "COM1:";
```

```
data fruit;
```

```

keep num type;
infile commdata unbuffered;
file commdata;
put "ready";
input totrecs records $;
if totrecs = . or records ne "RECORDS" then
do;
file log;
put "ERROR: Unable to determine
number of records to read.";
stop;
end;
do i = 1 to totrecs;
input num type $;
output;
put "NEXT";
end;
stop;
run;

```

この例では、FILENAME ステートメントで、デバイスタイプキーワード COMMPORT を使用しています。デバイスのアクセスプロトコルはファイルのアクセスプロトコルと多少異なるため、FILENAME ステートメントでは、必ず適切なデバイスタイプキーワードを予約済み物理名と組み合わせて使用してください。デバイスタイプキーワードを使用しないと、デバイスではなくファイルのアクセスプロトコルがデフォルトに設定されます。

FILENAME ステートメントで使用可能なデバイスタイプキーワードの詳細については、“Windows 版に固有の SAS ステートメント” (465 ページ) を参照してください。“通信ポートからデータを読み込む” (168 ページ) では通信ポートデバイスを使用するのアクセスプロトコルを説明します。

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 (154 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

概要: 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](#)” (165 ページ) を参照してください。

- “FILE ステートメントの使用” (161 ページ)
- “INFILE ステートメントの使用” (162 ページ)
- “%INCLUDE ステートメントの使用” (163 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つき 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。

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。

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は、“高度な外部 I/O の操作法” (165 ページ) で説明されています。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” (467 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、256 文字です。保存するデータのレコードが、256 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” (467 ページ) の LRECL=を参照してください。

デフォルトの 256 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は“高度な外部 I/O の操作法” (165 ページ) で説明されています。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” (478 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、256 文字です。読み込むデータのレコードが 256 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](#)” (478 ページ) の LRECL=を参照してください。

デフォルトの 256 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스팅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は次で説明されています。“[高度な外部 I/O の操作法](#)” (165 ページ) %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](#)” (476 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、256 文字です。読み込むプログラムのレコードが 256 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](#)” (476 ページ) の LRECL=を参照してください。

デフォルトの 256 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

概要: 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 \(154 ページ\)](#) 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は表示されません。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” (84 ページ) 関連情報: “プログラムエディタの使用” (110 ページ)

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は次に説明されています。“高度な外部 I/O の操作法” (165 ページ) FILE コマンドの完全な構文については、“FILE コマンド: Windows” (362 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);
```

この他に、ファイルを SAS セッションにコピーするには、開くダイアログボックスを使用する方法もあります。開くダイアログボックスでは、ファイルをコピーするほかに、コピーするプログラムを起動するなどの選択が可能です。ファイルをエディタ最も柔軟な方

法です。詳細については、次のファイルを開くを参照してください。“ファイルの保存” (84 ページ) 関連情報: “プログラムエディタの使用” (110 ページ)

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” (165 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” (366 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

概要: 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” (165 ページ)
- “外部ファイルへのデータの追加” (166 ページ)
- “ドライブマッピングの設定” (166 ページ)
- “各国語文字を含む外部ファイルの読み込み” (167 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、データを読み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 関数セクション(*SAS 関数と CALL ルーチン: リファレンス*)と PRINTTO プロシジャ(*Base SAS プロシジャガイド*)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したは更新を行うと、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 \(167 ページ\)](#) の情報と似ていません。

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=256
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。SAS 9.2 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 (WordPerfect 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。

通信ポートからデータを読み込む

概要: 通信ポートからデータを読み込む

マシンの通信(シリアル)ポートからデータを直接読み込むことができます。シリアル通信のパラメータを設定するには、Windows のコントロールパネルにあるポート構成ツールを使用して通信ポートを設定します。指定する通信パラメータは、各データ収集デバイスに固有です。

SAS の起動後に、次の例のように、FILENAME ステートメントをサブミットして、ファイル参照名を通信ポートに関連付けます。

```
filename test commport "com1:";
```

この FILENAME ステートメントは、ファイル参照名 TEST を定義し、COMMPORT デバイスタイプキーワードを使用して通信ポートを使用することを指定し、COM1:を指定しています。COM1:は予約済みの物理名です。

次に、ファイル参照名 TEST を使用して、COM1:のデータを SAS データセットに読み込みます。次の DATA ステップは、ファイルの終端を検出するまで、データを一度に 1 バイトずつ読み込みます(ファイルの終端の 16 進値は '1a'x です)。

```
data acquire;
infile test lrecl=1 recfm=f unbuffered;
input i $;
/* Read until you find an end-of-file. */
if i='1a'x then stop;
run;
```

通信ポートには、何回でもアクセスできますただし、何回も読み込めますが、一度にポートに書き込めるユーザーは 1 人に限定されます。

データ収集アプリケーションで役立つ関数に、SLEEP と WAKEUP の 2 つがあります。これらの関数を使用すると、プログラムの呼び出し時期を制御できます。たとえば、WAKEUP 関数を使用して、午前 2:00 時ちょうどにプログラムを開始することができます。これらの 2 つの関数に関する詳細については、“[SLEEP 関数 Windows](#)” (431 ページ) および “[WAKEUP 関数 Windows](#)” (433 ページ) を参照してください。

通信ポートのタイムアウト

通信ポートからの読み込み中にタイムアウトになると、デフォルトでは、ファイルの終端(EOF)がプログラムに返されます。COMTIMEOUT=オプションを使用すると、通信ポートのタイムアウト処理方法を指定できます。COMTIMEOUT=オプションは、FILENAME ステートメントで有効です。また、FILENAME ステートメントでは、COMMPORT デバイスタイプキーワードと併用する必要があります。

COMTIMEOUT=オプションには、次の値を指定できます。

EOF

タイムアウトになると、ファイルの終端文字が返されます。この動作はデフォルトです。この結果、現在の DATA ステップは終了します。

WAIT

データを受け取るまで待機するように通信ポートに指示します。つまり、この値を指定すると、タイムアウトが無効になります。この場合は、通信ポートにデータが送ら

れるまで、DATA ステップにはレコードが返されません。これによって、プログラムがループに陥る可能性があるため、この値を使用するときには注意が必要です。

ZERO

受信データがない場合は待機しません。

次の例の FILENAME ステートメントは、タイムアウトになった場合、長さが 0 バイトのレコードをプログラムに返すように指定しています。

```
filename test commport "com1" comtimeout=eof;
data test;
infile test length=linelen recfm=F eof=eof;
input @;
eof: if linelen ne 0 then input value;
else put 'Timeout reading from COM1:>';
run;
```

通信ポートのタイムアウト関連のオプション

通信ポートのタイムアウトに関するオプションには、次のようなオプションがあります。

RMULTI

読み込み操作のタイムアウト期間の合計を計算するために使用する乗数を、1000 分の 1 秒単位で指定します。この値は、読み込み操作ごとに、要求された読み込みバイト数と乗算されます。

RCONST

読み込み操作のタイムアウト期間の合計を計算するために使用する定数を、1000 分の 1 秒単位で指定します。この値は、読み込み操作ごとに、RMULTI と要求されたバイト数との積に加算されます。

WMULTI

書き込み操作のタイムアウト期間の合計を計算するために使用する乗数を、1000 分の 1 秒単位で指定します。この値は、書き込み操作ごとに、書き込みバイト数と乗算されます。

WCONST

書き込み操作のタイムアウト期間の合計を計算するために使用する定数を、1000 分の 1 秒単位で指定します。この値は、書き込み操作ごとに、WMULTI 数と書き込みバイト数との積に加算されます。

RINT

通信回線を介して、文字と文字を受信するまでの最大経過時間を、1000 分の 1 秒単位で指定します。

6 章

Windows 版での SAS 出力の管理

印刷	171
Windows 環境での SAS の印刷について	171
SAS ウィンドウからの印刷	172
印刷プレビュー	180
SAS 印刷フォームの使用	181
SAS コマンドを使用した印刷	183
プリンタへの DATA ステップの出力の送信	183
プリント出力のファイルへの送信	184
バッチモード印刷	184
デフォルトプリンタの詳細	185
印刷ジョブの取り消し	185
プロシジャ出力の Web ブラウザへの表示	186
プロシジャ出力の Web ブラウザへの表示について	186
HTML 出力のプリファレンスの構成	186
結果ビューアウィンドウの使用	187
プロシジャ出力と SAS ログをファイルに送る	188
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	188
名前を付けて保存ダイアログボックスの使用	188
PRINTTO プロシジャの使用	189
SAS システムオプションの使用	190
SAS ログ機能を使用したログメッセージの書き込み	190
グラフィックの作成	191
ディスプレイ上でのグラフィックの作成	191
グラフィックの印刷	192
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	195
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	197
補足	201

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使用法については、[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](#)” (611 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](#)” (191 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。

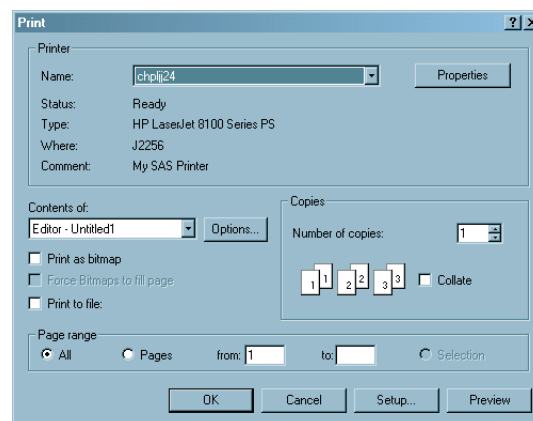
印刷オプションの設定

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、**ファイル** ⇒ **印刷** を選択します。

画面 6.1 印刷ダイアログボックス

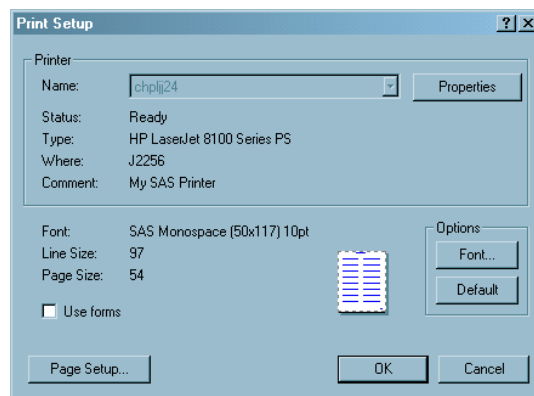


印刷設定ダイアログボックスでは、次の印刷オプションを設定します。

- 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、**ファイル** ⇒ **ページ設定**を選択します。

画面 6.2 印刷設定ダイアログボックス

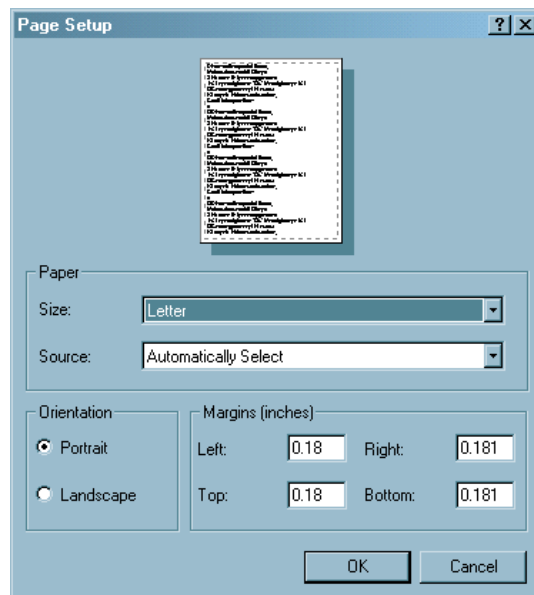


ページ設定ダイアログボックスでは、次の印刷オプションを設定します。

- 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、**ファイル** ⇒ **ページ設定**を選択します。

画面 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、**ファイル** ⇒ **印刷** ⇒ **プロパティ**を選択します。

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、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。

SAS の実行に使用する Windows 動作環境によっては、印刷ダイアログボックスに表示される項目が異なります。

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り決定され、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 P RTPERSISTDEFAULT 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 P RTPERSISTDEFAULT システムオプションを指定します。

SAS 起動時に SYSPRINT と P RTPERSISTDEFAULT 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に関する詳細については、“[SYSPRINT システム](#)

オプション: Windows” (605 ページ) および “PRTERSISTDEFAULT システムオプション: Windows” (576 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. 新しい値ボックスに、Windows のプリンタフォルダに表示されるプリンタ名を入力して、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。

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行サイズとページサイズ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. ファイル ⇒ 印刷設定 ⇒ フォントを選択します。
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 を右クリックして、ポップアップメニューから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。値はカッコ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” (606 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。

ページ設定ダイアログボックスのオプションと関連するシステムオプションを次の表に示します。

表 6.2 ページ設定の設定項目

ページ設定オプション	解説	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。 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。	左、右、上、下 のフィールドにインチ数を入力します。	LEFTMARGIN、 RIGHTMARGIN、 TOPMARGIN、 BOTTOMMARGIN
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て、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て、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。

特定の用紙サイズをサポートは、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。これ以外のサイズがサポートされている場合もあります。

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

LETTERS SMALL

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、14.875X11 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。印刷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
<ul style="list-style-type: none"> プリンタ 	<ul style="list-style-type: none"> 用紙サイズ 給紙方法 印刷の向き 余白設定 	<ul style="list-style-type: none"> フォント フォントスタイル 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で使用できます。このダイアログボックスを開くには、**印刷ダイアログボックスのオプション**をクリックします。

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を選択します。
SAS ウィンドウの印刷	ビットマップで印刷を選択します。 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を選択します。 内容ボックスで、全画面 (ビットマップ)を選択します。

ビットマップ形式	印刷ダイアログボックスの選択
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て、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、すべてを選択します。
- 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。

印刷プレビュー

ウィンドウのプレビュー法

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. ファイル ⇨ 印刷プレビューを選択します。

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 `d1gpptpreview` 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	次へまたは前へを使用します。
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	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](#)” (661 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
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. **ファイル** ⇒ **印刷設定**を選択します。
2. **フォームを使用する**チェックボックスを選択します。

印刷設定 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” (577 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、**フォームの使用**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;
```

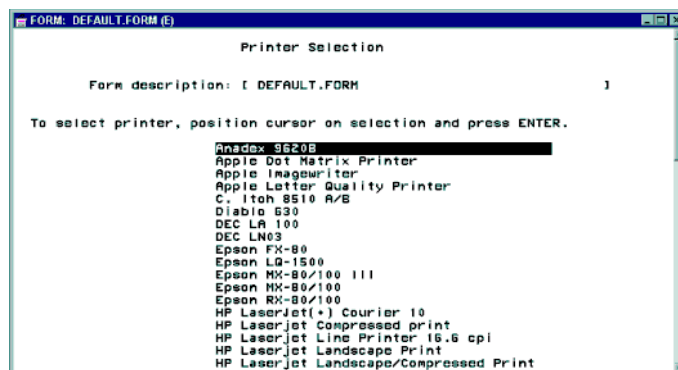
FSFORM コマンドの詳細については、SAS ヘルプとドキュメントを参照してください。

FORMFSFORM コマンドの発行後に表示される 1 番目のフレームは、Windows 環境で使用できるプリンタが示されるプリンタの選択フレームです。画面 6.4 (183 ページ) には、このフレームのデフォルトの情報が示されています。

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 を押します。
2. ツールメニューから、次のスクリーンと前のスクリーンを選択して、フレーム間を移動します。

画面 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、[画面 6.4 \(183 ページ\)](#) 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](#)” (605 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

プリント出力をファイルに送信するには、印刷ダイアログボックスのファイルへ出力チェックボックスを選択して、出力ファイル名を指定します。このアクションは、保存操作とは異なります。プリンタファイルには、印刷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。あくまでも、プリンタへの送信のみを目的としています。

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。この場合は、Windows のプリンタのプロパティダイアログボックスのポートタブのプリンタへは送信されません。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](#)” (605 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。

- **ファイル** ⇒ **印刷設定**を選択
- コマンドバーに DLGPRTSETUP を入力

SAS 起動時、PRTPERERSISTDEFAULT 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](#)” (174 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、**印刷の中止**ダイアログボックスの**キャンセル**をクリックして、印刷ジョブを取り消しできます。印刷の中止のダイアログボックスは、SAS がプリントファイルを出力先にスプーリングしている間のみ表示されます。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の中止ダイアログボックスが閉じてしまいます。

印刷の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	ファイル
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” (575 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 (IE) をインストールしている場合は、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、**プリファレンス**ダイアログボックスの**結果**タブと**Web** タブの各項目を設定します。

プリファレンスダイアログボックスを開くには、**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の**結果**タブを設定するには、次の操作を行います。

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**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、**フォルダ**テキストボックスにパスを入力するか、**参照**をクリックしてフォルダを指定します。

4. **スタイル**ボックス内をクリックして、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。“**TEMPLATE** プロシジャ: スタイルテンプレートの作成 プロシジャ” (*SAS Output Delivery System: ユーザーガイド*) を使用したスタイルの定義については、-->*SAS Output Delivery System: ユーザーガイド*を参照してください。
5. 結果をすぐに表示するには、**すぐに結果ビューアを表示する**を選択します。
6. **ブラウザ**の選択では、結果ビューアに HTML 出力を表示する場合は**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**を選択する必要があります。Microsoft Internet Explorer をインストールしていない場合は、**標準システムブラウザ**を選択する必要があります。

プリファレンスダイアログボックスの Web タブを設定するには、次の操作を行います。

- 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。
 - Windows レジストリに登録されたデフォルトブラウザに HTML 出力を表示するには、**標準システムブラウザを使用する**ラジオボタンを選択します。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。
 - デフォルトブラウザ以外の Web ブラウザに HTML を表示するには、**その他**ラジオボタンを選択します。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

プリファレンスダイアログボックスの**結果**タブで**内部ブラウザ**を選択した場合、ODS 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

フォントサイズを変更するには、二重になった A のボタンを選択します。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、**結果**ウィンドウからファイルを選択して表示できます。生成と同時に表示することもできます。そのためには、**プリファレンス**ダイアログボックスの**結果**タブで対応するオプションを選択します。

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選択します。
3. **ファイルを開く**ダイアログボックスで HTML ファイルを指定します。
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. **上書き保存**ツールバーボタンを選択します。
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。

プロシジャ出力と SAS ログをファイルに送る

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

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](#)” (150 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 コマンドを入力することによって呼び出します。
- 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”(84 ページ)を参照するか、プログラムエディタからの“ファイルの保存”(117 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](#)” (455 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。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](#)” (517 ページ) および “[ALTLOG システムオプション: Windows](#)” (516 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、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、プロシジャ出力と SAS ログの出力先が記録されます。

SAS ログ機能を使用したログメッセージの書き込み

SAS 9.3 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、形式を含む)
 - 診断レベルと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

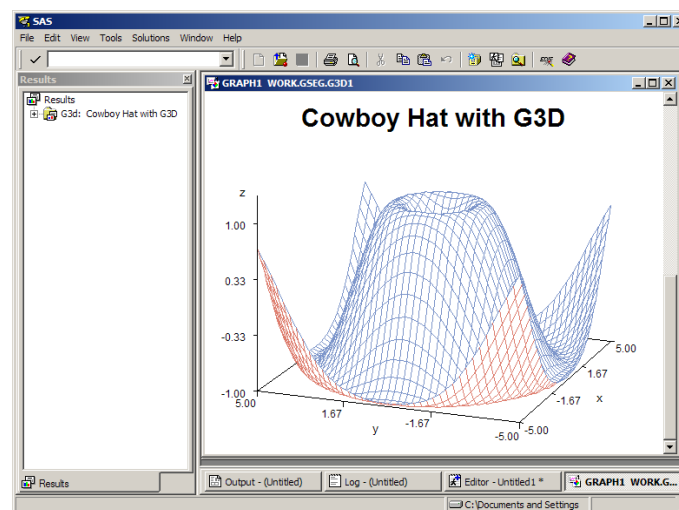
ほとんどの場合、SAS/GRAPH プロシジャを実行すると、出力はモニタに自動的に表示されます。SAS/GRAPH デバイスドライバを指定する必要はありません。グラフィック表示に関する情報は Windows の情報ファイルに格納されており、対話型の SAS セッション中に SAS によって自動的に使用されます。

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。

画面 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

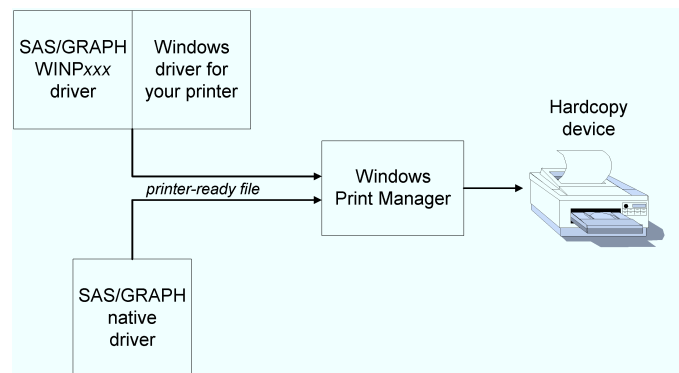
グラフィック印刷の概要

SAS/GRAPH の出力は、次の 2 通りの方法で印刷できます。

- “SAS/GRAPH 汎用(WINPxxx)ドライバの使用” (193 ページ) 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” (193 ページ) して、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。グラフィックストリームは出力デバイスに直接送信されます。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 6.1 (192 ページ) 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について説明します。

図 6.1 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” (194 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” (193 ページ)
- “SAS/GRAPH ネイティブプリンタドライバの使用” (193 ページ)
- “GRAPH ウィンドウからの印刷とプレビュー” (194 ページ)

- “SAS/GRAPH ネイティブドライバと WINPxxx ドライバの選択” (194 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. ファイルプルダウン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。

```
options sysprint='HP LaserJet III';
```

ただし、Windows で構成されているプリンタドライバ以外には、割り当てることができません。

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;
```

WINPxxx 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

```
goptions rotate=landscape
```

を指定するとします。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は、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 gsfname=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、WINPxxx 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 WINPxxx 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、WINPxxx 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](#)” (651 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

概要: 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し、SAS/GRAPH に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 (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](#)” (196 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](#)” (196 ページ) を参照してください。

ベクトルグラフィックをインポートするには、GIMPORT プロシジャを使用してコンピュータグラフィックメタファイル(CGM)ファイルをインポートします。インポートしたファイルは、GRSEG カatalogのエントリとして格納されます。この方法では、インポートしたグラフ内に個々のグラフィックオブジェクトが保持されます。これに対して、他の方法では、インポートしたグラフィックは編集できない単一ビットマップとして処理されます。

GIMPORT プロシジャの詳細については、*SAS/GRAPH: Reference* 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” (196 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” (196 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、編集メニューから貼り付けを選択します。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の形式	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 形式	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
PNG (Portable Network Graphics)	PNG
PCD (Photo CD image)	PCD
PBM (Portable Bitmap)	PBM

グラフィックファイルの形式	ファイル拡張子
XBM (X 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、ファイルメニューから**イメージのインポート**を選択します。
2. **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 で使用でき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 を指定してファ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

概要: 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](#)” (197 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 Windows アプリケーションへのグラフィックの貼り付け](#)” (198 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](#)” (199 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](#)” (201 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 SAS ヘルプとドキュメントを参照してください。

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	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
ESPI (Encapsulated PostScript Interchange)	PS
PNG (Portable Network Graphics)	PNG
PBM (Portable Bitmap)	PBM

GRAPH ウィンドウのグラフをエクスポートするには、次の操作を行います。

1. 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**イメージのエクスポート**を選択します。
2. **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。

グラフィックをイメージエディタから SAS IMAGE カタログエントリにエクスポートする方法は、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 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 を押すまたは編集メニューから**コピー**を選択します。

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。通常はその Windows アプリケーションの**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;
```

CGMxxxx 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。

filenameCGM 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 (199 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 9.3 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お問い合わせください。

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)

パッケージ	推奨ドライバ
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) CGMFRGA* (dgdevice カタログでは無効) 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) CGMGILFG* (dgdevice カタログでは無効) CGMMILFM* (dgdevice カタログでは無効)
Lotus Ami Pro 3.0	CGMAM3C* (dgdevice カタログでは無効)
DOS 用 Lotus Freelance	CGMFLALJ* (dgdevice カタログでは無効) CGMFLAPL* (dgdevice カタログでは無効) CGMFLAPT* (dgdevice カタログでは無効)
Window 2.0 用 Lotus Freelance	CGMFL2C* (dgdevice カタログでは無効)
Window 1.0 用 Lotus Freelance	CGMFLWA* (dgdevice カタログでは無効)
Lotus 1-2-3 4.0	CGM123C* (dgdevice カタログでは無効)
Lotus Office 97	CGMLT97L CGMLT97P
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、 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) CGMWPCAP* (dgdevice カタログでは無効) CGMWPGA (dgdevice カタログでは無効) CGMWPGAP* (dgdevice カタログでは無効) CGMWPCA (dgdevice カタログでは無効) CGMWPCAP* (dgdevice カタログでは無効)

パッケージ	推奨ドライバ
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) CGMWPGA* (dgdevice カタログでは無効) CGMWPCA* (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、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 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](#)” (651 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	203
ハードウェアの注意点の概要	203
プロセッサの速度	204
メモリ	204
I/O に必要なディスクスペース	205
グラフィックアダプタ	205
パフォーマンスを最適化する Windows の機能	206
SAS の応答性の制御	206
複数プロセッサに対応した I/O 拡張	207
メモリベースのライブラリ	208
パフォーマンスを最適化する SAS の機能	213
ネットワークのパフォーマンスに関する注意点	214
高度なパフォーマンス調整メソッド	214
概要: 高度なパフォーマンス調整メソッド	214
SORT プロシジャのパフォーマンス向上	214
データセットのサイズの計算	216
対話型処理の効率の向上	218

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 に使用可能なディスク容量
- グラフィックアダプタ

この中には、構成や SAS の実行方法によってはあてはまらない要素もあります。詳細は、システム管理者に問い合わせてください。

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。

SAS を実行するプロセッサの必要条件は、動作環境によって異なります。

- 32 ビット環境の場合は、Intel (または Intel 互換) の SSE2 命令セットをサポートする Pentium 4 クラスのプロセッサが搭載された PC が必要です。
- 64 ビット環境の場合は、Intel64 または AMD64 プロセッサが必要です。
- Itanium 64 ビット環境の場合は、Intel Itanium II 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。使用可能なメモリ量が大きいシステムでは、スワッピングをせずに大量のデータを処理することができます。スワッピングでは、メモリにロードできないデータを格納するために、ハードドライブ上の一時スペースを使用します。しかし一時ファイルの処理などのシステム操作は、ハードドライブよりメモリのほうが高速に処理できます。したがって、使用可能なメモリ量が大きいほど、この種の操作のためにハードドライブにアクセスする回数は減ります。

必要最小メモリ量は、動作環境によって異なります。

表 7.1 32 ビット環境で必要なメモリ量

動作環境	必要なメモリ量
Windows XP Professional	<ul style="list-style-type: none"> • 512MB 以上(パフォーマンス向上のため、より多くのメモリの搭載を推奨します。) • 512MB 以上のスワップファイルスペース
Windows Vista, Server 2003, Server 2008 R1, Windows 7	<ul style="list-style-type: none"> • 2GB 以上(パフォーマンス向上のため、より多くのメモリの搭載を推奨します。) • 2GB 以上のスワップファイルスペース

表 7.2 64 ビット環境で必要なメモリ量

動作環境	必要なメモリ量
Windows Server 2003, 64 ビット版 Vista, Windows 7, Server 2003, 2008, 2008 R2	<ul style="list-style-type: none"> • 1GB 以上(パフォーマンス向上のため、より多くのメモリの搭載を推奨します。) • 1GB 以上のスワップファイルスペース

I/O に必要なディスクスペース

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の 3 つです。

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。一般に、SCSI コントローラおよび EIDE コントローラでは、バススピードが高速化します。高速の SCSI バスまたは EIDE バスを適切な SCSI ドライブまたは EIDE ドライブとともに使用すると、他の種類のコントローラとハードドライブを使用するよりも、パフォーマンスが向上します。現在、最も I/O のパフォーマンスが高いのは、SCSI コントローラです。

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 管理者が、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すための 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。

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。

グラフィックアダプタ

SAS の機能には大量のグラフィックメモリを使用するものもあるため、どの種類のグラフィックアダプタを使用するかによって、パフォーマンスは大きく異なります。同じ容量のグラフィックメモリを使用した場合、一般に PCI アダプタより AGP アダプタのほうが高速です。ただし使用可能なアダプタの種類は、マザーボードによって異なります。

アダプタで使用可能なメモリ量は、グラフィックがレンダリングされる速度に影響します。通常はメモリが大きいほど、パフォーマンスは向上します。

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。詳細は、“[パフォーマンスツール](#)” (234 ページ) を参照してください。

Windows Vista でのアプリケーションパフォーマンスの最適化

Windows Vista および Server 2008 R1 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. システムとメンテナンスをクリックします。
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. 詳細設定タブを選択します。
6. パフォーマンスボックスの設定をクリックして、詳細設定タブを選択します。
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。

Windows 7 および Server 2008 R2 でのアプリケーションパフォーマンスの最適化

Windows 7 および Server 2008 R2 上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選択します。
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. **詳細設定**タブを選択します。
6. パフォーマンスボックスの**設定**をクリックして、**詳細設定**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。

Windows XP でのアプリケーションパフォーマンスの最適化

Windows XP 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**をクリックします。
3. **詳細設定**タブを選択します。
4. パフォーマンスボックスの**設定**をクリックして、**詳細設定**タブを選択します。
5. 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。
6. 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。
7. **OK** をクリックします。

Windows Server 2003 でのアプリケーションパフォーマンスの最適化

Windows Server 2003 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**をクリックします。
3. **詳細設定**タブを選択します。
4. パフォーマンスボックスの**設定**をクリックします。
5. 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、プロセッサのスケジュールボックスの**プログラム**を選択します。
6. 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、プロセッサのスケジュールボックスの**バックグラウンド サービス**を選択します。
7. **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 ज्या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。
- SMP ボックス(RAID、RAM)内のサポートハードウェアにより、通常、どのアプリケーションのパフォーマンスも向上します。

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。

32 ビット版の Windows 動作環境では、動作環境に 2 ギガバイト(GB)の物理メモリを使用します。残りの 2 GB の物理メモリはアプリケーションで使用できるようにしておきます。PC またはサーバーのメモリが 4 GB を超えている場合は、4 GB を超えるメモリが拡張メモリとして定義されるため、すべてアプリケーションで使用できます。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、拡張メモリはサポートされません。拡張メモリがサポート、インストールされていない動作環境では、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。32 ビット環境では、基本メモリは 4 GB 以下のメモリとして定義されます。64 ビット環境では、すべてのメモリが基本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と、ディスクへの入力およびディスクからの出力が削減されるため、SAS のパフォーマンスが向上します。次に、メモリベースのライブラリの用途を示します。

- WORK ライブラリの記憶域
- 高容量の I/O 力がある SAS ライブラリの処理
- 巨大な SAS ライブラリのキャッシュ

拡張メモリは、32 ビット版 Windows Server 2003 ファミリーおよび 32 ビット版 Server 2008 をサポートするのに使われます。従来のメモリは、その他すべてのサポートされた環境での MEMLIB および MEMCACHE システムオプションをサポートするのに使われます。

これらのサーバーでは、メモリベースのライブラリに同時にアクセスして複数の処理を行えます。

動作環境の設定が終了したら、MEMLIB システムオプションおよび MEMCACHE システムオプション、LIBNAME ステートメントの MEMLIB オプションを使用して、メモリベースのライブラリにアクセスします。

Windows 2003 Server 以降でメモリベースのライブラリを使用する場合の設定条件

Windows 2003 のメモリマネージャでは、Windows 2003 Enterprise Server の 8 GB までの物理 RAM にアクセスできます。Windows 2003 Datacenter Server の場合は、32 GB までの物理 RAM にアクセスできます。

Windows 2003 Server 以降で拡張メモリを使用するには、動作環境が次の必要条件を満たす必要があります。

- Intel または Intel 互換の 32 ビットプロセッサ(Pentium Pro 以降)。

- 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。
- パラメータ/PAE が、boot.ini ファイルのサーバー起動行に追加されています。
- Windows 版 SAS 9.3

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。次を参照してください。[“MEMMAXSZ システムオプション: Windows” \(559 ページ\)](#)
- 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。次を参照してください。[“MEMBLKSZ システムオプション: Windows” \(557 ページ\)](#)

Windows 版 SAS 9.3 では、利用可能なメモリを使用して MEMLIB オプションと MEMCACHE オプションをサポートします。

注意:

システムメモリの上限を超えると、システムエラーが発生します。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。

Windows XP Professional でメモリベースのライブラリを使用する場合の設定条件

Windows XP Professional の物理 RAM は、4 GB に制限されています。メモリベースのライブラリは、物理メモリが 4 GB を超えるサーバー環境でのパフォーマンスを最適化します。この動作環境では、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。Windows XP Professional 環境、Windows Vista 環境、Windows 7 環境にこれらのオプションを指定すると、パフォーマンスが向上する場合もありますが、向上しない場合もあります。

Windows XP Professional、Windows Vista、Windows 7 でメモリベースのライブラリを使用するには、動作環境が次の必要条件を満たす必要があります。

- Intel または Intel 互換の 32 ビットプロセッサ(Pentium Pro 以降)
- 4 GB の RAM
- Windows 版 SAS 9.3

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。次を参照してください。[“MEMMAXSZ システムオプション: Windows” \(559 ページ\)](#)
- 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。次を参照してください。[“MEMBLKSZ システムオプション: Windows” \(557 ページ\)](#)

Windows 版 SAS 9.3 では、利用可能なメモリを使用して MEMLIB オプションと MEMCACHE オプションをサポートします。

注意:

システムメモリの上限を超えると、システムエラーが発生します。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。

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:

システムメモリの上限を超えると、システムエラーが発生します。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。

- Intel または Intel 互換の 64 ビットプロセッサ(Itanium 以降)
- 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 9.3

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アベースのライブラリに使用するように設定します。次を参照してください。[“MEMMAXSZ システムオプション: Windows” \(559 ページ\)](#)
- 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)をメモリアベースのライブラリに使用するように設定します。次を参照してください。[“MEMBLKSZ システムオプション: Windows” \(557 ページ\)](#)

ローカルセキュリティの設定

Windows XP Professional 以降の 32 ビット環境では、メモリ内のページのロックを設定して、SAS を実行する個々のユーザーが拡張メモリにアクセスできるようにする必要があります。正しい権限を持たない場合は、ログに警告メッセージが表示されます。

システム管理者は、スタートメニューからローカルセキュリティを設定できます。この値を設定するには、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. **管理ツール**をダブルクリックします。
3. **ローカル セキュリティ ポリシー**をダブルクリックします。[ローカル セキュリティ設定]ウィンドウが開きます。
4. **ローカル ポリシー**をダブルクリックします。
5. **ユーザー権利の割り当て**をダブルクリックします。
6. **メモリ内のページのロック**をダブルクリックします。**ローカル セキュリティ ポリシーの設定**ウィンドウが開きます。
7. **追加**をクリックします。**ユーザーまたはグループの選択**ウィンドウが開きます。
8. 拡張メモリにアクセスして SAS を実行する必要があるユーザーグループのユーザー ID または名前を選択します。
9. **ユーザーまたはグループの選択**ウィンドウで **OK** をクリックします。
10. **ローカル セキュリティ ポリシー**ウィンドウで **OK** をクリックします。

注: ローカル セキュリティ ポリシーウィンドウには、ローカル ポリシーの設定チェックボックスと有効なポリシーの設定チェックボックスの 2 つのチェックボックスが表示されることがあります。追加したユーザーの有効なポリシーの設定チェックボックスが選択されていない場合は、新しいセキュリティ設定が有効にするためにコンピュータを再起動する必要があります。

メモリアベースのライブラリとして SAS ライブラリを処理する

メモリア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。

メモリアベースの 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ア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ア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ア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;
```

詳細は、次を参照してください。“MEMLIB システムオプション: Windows” (559 ページ) および “LIBNAME ステートメント: Windows” (481 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” (558 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” (213 ページ) および “SGIO システムオプション: Windows” (588 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 9.3 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。ファイル入出力を減らす方法には次のものがあります。
 - 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みます。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](#)” (588 ページ)

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。

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” \(214 ページ\)](#)
- [“データセットのサイズの計算” \(216 ページ\)](#)
- [“対話型処理の効率の向上” \(218 ページ\)](#)

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](#)” (457 ページ) を参照してください。

- “SORTSIZE オプション” (215 ページ)
- “TAGSORT オプション” (215 ページ)
- “ソートされるファイルの場所の選択” (215 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を使用されます。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、256MB (メガバイト) 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

指定したデータセットが物理的にソートされる場所は、データセット名の参照方法や、PROC SORT ステートメントで OUT=オプションを使用するかどうかによって異なります。ソートに必要なディスク容量が足りない場合、ソートされる場所を把握する必要があります。

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、WORK データライブラリです。UTILLOC オプションに複数の場所が指定されていると、2 番目の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。

PROC SORT ステートメントに OVERWRITE オプションを指定すると、入力データセットはソート済みデータセットに置き換わります。

PROC SORT ステートメントに OVERWRITE オプションを指定しない場合、.sas7butl というファイル拡張子を持つ第 2 のファイルが作成されます。ソートが正常に終了すると、このファイル名は、.sas7bdat ファイル拡張子を含むソート済みファイルのデータセット名に変更されます。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 C:ドライブの MYDATA 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、最初のソートが SASWORK サブディレクトリで行われ、2 番目のソートが F:ドライブの JANDATA 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*内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. 1 オブザベーションと必要な数の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 = $1 + (\text{floor}(\text{オブジェクトの数} / \text{ページあたりの最大オブジェクト}))$
- サイズ = $1024 + (\text{データセットページサイズ} * \text{データページの数})$

(floor は端数を切り捨てて、最も近い整数にする機能を表します。)

[アウトプット 7.1 \(217 ページ\)](#) 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 = $1 + (\text{floor}(1/101))$
- 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場合があります。

2 部

SAS を他の Windows アプリケーションとともに使用する

8 章	Windows 版での Lotus Notes を使用した SAS データの配布	221
9 章	Windows 版 SAS で Windows システムツールの使用	229
10 章	Windows 版 SAS/AF ソフトウェアでの OLE の使用	251
11 章	Windows 版で OLE を使用し、他のアプリケーションから SAS を制御する	277
12 章	Windows 版で動的データ交換を使用する	285
13 章	Windows 版で名前付きパイプと名前の付いていないパイプを使用する ..	297
14 章	Windows 版 SAS から外部 DLL にアクセスする	309
15 章	Windows 版 SAS/AF のプログラミングで特に注意すべき点	331

8 章

Windows 版での Lotus Notes を使用した SAS データの配布

SAS での Lotus Notes の使用について	221
NOTESDB Engine	221
クライアントの必要条件	221
DATA ステップと SCL コードを使用し、Lotus Notes データベースに入力する ...	222
新しい Notes ドキュメントの作成	222
Notes ドキュメント向けの SAS/GRAPH 出力を用意する	226
NotesSQL ODBC ドライバーを用いて SAS を使用する	227
既存の Notes ドキュメントから情報を取り込む	228

SAS での Lotus Notes の使用について

NOTESDB Engine

SAS では、クライアントユーザーが既存の Notes データベースに新規 Notes ドキュメントを追加できるように、アクセスエンジン NOTESDB を提供しています。このエンジンは、Lotus Notes クライアントバージョン 8.5.2 以降との互換性があります。

NOTESDB アクセスエンジンでは、データベースを新規作成したり、既存の Notes ドキュメントから情報を取り込むことはできません。しかし NotesSQL ODBC Driver を使用すれば、Notes データベースからデータを取り込むことができます。これらの詳細については、“[既存の Notes ドキュメントから情報を取り込む](#)” (228 ページ) を参照してください。

サーバーでパスワードの入力が要求されるため、NOTESDB エンジンをバッチジョブとして使用するプログラムをスケジュールしないでください。

クライアントの必要条件

SAS で Lotus Notes を使用するための必要条件は、次のとおりです。

- Lotus Notes のクライアントバージョンと、Notes の有効なユーザー ID 認証が、NOTESDB エンジンを使用するマシンにインストールされている必要があります。ただし、SAS が Lotus Notes にアクセスするために、Notes を実行する必要はありません。SAS 経由で Notes にアクセスするには、パスワードの入力が求められます。
- Lotus Notes のディレクトリが、システムパスに含まれている必要があります。**システムプロパティダイアログボックスの Advanced タブ**内に PATH システム環境変数を加えることで、システムパスを更新します。

DATA ステップと SCL コードを使用し、Lotus Notes データベースに入力する

新しい Notes ドキュメントの作成

Notes と対話するための SAS ステートメント

Notes データベースと対話する DATA ステップと SCL コードは、次の要素から構成されます。

- NOTESDB デバイスタイプキーワードを含む FILENAME ステートメント
- データディレクティブと Notes データベースに追加するデータを含む PUT ステートメント
- Notes データベースへのデータの配信タイミングを制御する操作ディレクティブを含む PUT ステートメント

Lotus Notes データベースに入力する構文

FILENAME *ファイル参照* NOTESDB;

説明

ファイル参照

有効なファイル参照名です。

NOTESDB

Lotus Notes データベースの使用を示すデバイスタイプキーワード。

DATA ステップでは、使用するデータベースと送信するデータを定義するデータディレクティブを含む PUT ステートメントを使用します。

注: Notes データベースにアクセスするために指定したディレクティブでは大文字と小文字は区別されませんが、これらのディレクティブを使用して指定するフィールドでは大文字と小文字が区別されます。また、1 つの PUT ステートメントに含めることができるディレクティブは 1 つのみです。ディレクティブは、感嘆符で区切り、一重引用符で囲みます。

次のデータディレクティブを使用して、データベースの場所と、データベースに追加するデータを指定します。

!NSF_SERVER! *サーバー名*

アクセスする Notes サーバーを指定します。サーバー名は、Lotus Notes サーバーを表します。このディレクティブを指定しない場合、SAS では使用しているローカルシステムをデータベースのソースとして使用します。このディレクティブを複数指定した場合は、最新の PUT ステートメントで指定されたサーバーが使用されます。

注: SAS 経由で Notes サーバーにアクセスするには、パスワードの入力が求められます。

!NSF_DB! *database-filename*

アクセスする Notes データベースファイルを指定します。ローカルのデータベースにアクセスすると、SAS では Notes データディレクトリにあるデータベースを検索します。見つからない場合は、システムパスを検索します。または、データベースへの完全修飾されたパスを指定することもできます。SAS から Notes データベースにアクセスするには、このディレクティブを含む Notes データベースファイルを指定

する必要があります。このディレクティブを複数指定した場合は、最新の PUT ステートメントで指定されたデータベースが使用されます。

!NSF_FORM! *form-name*

追加されたノートの表示時に、Notes で使用する形式を指定します。このディレクティブを指定しない場合、Notes はデフォルトのデータベースフォームを使用します。このディレクティブを複数指定した場合は、!NSF_FORM!ディレクティブを含む最新の PUT ステートメントで指定されたフォームが使用されます。ディレクティブを使います。

!NSF_ATTACH! *ファイル名*

追加されたノートにファイルを添付します。SAS は、Notes データディレクトリにあるファイルを検索します。見つからない場合は、システムパスを検索します。または、ファイルへの完全修飾されたパスを指定することもできます。添付できるファイルは、!NSF_ATTACH!ディレクティブを含む PUT ステートメント 1 つにつき 1 つです。ディレクティブ 複数のファイルを添付するには、ファイルごとに異なる PUT ステートメント(!NSF_ATTACH!ディレクティブを含む)を使用してください。各ファイルへのディレクティブ

!NSF_FIELD! *フィールド名* *フィールド名*

指定されたフィールド名に値を追加します。SAS は、フィールドの正しいフォーマットを検出し、それに従ってデータをフォーマットします。SAS では、ラインフィードやキャリッジリターンはすべて取り除かれます。Notes でドキュメントを正しく表示するには、これらの制御文字を挿入しないでください。!NSF_FIELD!ディレクティブを含む PUT ステートメントでは、複数の PUT ステートメントに同じフィールド名が指定されている場合は、フィールドの情報が連結されます。ディレクティブと、同じフィールド名はそのフィールドの情報が連結されます。また、ディレクティブを含まない PUT ステートメントは、最後にサブミットされたフィールド名に連結されます。また、!NSF_FIELD!ディレクティブを含む PUT ステートメントがサブミットされていない場合は、ディレクティブを含まない PUT ステートメントは無視されます。ディレクティブは前もってサブミットされています。

次の種類のフィールドを追加することができます。これらのフィールドは編集可能です。

- テキスト
- 数値
- キーワード

Lotus Notes によってフォーマットされるテキストを追加することができます。また、次の構文を使用すると、Windows ビットマップ形式のビットマップを追加することもできます。

!NSF_FIELD! *フィールド名* <*bitmap* *ファイル名*>

Notes データベースを操作するには、次の操作ディレクティブを使用します。

!NSF_ADD!

DATA ステッププログラムから、ただちにドキュメントを Notes データベースに追加します。

!NSF_ABORT!

データストリームの終了時にノートを追加しません。デフォルトでは、ドライバは、FILE ステートメントを使用する SAS プログラムが終了するたびにノートを追加しません。

!NSF_CLR_FIELDS!

!NSF_FIELD!ディレクティブによって指定されたすべてのフィールド値をクリアします。ディレクティブ このディレクティブを!NSF_ADD!と一緒に使用すると、複数のノ

ートを複数のデータベースに追加するために、ループ付きの DATA ステッププログラムを簡単に作成することができます。

```
!NSF_CLR_ATTACHES!
```

!NSF_ATTACH!!ディレクティブによって指定されたすべてのフィールド値をクリアします。ディレクティブこのディレクティブを!NSF_ADD!と一緒に使用すると、複数のノートを複数のデータベースに追加するために、ループ付きの DATA ステッププログラムを簡単に作成することができます。

注: ディレクティブを含まない PUT ステートメントの内容は、最新のフィールド値に関連するデータに連結されます。

Lotus Notes データベースへの入力例

次の例では、Lotus Notes で作成された Business Card Request データベースを使用します。この DATA ステップでは、データベースに新規ドキュメントを作成し、そのすべてのフィールドに値を入力します。

例のコード 8.1 Business Card Request データベースの使用

```
01 filename reqcard NOTESDB;
02 data _null_;
03 file reqcard;
04 put '!NSF_DB! examples\buscard.nsf!';
05 put '!NSF_FIELD!Status! Order!';
06 put '!NSF_FIELD!Quantity! 500!';
07 put '!NSF_FIELD!RequestedBy! Systems!';
08 put '!NSF_FIELD!RequestedBy_CN! Jane Doe!';
09 put '!NSF_FIELD!NameLine_1! Jane Doe!';
10 put '!NSF_FIELD!NameLine_2! Developer!';
11 put '!NSF_FIELD!AddressLine_1! Software R Us!';
12 put '!NSF_FIELD!AddressLine_2! 123 Silicon Lane!';
13 put '!NSF_FIELD!AddressLine_3! Garner, NC 27123!';
14 put '!NSF_FIELD!AddressLine_4! USA!';
15 put '!NSF_FIELD!PhoneLine_1! (910) 777-3232!';
16 run;
```

行 1 は、FILENAME ステートメントを使用して、ファイル参照名を割り当てます。ここでは、通常のファイルのかわりに Notes を参照します。NOTESDB は、Lotus Notes のデバイスタイプです。行 3 は、割り当てたファイル参照名を PUT ステートメントの出力先に設定します。行 4 は、開く Notes データベースを示します。行 5 から 15 は、作成する新規の Notes ドキュメントのフィールドとフィールド値を指定します。Status はフィールド名、Order は特定のドキュメントの Status フィールドの入力値です。行 16 は、これらの SAS ステートメントを実行します。新規の Notes ドキュメントが、Business Card Request データベースに作成されます。

次の例では、SALES データセットのオブザベーションを使用して、qrtsales.nsf データベースに新規ドキュメントを作成し、作成したドキュメントの Sales、Change、Comments フィールドに書き込みます。

例のコード 8.2 データセットからの新規ドキュメントの作成

```
01 data sasuser.sales;
02 length comment $20;
03 format comment $char20.;
04 input sales change comment $ 12-31;
05 datalines;
06 123472 342 Strong Increase
```

```

07 423257 33 Just enough
09 218649 4 Not high enough
09 ;
10 run;
11 filename sales NOTESDB;
12 data _null_;
13 file sales;
14 set sasuser.sales;
15 put '!NSF_DB! qrtsales.nsf';
16 put '!NSF_FORM! Jansales';
17 put '!NSF_ADD!';
18 put '!NSF_FIELD!Sales !' sales;
19 put '!NSF_FIELD!Change!' change;
20 put '!NSF_FIELD!Comments!' comment;
21 put '!NSF_CLR_FIELDS!';
22 run;

```

行 11 は、FILENAME ステートメントを使用して、ファイル参照名を割り当てます。ここでは、通常のファイルのかわりに Notes を参照します。NOTESDB は、Lotus Notes のデバイスタイプです。行 13 は、割り当てたファイル参照名を PUT ステートメントの出力先に設定します。行 15 の NSF_DB データディレクティブは、開く Notes データベースを示します。行 18 から 20 は、作成する新規の Notes ドキュメントのフィールドとフィールド値を指定します。sales はフィールド名、sales は特定のドキュメントのステータスフィールドの入力値です。行 22 は、これらの SAS ステートメントを実行します。新規の Notes ドキュメントが、Sales データベースに作成されます。

Business Card Request データベースの例を発展させると、操作ディレクティブとデータディレクティブを使用して、1 つの DATA ステップや SCL コードで、複数の Notes ドキュメントを作成することができます。次の例では、1 つの DATA ステップ内で複数の Notes ドキュメントを作成する方法を説明します。

例のコード 8.3 1 つの DATA ステップ内で複数の Notes ドキュメントを作成する

```

01 filename reqcard NOTESDB;
02 data _null_;
03 file reqcard;
04 put '!NSF_DB!Examples\buscard.nsf';
05 put '!NSF_FIELD!Status! Order';
06 put '!NSF_FIELD!Quantity! 500';
07 put '!NSF_FIELD!RequestedBy!Systems';
08 put '!NSF_FIELD!RequestedBy_CN! Jane Doe';
09 put '!NSF_FIELD!NameLine_1! Jane Doe';
10 put '!NSF_FIELD!NameLine_2! Developer';
11 put '!NSF_FIELD!AddressLine_1! Software R Us';
12 put '!NSF_FIELD!AddressLine_2! 123 Silicon Lane';
13 put '!NSF_FIELD!AddressLine_3! Garner, NC 27123';
14 put '!NSF_FIELD!AddressLine_4! USA';
15 put '!NSF_FIELD!PhoneLine_1! (910) 555-3232';
16 put '!NSF_ADD!';
17 put '!NSF_CLR_FIELDS!';
18 put '!NSF_FIELD!Status! Order';
19 put '!NSF_FIELD!Quantity! 10';
20 put '!NSF_FIELD!RequestedBy! Research and Development';
21 put '!NSF_FIELD!RequestedBy_CN! John Doe';
22 put '!NSF_FIELD!NameLine_1! John Doe';
23 put '!NSF_FIELD!NameLine_2! Analyst';
24 put '!NSF_FIELD!AddressLine_1! Games Inc';
25 put '!NSF_FIELD!AddressLine_2! 123 Software Drive';

```

```

26 put '!NSF_FIELD!AddressLine_3! Cary, NC 27511';
27 put '!NSF_FIELD!AddressLine_4! USA';
28 put '!NSF_FIELD!PhoneLine_1! (910) 555-3000';
29 run;

```

行 1 は、FILENAME ステートメントを使用して、ファイル参照名を割り当てます。ここでは、通常のファイルのかわりに Notes を参照します。NOTESDB は、Lotus Notes のデータベースタイプです。行 3 は、割り当てたファイル参照名を PUT ステートメントの出力先に設定します。行 4 は、開く Notes データベースを示します。行 5 から 15 は、作成する新規の Notes ドキュメントのフィールドとフィールド値を指定します。ステータスはフィールド名、Order はこのドキュメントステータス フィールドの入力値です。行 16 は、新規の Notes ドキュメントを作成します。行 17 は、これまでの行の!NSF_FIELD!データディレクティブで使用されたフィールド値をクリアします。前の行のデータディレクティブ行 18 から 28 は、2 番目に作成する Notes ドキュメントのフィールドとフィールド値を指定します。ステータスはフィールド名、Order は 2 番目のドキュメントステータス フィールドの入力値です。行 29 は、これらの SAS ステートメントを実行します。2 番目の Notes ドキュメントが、Business Card Request データベースに作成されます。

一度だけの!NSF_DB!データディレクティブは先に挙げた例で発行されます。デフォルトでは、2 番目の Notes ドキュメントは、行 4 で指定された!NSF_DB!データディレクティブが参照するデータベースに作成されます。2 番目の Notes ドキュメントを別のデータベースに作成するには、行 18 を実行する前に、新たなデータベースファイル名が指定された!NSF_DB!データディレクティブを発行する必要があります。この例に追加された重要な要素は、行 16 と行 17 の操作ディレクティブです。

注: すべてのディレクティブで大文字と小文字が区別されるわけではありません。ただし、データディレクティブの後に置かれるフォーム名やフィールド名などの値では、大文字と小文字が区別されます。

Notes ドキュメント向けの SAS/GRAPH 出力を用意する

SAS/GRAPH 出力は、NOTESDB アクセスエンジン経由で Notes ドキュメントに渡されます。!NSF_FIELD!データディレクティブの構文を若干変更するだけで、SAS/GRAPH 出力が Notes ドキュメントのリッチテキスト形式(RTF)フィールドに送られます。データディレクティブにより、SAS/GRAPH 出力は Notes ドキュメント内の RTF フィールドへと送られます。その手順は次のとおりです。

- SAS/GRAPH 出力を、ビットマップファイル形式にエクスポートします。
- 変更した!NSF_FIELD!を使います。RTF フィールドにビットマップファイル名の値を割り当てるためのデータディレクティブ構文は次のとおりです。

```
!NSF_FIELD! RTF-フィールド名 < bitmap ファイル名
```

次の例では、変更した構文を使用します。

注: この例では、Electronic Library サンプルデータベースを使用します。

例のコード 8.4 Notes ドキュメントへの SAS/GRAPH 出力をエクスポート

```

01 filename myfile 'test1.bmp';
02 goptions device=bmp gsfname=myfile gsfmode=replace;
03 title1 'US Energy Consumption for 1955-1988';
04 proc gplot data=3Dsampsio.energy1;
05 plot consumed*year / des=3D'D0319U01-1';
06 run;
07 quit;

08 filename newdoc NOTESDB;
09 data _null_;

```

```

10 file newdoc;
11 put '!NSF_DB!Examples\hrdocs.nsf';
12 put '!NSF_FIELD!Subject! US Energy Consumption';
13 put '!NSF_FIELD!Categories! Office Services';
14 put '!NSF_FIELD!Body! US Energy Consumption for 1955-1988';
15 put '!NSF_FIELD!Body<c:\usenergy.bmp';
16 run;

```

行 1-6 は、サンプルデータセットを使用して SAS/GRAPH 出力を生成するために、SAS/GRAPH サンプルから取り込まれたコードです。行 8 は、FILENAME を使用して、ファイル参照名を割り当てます。ここでは、通常のファイルのかわりに Notes を参照します。NOTESDB は、Lotus Notes のデバイスタイプです。行 10 は、割り当てたファイル参照名を PUT ステートメントの出力先に設定します。行 11 は、開く Notes データベースを示します。行 12 から 14 は、作成する新規の Notes ドキュメントのフィールドとフィールド値を指定します。Subject はフィールド名、US Energy Consumption はこのドキュメントの Subject フィールドの入力値です。行 15 は、Body フィールドに usenergy.bmp ビットマップファイルを表示することを示します。このため、フィールド名とフィールド値の区切りに、感嘆符(!)ではなく不等号(<)が使用されます。行 16 は、これらの SAS ステートメントを実行します。新規の Notes ドキュメントが、Electronic Library データベースに作成されます。

前述の例では、Detailed フィールドは RTF フィールドです。RTF フィールドが使用されている場合は、データとビットマップを挿入することができます。

NotesSQL ODBC ドライバーを用いて SAS を使用する

SAS では、Notes データベース内の既存の Notes ドキュメントに関する情報を取り込めるように、SAS/ACCESS to ODBC パススルーエンジンが提供されています。テキストフィールドのみ取り込むことができます。図形データは取り込めません。

注: Notes ODBC ドライバーとデータソースを構成する必要があります。

次のソフトウェアが必要です。

- Lotus Notes Client Version 7.0.2 以降(32 ビットのみ)
- ODBC ドライバー NotesSQL 3.02 (32 ビット)

NotesSQL は、Lotus から提供される ODBC ドライバーです。<http://www.lotus.com> から無料でダウンロードできます。

ソフトウェアの入手後、次の操作を行います。

1. NotesSQL ODBC ドライバーを設定します。

このドライバの設定方法については、Lotus 提供のファイル(.nfs)を参照してください。

2. ODBC データソースを構成します。

Lotus Notes ODBC 2.0 設定画面を完了する必要があります。次のフィールドに適切な情報を追加します。

注: 丸括弧内に例を示します。

- データソース名(buscard)
- 説明(Test Notes Access)
- サーバーまたはデータベース名(c:\notes\data\buscard.nsf)
- NotesSQL オプションセットアップ(次のフィールドから構成されます)
 - テキストフィールドの最大長(254)

- テーブルの最大数(20)
- サブクエリの最大数(20)
- リッチテキストフィールドの最大長(512)
- SQL ステートメント長(4096)
- スレッドのタイムアウト秒(60).

設定画面の入力が完了したら、OK をクリックします。

既存の Notes ドキュメントから情報を取り込む

SAS/ACCESS to ODBC パススルーエンジンを使用すると、Notes データベース内の既存の Notes ドキュメントに関する情報を取り込むことができます。例のコード 8.5 (228 ページ) では DATA ステップを使って Business Card Request データベースから、どのように情報を取り込むかの一例を示します。

例のコード 8.5 ODBC を使用して、既存の Notes ドキュメントから情報を取り込む

```
01 proc sql;
02 connect to ODBC ("dsn=3Dbuscard");
03 create table sasuser.buscard as
04 select * from connection to
05 ODBC (select * from All_Requests_By_Organization);
06 disconnect from ODBC;
run;
```

行 1 は、SQL ビューとテーブルを操作する SQL ステートメントを処理します。行 2 は、ODBC に接続します。3Dbuscard データソースを使用するために、SAS/ACCESS to ODBC Driver および NotesSQL ODBC Driver を経由して Notes に接続します。行 3、4、5 は、`All_Requests_By_Organization` と呼ばれる Notes Business Card Request テーブルから取り込んだデータから、テーブルと `sasuser.buscard` を作成します。これは、Business Card Request データベースに割り当てられたデフォルトビューです。行 6 は、ODBC から切断し、Notes データベースへの接続を終了します。行 7 は、これらの SAS ステートメントを実行します。`buscard` という名前の新規データセットが、`Sasuser` ライブラリに作成されます。

この他に、SQL クエリウィンドウを使用して、Notes データベース内の使用可能なテーブルを参照する方法もあります。SQL クエリウィンドウは SAS のコンポーネントです。この対話型インターフェイスを使用すると、プログラミングステートメントを書くことなく簡単にクエリを作成することができます。SQL クエリウィンドウを呼び出すには、コマンドラインから `QUERY` コマンドを発行します。

PROC SQL 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。

9 章

Windows 版 SAS で 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	229
イベントビューアアプリケーションログ	230
Windows Vista を使用してアプリケーションログにアクセス	230
Windows 7 を使用したアプリケーションログへのアクセス	230
Windows Server 2003 を使用してアプリケーションログにアクセス	230
Windows XP を使用してアプリケーションログにアクセス	231
SAS イベントの表示	231
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	231
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	233
パフォーマンスツール	234
パフォーマンスモニタの使用理由について	234
Windows パフォーマンスモニタの起動	234
パフォーマンスカウンタとオブジェクト	235
パフォーマンスモニタとシステムモニタの SAS カウンタ	235
モニタする SAS カウンタの選択	235
DATA ステップ、PROC SORT、PROC SQL のモニタリング例	236
DATA ステップと PROC SORT ステップ間のパフォーマンス検証	236
PROC SQL クエリの検証	238
SAS を Windows のサービスとして起動する	240
Windows のサービスとしての SAS の起動について	240
SAS サービス構成ユーティリティの起動	240
初期化ファイルの作成	240
SAS サービスのインストール	247
SAS サービスの起動	249
SAS サービスの削除	249

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。詳細は、“[イベントビューアアプリケーションログ](#)” (230 ページ) を参照してください。

SAS セッションを監視し、問題の診断とセッションの調整に必要な情報を入手するには、Windows XP と Window Server 2003 では Windows システムモニタを使用します。詳細は、“パフォーマンスツール” (234 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。詳細は、“SAS を Windows のサービスとして起動する” (240 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 Vista を使用してアプリケーションログにアクセス

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. システムとメンテナンスを選択します。
3. 管理ツールで、イベントビューアを選択します。

注: Windows が継続の許可を必要としているというメッセージが表示される場合、**続行**を選択してください。

4. ツリービューで、[Windows ログ]を選択します。
5. [アプリケーションログ]を選択します。

他の方法でイベントビューアを起動するには、**プログラムとファイルの検索**ダイアログボックスで eventvwr と入力し、OK をクリックします。

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]で、[イベントビューア]を選択します。

注: Windows が継続の許可を必要としているというメッセージが表示される場合、**続行**を選択してください。

4. ツリービューで、[Windows ログ]を選択します。
5. [アプリケーションログ]を選択します。

他の方法でイベントビューアを起動するには、**プログラムとファイルの検索**ダイアログボックスで eventvwr と入力し、OK をクリックします。

Windows Server 2003 を使用してアプリケーションログにアクセス

Windows Server 2003 を使用して、イベントビューアのツリービューからアプリケーションログにアクセスするには、次の操作を行います。

1. まず、**スタート** ⇒ **設定** ⇒ **コントロールパネル**と選択します。
2. そして、**管理ツール**をダブルクリックします。

3. 次に、**イベントビューア**をダブルクリックします。
4. ツリービューで、**アプリケーションログ**を選択します。

他の方法でイベントビューアを起動するには、ファイル名を指定して実行ダイアログボックスで `eventvwr` と入力し、**OK** をクリックします。

Windows XP を使用してアプリケーションログにアクセス

Windows XP を使用して、イベントビューアのツリービューからアプリケーションログにアクセスするには、次の操作を行います。

1. まず、**スタート** ⇨ **コントロールパネル**と選択します。
2. そして、**管理ツール**をダブルクリックします。
3. 次に、**イベントビューア**をダブルクリックします。
4. ツリービューで、**アプリケーションログ**を選択します。

他の方法でイベントビューアを起動するには、**ファイル名を指定して実行**ダイアログボックスで `eventvwr` と入力し、**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は、イベントソースとして **SAS Job Spawner** が表示されます。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、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相当します。

```
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 9.1 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	“ERROR”
警告	“WARNING”
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。

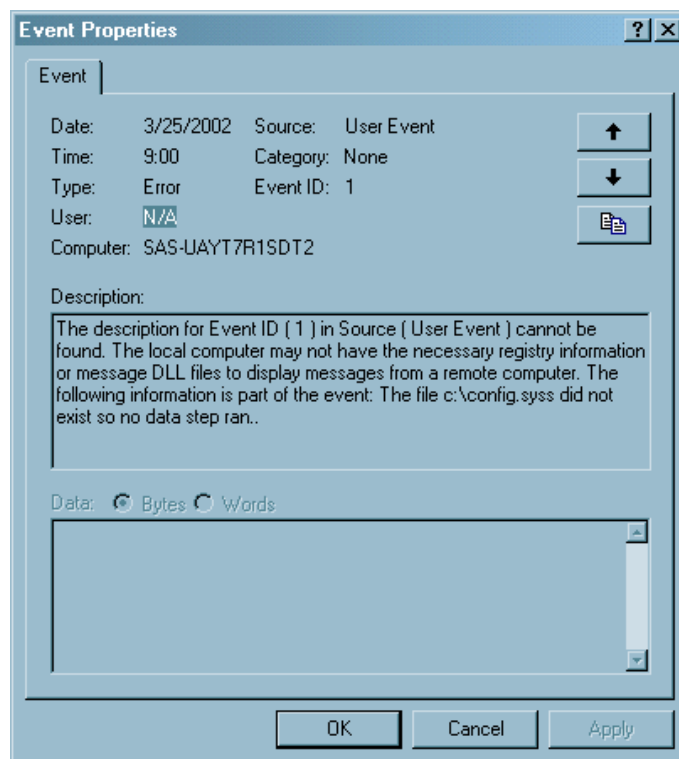
次の例では、SAS によって長い処理が実行される前に、セマフォファイルの存在がチェックされます。

```
%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の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。

画面 9.1 [イベントの詳細]ダイアログボックスでのユーザーメッセージの表示



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、システムモニタ (Windows XP、Windows Vista、Windows 7、Windows Server 2008、Windows Server 2003)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見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。Disk WriteFile Bytes Written/Sec カウンタおよび Disk WriteFile Bytes Written Total カウンタをモニタリングすると、ジョブが動作していることを確認できます。

Windows パフォーマンスモニタの起動

ファイル名を指定して実行ダイアログボックスに `perfmon` と入力すると、Windows Vista、Windows 7、Windows Server 2008、Windows Server 2003、Windows XP を使用している場合はパフォーマンスウィンドウが開きます。

パフォーマンスモニタおよびパフォーマンスウィンドウは、管理ツールフォルダからアクセスすることもできます。

注: Windows x64 システムで 32 ビットバージョンの SAS パフォーマンスカウンタを使用するには、32 ビットバージョンの SAS の起動後、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 エクスプローラを表示して、32 ビットアプリケーションが含まれる `c:\Windows\SysWOW64` フォルダを開きます。32 ビットバージョンの PerfMon は、このフォルダにあります。32 ビットバージョンの SAS のパフォーマンスカウンターにアクセスできます。

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、共有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。システムモニタの詳細については、『Windows XP リソースキット』を参照してください。

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 9.2 SAS カウンタ選択の手順

Windows XP を使用している場合

- SAS を起動します。
- パフォーマンスウィンドウを開きます。
- 追加ボタン(+)をクリックします。
- パフォーマンスオブジェクトリストから Process を選択します。
- インスタンスリストから SAS を選択します。
- モニタリングするカウンタごとに、リストからカウンタを選択し、**追加**をクリックします。
- モニタリングするカウンタの選択が完了したら、**閉じる**をクリックします。

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。SAS インスタンスは、SAS PID number の形式で表示されます。PID number は、SAS セッションのプロセス識別子です。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. 次の SAS カウンタを追加します。
 - Disk ReadFile Bytes Read/Sec
 - Disk WriteFile Bytes Written/Sec
 - Disk SetFilePointer/Sec
4. Process オブジェクトを選択します。
5. 次の Process カウンタを追加します。
 - %Processor Time
 - %User Time
 - %Privileged Time
6. **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。

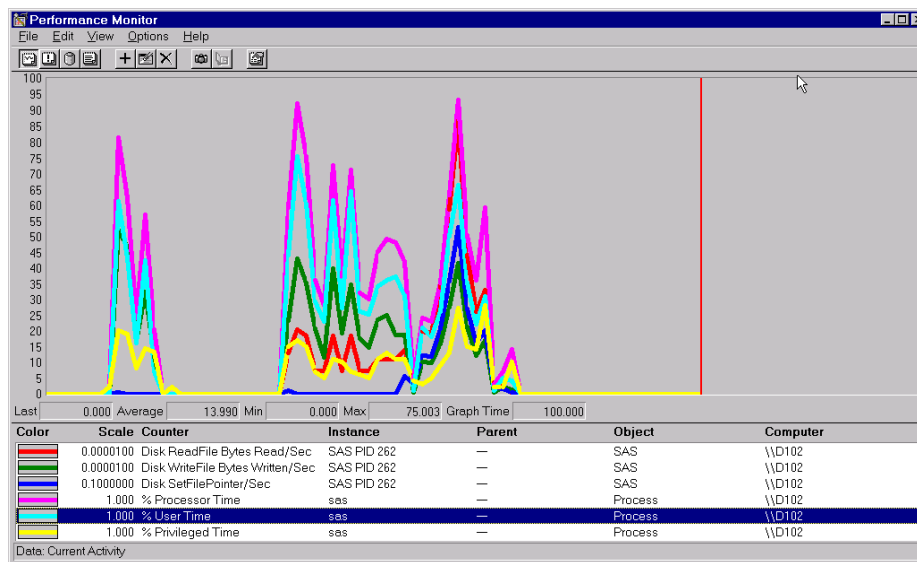
```

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;

```

次のコードをサブミット後、パフォーマンスモニターまたはシステムモニターは [画面 9.2](#) (237 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。

画面 9.2 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. 編集メニューの**表示をクリア**を選択するか、**表示をクリア**ツールバーボタンをクリックして、パフォーマンスモニタグラフをクリアします。
3. ステップ 3 のコードをサブミットして、**画面 9.3 (239 ページ)** のようなグラフを表示させます。

```
/* 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. */

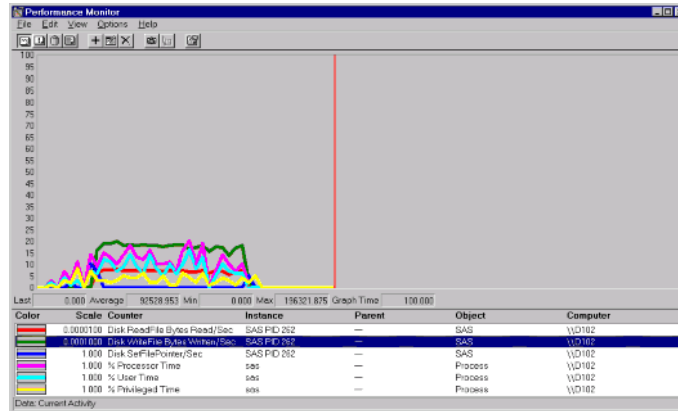
PROC SQL;
create table sample.yz as
select y,z
from sample.a
where x > 0.5;
quit;
```

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。

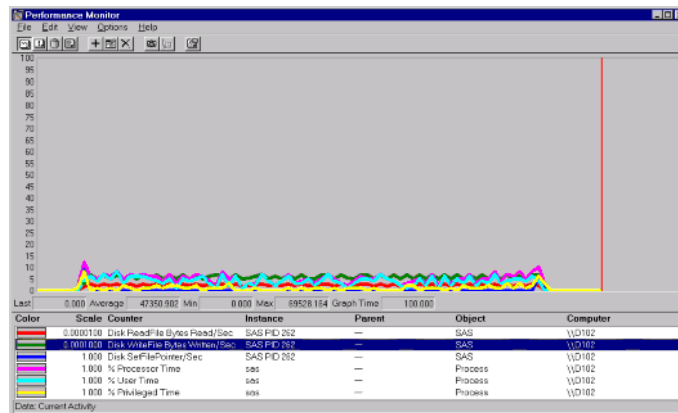
1. ステップ 1 をもう一度サブミットします。
2. グラフをクリアします。

3. ステップ 3 を再サブミットして、画面 9.4 (239 ページ) のようなグラフを表示させます。

画面 9.3 インデックスを使用した PROC SQL クエリのパフォーマンス



画面 9.4 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。画面 9.4 (239 ページ)、画面 9.3 (239 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、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を示しています。

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。Disk WriteFile Bytes Written/Sec 画面 9.3 (239 ページ) における平均値は 92528.953 でした。この値を、平均値が 47350.902 であった画面 9.4 (239 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に定義されている場合は、`net start` コマンドを使用してアプリケーションから起動することも、Windows のサービスダイアログボックスを使用して起動することもできます。

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。詳細は、“[SAS サービスの起動](#)” (249 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 ⇒ 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と選択します。
- 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\9.3\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)ファイルを作成し、インストールする必要があります。INI ファイルは Unicode ファイルです。次の構成が指定されます。

- SAS サービスの名前を指定します。
- サービスを自動で起動するか手動で起動するかを指定します。
- SAS へのパスと、SAS の作業フォルダを定義します。

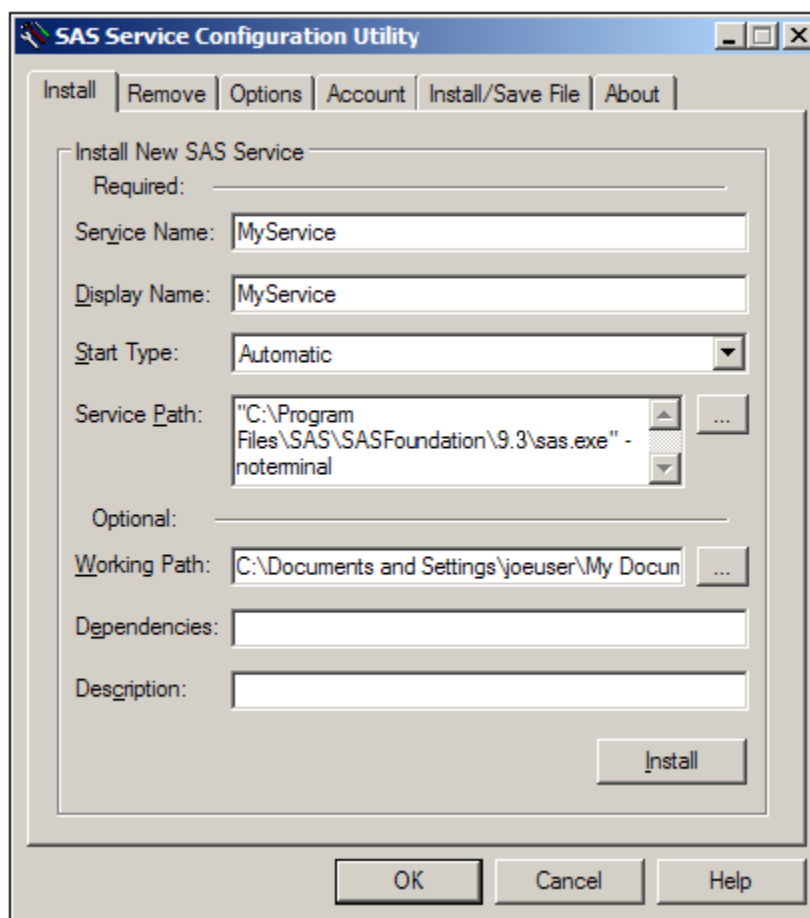
- アプリケーションから SAS サービスへのアクセスのレベルを指定します。
- このサービスを起動する前に起動する必要がある Windows の他のサービスを指定します。
- SAS がサービスとして起動しない場合に、Windows で完了すべき処理を定義します。
- SAS サービスをシステムアカウントにするかローカルアカウントにするかを指定します。
- ユーザーに SAS デスクトップからの操作を許可するかどうかを指定します。

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。SSCU GUI を使用すると、必須の値を指定するだけで、.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 エンコードファイルではなく、Unicode ファイルとして保存する必要があります。[表 9.3 \(244 ページ\)](#) テキストエディタで SAS サービス INI ファイルを作成するときに指定する設定を説明します。

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。設定が終了したら、**ファイルのインストール/保存タブ**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。

画面9.5 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

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。デフォルトは手動です。手動を指定すると、別のプロセスによってサービスが起動されます。自動を指定すると、システム起動時にサービスが自動的に起動されます。無効を指定すると、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。新たにインストールする場合、デフォルトのパスは SAS のインストールパスで、その後コマンド sas.exe - noterminal が付きます。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\SAS\SASFoundation\9.3\

で、その後に `sas.exe -noterminal` が付きます。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、...をクリックしてください。(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 Vista、Windows 7、Windows Server 2008、Windows Server 2008 R2 では、ユーザープロファイルは `c:\Users\ユーザーID\Documents\My SAS Files\9.3` に置かれます。その他すべての Windows 動作環境では、ユーザープロファイルは `c:\Documents and Settings\ユーザーID\My Documents\My SAS Files\9.3` に置かれます。作業フォルダを選択するには、...をクリックしてください。(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は、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。たとえば、NetDDE|NetDDEsdm のように入力します。

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。

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、INI ファイル内で特定のアクセスタイプに 1 (TRUE) が設定されます。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 9.3 \(244 ページ\)](#) を参照してください。アクセスレベルは、次のとおりです。

読み込みにより、アプリケーションでは**応答、クエリの構成、クエリのステータス**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 ファイルでは `AccessInterrogate=`、`AccessQryCfg=`、および `AccessQryStatus=` に 1 (TRUE) が設定されます。

書き込みにより、アプリケーションでは、**構成の変更**アクセスタイプが設定できます。書き込みアクセスレベルを選択すると、INI ファイルでは `AccessChgCfg=` 設定に 1 (TRUE) が設定されます。

実行により、アプリケーションでは、**応答、一時停止/再開、サービスの開始、サービスの停止、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 ファイルでは `AccessInterrogate=`、

AccessPauseCont=、AccessStart=、AccessStop、および AccessUserDefCtrl=に 1 (TRUE) が設定されます。

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。デスクトップからユーザーがサービス进行操作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 ファイルでは ServiceStartName=に LocalSystem が設定されます。

このアカウント

は、サービスは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、アカウント名をフィールドに入力します。次にパスワードフィールドおよびパスワードの入力確認フィールドにパスワードを入力します。

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

インストールする初期化(INI)ファイルを指定するには、ファイルからインストールボタンをクリックします。

ファイルに設定を保存

SAS サービス構成ユーティリティの GUI で指定した設定をファイルに保存するには、ファイルに設定を保存ボタンをクリックします。

ファイル内容を表示

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は、ファイル内容を表示ボックスを選択します。

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 9.3 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

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
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 SERVICE_DEMAND_START SERVICE_DISABLED	SERVICE_DEMAND_START	開始タイプ
依存関係=	X	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	X	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=	X	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。	作業ディレクトリは、Windows Vista、Windows 7、Windows Server 2008 では c:\Users\user ID に、また、Windows XP、Windows Server 2003 では、c:\Documents and Settings\user ID にあります。	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 SERVICE_ERROR_NORMAL SERVICE_ERROR_SEVERE SERVICE_ERROR_CRITICAL	SERVICE_ERROR_NORMAL	エラーコントロール
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) 0(FALSE)	0(FALSE)	対話型処理

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
AccessChgCfg=	Yes	SAS サービス構成を修正する。	1(TRUE) 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) 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) 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) 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) 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) 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) 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) 0(FALSE)	1(TRUE)	コントロールの定義
ServiceStart Name=	X	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=	X	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、**my`pw`** 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}bX1wdw==` が表示されます。次に、INI ファイル内に Password=設定の値として `{sas001}bX1wdw==` を指定します。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。SSCU 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. **ファイルを開く**ダイアログボックスから、INI ファイルを選択します。
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 9.4 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。
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。

数値コード	エラーコード	説明
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 サービスが手動で起動するように設定されている場合は、アプリケーションから `net start` コマンドを使用するか、サービスダイアログボックスを使用してサービスを起動することができます。

SAS サービスをサービスダイアログボックスを使用して起動するには、次の操作を実行します。

- Windows Vista では、**スタート** ⇒ **コントロールパネル** ⇒ **システムとメンテナンス** ⇒ **管理ツール** ⇒ **サービス**と選択します。

Windows 7 や 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**と選択します。

その他すべての Windows 動作環境では、**スタート** ⇒ **設定** ⇒ **コントロールパネル** ⇒ **管理ツール** ⇒ **サービス**と選択します。

- サービスリストボックスから、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. **既存のサービスの削除**リストボックスから、SAS サービスを選択します。
3. **削除**をクリックします。
4. サービスの削除を確認するには、**はい**を**サービスの削除**ダイアログボックスで選択します。

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。

10 章

Windows 版 SAS/AF ソフトウェアでの OLE の使用

OLE について	252
SAS/AF カタログの互換性	252
OLE オブジェクトを FRAME エントリに挿入する	252
FRAME エントリへの OLE オブジェクトの挿入について	252
OLE オブジェクトの挿入	253
クリップボードからの OLE オブジェクトの貼り付け	254
HSERVICE エントリから OLE オブジェクトの読み込み	254
ドラッグによる OLE オブジェクトの挿入	255
FRAME エントリ内での OLE オブジェクトの編集	256
OLE Verb の起動	257
リンクされた OLE オブジェクトを使用する	258
概要: リンクされた OLE オブジェクトを使用する	258
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	258
リンクされたオブジェクトの自動更新	259
OLE オブジェクトの変換	259
OLE オブジェクトとアプリケーションの自動化	261
概要: OLE オブジェクトとアプリケーションの自動化	261
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	262
値のプロパティの使用	263
OLE サーバーメソッドのオプションパラメータの指定	264
外部 OLE オートメーションインスタンスの作成	264
例: Microsoft Excel スプレッドシートへの SAS データの入力	266
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	268
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	268
OLE コントロールを FRAME エントリに挿入	269
OLE コントロールの登録	269
OLE コントロールのプロパティへのアクセス	269
SCL メソッドを使用した OLE コントロールの操作	271
OLE コントロールイベントへの応答	271

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 9.2 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” (646 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 9.3 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 9.3 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、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。

この他に、**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方法もあります。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](#)” (258 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、**ビュー** ⇨ **プロパティウィンドウ**と選択します。

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。

エントリフィールドにオブジェクトエントリの名前を入力します。2 レベルの HSERVICE 名を利用できます。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 編集セッションを**保存**または**終了**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、形式を選択して貼り付けダイアログボックスで**リンクの貼り付け**を選択します。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](#)” (258 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場合、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しないと検索できません。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てから、**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** と選択します。**プロパティ**ボックスでオブジェクトを選択し、**オブジェクト属性**をクリックします。
8. **エントリフィールド**にオブジェクトエントリの名前を入力します。2レベルの HSERVICE 名を利用できます。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 編集セッションを**保存**または**終了**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. カーソルを空白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。

- OLE-オブジェクト属性の読み込みウィンドウで、**エントリフィールド**に HSERVICE エントリの名前を入力します。2 レベルの HSERVICE 名を利用できます。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。選択ウィンドウを使用してエントリを検索するには、**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。

- 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。
- 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。
- 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](#)” (255 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しないでください。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

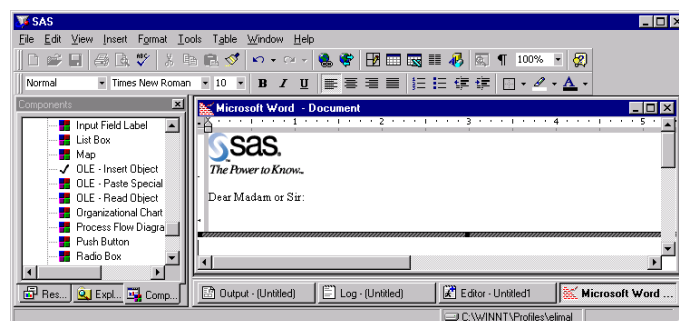
OLE 2.0 の最も優れた機能の 1 つに、ビジュアル編集があります。これは、埋め込まれているオブジェクトを、他のアプリケーションへと切り替えることなく、埋め込み先で編集できる機能です。

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、右クリックして、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、[画面 10.1 \(256 ページ\)](#) 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。

画面 10.1 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。大部分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 (BUILD: DISPLAY モードで OLE オブジェクトを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)アクションと**開く** (オープン編集)アクションを認識します。メディアクリップオブジェクトは、**再生**アクションと**編集**アクションを認識します。

有効な動詞のリストを表示するには、オブジェクトのオブジェクト属性ダイアログボックスで**関連する動詞**をクリックします。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、OLE オブジェクト `mediaobj` の動詞**再生**を起動します。

```
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_” \(636 ページ\)](#)”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。

- 自動更新 - リンクオブジェクトが参照するソースファイルが更新された場合。(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)。
- 手動更新 - リンクダイアログボックスで**今すぐ更新**を選択した場合か、または SCL で `_UPDATE_` メソッドを使用した場合。
- 手動更新 - 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された場合。

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。

- ビジュアル編集に対して、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” \(256 ページ\)](#)”を参照してください。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。

もし、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、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する必要があります。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

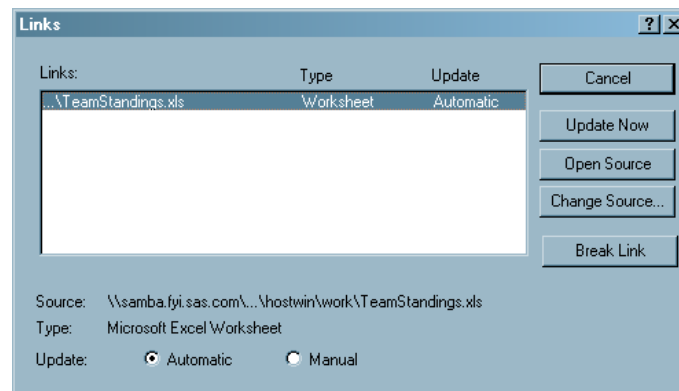
リンクダイアログボックスを使用してリンク情報を更新するには、次の操作を行います。画面 [10.2 \(259 ページ\)](#) に示されています。

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、オブジェクトの種類がメニューの一番下の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、その中に、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表示されます (FRAME エントリ内にリンクオブジェクトが存在しない場合は、**リンク**項目は無効になっています)。
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');
```

画面 10.2 [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_”](#) (641 ページ) を参照してください。

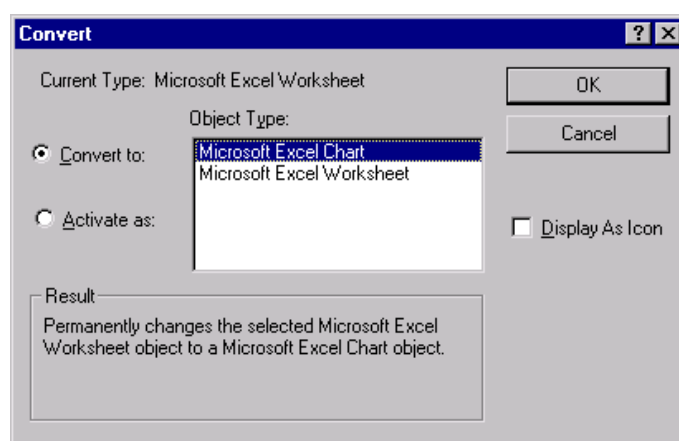
OLE オブジェクトの変換

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。[画面 10.3](#) (260 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。たとえば、システム上にオブジェクトを変換できるサーバーアプリケーションがある場合、Microsoft Excel オブジェクトを Lotus 1-2-3 オブジェクトに変換できます。この種類の変換はそのまま保持されます。
- 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。たとえば、Microsoft Excel をサーバーとして使用して Lotus 1-2-3 オブジェクトを再生するように選択できます。これにより、オブジェクトを Excel オブジェクトと同じように編集できます。オブジェクトは、Lotus 1-2-3 オブジェクトとして保持されます。変換設定を変更するまで、その後作成する Lotus 1-2-3 OLE オブジェクトはすべて、Excel を OLE サーバーとして使用します。

画面 10.3 [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。

オブジェクトをアイコンで表示するのか、内容で表示するのかを切り替えるには、**アイコンで表示**を使用します。

オブジェクトを他のサーバーを使用して再生するには、**再生方法**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。各種オブジェクトは、それぞれ独自のメソッドおよびプロパティが定義されています。使用できる SCL メソッドは次にリストされています。表 10.1 (261 ページ) そして、詳細は、“OLE クラスメソッドの要約” (633 ページ) に記述されています。

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 10.1 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
<code>_COMPUTE_</code>	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
<code>_DO_</code>	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)。

OLE オートメーションメソッド	説明
<code>_GET_PROPERTY_</code>	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
<code>_GET_REFERENCE_ID_</code>	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
<code>_IN_ERROR_</code>	オブジェクトのエラーステータスを返します
<code>_NEW_</code>	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。
<code>_SET_PROPERTY_</code>	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 リストとして渡される 1 次元配列が受信できます。
- 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、\WINNT\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_`” (640 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 10.2 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 <code>_NEW_</code> メソッドを使用することにより作成。

SAS OLE オブジェクト	SAS OLE 自動化オブジェクト
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。
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 10.3 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。	<pre>LAUNCHXL: hostc1 = loadclass('sashelp.fsp.hauto'); call send(hostc1, '_NEW_', excelobj, 0, 'Excel.Application'); call send(excelobj, '_SET_PROPERTY_', 'Visible', 'True'); return;</pre>
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。	<pre>CREATEWS: call send(excelobj, '_GET_PROPERTY_', 'Workbooks', wsobj); call send(wsobj, '_DO_', 'Add'); call send(excelobj, '_GET_PROPERTY_', 'ActiveSheet', wsobj);</pre>
SAS データセットを開きます。	<pre>dsid=open('sasuser.class','i'); call set(dsid); rc=fetch(dsid); nvar=attrn(dsid, 'NVAR'); nobs=attrn(dsid, 'NOBS');</pre>

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。	<pre>do col=1 to nvar; call send(wsobj, '_COMPUTE_', 'Cells',1,col,retcell); var=varname(dsid,col); call send(retcell, '_SET_PROPERTY_', 'Value',var); end; do while (rc ne -1); do row = 1 to nobs; do col = 1 to nvar; r=row+1; call send (wsobj, '_COMPUTE_', 'Cells', r ,col,retcell); if vartype(dsid,col) eq 'N' then do; varn=getvarn(dsid,col); call send(retcell, '_SET_PROPERTY_', 'Value' ,varn); end; else do; varc=getvarc(dsid,col); call send(retcell, '_SET_PROPERTY_', 'Value' ,varc); end; end; rc=fetch(dsid); end; end; dsid=close(dsid); return;</pre>
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。	<pre>QUITXL: call send(excelobj, '_GET_PROPERTY_', 'ActiveWorkbook', awbobj); call send(awbobj, '_DO_', 'Close', 'False'); call send(excelobj, '_DO_', 'Quit'); call send(excelobj, '_TERM_'); return;</pre>

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、その後で、生成された Visual Basic コードを参照すると便利です。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 10.4 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
<pre>Excel を起動して表示します Set excelobj = CreateObject("Excel.Application") excelobj.Visible = True</pre>	<pre>hostcl = loadclass('sashelp.fsp.hauto'); call send (hostcl, '_NEW_', excelobj, 0, 'Excel.Application'); call send (excelobj, '_SET_PROPERTY_', 'Visible', 'True');</pre>

Visual Basic のコード	SCL の OLE オートメーション
<pre>新しいワークシートを作成します Dim wbsobj, wsobj As Object Set wbsobj = excelobj.Workbooks wbsobj.Add Set wsobj = excelobj.ActiveSheet</pre>	<pre>call send(excelobj, '_GET_PROPERTY_', 'Workbooks', wbsobj); call send(wbsobj, '_DO_', 'Add'); call send(excelobj, '_GET_PROPERTY_', 'ActiveSheet', wsobj);</pre>
<pre>セルの値を設定します wsobj.Cells(row + 1, col).Value =var</pre>	<pre>r=row+1; call send(wsobj, '_COMPUTE_', 'Cells', r, col, retcell); call send(retcell, '_SET_PROPERTY_', 'Value' ,var);</pre>
<pre>Excel アプリケーションオブジェクトを閉じます excelobj.ActiveWorkbook.Close (False) excelobj.Quit</pre>	<pre>call send(excelobj, '_GET_PROPERTY_', 'ActiveWorkbook', awbobj); call send(awbobj, '_DO_', 'Close', 'False'); call send(excelobj, '_DO_', 'Quit'); call send(excelobj, '_TERM_');</pre>

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す。16 ビットコンテナのみをサポートする Windows アプリケーションでは 32 ビットコンテナは使用できません。また、SAS は、32 ビットコンテナであるため、16 ビットコントロールは使用できません。

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. 選択リストから **OLE - オブジェクトの挿入**を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開きます。**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。“**OLE コントロールの登録**” (269 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. 次に記述されているステップ 1 から 4 を完了してください。“**OLE コントロールを FRAME エントリに挿入**” (269 ページ) 登録済みコントロールのリストでオブジェクトの挿入ダイアログボックスを呼び出すには、次の操作を実行してください。
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

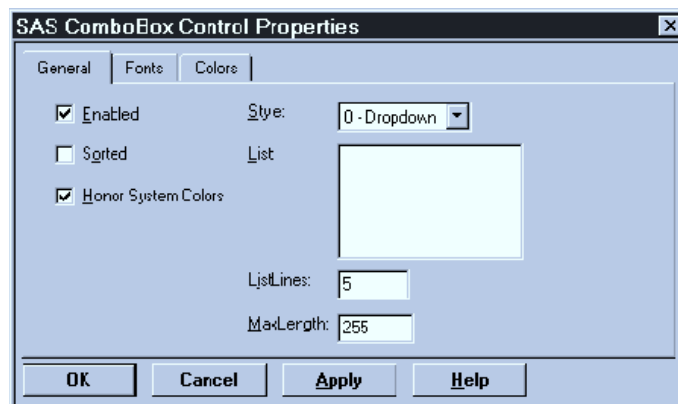
OLE コントロールのプロパティは、SCL メソッドを使用して設定または取得できます。OLE コントロールの中には、プロパティページからプロパティの一部にアクセスできるものがあります。プロパティページを使用すると、データを対話的に設定または取得できます。

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で右クリックし、プロパティポップアップメニューから、次の操作ができます。コントロールのプロパティページが表示されます。画面 10.4 (270 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。

画面 10.4 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の

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せん。

- “OLE コントロールイベントへの SCL コードの割り当て” (271 ページ)
- “イベントから引数値を取り込む” (272 ページ)
- “例: SCL コードへの OLE コントロールイベントのマッピング” (273 ページ)
- “例: OLE カスタムコントロールのサブクラス化” (273 ページ)
- “コンボボックスリストへのアイテムの追加” (274 ページ)
- “コンボボックスのアイテムの検索” (274 ページ)
- “コントロールのテキスト値の取り込み” (274 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

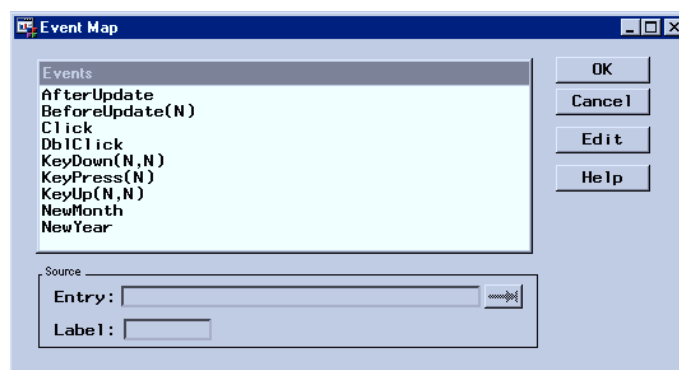
OLE コントロールイベントの発生時に実行する SCL コードを割り当てるには、次の操作を行います。

- 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。
- オブジェクトのポップアップメニューから**オブジェクト属性**を選択します。
- オブジェクト属性ダイアログボックスで**イベントマッピング**を選択します。次に示されるように、イベントマッピングダイアログボックスが表示されます。[画面 10.5 \(272 ページ\)](#)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 エントリとともに保存される、同じ SCL ソースエントリも指定できません。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。

画面 10.5 [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ピング

次のいずれかの方法で、OLE コントロールイベントをマッピングできます。

- イベントマッピングウィンドウのイベントごとに、それぞれ異なる操作を実行するコードを含む、SCL コードの各ラベルセクションにマッピングします。
- イベントマッピング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ピングします。_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 つのラベルにマッピングします。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 つのラベルにマッピングされる場合に、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('DbClick') put 'DbClick 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、OLE - オブジェクトの挿入を選択するのではなく、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。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);
```


11 章

Windows 版で OLE を使用し、他のアプリケーションから SAS を制御する

SAS の自動化について	277
SAS のインスタンスの作成	278
SAS セッションからのフィードバックの取得	278
OLE を用いた SAS の自動化の例	279
Visual Basic の使用例	279
SAS オートメーションオブジェクトの作成	279
SAS セッションがビジーかどうかを特定	279
SAS セッションの表示/非表示の切り替え	280
SAS セッションのメイン SAS ウィンドウタイトルの設定	280
SAS ライブラリの割り当てと SAS プロシジャの実行	280
SAS セッションの終了	280
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	280
ディクショナリ	281
コマンドメソッド	281
QueryWindow メソッド	281
Quit メソッド	282
Submit メソッド	282
Top メソッド	282
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	283

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 Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
```

```
Private Sub Command1_Click()
Dim OleSAS as Object
Set 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",
0, 0)
x=1
EndIf
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” (279 ページ)
- “SAS セッションがビジーかどうかを特定” (279 ページ)
- “SAS セッションの表示/非表示の切り替え” (280 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” (280 ページ)
- “SAS ライブラリの割り当てと SAS プロシジャの実行” (280 ページ)
- “SAS セッションの終了” (280 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", 0, 0)
Else
Response = MsgBox("SAS Session is Idle",
vbOKOnly, "SAS Session", 0, 0)

```

```
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。

```
OleSAS.Visible = false
```

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、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、指定されたウィンドウが自動化された SAS セッションに開かれているかどうかを **True** または **False** のどちらかで返します。存在するウィンドウが非表示になっている場合でも、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する必要があります。
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.QueryWindow("build")) Then
```

```
OleSAS.Command("build")  
EndIf
```

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  
OleSAS.Submit("libname afapp 'f:\sas\afapp';")  
OleSAS.Command("af c=afapp.bigapp.main.frame")
```

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
OleSAS.Top
```

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 が終了する前に、確認ボックスは表示されません。値が 1 の場合は、SAS が終了する前に、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。デフォルトの操作は、[プリファレンス]ダイアログボックスなど、SAS の終了方法を定義する別な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。

12 章

Windows 版で動的データ交換を使用する

動的データ交換(DDE)の概要	285
SAS での DDE 構文	286
DDE 外部ファイルの参照	287
DDE 外部ファイルの参照の概要	287
DDE トリプレットの使用	287
DDE を使用して他のアプリケーションを制御する	288
DDE の例	288
DDE 例の概要	288
X コマンドを使用して DDE サーバーを開く	288
DDE を使用して、Microsoft Excel にデータを書き込む	289
DDE を使用して、Microsoft Word にデータを書き込む	289
DDE を使用して、Microsoft Excel からデータを読み込む	289
DDE を使用して、Microsoft Word からデータを読み込む	290
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	290
DDE で NOTAB オプションを使用する	291
DDE HOTLINK の使用	292
!DDE_FLUSH 文字列を使用してデータを動的に転送する	293
マクロ変数を使用した DDE コマンドの発行	293
欠損データの読み取り	294

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。

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” (252 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 *ファイル参照* DDE 'DDE トリプレット' <DDE オプション>;

説明

ファイル参照

有効なファイル参照名です(“外部ファイルの参照”(150 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 トリプレット'

DDE 外部ファイルの名前です。

DDE オプション

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”(288 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”(292 ページ)を参照してください。

LRECL=レコード長

レコード長をバイト単位で指定します。Windows 版では、デフォルト値は 256 です。レコード長には、1 から 1,073,741,823 (1GB)の値を指定できます。

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”(291 ページ)を参照してください。

RECFM=レコードの形式

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 256 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 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。

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”(294 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 トリプレットについては、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。

```
'アプリケーション名|トピック|アイテム'
```

説明

アプリケーション名

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。

トピック

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。

アイテム

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、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 ⇒ DDEトリプレットと選択します。

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 名を空白にします。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” (261 ページ) を参照してください。

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 come 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、空白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 変数に空白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。空白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せん。

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 が処理された時点で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つ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 が解決されず、%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 - 3 と行 10 - 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ぎます。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を使用されます。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。

13 章

Windows 版で名前付きパイプと名前の付いていないパイプを使用する

パイプの概要	297
名前の付いていないパイプの使用	298
名前の付いていないパイプについて	298
名前の付いていないパイプの構文	298
リダイレクトの順序の使用	299
名前の付いていないパイプの例	299
名前付きパイプの使用	300
名前付きパイプについて	300
名前付きパイプの構文	300
CALL RECONNECT ルーチンの使用	302
SCL の名前付きパイプの使用	302
名前付きパイプの例	302

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。

パイプには、一方向通信と双方向通信があります。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ファイルハンドル番号を示します。

表 13.1 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](#)” (299 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 *ファイル参照* PIPE '*プログラム名*' *オプションリスト*;

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。

ファイル参照

有効なファイル参照名です。“[外部ファイルの参照](#)” (150 ページ) に詳細があります。

パイプ

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。

プログラム名

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'
```

オプションリスト

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](#)” (469 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。ホストオブシ

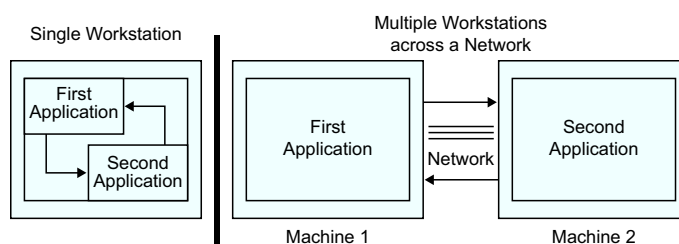
ン `CONSOLE=MIN` が指定されているため、`STOCKMKT.EXE` プログラムを実行するため開かれるコマンドプロンプトウィンドウが、最小化されて開かれます。`INFILE` ステートメントによって、SAS から `STOCKMKT.EXE` プログラムが呼び出され、このプログラムからパイプを介してデータが読み込まれます。Windows タスクバーにコマンドプロンプトウィンドウボタンが表示されますが、ユーザーが `STOCKMKT.EXE` プログラムの実行に気付くことなく、このプログラムは終了します。ファイル参照名 `STOCKS` はすでに名前の付いていないパイプとして定義されているため、`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 13.1 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ることで、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 つのプロセスをサーバーとして定義し、それ以外の 1 つまたは複数のプロセスをクライアントとして定義します。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 ファイル参照 **NAMEPIPE** 'パイプ仕様 <名前付きパイプオプション>;

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。

ファイル参照

有効なファイル参照名です。“外部ファイルの参照” (150 ページ) に詳細があります。

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。

パイプ仕様

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。

\\サーバー名PIPE\パイプ名

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。

前付きパイプオプション

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を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 が指定されたときとみなされます。

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” (307 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 13.2 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)

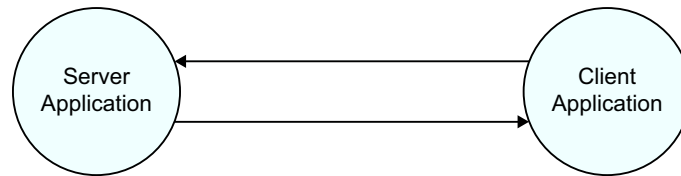
名前付きパイプの例**名前付きパイプ例の概要**

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 13.2 (303 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13.2 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MIT 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ることで、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。

```

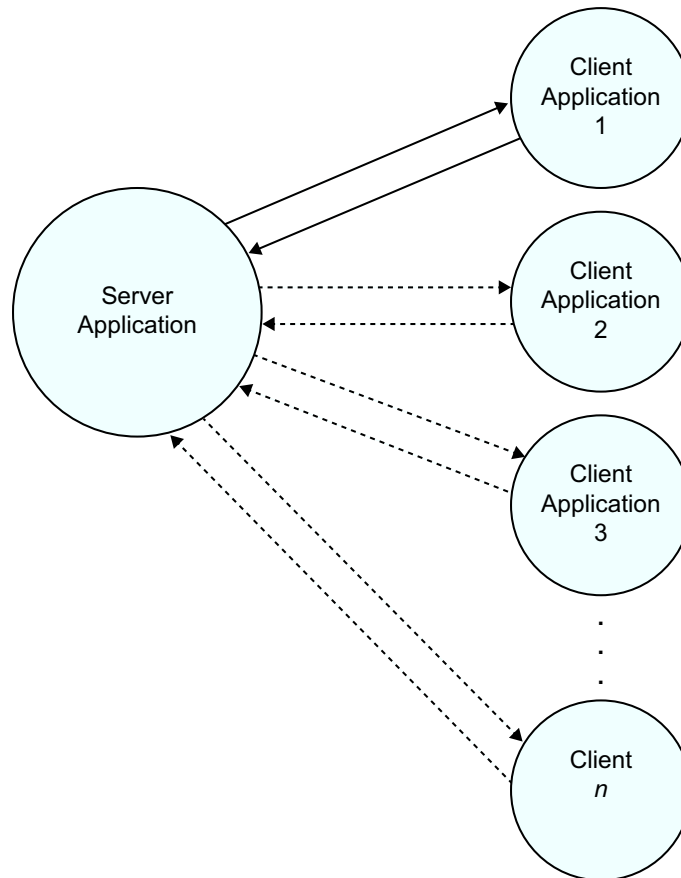
filename results namepipe '\\.\pipe\results'
server retry=60;
proc printto print=results new;
run;
  
```

```
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つのサーバーで複数のクライアントを接続できます。この場合、名前付きパイプの構成は [図 13.3 \(304 ページ\)](#) に示すようなものになります。

図 13.3 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 totsals;
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;

```

14 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	309
SASCBTBL 属性テーブル	310
SASCBTBL 属性テーブルの概要	310
属性テーブルの構文	311
属性テーブルの重要性	314
外部 DLL を使用する場合に特に注意すべき点	315
PEEKLONG 関数を使用して文字列引数にアクセスする	315
効率的に外部 DLL にアクセス	316
SAS 変数を構造引数としてグループ化	317
MODULE の引数として定数と式を使用	318
MODULE 引数で使用する出力形式および入力形式の指定	319
MODULE ログメッセージについて	323
例	325
文字列引数の更新	325
引数を値として渡す	326
PEEKCLONG を使用して返されたポインタにアクセスする	326
構造の使用	327
DLL ルーチンを PROC IML から起動	328

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” (310 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、IML プロシジャステップ内では MODULEI、MODULEIN、MODULEIC を使用することもできます。MODULE 関数は次に記述されています。“MODULE 関数: Windows” (428 ページ)。

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>
  <TRANPOSE=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” (428 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ことを示します。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と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があります。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

```
STGPTR=3536768 STGPTR=0035F780
```

を発行します。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、“PEEKCLONG 関数”(SAS 関数と CALL ルーチン: リファレンス)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

```
* 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;
```

のような基本的な形式を取ったとします。それから、DATA ステップが

```
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;
```

のようなものだとします。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、外部ルーチン呼び出しの際、ポインタを構造に渡す必要があります。構造の一部がルーチンの入力として使用されることもあり、また、他の部分がルーチンによって置換または埋め込まれることもあります。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

```
HWND GetForegroundWindow(VOID);
BOOL GetClientRect(HWND hWnd, LPRECT lprc);
```

です。C 言語では、ルーチン呼び出しコードは

```
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);
```

になります。

MODULE を使用してこれらのルーチン呼び出すには、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.;
```

と次の DATA ステップ

```
filename sascbtbl 'sascbtbl.dat';
data _null_;
hwnd=modulen('GetForegroundWindow');
call module('GetClientRect',hwnd,left,
top,right,bottom);
put left= top= right= bottom=;
run;
```

を使用します。

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ことを示します。

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:

```
* 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 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、文字列へと渡すことの詳細については、次を参照してください。“[\\$BYVALw. 出力形式](#)” (322 ページ)。

C 言語形式

表 14.1 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 および Itanium SAS)
unsigned int	PIB4.
unsigned short	PIB2.
unsigned long	PIB4.
char[w]	\$CHAR _w .または\$CSTR _w .(次を参照してください。“\$CSTR _w . 出力形式” (322 ページ))

FORTRAN 言語形式

表 14.2 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、引数がディスクリブ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 14.3 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 14.4 COBOL 言語形式

COBOL 言語の型	SAS 出力形式/入力形式	説明
PIC Sxxxx BINARY	IBw.	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 Sxxxx	Fw.	出力可能数値
PIC yyyy	\$CHARw.	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 14.5 COBOL 言語形式(ゾーン 10 進とパック 10 進)

COBOL 言語の型	SAS 出力形式/入力形式	説明
PIC Sxxxx DISPLAY	ZDw.	ゾーン 10 進
PIC Sxxxx PACKED-DECIMAL	PDw.	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 S370Fxxx 入出力形式と連結して使用可能です。S370F エlement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 14.6 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	S370FZDUw.	符号なしゾーン 10 進
PIC Sxxxx DISPLAY SIGN LEADING	S370FZDLw.	前符号付きゾーン 10 進
PIC Sxxxx DISPLAY SIGN LEADING SEPARATE	S370FZDSw.	分離した前符号付きゾーン 10 進
PIC Sxxxx DISPLAY SIGN TRAILING SEPARATE	S370FZDTw.	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

\$CSTR w . 出力形式

文字引数をヌル終端文字列として渡すには、\$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空白文字が検索され、この最後の非空白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、次の属性テーブルエントリがあるとします。

```
* 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、指定された長さの空白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。

\$BYVAL w . 出力形式

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 14.1 \(324 ページ\)](#) に示される行がログに出力されます。

アウトプット 14.1 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
0000000000002C400000000000002E40000000000003040000000000003140000000000003240
00000000000038400000000000003940000000000003A40000000000003B40000000000003C40
0000000000004140000000000080414000000000
---ROUTINE changi LOADED AT ADDRESS 886119B8 (PARMLIST AT 886033A0)--- PARM 1
06000000 <CALL-BY-VALUE>
PARM 2 88604720
0E0000000F00000010000000110000001200000018000000190000001A0000001B0000001C000000
22000000230000002400000025000000260000002C0000002D0000002E0000002F00000030000000
---VALUES UPON RETURN FROM changi ROUTINE--- PARM 1 06000000
<CALL-BY-VALUE>
PARM 2 88604720
140000001F0000002A0000003500000040000000820000008D00000098000000A3000000AE000000
F0000000FB00000006010000110100001C0100005E01000069010000740100007F0100008A010000
---VALUES UPON RETURN FROM MODULEIN ROUTINE--- NUM PARM 3 885E0AE0
0000000000001840
NUM PARM 4 885E07F0
00000000000034400000000000003F40000000000004540000000000804A40000000000005040
000000000406040000000000A061400000000006340000000000606440000000000C06540
000000000006E40000000000606F4000000000

```

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。この最初の 2 つの引数についてのみ、対応する ASCII の値が出力されています。これは、他の引数については、読み取り不可能なバイナリデータが含まれ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 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

```
routine GetTempPathA
minarg=2
maxarg=2
stackpop=called
returns=long;
arg 1 input byvalue format=pib4.;
arg 2 update format=$cstr200.;
```

を示します。

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せん。

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 sasctbl "sasctbl.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ブ音を出します。

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,
LPTSTR 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。

PEEKCLONG 関数に関する詳細は次を参照してください。“PEEKCLONG 関数” (SAS 関数と CALL ルーチン: リファレンス) および “PEEKLONG 関数” (SAS 関数と CALL ルーチン: リファレンス).

構造の使用

“SAS 変数を構造引数としてグループ化” (317 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、 4×5 の行列を作成します。各セルは、 $10x+y+3$ 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 $(10 \cdot 1) + 2 + 3$ 、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、 $100x+10y$ 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 x_1 、 x_2 、 y_1 、 y_2 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
```


15 章

Windows 版 SAS/AF のプログラミングで特に注意すべき点

SAS の表示画面と機能の制御	331
メイン SAS ウィンドウの制御	332
概要: メイン SAS ウィンドウの制御	332
メイン SAS ウィンドウを制御する SAS システムオプション	332
メイン SAS ウィンドウを制御する SAS コマンド	333
SAS から外部 DLL にアクセスする	336
カスタムツールバーコントロールの設計、保存、ロード	336
SAS/AF アプリケーションの自動起動	337
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	337

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 15.1 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、非表示したりにします。
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、非表示したりにします。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 15.2 (333 ページ) SAS メインウィンドウの表示と機能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 15.2 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	リリース情報 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	メールの送信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。
TOOLEEDIT	ツールバーまたはツールボックスのユーザー設定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	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、IML プロシジャステップ内で 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、“[Windows 版 SAS から外部 DLL にアクセスする](#)” (309 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](#)” (69 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、TOOLS SWITCH 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](#)” (597 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

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](#)”
(612 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 版に固有の SAS 言語の 機能

16 章	Windows 版に固有のデータセットオプション.....	341
17 章	Windows 版に固有の SAS コマンド.....	343
18 章	Windows 版に固有の SAS 出力形式.....	395
19 章	Windows 版に固有の SAS 関数と CALL ルーチン.....	407
20 章	Windows 版に固有の SAS 入力形式.....	435
21 章	Windows 版に固有の SAS プロシジャ.....	445
22 章	Windows 版に固有の SAS ステートメント.....	465
23 章	Windows 版に固有の SAS システムオプション.....	491
24 章	Windows 版での変数の長さ精度.....	621
25 章	Windows 版に固有の SAS マクロ機能.....	625

16 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	341
ディクショナリ	341
SGIO データセットオプション: Windows	341

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](#)

17 章

Windows 版に固有の SAS コマンド

Windows 版に固有の SAS コマンド	344
Windows 版に固有の SAS コマンドの概要	344
Windows 動作環境でサポートされないコマンド	345
ディクショナリ	345
AUTOSCROLL コマンド: Windows	345
AWSMAXIMIZE コマンド: Windows	346
AWSMINIMIZE コマンド: Windows	346
AWSRESTORE コマンド: Windows	347
CAPS コマンド: Windows	347
COLOR コマンド: Windows	348
COMMAND コマンド: Windows	348
CUT コマンド: Windows	350
DLGABOUT コマンド: Windows	350
DLGCDIR コマンド: Windows	351
DLGCOLUMNSIZE コマンド: Windows	351
DLGCOLUMNSORT コマンド: Windows	351
DLGCONVERT コマンド: Windows	352
DLGENDR コマンド: Windows	352
DLGFIND コマンド: Windows	353
DLGFONT コマンド: Windows	353
DLGLIB コマンド: Windows	353
DLGLINKS コマンド: Windows	354
DLGOPEN コマンド: Windows	354
DLGPAGESETUP コマンド: Windows	356
DLGPREF コマンド: Windows	356
DLGPRT コマンド: Windows	357
DLGPRTPREVIEW コマンド: Windows	358
DLGPRTSETUP コマンド: Windows	358
DLGREPLACE コマンド: Windows	359
DLGRUN コマンド: Windows	359
DLGSAVE コマンド: Windows	360
DLGSMAIL コマンド: Windows	361
FILE コマンド: Windows	362
FILEOPEN コマンド: Windows	364
FILL コマンド: Windows	364
GSUBMIT コマンド: Windows	365
HOME コマンド: Windows	365
ICON コマンド: Windows	366
INCLUDE コマンド: Windows	366
PMENU コマンド: Windows	368
SAVE コマンド: Windows	369

STORE コマンド: Windows	369
SUBTOP コマンド: Windows	370
TOOLCLOSE コマンド: Windows	371
TOOLEEDIT コマンド: Windows	371
TOOLLARGE コマンド: Windows	371
TOOLLOAD コマンド: Windows	372
TOOLSWITCH コマンド: Windows	373
TOOLTIPS コマンド: Windows	373
WATTACH コマンド: Windows	374
WATTENTION コマンド: Windows	375
WAUTOSAVE コマンド: Windows	375
WBROWSE コマンド: Windows	376
WCOPY コマンド: Windows	376
WCUT コマンド: Windows	377
WDOCKVIEW コマンド: Windows	377
WDOCKVIEWMINIMIZE コマンド: Windows	378
WDOCKVIEWRESIZE コマンド: Windows	378
WDOCKVIEWRESTORE コマンド: Windows	379
WEDIT コマンド: Windows	379
WEMAILFMT コマンド: Windows	380
WEXITSAVE コマンド: Windows	381
WFILE コマンド: Windows	381
WHIDECURSOR コマンド: Windows	382
WHSBAR コマンド: Windows	382
WINSERT コマンド: Windows	383
WMENUPOP コマンド: Windows	383
WMRU コマンド: Windows	384
WNAVKEYUNMARK コマンド: Windows	385
WNEWTITLE コマンド: Windows	385
WNEXTEDIT コマンド: Windows	386
WPASTE コマンド: Windows	386
WPGM コマンド: Windows	387
WPOPUP コマンド: Windows	387
WRTFSAVE コマンド: Windows	387
WSCREENTIPS コマンド: Windows	388
WSTATUSLN コマンド: Windows	389
WUNDO コマンド: Windows	390
WVSBAR コマンド: Windows	390
WWINDOWBAR コマンド: Windows	391
X コマンド: Windows	392
ZOOM コマンド: Windows	392

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呼び出しもう 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せん。

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**タブでスクロールオプションを設定することもできます。

関連項目:

- “[セッションプリファレンスの設定](#)” (60 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>

必須引数

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。

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>

必須引数

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。

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>

必須引数

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。

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 を押す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* <color | 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。

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。

強調表示

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>

必須引数

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。

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、SORT 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” (61 ページ)
- “DLGPREF コマンド: Windows” (356 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>

必須引数

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 COMMAND コマンド」
- “クリップボードの使用” (57 ページ)
- “WCUT コマンド: Windows” (377 ページ)

DLGABOUT コマンド: Windows

SAS のリリース情報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

メニューからリリース情報ダイアログボックスを表示するには、ヘルプメニューを選択し、リリース情報を選択します。

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で、別の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” \(585 ページ\)](#)

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。

関連項目:

- [“一覧ビューの個々の列のサイズ調整” \(xix ページ\)](#)

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。

関連項目:

[“指定した列を基準にした、ウィンドウの一覧ビューの並べ替え” \(xviii ページ\)](#)

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変換できます。変換できる種類は、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:

SAS/AF ソフトウェアでの OLE の使用セクション内の OLE のオブジェクトの変換を参照してください。参照先: *Windows 版 SAS*

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されます。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。**プリファレンス**ダイアログボックスの**全般**タブで**終了の確認をする**が選択されていない場合は、DLGENDR コマンドを入力した時点で SAS が終了します。

関連項目:

[“セッションプリファレンスの設定” \(60 ページ\)](#)

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” \(359 ページ\)](#)

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。

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、自動でライブラリを起動することもできます。

関連項目:

SAS エクスプローラー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オブジェクトはありません。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:

SAS/AF ソフトウェアでの OLE の使用セクション内の OLE のリンクオブジェクトを参照してください。参照先: Windows 版 SAS

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'>
```

必須引数

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。

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 1 つ以上のフィルタを指定します。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**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 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=を指定したときのみ有効です。たとえば、このコマンド

```
dlgopen longfilter="*.txt" replace
```

により、**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

ダイアログボックスが表示されるときに、**サブミット**チェックボックスを選択するかどうかを指定します。**サブミット**チェックボックスでは、開いたファイルの内容を SAS プログラムとして直ちにサブミットするかどうかを指定できます。デフォルトでは、[サブミット]チェックボックスは選択されていません。ファイルを開いたときに、そのファイルを自動的にサブミットするには、**プリファレンス**ダイアログボックスの**全般**タブで**開いたファイルの内容をサブミットする**を選択します。

IMPORT

インポートインポートダイアログボックスを表示して、グラフィックファイルを SAS セッションに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” \(196 ページ\)](#)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。ある場合は、デフォルトの INCLUDE コマンドを起動する代わりに、**ファイルを開く**ダイアログボックスで項目を開くコマンドを起動します。

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するコマンドを指定します。たとえば、そのコマンド

```
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ボックスを使用して、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**拡張エディタを使用するオプション**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を表示するには、ファイルメニューを選択して**開く**を選択します。

関連項目:

- “ファイルを開く” (110 ページ)
- “セッションプリファレンスの設定” (60 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” (195 ページ)
- “FILEOPEN コマンド: Windows” (364 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、用紙のサイズ、用紙トレイ、印刷の向き、余白などのページの属性を定義できます。

関連項目:

- “印刷ページの設定” (176 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の設定” \(60 ページ\)](#)

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>

必須引数

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。

ACTIVEBITMAP

印刷ダイアログボックスを表示せず、アクティ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で、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表示せず、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で、ビットマップ印刷のみ許可します。

CLIPBITMAP

印刷ダイアログボックスを表示せず、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表示せず、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表示せず、デフォルトの設定で印刷します。

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表示せず、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。テキスト印刷がサポートされていない場合は、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” \(172 ページ\)](#)

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>

必須引数**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。出力をプレビューするには、DLGPRT VERIFY コマンドを発行し、次にプレビューをクリックします。

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:

- [“印刷プレビュー” \(180 ページ\)](#)
- [“印刷” \(171 ページ\)](#)

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>

必須引数**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表示しません。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のみ使用します。

詳細

印刷設定ダイアログボックスでは、印刷に使用するプリンタの指定、SAS フォームを使用した印刷の指定、および情報を印刷する方法(用紙の向き、余白、フォントなど)の設定が可能です。

関連項目:

- [“プリンタの変更” \(174 ページ\)](#)
- [“印刷フォントの変更” \(175 ページ\)](#)
- [“印刷ページの設定” \(176 ページ\)](#)

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では、テキスト文字列を検索し、別のテキスト文字列に置き換えます。

関連項目:

- [“DLGFIND コマンド: Windows” \(353 ページ\)](#)

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 で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'>
```

必須引数

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。

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、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 *.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せんが、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、このコマンド

```
dlgsave longfilter="*.txt" replace
```

により、**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表示して、グラフィックファイルを SAS セッションから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” (197 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、デフォルトの FILE コマンドではなく、**名前を付けて保存**ダイアログボックスで**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で選択したファイルに適用するコマンドを指定します。たとえば、そのコマンド

```
dlgsave altcmd='prtfile'
```

により、**名前を付けて保存**ダイアログボックスで選択したファイルが、現在の印刷ファイルに設定されます。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を表示するには、ファイルメニューを選択して**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**” (84 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**” (117 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 によってメールの送信ダイアログボックスが開きます。EMAILDLG オプションの値が `native` の場合は、DLGSMAIL によって MAPI 準拠の電子メールのダイアログボックスが開きます。

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” (380 ページ)
- “EMAILDLG システムオプション: Windows” (532 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。

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、20 章: “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。

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。デフォルトの設定は 8192 です。最大 1MB まで指定できます。

IGNOREDOEOF

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 版では、デフォルト値は 256 です。*record-length* には、1 から 1,073,741,823 (1GB)の値を指定できます。

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 256 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 | 可変の形式を指定します。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を確認され、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”
- “ENCODING=オプション” (*SAS 各国語サポート(NLS): リファレンスガイド*) *SAS 各国語サポート(NLS): リファレンスガイド Windows 版 SAS*
- 外部ファイルの使用セクション内の外部ファイルの参照を参照してください。参照先: *Windows 版 SAS*

- 外部ファイルの使用チャプター内の FILE コマンドの使用を参照してください。参照先: *Windows 版 SAS*
- これらのオプション使用の例については、外部ファイルの使用セクション内の上級外部入出力テクニックを参照してください。参照先: *Windows 版 SAS*

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* を指定しない場合には、**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” \(83 ページ\)](#)
- [“DLGOPEN コマンド: Windows” \(354 ページ\)](#)

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内の GSUBMIT コマンドの使用を参照してください。参照先: *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>

必須引数**引数なし**

アクティブウィンドウを最小化します。

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。

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、20 章: “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。

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-sizeBLK=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。

IGNOREDOEOF

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 版では、デフォルト値は 256 です。record-length には、1 から 1,073,741,823 (1GB)の値を指定できます。

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 256 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	可変の形式を指定します。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=オプション” (SAS 各国語サポート(NLS): リファレンスガイド) SAS 各国語サポート(NLS): リファレンスガイド
- “外部ファイルの参照” (150 ページ)
- 外部ファイルの使用セクション内の外部ファイルの参照を参照してください。参照先: Windows 版 SAS
- “INCLUDE コマンドの使用” (164 ページ)
- 外部ファイルの使用セクション内の INCLUDE コマンドの使用を参照してください。参照先: Windows 版 SAS
- これらのオプション使用の例については、外部ファイルの使用セクション内の上級外部入出力テクニックを参照してください。参照先: Windows 版 SAS

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>

必須引数

引数なし

コマンド行のオン/オフを切り替えます。

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” (348 ページ)
- “WMENUPOP コマンド: Windows” (383 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>

必須引数**(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。

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>

必須引数**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” \(57 ページ\)](#)

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 >

必須引数**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。

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

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チェックボックスが選択されている場合、SUBTOP コマンドの発行時にサブミットされた行はすべてウィンドウから削除されます。

関連項目:

[“プログラムのサブミット” \(94 ページ\)](#)

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。

TOOLEEDIT 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

TOOLEEDIT <*library.catalog.entry*>

必須引数

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。

library.catalog.entry

編集する TOOLBOX エントリを指定します。

詳細

TOOLEEDIT コマンドは、*library.catalog.entry* で指定された TOOLBOX エントリでツールのユーザー設定ダイアログボックスを起動します。TOOLBOX エントリを指定しない場合、現在ロードされているツールセットを編集する[ツールのユーザー設定]ダイアログボックスが起動されます。

関連項目:

[“ツールバーのカスタマイズ” \(69 ページ\)](#)

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>

必須引数

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。

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” \(74 ページ\)](#)

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>

必須引数

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。

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ツールバーをカスタマイズし保存する” \(73 ページ\)](#)

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の設定” \(60 ページ\)](#)
- [“TOOLEEDIT コマンド: Windows” \(371 ページ\)](#)

TOOLTIPS コマンド: Windows

ツールチップ機能のオン/オフを切り替えます。

Windows 固有: すべて

構文

TOOLTIPS <ON | OFF>

必須引数

引数なし

ツールチップ機能のオン/オフを切り替えます。

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” \(388 ページ\)](#)

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>

必須引数

引数なし

添付モードのオン/オフを切り替えます。

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のファイル形式には、テキスト形式または RTF 形式のどちらかが使用されます。このファイル形式は、WEMAILFMT コマンドまたはプリファレンスダイアログ

ボックスの設定に応じて決定されます。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。

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**で切り替えることもできます。

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” (380 ページ)
- “セッションプリファレンスの設定” (60 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*>

必須引数

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。

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タブで自動保存機能を設定することもできます。

関連項目:

- “セッションプリファレンスの設定” (60 ページ)
- “AUTOSAVELOC=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)

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”>

必須引数

引数なし

プリファレンスダイアログボックスの Web タブで定義した Web ブラウザを起動します。

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 ブラウザによって開かれる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 タブで変更できます。

関連項目:

- “セッションプリファレンスの設定” (60 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” \(369 ページ\)](#)

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” \(350 ページ\)](#)

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>

必須引数

引数なし

ドッキングビューのオン/オフを切り替えます。

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 の[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の設定” (60 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” (379 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 **ウィンドウ** ⇨ **サイズドッキングビュー** 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。

関連項目:

[“メイン SAS ウィンドウのドッキングビューのサイズ変更” \(xviii ページ\)](#)

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

WDOCKVIEWRESTORE 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” \(378 ページ\)](#)
- [“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>

必須引数

引数なし

拡張エディタウィンドウを開きます。

“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**拡張エディタを使用する**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、**拡張エディタを使用する**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の設定” \(60 ページ\)](#)
- [“Windows 環境での SAS エディタの使用” \(81 ページ\)](#)

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、**プリファレンス**ダイアログボックスの全般タブにある**ウィンドウの内容を電子メールに添付する**チェックボックスが更新されます。

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” \(374 ページ\)](#)
- [“DLGSMail コマンド: Windows” \(361 ページ\)](#)

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>

必須引数

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。

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**ページで切り替えることもできます。

関連項目:

[“セッションプリファレンスの設定” \(60 ページ\)](#)

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](#)” (84 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](#)” (117 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>

必須引数

引数なし

カーソルの表示と非表示を切り替えます。

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。プリファレンスダイアログボックスの**詳細**タブで、WHIDECURSOR の設定を切り替えることもできます。

関連項目:

[“セッションプリファレンスの設定” \(60 ページ\)](#)

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>

必須引数

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。

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タブで切り替えることもできます。

関連項目:

[“セッションプリファレンスの設定” \(60 ページ\)](#)

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>

必須引数

引数なし

挿入モードのオン/オフを切り替えます。

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

キーボードの Insert キーまたはプリファレンスダイアログボックスの編集タブの上書き入力するオプションで、この設定を切り替えることもできます。

関連項目:

[“セッションプリファレンスの設定” \(60 ページ\)](#)

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>

必須引数

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。

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: ウィンドウ” (523 ページ)
- “PMENU コマンド: Windows” (368 ページ)
- “WPOPUP コマンド: Windows” (387 ページ)

WMRU コマンド: Windows

ファイルメニューに最近使用したファイル名を保持します。

Windows 固有: すべて

構文

WMRU <<ON> <NUM=*number-of-filenames*> <CASCADE> > | <OFF>

必須引数**引数なし**

ファイルの一覧のオン/オフを切り替えます。

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

最近使用したファイルの一覧を、ファイルメニューの**最近使ったファイルサブメニュー**から表示できるようにします。

OFF

ファイルの一覧をオフします。

詳細

ファイルを開くダイアログボックス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いたり保存すると、そのファイル名が、ファイルメニューや**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の一覧に追加されます。SAS の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**から最近使用したファイルの名前を選択することにより、SAS のエディタウィンドウにその

ファイルを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。

これらの設定は、**プリファレンス**ダイアログボックスの**全般**ページで設定することもできます。

関連項目:

[“セッションプリファレンスの設定” \(60 ページ\)](#)

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>

必須引数

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の設定を切り替えます。

ON

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を有効にします。

OFF

ナビゲーションキーによるマーク解除を有効にするの設定を無効にします。

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の設定を使用するには、**ツール** ⇨ **オプション** ⇨ **プリファレンス** ⇨ **編集**と選択していきます。

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の設定” \(60 ページ\)](#)

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” \(387 ページ\)](#)

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、**拡張エディタを使用する**チェックボックスの設定によって異なります。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**から使用できます。**拡張エディタを使用する**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。**拡張エディタを使用する**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は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” \(386 ページ\)](#)

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” \(383 ページ\)](#)

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。

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ファイルの種類を選んで保存リストから.rtf ファイルを選択するときと同じ処理を実行します。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>

必須引数

引数なし

スクリーンチップのオン/オフを切り替えます。

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。スクリーンチップの設定は、**プリファレンスダイアログボックスの表示タブ**で切り替えることもできます。

関連項目:

- “TOOLTIPS コマンド: Windows” (373 ページ)
- “セッションプリファレンスの設定” (60 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> >
```

必須引数

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。

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。プリファレンスダイアログボックスの表示ページで、ステータスバーを切り替えることもできます。

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の設定” \(60 ページ\)](#)

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存在する場合、WUNDO を実行すると、メニューから元に戻すを選択したときと同じ操作が行なわれます。[元に戻す]メニュー項目がない場合は、UNDO コマンドが実行されません。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>

必須引数

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。

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タブ**で切り替えることもできます。

関連項目:

[“セッションプリファレンスの設定” \(60 ページ\)](#)

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>

必須引数

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。

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せん。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。

プリファレンスダイアログボックスの表示タブで、ウィンドウバーの表示と非表示を切り替えることもできます。

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” \(64 ページ\)](#)

- “セッションプリファレンスの設定” (60 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 <'コマンド'>;

必須引数

引数なし

DOS コマンドウィンドウを開きます。

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” (616 ページ)
- “XSYNC システムオプション: Windows” (618 ページ)
- “XWAIT システムオプション: Windows” (619 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>

必須引数

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

18 章

Windows 版に固有の SAS 出力形式

Windows 版に固有の SAS 出力形式	395
バイナリデータの書き込み	395
以前のリリースのユーザー定義入力形式を SAS 9.3 で利用する	396
ディクショナリ	397
HEXw 出力形式: Windows	397
\$HEXw 出力形式: Windows	398
IBwd 出力形式: Windows	399
PDwd 出力形式: Windows	400
PIBwd 出力形式: Windows	402
RBwd 出力形式: Windows	403
ZDwd 出力形式: Windows	404

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 出力形式は完全に *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](#)” (395 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 9.3 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 9.3 で利用する](#)” (396 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 big-endian と呼ばれる順序で格納されます。Intel ベース PC や IBM 互換 PC、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。

- *IBwd*, *PDwd*, *PIBwd* と *RBwd* 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 byte-ordering システムを使用します。
- *S370FIBw.d*, *S370FPDwd*, *S370FRBwd* および *S370FPIBwd* 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。

PIBw に基づいてデータを書き込むプログラムを一例として説明します。*d* 出力形式このプログラムを PC 上で実行すると、データはバイト反転(リトルエンディアン)で書き出されます。そのとき、PC 上では、*PIBw* を使うもう 1 つの SAS プログラムを実行します。データを読み込むための *d* 入力形式。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 PC 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、*S370FPIBw.d* 形式を使用してデータを書き込んでください。マイクログ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、*S370FPIBw.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 9.3 で利用する

バージョン 6.12,7,8 の出力形式(カタログ)は、32 ビットの Windows 版 SAS 9 で読み込むことができます。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 を使用している場合、SAS ライブラリを移行するには、PROC MIGRATE 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

HEXw 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. 出力形式” ([SAS 出力形式と入力形式: リファレンス](#))

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。 w 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](#)” (398 ページ)

入力形式:

- “[HEXw.入力形式: Windows](#)” (438 ページ)

\$HEXw 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

\$HEXw

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

\$HEXw 出力形式は、HEXw 出力形式と同様に、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 版では、\$HEXw 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” \(397 ページ\)](#)

入力形式:

- [“\\$HEXw 入力形式: Windows” \(438 ページ\)](#)

IBwd 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 出力形式” (SAS 出力形式と入力形式: リファレンス)

構文

IBwd

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* 値を指定する場合、IBw.d 出力形式は数を 10^d をかけます。それから、その値に整数バイナリ出力形式を適用します。

範囲 0–10

詳細

IBwd 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。

PC へ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
```

IB4.出力形式を指定すると、次の値が書き出されます。

```
01 00
```


d

倍率を指定します。*d* 値を指定する場合、PD*w.d* 出力形式は数を 10^d をかけます。それから、その値にパック 10 進出力形式を適用します。

範囲 1-31

詳細

PDwd 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

数値が大きすぎてフォーマットできない場合、(例:

```
data a;
x = 1e308;
y = put(x, PD16.2);
put y = hex16.;
run;
```

)。結果は

```
y=0099999999999999
```

です。

関連項目:

入力形式:

- “PD*w.d* 入力形式: Windows” (440 ページ)
- “バイナリデータの書き込み” (395 ページ)

PIBwd 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

PIBwd

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* 値を指定する場合、PIBw.*d* 出力形式は数に 10^d をかけます。それから、その値に正数の整数バイナリ出力形式を適用します。

範囲 0-10

詳細

PIBwd 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、PIBwd 出力形式は-1 の表示に整数を書き込みます。

PCにこ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 = 9999999999999999999;
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:

- “ビッグエンディアンプラットフォームとリトルエンディアンプラットフォーム上でのバイナリ整数データのバイトオーダーリング” (SAS 出力形式と入力形式: リファレンス 1 章)
- “ビッグエンディアンプラットフォームとリトルエンディアンプラットフォーム上でのバイナリ整数データのバイトオーダーリング” (SAS 出力形式と入力形式: リファレンス 3 章)

入力形式:

- “PIBw.d 入力形式: Windows” (441 ページ)
- “バイナリデータの書き込み” (395 ページ)

RBwd 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 出力形式” (SAS 出力形式と入力形式: リファレンス)

構文

RBwd

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* 値を指定する場合、RBw.d 出力形式は数に^{*d*}をかけます。それから、その値にバイナリ実数出力形式を適用します。

範囲 0–10

詳細

RBwd 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。科学的計算の数値データは、通常、浮動小数点表記で表されます (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

数値が大きすぎてフォーマットできない場合、(例:

```
data a;
x = 1e308;
y = put(x, RB8.2);
put y = hex16.;
run;
```

)。結果は

```
y=0000000000D1FFFF
```

です。

関連項目:

- “ビッグエンディアンプラットフォームとリトルエンディアンプラットフォーム上でのバイナリ整数データのバイトオーダーリング” (SAS 出力形式と入力形式: リファレンス 1 章)

入力形式:

- “ビッグエンディアンプラットフォームとリトルエンディアンプラットフォーム上でのバイナリ整数データのバイトオーダーリング” (SAS 出力形式と入力形式: リファレンス 3 章)
- “RBw.d 入力形式: Windows” (443 ページ)
- “バイナリデータの書き込み” (395 ページ)

ZDwd 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 出力形式” (SAS 出力形式と入力形式: リファレンス)

構文

ZDwd

必須引数

”

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

ZDwd 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 18.1 最後のバイト変換

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

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、(例:

```
data a;
x = 1e308;
y = put(x, ZD32.2);
put y = hex16.;
run;
```

)結果は 39 の数列になります。

```
y=3939393939393939
```

関連項目:

[“ZDw.d 入力形式: Windows” \(443 ページ\)](#)

19 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	407
概要: Windows 版に固有の SAS 関数と CALL ルーチン	407
ディクショナリ	408
BYTE 関数: Windows	408
CALL SOUND ルーチン: Windows	408
CALL SYSTEM ルーチン: Windows	409
COLLATE 関数: Windows	411
DINFO 関数: Windows	412
DOPEN 関数: Windows	413
DOPTNAME 関数: Windows	414
DOPTNUM 関数: Windows	415
FDELETE 関数: Windows	415
FEXIST 関数: Windows	416
FILEEXIST 関数: Windows	416
FILENAME 関数: Windows	417
FILeref 関数: Windows	418
FINFO 関数: Windows	419
FOPTNAME 関数: Windows	422
FOPTNUM 関数: Windows	424
LIBNAME 関数: Windows	425
MCIPLSLP 関数: Windows	426
MCIPISTR 関数: Windows	427
MODULE 関数: Windows	428
PEEKLONG 関数: Windows	430
RANK 関数: Windows	431
SLEEP 関数: Windows	431
TRANSLATE 関数: Windows	432
WAKEUP 関数: Windows	433

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。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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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'`) 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

周波数

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。

期間

音声の期間をミリ秒単位で指定します。デフォルト値は-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 ルーチン” (SAS 関数と CALL ルーチン: リファレンス)

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 ルーチンの場合に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、C:\MYDIR、C:\JUNK2、C:\XYZ の 3 つのディレクトリが作成されます。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” \(392 ページ\)](#)
- [“XSYNC システムオプション: Windows” \(618 ページ\)](#)
- [“XWAIT システムオプション: Windows” \(619 ページ\)](#)

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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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。

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。

長さ

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長である 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* の長さとして 56 文字が指定されているため、COLLATE 関数は照合順序の最初の 56 番目の文字までを返します。

```
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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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=dopen("tmpdir");
put did=;
numopts=doptnum(did);
put numopts=;
do i = 1 to numopts;
optname = doptname(did,i);
put i= optname=;
optval=dinfo(did,optname);
put optval=;
end;
run;

```

ログ19.1 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” \(413 ページ\)](#)

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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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](#)” (413 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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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。指定できるディレクトリの情報項目数は 1 つだけのため、DOPTNUM 関数は値として 1 を返します。

例

DOPTNUM 関数の使用例については、“[例: ディレクトリ情報の取得](#)” (413 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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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” (413 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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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。

Windows 版では、*filename* デバイスタイプによって異なります。表 5.2 (159 ページ) に各デバイスに関する情報が記載されています。指定するファイル名は、引用符で囲む必要があります。

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。ファイル参照名として FILENAME ステートメントの *device-type* 引数 (469 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 ステートメントに指定できるすべてのオプションを使用できます。FILENAME ステートメントの [host-option-list \(469 ページ\)](#) を参照してください。

dir-ref

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。

詳細

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](#)” (413 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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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となります。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](#)” (413 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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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せん。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 "Fileref curdirfl exist? rc should be 0 (no); " rc=;
/* assign fileref */
rc=filename("curdirfl", "c:\tmp333");
/* RC=0 indicates success in assigning fileref */
put "Fileref assigned - rc should be 0; " rc=;
rc=fexist ("curdirfl");
/* Does file which "curdirfl" points to exist? No = 0 */
/* Assigning a fileref doesn't create the file. */
put "File still doesn't exist - rc should be 0; " rc=;
rc=fileref ("curdirfl");
/* Does fileref "curdirfl" exist? */
/* Negative means fileref exists, but file does not */
/* Positive means fileref does not exist */
/* Zero means both fileref and file exist */
put "Fileref now exists - rc should be negative; " rc=;
put;
/* Does the file that the fileref points to exist? Should be no. */
if ( fileexist (".\tmp333") ) then
/* if it does, open it for input */
do;
put "Open file for input";
fid=fopen ("curdirfl", "i") ;
end;
else /* most likely scenario */
do;
put "Open file for output";
fid=fopen ("curdirfl", "o");
end;
/* fid should be non-zero. 0 indicates failure. */
put "File id is: " fid=;
numopts = foptnum(fid);
put "Number of information items should be 6; " numopts=;
do i = 1 to numopts;
optname = foptname (fid,i);
put i= optname=;
optval = finfo (fid, optname);
put optval= ;
end;
rc=fclose (fid);
rc=fdelete ("curdirfl");
put "Closing the file, rc should be 0; "
rc=; run;

```

ログ19.2 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;
297 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: " 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=256
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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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 19.1 ファイルの情報項目

<i>nval</i>	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19.3 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=256
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 関数” (SAS 関数と CALL ルーチン: リファレンス)

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](#)” (413 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: ' ' ライブラリ参照名(一重引用符に囲まれたスペース)の動作

参照項目: “LIBNAME 関数” (SAS 関数と CALL ルーチン: リファレンス)

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。

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、MCPISTR 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、MCPISTR 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.avi
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” \(427 ページ\)](#)

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” \(426 ページ\)](#)

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(<cntl> ,module,arg-1,arg-2... ,arg-n);
```

```
num=MODULEN(<cntl> ,module,arg-1,arg-2... ,arg-n);
```

```
char=MODULEC(<cntl> ,module,arg-1... ,arg-2,arg-n);
```

```
CALL MODULEI <cntl> ,modulearg-1,arg-2... ,arg-n);
```

```
num=MODULEIN(<cntl> ,module,arg-1,arg-2... ,arg-n)
```

```
char=MODULEIC(<cntl> ,module,arg-1,arg-2... ,arg-n);
```

必須引数

cntl

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。このオブ

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。E オプションを指定すると、E オプションが暗黙的に指定されます。

- 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、より優先度の高い E オプション)を指定しない場合、MODULE 関数は、"関数の引数は無効です。"というエラーメッセージのみ作成します。通常、この情報だけでは、エラーの原因を特定するには不十分です。
- Sx 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 ルーチンの構文、属性ファイルの形式および提供されている SAS 出力形式と入力形式についての簡単なヘルプ情報を提供します。

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

使用する外部モジュールの名前です。DLL 名と、ルーチン名または順序値をカンマで区切って指定します。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い場合、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。

arg-1, arg-2, ...arg-n

要求された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 内にはない)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:

SAS から外部 DLL へのアクセスについてのセクション内の SASCBTBL 属性テーブルを参照してください。参照先: *Windows 版 SAS*

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

Windows 固有: すべて

参照項目: “PEEKLONG 関数” (SAS 関数と CALL ルーチン: リファレンス) および “PEEKCLONG 関数” (SAS 関数と CALL ルーチン: リファレンス)

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*>)

詳細

address

メモリアドレスにする文字式を指定します。

length

文字データの長さを指定します。

比較

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 関数” (SAS 関数と CALL ルーチン: リファレンス)

構文

RANK(*x*)

必須引数

x

ASCII 照合順序の文字を含む文字式を指定します。*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')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 関数” (SAS 関数と CALL ルーチン: リファレンス)

構文

SLEEP(*n*<,unit>)

必須引数

n

DATA ステップの実行を停止する時間を秒単位で指定します。*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。

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” \(590 ページ\)](#)

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 関数” (SAS 関数と CALL ルーチン: リファレンス)

構文

TRANSLATE(*source*,*to-1*,*from-1* <,...*to-n*,*from-n*>)

必須引数

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
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。過去の時刻を指定した場合、WAKEUP 関数は指定された時刻が再び到来す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” (590 ページ)

20 章

Windows 版に固有の SAS 入力形式

Windows 版に固有の SAS 入力形式	435
Windows 版に固有の SAS 入力形式の概要	435
バイナリデータの読み取り	436
以前のリリースのユーザー定義入力形式を SAS 9.3 用に変換する	437
以前のリリースのユーザー定義入力形式から SAS 9.3 用への変換について ..	437
バージョン 6 のユーザー定義入力形式の変換	437
バージョン 5 のユーザー定義入力形式の変換	437
ディクショナリ	438
HEXw 入力形式: Windows	438
\$HEXw 入力形式: Windows	438
IBw.d 入力形式: Windows	439
PDw.d 入力形式: Windows	440
PIBw.d 入力形式: Windows	441
RBw.d 入力形式: Windows	443
ZDw.d 入力形式: Windows	443

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 入力形式は完全に *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](#)” (436 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 9.3 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 9.3 用に変換する](#)” (437 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 をはじめとする多くの UNIX システムでは、バイトは big-endian と呼ばれる順序で格納されます。Intel ベース PC や IBM 互換 PC、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。

- *IBw.d*、*PDw.d*、*PIBw.d* および *RBw.d* 入出力形式は、ネイティブモード、すなわち、コンピュータ標準であるバイト順序システムを使用して読み込みと書き込みを行います。
- *S370FIBw.d*、*S370FPDw.d*、*S370FRBw.d* および *S370FPIBw.d* 入出力形式は、コンピュータのネイティブモードにかかわらず、IBM 370 標準に基づく読み込みと書き込みを行います。これらの入力形式と出力形式を使用すると、数字データがどれだけ保存されているにかかわらず、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、*PIBw.d* 出力形式に基づいてデータを書き込むプログラムを一例として説明します。このプログラムを PC 上で実行すると、データはバイト反転(リトルエンディアン)で書き出されます。その後、*PIBw.d* 入力形式を使用する他の SAS プログラムを実行するマイクロコンピュータ上で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 PC 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 シリーズのマイクロ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、*S370FPIBw.d* 出力形式でデータを書き込みます。これで、マイクロコンピュータでも、IBM 370 モードでデータの保存が可能になります。それから、*S370FPIBw.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 9.3 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 9.3 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 9.3 形式に変換してから SAS 9.3 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 9.3 で、PROC FORMAT ステートメントに CNTLIN=オプションを指定し、SAS 9.3 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 9.3 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 9.3 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 9.3 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 にアクセスする必要があります。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 9.3 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

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 CIMPORT プロシジャを使用して SAS 9.3 形式に変換することをお勧めします。

セットを転送します。次に、SAS 9.3 で PROC FORMAT ステートメントに CNTLIN= オプションを指定して、SAS 9.3 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

HEXw.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。

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. 入力形式” (SAS 出力形式と入力形式: リファレンス)

構文

HEXw.

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

HEXw.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)。これは、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ために HEXw.入力形式を使用できることを意味します。

関連項目:

- “\$HEXw.入力形式: Windows” (438 ページ)
- “HEXw 出力形式: Windows” (397 ページ)

\$HEXw.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. 入力形式” (SAS 出力形式と入力形式: リファレンス)

構文

\$HEXw.

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

\$HEXw.入力形式は、各 16 進数文字が 1 バイトを占める値を読み込むという点で HEXw 入力形式と似ています。入力データが印刷可能な文字に限定されている場合は、\$HEXw.入力形式を使用して 16 進情報を文字変数にエンコードします。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” (438 ページ)
- “\$HEXw 出力形式: Windows” (398 ページ)

IBw.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 入力形式” (SAS 出力形式と入力形式: リファレンス)

構文

IBw.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 で除算されます。

IBw.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。

PC の固定小数点数値の詳細については、Intel 社の開発者向け Web サイトを参照してください。

比較

IBw.d 入力形式および PIBw.d 入力形式は異なる結果を返します。IBw.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、PIBw.d 入力形式は正の数だけに使われ、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

01 80

のバイトスワップ値が含まれるとします。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:

- [“IBwd 出力形式: Windows” \(399 ページ\)](#)
- [“バイナリデータの読み取り” \(436 ページ\)](#)

PDw.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 入力形式” (SAS 出力形式と入力形式: リファレンス)

構文

PDw.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:

- [“PDwd 出力形式: Windows” \(400 ページ\)](#)
- [“バイナリデータの読み取り” \(436 ページ\)](#)

PIBw.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 入力形式” (SAS 出力形式と入力形式: リファレンス)

構文

PIBw.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” (439 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

PIBw.d 入力形式および IBw.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。IBw.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、PIBw.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 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:

- “PIBwd 出力形式: Windows” (402 ページ)
- “バイナリデータの読み取り” (436 ページ)

RBw.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 入力形式” (SAS 出力形式と入力形式: リファレンス)

構文

RBw.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

RBw.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。科学的計算では、数値は通常、浮動小数点数で格納します (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:

- “RBwd 出力形式: Windows” (403 ページ)
- “バイナリデータの読み取り” (436 ページ)

ZDw.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 入力形式” (SAS 出力形式と入力形式: リファレンス)

構文

ZDw.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 20.1 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:

[“ZDwd 出力形式: Windows” \(404 ページ\)](#)

21 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	445
ディクショナリ	445
CATALOG プロシジャ: Windows	445
CIMPORT プロシジャ: Windows	446
CONTENTS プロシジャ: Windows	447
CONVERT プロシジャ: Windows	449
CPORT プロシジャ: Windows	451
DATASETS プロシジャ: Windows	452
OPTIONS プロシジャ: Windows	453
PMENU プロシジャ: Windows	454
PRINTTO プロシジャ: Windows	455
SORT プロシジャ: Windows	457

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TRIES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: “CATALOG プロシジャ” (*Base SAS プロシジャガイド*)

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。完全な構文と説明については、CATALOG プロシジャを参照してください。参照先: “CATALOG プロシジャ” (Base SAS プロシジャガイド)

ファイル参照

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した場合、ファイル MYFILE.LST (SASUSER.PROFILE の内容のリストを含む)が作成され、作業ディレクトリに保存されます。

```
proc catalog catalog=sasuser.profile;
contents file=myfile;
run;
```

CIMPORT プロシジャ: Windows

CIMPORT 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: “CIMPORT プロシジャ” (Base SAS プロシジャガイド)

構文

PROC CIMPORT *宛先=libref* | *<libref>* *member-name* <オプション>;

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。完全な構文と説明については、の CIMPORT プロシジャ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、CIMPORT 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 CIMPORT プロシジャを組み合わせると、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进行处理しますが、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

- 移送するファイルが、CPORT プロシジャによってバイナリ形式で作成されたファイルでない場合、CIMPORT プロシジャはそのファイルを読み込むことができず、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” (451 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> および *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: . “CONTENTS プロシジャ” (*Base SAS プロシジャガイド*)

構文

CONTENTS プロシジャ <オプション>;

必須引数

オプション

利用可能なオプションの説明については、*Base SAS プロシジャガイド* の CONTENTS プロシジャ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。ただし、Engine/Host Dependent Information の出力は、動作環境およびエンジン

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 21.1 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” (216 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* <オプション>

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 デクシヨ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 ファイルのデクシヨ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” (450 ページ) を参照してください。BMDP、OSIRIS、SPSS Engine では、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ます。このため、変更によって各システムファイルと、現バージョンの CONVERT プロシジャ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、末尾以外の空白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 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: “CPORT プロシジャ” (Base SAS プロシジャガイド)

構文

```
PROC CPORT source-type=libref| <libref> member-name<オプション>;
```

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: “CPORT プロシジャ” (Base SAS プロシジャガイド)

ライブラリ参照

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” (446 ページ)
- "オペレーティングシステム間の SAS ファイルの移動"については、SAS のファイルの使用を参照してください。参照先: *Windows 版 SAS*
- MIGRATE プロシジャとリリース間の互換性については、次を参照してください。
<http://www.sas.com/sas.com/migration>

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: “DATASETS プロシジャ” (*Base SAS プロシジャガイド*)

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、*Base SAS プロシジャガイド*の DATASETS プロシジャ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 21.1 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” \(447 ページ\)](#)

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: [“OPTIONS プロシジャ” \(SAS システムオプション: リファレンス\)](#)

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、*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。

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 21.2 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 行ですが、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](#)” (493 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](#)” (493 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: “PMENU プロシジャ” (*Base SAS プロシジャガイド*)

構文

```

PROC PMENU <CATALOG=<libref.> catalog> <DESC 'entry-description'>;
  ITEM command <オプション>;
  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、“PMENU プロシジャ” (*Base SAS プロシジャガイド*)の PMENU プロシジャ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は、編集メニューで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: “PRINTTO プロシジャ” (*Base SAS プロシジャガイド*)

構文

PROC PRINTTO <オプション>;

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。
- 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。完全な構文と説明については、PRINTTO プロシジャを参照してください。参照先: *Base SAS プロシジャガイド*

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: “SORT プロシジャ” (*Base SAS プロシジャガイド*)

構文

```
PROC SORT <オプション> <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: *Base SAS プロシジャガイド*

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。

nK
メモリの量を 1KB 単位で指定します。

nM
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 256MB 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 プロシジャに関する詳細については、*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。独自の交換テーブルを作成すると、交換テーブルは 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 エクスプローラーを用いて 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;
```

代わりに、ビューメニューからライブラリを選択し、Sashelp ライブラリをダブルクリックしてから、HOST カタログをダブルクリックします。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 の入力エントリは照合順序です。

特定の変換テーブルの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](#)” (595 ページ)、“[SORTCUT システムオプション: Windows](#)” (591 ページ)および“[SORTCUTP システムオプション: Windows](#)” (592 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](#)” (593 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 21.1 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

タスク	SORTANOM オプション
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](#)” (590 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](#)” (594 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](#)” (595 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 オプションに関する詳細については、BY ステートメント(*SAS ステートメント: リファレンス* 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 21.2 ダミーの BY 変数を使用してビューを作成する

<pre>The SAS System Obs name age 1 ALBERT 10 2 anne 35 3 BRIAN 45 4 bridget 23 5 janet 5 6 JUAN 90</pre>
--

関連項目:

- “TRANTAB プロシジャ” (SAS 各国語サポート(NLS): リファレンスガイド)
- “SORTANOM システムオプション: Windows” (590 ページ)
- “SORTCUT システムオプション: Windows” (591 ページ)
- “SORTCUTP システムオプション: Windows” (592 ページ)
- “SORTDEV システムオプション: Windows” (593 ページ)
- “SORTPARM システムオプション: Windows” (594 ページ)
- “SORTPGM システムオプション: Windows” (595 ページ)
- “SORTSIZE システムオプション: Windows” (596 ページ)

- SORT プロシジャのパフォーマンス向上 (パフォーマンス考慮)を参照してください。
参照先: *Windows 版 SAS*

22 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	465
ディクショナリ	465
ABORT ステートメント: Windows	465
ATTRIB ステートメント: Windows	466
FILE ステートメント: Windows	467
FILENAME ステートメント: Windows	469
FOOTNOTE ステートメント: Windows	475
%INCLUDE ステートメント: Windows	476
INFILE ステートメント: Windows	478
LENGTH ステートメント: Windows	480
LIBNAME ステートメント: Windows	481
SYSTASK ステートメント: Windows	484
TITLE ステートメント: Windows	487
WAITFOR ステートメント: Windows	487
X ステートメント: Windows	488

Windows 版に固有の SAS ステートメント

SAS ステートメントは、SAS への一種類のディレクティブです。SAS ステートメントを使用して、特定の操作の実行を要求したり、次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 ステートメント” (*SAS ステートメント: リファレンス*)

構文

ABORT <ABEND | RETURN> <n>;

必須引数

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。

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:

[“リターンコードと完了ステータス” \(643 ページ\)](#)

ATTRIB ステートメント: Windows

変数に出力形式、入力形式、ラベル、長さに関連付けます。

該当要素: DATA ステップ

Windows 固有: 長さの指定

参照項目: “ATTRIB ステートメント” (SAS ステートメント: リファレンス)

構文

ATTRIB *variable-list-1* 属性リスト-1...<*variable-list-n* 属性リスト-*n*>;

必須引数

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 ステートメント” (SAS ステートメント: リファレンス) 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。

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

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 ステートメント” (SAS ステートメント: リファレンス)

構文

FILE *file-specification*<ENCODING=*エンコーディング値*> <*option-list*> <*host-option-list*>;

必須引数

ファイル指定

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ファイル指定形式のいずれかを使用します。

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。

ENCODING=*エンコーディング値*

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を参照してください。(SAS 各国語サポート(NLS): リファレンスガイド)

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 を使用した他のアプリケーションの制御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。

IGNOREDOEOF

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=レコード長

レコード長をバイト単位で指定します。Windows 版では、デフォルト値は 256 です。レコード長には、1 から 1,073,741,823 (1GB)の値を指定できます。

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 オプションの使用を参照してください。

RECFM=レコード形式

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 256 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 可変長形式を指定します。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” \(302 ページ\)](#)
- [Windows 版 SAS 動的データ交換の使用の DDE の例を参照してください](#)

FILENAME ステートメント: Windows

外部ファイルまたは論理ファイルデバイスに SAS ファイル参照名を関連付けます。

該当要素: SAS プログラム内の任意の場所

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 ステートメント” (SAS ステートメント: リファレンス)

構文

FILENAME *ファイル参照* <デバイスタイプ> '外部ファイル' <ENCODING='エンコーディング値'> <host-

FILENAME *fileref device-type* <外部ファイル> <ENCODING=エンコーディング値> <host-option-list>

FILENAME ファイル参照<デバイスタイプ> ('ディレクトリ-1'<... ディレクトリ-n'>) <ENCODING=エンコーディング値> <host-option-list>;

必須引数

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 ステートメント” (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。

ファイル参照

有効なファイル参照名(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)。

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。

メンバ名構文(集計構文)でのファイル参照名の使用例については、外部ファイルの使用のディレクトリへのファイル参照名の割り当てを参照してください。参照先: Windows 版 SAS。SAS でファイル参照名によるファイルへのアクセス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” (158 ページ)を参照してください。

デバイスタイプ

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 SSH クライアントソフトウェアを構成 次の値が有効な値です。

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。

COMMPORT

通信ポートからデータを読み込み、書き込みます。

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” (286 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。

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。

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](#)” (300 ページ)を参照してください。

NOTESDB

Lotus Notes データベースにデータを書き込みます。詳細については、“[DATA ステップと SCL コードを使用し、Lotus Notes データベースに入力する](#)” (222 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](#)” (298 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](#)” (605 ページ)を参照してください。

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](#)” (165 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](#))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。また、Windows 版では、デバイスタイプキーワードとして DISK を指定することはできますが、ディスクファイルが標準であるため、無視されます。

外部ファイル

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](#)” (150 ページ)を参照してください。

ENCODING='エンコーディング値

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、を参照してください。これらのオプションを使用する方法については、SAS 言語要素中のエンコーディング値を で参照してください。(SAS 各国語サポート(NLS): リファレンスガイド)

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。

BAUD=

ボーレートを設定します。ボーレートの値は、通信機器によって異なります。値は整数にする必要があります。このホストオプションは、デバイスタイプキーワードの COMMPORT を指定しているときにのみ有効です。

BITS=

送信ビットを設定します。値の範囲は 5 - 8 です。このホストオプションは、デバイスタイプキーワードの COMMPORT を指定しているときにのみ有効です。

8250 シリアルポートの場合、5 データビットと 2 ストップビット、6、7、8 データビットと 1.5 ストップビットの組み合わせは無効です。

BLKSIZE=block-sizeBLK=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” (287 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” (288 ページ)を参照してください。

COMTIMEOUT=値

通信ポートのタイムアウトの処理方法を制御します。通信ポートで一定時間(通常は 60 秒間)データ処理がないと、タイムアウトになります。COMTIMEOUT=オプションには、次の値を指定できます。

- EOF** タイムアウト発生時にファイル終端(EOF)文字を返します。この動作はデフォルトです。EOF 文字が返されると、現在の DATA ステップは終了します。
- WAIT** データを受け取るまで待機するように通信ポートに指示します。つまり、この値を指定すると、タイムアウトが無効になります。この場合は、通信ポートにデータが送られるまで、DATA ステップにはレコードが返されません。プログラムが無限ループになる可能性があるため、この値の使用には注意が必要です。
- ZERO** タイムアウト発生時にレコード長 0 バイトを返します。ただし、DATA ステップは終了せず、データの読み取りが再試行されません。

このホストオプションは、デバイスタイプキーワードの COMMPORT を指定しているときにのみ有効です。

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。

DROPNUL=

受信時にヌルバイトを削除します。次の値が有効です。

- ON** 受信時にヌルバイトを削除します。
- OFF** 受信時にヌルバイトを削除しません。デフォルト値は OFF です。

このホストオプションは、デバイスタイプキーワードの COMMPORT を指定しているときにのみ有効です。例を次に示します。

```
filename portin commport 'com1:' droppnull=off;
```

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。

FLOW=

送信制御のフローを制御します。有効な値は、XONXOFF、DTRDSR、RTSCTS です。このホストオプションは、デバイスタイプキーワードの COMMPORT を指定しているときにのみ有効です。

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](#)”(292 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=レコード長

レコード長をバイト単位で指定します。Windows 版では、デフォルト値は 256 です。レコード長には、1 から 1,073,741,823 (1GB)の値を指定できます。

また、の使用例を参考にして、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、“[DDE で NOTAB オプションを使用する](#)” (291 ページ)を参照してください。

PARITY=

パリティチェックビットを設定します。値には、NONE、ODD、EVEN、MARK、SPACE を指定できます。このホストオプションは、デバイスタイプキーワードの COMMPORT を指定しているときにのみ有効です。

RECFM=レコード形式

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、デフォルト設定では、一度に 256 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 | 可変長形式を指定します。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。

RCONST=*seconds*

初回の読み取りタイムアウト値を、1000 分の 1 秒単位で指定します。デフォルトは 8 秒です。このホストオプションは、デバイスタイプキーワードの COMMPORT を指定しているときにのみ有効です。

RMULTI=seconds

2 回目以降の読み取りタイムアウト値を、1000 分の 1 秒単位で指定します。このホストオプションは、デバイスタイプキーワードの COMMPORT を指定しているときにのみ有効です。

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。

STOP=

ストップビットを設定します。値には、ONE、TWO、ONEHALF を指定できません。このホストオプションは、デバイスタイプキーワードの COMMPORT を指定しているときにのみ有効です。

8250 シリアルポートの場合、5 データビットと 2 ストップビット、6、7、8 データビットと 1.5 ストップビットの組み合わせは無効です。

WCONST=seconds

初回のタイムアウト値を、1000 分の 1 秒単位で指定します。このホストオプションは、デバイスタイプキーワードの COMMPORT を指定しているときにのみ有効です。

WMULTI=seconds

2 回目以降のタイムアウト値を、1000 分の 1 秒単位で指定します。このホストオプションは、デバイスタイプキーワードの COMMPORT を指定しているときにのみ有効です。

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” \(165 ページ\)](#)

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 ステートメント” (SAS ステートメント: リファレンス)

構文

FOOTNOTE <n> <'テキスト'|“テキスト”>;

必須引数

n
フットノートに使用する行を指定します。

テキスト

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 ステートメント” (SAS ステートメント: リファレンス)

構文

%INCLUDE ソース </<ENCODING='エンコーディング値'> <ホストオプション> >;

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 ステートメント” (SAS ステートメント: リファレンス) の %INCLUDE ステートメントを参照してください。

ソース

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。Windows 版では、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。“[外部ファイルの参照](#)” (150 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。

ENCODING='エンコーディング値'

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。

有効なエンコードの値については、20 章: “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。

ホストオプション

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=ブロックサイズ BLK=ブロックサイズ

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。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。

IGNOREDOEOF

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=レコード長

レコード長をバイト単位で指定します。Windows 版では、デフォルト値は 256 です。レコード長には、1 から 1,073,741,823 (1GB)の値を指定できます。

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](#)” (291 ページ)を参照してください。

RECFM=レコード形式

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 256 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 可変長形式を指定します。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せん。

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 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。

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 ステートメント”(SAS ステートメント: リファレンス)

構文

INFILE *ファイル指定* <ENCODING=*エンコーディング値*> <*options*> <*ホストオプション*>;

必須引数

ファイル指定

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。ファイル指定引数には“外部ファイルの参照”(150 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。

ENCODING=*エンコーディング値*

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、20 章: “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。

ホストオプション

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=ブロックサイズ **BLK=ブロックサイズ**

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](#)” (292 ページ)を参照してください。

IGNOREDOEOF

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=レコード長

レコード長をバイト単位で指定します。Windows 版では、デフォルト値は 256 です。レコード長には、1 から 1,073,741,823 (1GB)の値を指定できます。

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](#)” (291 ページ)を参照してください。

RECFM=レコード形式

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 256 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 可変長形式を指定します。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” (302 ページ)
- “DDE の例” (288 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 ステートメント” (SAS ステートメント: リファレンス)

構文

```
LENGTH <variable-1> <...variable-n> <$> <長さ> <DEFAULT=n>;
```

必須引数

長さ

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 版での変数の長さ精度” \(621 ページ\)](#)

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 ステートメント” (SAS ステートメント: リファレンス)

構文

```
LIBNAME libref<engine> ('(SAS-data-library-1' <,...'SAS-data-library-n'> )' <MEMLIB> ,  
<FILELOCKWAIT> ;
```

```
LIBNAME ライブラリ参照 _ALL_ LIST;
```

```
LIBNAME ライブラリ参照 _ALL_ CLEAR;
```

必須引数

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 ステートメント” (SAS ステートメント: リファレンス)

ライブラリ参照

有効なライブラリ参照名を指定します。SAS ステートメント: リファレンスに記載されています。

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、SAS 9.3 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 を使用できません。
- 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せん。
- 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” (124 ページ) を参照してください。

SAS- データライブラリ

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カッコで囲みます。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” (134 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。ESMA 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” (208 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きません。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 0
オル
ト

範 0 – 600
囲

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。

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” (535 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” (132 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 ライブラリ参照 <engine> 'SAS- データライブラリ

LIBNAME ライブラリ参照 <engine> ('SAS- データライブラリ-1' <,...'SAS- データライブラリ-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 22.1 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\SAS\SASFoundation\9.2\nls\en\SASCFG
File Name= C:\Program Files\SAS\SASFoundation\9.2\nls\en\SASCFG
-Level 2-
Engine= V9
Physical Name= C:\Program Files\SAS\SASFoundation\9.2\core\sashelp
File Name= C:\Program Files\SAS\SASFoundation\9.2\core\sashelp
. . .
-Level 27-
Engine= V9
Physical Name= C:\Program Files\SAS\SASFoundation\9.2\webhound\sashelp
File Name= C:\Program Files\SAS\SASFoundation\9.2\webhound\sashelp
2 run;

```

関連項目:

- “LIBNAME ステートメント, SASEDOC” (*SAS Output Delivery System: ユーザーガイド*)
- “LIBNAME ステートメント” (*SAS ステートメント: リファレンス*)

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 “オペレーティングシステムコマンド” <WAIT | NOWAIT> <TASKNAME=タスク名> <MNAME=name-var> <STATUS=stat-var> <SHELL=<=“シェルコマンド”>> ;

SYSTASK LIST <_ALL_ | タスク名> <STATE> <STATVAR> ;

SYSTASK KILL タスク名 <タスク名...> ;

必須引数**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。

オペレーティングシステムコマンド

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レ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=タスク名

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-var

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-var

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<="シェルコマンド">

オペレ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 myprogl.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acro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は、`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](#)” (487 ページ)
- “[X ステートメント: Windows](#)” (488 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 ステートメント” (SAS ステートメント: リファレンス)

構文

```
TITLE <n> <'テキスト'|“テキスト”>;
```

必須引数

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_> タスク名<タスク名...> <TIMEOUT=seconds>;
```

必須引数

タスク名

待機対象のタスクの名前を指定します。タスク名の詳細は、“SYSTASK ステートメント: Windows” (484 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する場合は、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、次のような場合に、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” (484 ページ)
- “X ステートメント: Windows” (488 ページ)
- “XWAIT システムオプション: Windows” (619 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 ステートメント” (SAS ステートメント: リファレンス)

構文

X <'コマンド'>;

必須引数

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もあります。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。

コマンド

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は、[“XWAIT システムオプション: Windows” \(619 ページ\)](#)および[“XSYNC システムオプション: Windows” \(618 ページ\)](#)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” \(392 ページ\)](#)
- [“XSYNC システムオプション: Windows” \(618 ページ\)](#)
- [“XWAIT システムオプション: Windows” \(619 ページ\)](#)
- [“CALL SYSTEM ルーチン: Windows” \(409 ページ\)](#)

- SYSEXEC ステートメント“マクロステートメント” (628 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” (68 ページ)

23 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	493
Windows 環境下での SAS システムオプションの概要	493
制限オプション	494
SAS システムオプション設定の表示	494
SAS システムオプション設定の変更	495
概要: SAS システムオプション設定の変更	495
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	495
SAS システムオプションでライブラリを連結する構文	496
OPTIONS ステートメントのシステムオプションの構文	496
複数の場所に設定されたシステムオプションの処理	497
Windows のシステムオプション概要	497
ディクショナリ	515
ACCESSIBILITY システムオプション: Windows	515
ALTLOG システムオプション: Windows	516
ALTPRINT システムオプション: Windows	517
APPEND システムオプション: Windows	518
AUTHPROVIDERDOMAIN システムオプション: Windows	519
AUTHSERVER システムオプション: Windows	519
AUTOEXEC システムオプション: Windows	520
AWSCONTROL システムオプション: ウィンドウ	521
AWSDEF システムオプション: Windows	522
AWSMENU システムオプション: ウィンドウ	523
AWSMENUMERGE システムオプション: ウィンドウ	523
AWSTITLE システムオプション: Windows	524
BUFNO システムオプション: Windows	524
BUFSIZE システムオプション: Windows	526
CATCACHE システムオプション: Windows	527
CLEANUP システムオプション: Windows	527
COMDEF システムオプション: Windows	528
CONFIG システムオプション: Windows	529
DEVICE システムオプション: Windows	530
ECHO システムオプション: Windows	531
EMAILDLG システムオプション: Windows	532
EMAILSYS システムオプション: Windows	532
ENGINE システムオプション: Windows	533
ENHANCEDEDITOR システムオプション: Windows	534
FILELOCKWAITMAX=システムオプション: Windows	535
FILTERLIST システムオプション: Windows	536
FONT システムオプション: Windows	537

FONTALIAS システムオプション: Windows	538
FONTSLC システムオプション: Windows	539
FORMCHAR システムオプション: Windows	540
FULLSTIMER システムオプション: Windows	541
HELPHOST システムオプション: Windows	542
HELPINDEX システムオプション: Windows	543
HELPLOC システムオプション: Windows	544
HELPREGISTER システムオプション: Windows	545
HELPTOC システムオプション: Windows	547
HOSTPRINT システムオプション: Windows	548
ICON システムオプション: Windows	549
INITSTMT システムオプション: Windows	549
INSERT システムオプション: Windows	550
JREOPTIONS システムオプション: Windows	551
LINESIZE システムオプション: Windows	552
LOADMEMSIZE システムオプション: Windows	553
LOG システムオプション: Windows	554
MAPS システムオプション: Windows	555
MAXMEMQUERY システムオプション: Windows	556
MEMBLKSZ システムオプション: Windows	557
MEMCACHE システムオプション: Windows	558
MEMLIB システムオプション: Windows	559
MEMMAXSZ システムオプション: Windows	559
MEMSIZE システムオプション: Windows	560
MSG システムオプション: Windows	562
MSGCASE システムオプション: Windows	562
MSYMTABMAX システムオプション: Windows	563
MVARSIZE システムオプション: Windows	564
NEWS システムオプション: Windows	565
NUMKEYS システムオプション: Windows	565
NUMMOUSEKEYS システムオプション: Windows	566
OBS システムオプション: Windows	567
PAGENO システムオプション: Windows	568
PAGESIZE システムオプション: Windows	568
PAPERTYPE システムオプション: Windows	569
PATH システムオプション: Windows	570
PFKEY システムオプション: Windows	571
PRIMARYPROVIDERDOMAIN=システムオプション: Windows	573
PRINT システムオプション: Windows	574
PRNGETLIST システムオプション: Windows	574
PRTABORTDLGS システムオプション: Windows	575
P RTPERSISTDEFAULT システムオプション: Windows	576
PRTSETFORMS システムオプション: Windows	577
REALMEMSIZE システムオプション: Windows	578
REGISTER システムオプション: Windows	579
RESOURCESLOC システムオプション: Windows	579
RSASUSER システムオプション: Windows	580
RTRACE システムオプション: Windows	581
RTRACELOC システムオプション: Windows	582
SASAUTOS システムオプション: Windows	582
SASCONTROL システムオプション: Windows	583
SASHELP システムオプション: Windows	584
SASINITIALFOLDER システムオプション: Windows	585
SASUSER システムオプション: Windows	586
SCROLLBARFLASH システムオプション: Windows	587
SET システムオプション: Windows	587

SGIO システムオプション: Windows	588
SLEEPWINDOW システムオプション: Windows	590
SORTANOM システムオプション: Windows	590
SORTCUT システムオプション: Windows	591
SORTCUTP システムオプション: Windows	592
SORTDEV システムオプション: Windows	593
SORTNAME システムオプション: Windows	594
SORTPARM システムオプション: Windows	594
SORTPGM システムオプション: Windows	595
SORTSIZE システムオプション: Windows	596
SPLASH システムオプション: Windows	597
SPLASHLOC システムオプション: Windows	597
STIMEFMT システムオプション: Windows	598
STIMER システムオプション: Windows	602
SYSGUIFONT システムオプション: Windows	603
SYSIN システムオプション: Windows	604
SYS Parm システムオプション: Windows	604
SYS PRINT システムオプション: Windows	605
SYS PRINT FONT システムオプション: Windows	606
TOOLDEF システムオプション: Windows	609
UNIVERSALPRINT システムオプション	609
UPRINTMENUMSWITCH システムオプション: Windows	611
USER システムオプション: Windows	611
USERICON システムオプション: Windows	612
VERBOSE システムオプション: Windows	613
WEBUI システムオプション: Windows	614
WINDOWSMENU システムオプション: Windows	614
WORK システムオプション: Windows	615
XCMD システムオプション: Windows	616
XMIN システムオプション: Windows	617
XSYNC システムオプション: Windows	618
XWAIT システムオプション: Windows	619

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 (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) の使用、OPTIONS ステートメント (SAS プログラム上または SAS autoexec ファイル上のどちらか)、システムオプションウインドウまたは SCL プログラム上 “Windows のシステムオプション概要” (497 ページ) では、各 SAS システムオプションがどこで設定できるかの情報を提供します。

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、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。制限オプションに関する更に詳細な情報は“制限されたオプション”(SAS システムオプション: リファレンス 1 章)を参照してください。

- COMDEF
- FILELOCKWAITMAX
- MEMCACHE
- MEMLIB
- PATH
- RESOURCESLOC
- SASCONTROL
- SGIO
- TOOLDEF

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 ⇒ システムと選択するか、コマンドエリアにて options を入力して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。この 2 つの場合では構文が異なり、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 ステートメントに関する詳細情報は *SAS ステートメント: リファレンス* を参照してください。

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 SAS コマンドは、SAS を起動し自動呼出しマクロが C:\SAS\CORE\SASMACRO ディレクトリに保存されていることを示します。

```
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 構成ファイルに関する詳細な情報は“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、全体をカッコで囲みます。

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 ステートメントに関する詳細情報は *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。OPTION ステートメントにてシステムオプションが指定できるかどうかを確認するには、表 23.1 (498 ページ) でオプション名検索してください。オプションの指定箇所や PROC OPTIONS

```
option=optname define;
run;
```

の使用など SAS のシステムオプションに関するすべての情報がそこに記載されています。

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。

Windows のシステムオプション概要

表 23.1 (498 ページ) では、Windows 版の SAS を使用するユーザー向けに提供されているすべてのシステムオプションが記載されています。これらのオプションの多くはシステムに依存する動作を持たず、SAS システムオプション: リファレンスにて詳細に説明されています。それ以外のオプションは、Windows でのみ使用できるものであり、すべてここで説明されています。いくつかのシステムオプションは次と SAS システムオプション: リファレンスにて説明されています。

注: SAS システムオプション: リファレンスでのいくつかのシステムオプションは、システムオプションが更なる操作環境の情報をもち、操作環境のための SAS ドキュメントを参照していることを示します。そのようなシステムオプションが Windows 版 SAS で説明されていない場合は、SAS システムオプション: リファレンスで説明されているようにシステムオプションを使用します。

システムオプションに関する詳細情報の参照先は、次の凡例を参考してください。

アクセス

SAS/ACCESS for Relational Databases: Reference

AppSrv

SAS Intelligence Platform: Application Server Administration Guide

ARM

SAS Interface to Application Response Measurement (ARM): Reference

CAM

Communications Access Methods for SAS/CONNECT and SAS/SHARE

Comp

このセクションにてオプションが完全に説明されていることを意味します。

Conn	<i>SAS/CONNECT User's Guide</i>
DQ	<i>SAS Data Quality Server: Reference</i>
DST	<i>SAS の暗号化</i>
ログ	<i>SAS ログ機能: 構成とプログラミングリファレンス</i>
SOR	<i>SAS システムオプション: リファレンス</i>
MACRO	<i>SAS マクロ言語: リファレンス</i>
Meta	<i>SAS Language Interfaces to Metadata</i>
NLS	<i>SAS 各国語サポート(NLS): リファレンスガイド</i>
Share	<i>SAS/SHARE User's Guide</i>
SPD エンジン	<i>SAS Scalable Performance Data Engine: リファレンス</i>
Web	オプションが SAS ウェブサイトに掲示されているドキュメントに説明されていることを示します(support.sas.com)。

表 23.1 SAS システムオプションの要約

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
ACCESSIBILITY	STANDARD	X	X			Comp
ALTLOG 変数	NOALTLOG	X	X			Comp
ALTPRINT 変数	NOALTPRINT	X	X			Comp
APPEND	なし	X	X	X	X	SOR
APPLETLOC	なし	X	X	X	X	SOR
ARMAGENT	なし	X	X	X	X	ARM
ARMLOC	ARMLOG.LOG	X	X	X	X	ARM
ARMSUBSYS	ARM_NONE	X	X	X	X	ARM
AUTHPROVIDER-DOMAIN	AUTHPP	X	X			SOR、Comp
AUTHSERVER	ローカル接続で信頼できるサーバー	X	X	X	X	Comp

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
AUTOEXE 変数	ファイルがあれば AUTOEXEC.SAS。そうでないときはなし。	X	X			Comp
AUTOSAVELOC	なし	X	X	X	X	SOR
AUTOSIGNON	NOAUTOSIGNON	X	X	X	X	Conn
AWSCONTROL	TITLE SYSTEMMENU MINMAX	X	X	X	X	Comp
AWSDEF	80%の画面高さ と幅	X	X	X	X	Comp
AWSMENU	AWSMENU	X	X	X	X	Comp
AWSMENUMERGE	AWSMENUMERGE	X	X	X	X	Comp
AWSTITLE 引数	なし	X	X			Comp
BINDING 引数	DEFAULT	X	X	X	X	SOR
BOMFILE	BOMFILE	X	X	X	X	NLS
BOTTOMMARGIN	0.000IN	X	X	X	X	SOR
BUFNO 引数	1	X	X	X	X	SOR、 Comp
BUFSIZE 引数	0	X	X	X	X	SOR、 Comp
BYERR	BYERR	X	X	X	X	SOR
BYLINE	BYLINE	X	X	X	X	SOR
BYSORTED	BYSORTED	X	X	X	X	SOR
CAPS	NOCAPS	X	X	X	X	SOR
CARDIMAGE	NOCARDIMAGE	X	X	X	X	SOR
CATCACHE 引数	0	X	X			SOR、 Comp
CBUFNO 引数	0	X	X	X	X	SOR
CENTER	CENTER	X	X	X	X	SOR
CGOPTIMIZE	3	X	X	X	X	SOR

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
CHARCODE	NOCHARCODE	X	X	X	X	SOR
CLEANUP	CLEANUP	X	X	X	X	SOR、 Comp
CMDMAC	NOCMDMAC	X	X	X	X	マクロ
COMPLIB 引数	なし	X	X	X	X	SOR
CMPMODEL	BOTH	X	X	X	X	SOR
CMPOPT 引数	<ul style="list-style-type: none"> • NOEXTRAMATH • NOMISSCHECK • NOPRECISE • NOGUARDCHECK • NOFUNCDIFFERENCING 	X	X	X	X	SOR
COLLATE	NOCOLLATE	X	X	X	X	SOR
COLORPRINTING	COLORPRINTING	X	X	X	X	SOR
COMAMID 引数	TCP	X	X		X	接続、 共有、 CAM
COMAUX1 引数	なし	X	X			CAM
COMAUX2 引数	なし	X	X			CAM
COMDEF 引数	BOTTOM CENTER	X	X			Comp
COMPRESS 引数	NO	X	X	X	X	SOR
CONFIG 引数	!sasroot \SASV9.CFG	X	X			Comp
CONNECTPERSIST	YES	X	X			Conn
CONNECTREMOTE 引数	なし	X	X	X	X	Conn
CONNECTSTATUS	CONNECTSTATUS	X	X	X	X	Conn
CONNECTWAIT	CONNECTWAIT	X	X	X	X	Conn
COPIES 引数	1	X	X	X	X	SOR

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
CPUCOUNT 引数	1	X	X	X	X	SOR
CPUID	CPUID	X	X			SOR
DATASTMTCHK 引数	COREKEYWORDS	X	X	X	X	SOR
DATE	DATE	X	X	X	X	SOR
DATESTYLE	MDY	X	X	X	X	SOR
DBCS	NODBCS	X	X			NLS
DBCSLANG	NODBCSLANG	X	X			NLS
DBCSTYPE	WINDOWS	X	X			NLS
DBIDIRECTEXEC	NODBIDIRECTEXEC	X	X	X	X	アクセス
DBSLICEPARM	THREADED_APPS、2	X	X	X	X	アクセス
DBSRVTP	NONE	X	X			アクセス
DEFLATION	6	X	X	X	X	SOR
DMSEXP	NODMSEXP	X	X			SOR
DMSGMLINESIZE	136	X	X			SOR
DETAILS	NODETAILS	X	X	X	X	SOR
DEVICE 引数	なし	X	X	X	X	SOR、Comp
DFLANG 引数	ENGLISH	X	X	X	X	NLS
DKRCOND 引数	ERROR	X	X	X	X	SOR
DKROCOND 引数	WARN	X	X	X	X	SOR
DLDMGACTION	FAIL はバッチモード、REPAIR はインタラクティブモード	X	X	X	X	SOR
DMR	NODMR	X	X			SOR、Conn
DMS	DMS	X	X			SOR
DMSLOGSIZE 引数	99999	X	X			SOR

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
DMSOUTSIZE 引数	99999	X	X			SOR
DMSSYNCHK	NODMSSYNCHK	X	X	X	X	SOR
DQLOCALE 引数	なし	X	X	X	X	DQ
DQOPTIONS	なし	X	X			DQ
DQSETUPLOC 引数	なし	X	X	X	X	DQ
DSNFERR	DSNFERR	X	X	X	X	SOR
DTRESET	NODTRESET	X	X	X	X	SOR
DUPLEX 引数	NODUPLEX	X	X	X	X	SOR
ECHO 引数	NOECHO	X	X			Comp
ECHOAUTO	NOECHOAUTO	X	X			SOR
EMAILAUTHPROTOCOL 引数	NONE	X	X			SOR
EMAILDLG 引数	NATIVE	X	X			Comp
EMAILFROM	NOEMAILFROM	X	X	X	X	SOR
EMAILHOST 引数	LOCALHOST	X	X			SOR
EMAILID 引数	なし	X	X			SOR
EMAILPORT 引数	25	X	X			SOR
EMAILPW 引数	なし	X	X			SOR
EMAILSYS 引数	MAPI	X	X			Comp
ENCODING	WLATIN1	X	X			NLS
ENGINE 引数	V9	X	X			SOR、Comp
ENHANCEDEDITOR	ENHANCEDEDITOR	X	X			Comp
ERRORABEND	NOERRORABEND	X	X	X	X	SOR
ERRORBYABEND	NOERRORBYABEND	X	X	X	X	SOR
ERRORCHECK 引数	NORMAL	X	X	X	X	SOR
ERRORS 引数	20	X	X	X	X	SOR

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
EXPLORER	NOEXPLORER	X	X			SOR
FILELOCKWAITMAX	600	X	X			Comp
FILESYNC	SAS	X	X			SOR
FILTERLIST 引数	なし	X	X			Comp
FIRSTOBS 引数	1	X	X	X	X	SOR
FMterr	FMterr	X	X	X	X	SOR
FMTSEARCH 引数	WORK ライブラリ	X	X	X	X	SOR
FONT 引数	Sasfont 8	X	X	X	X	Comp
FONTALIAS 引数	可変	X	X			Comp
FONTEMBEDDING	FONTEMBEDDING	X	X	X	X	SOR
FONTRENDERING	FREETYPE-POINTS	X	X	X	X	SOR
FONTSLoc 引数	!sasroot\core\nresource	X	X			SOR、Comp
FORMCHAR 引数	(SASV9.CFG を参照)	X	X	X	X	SOR、Comp
FORMDLIM 引数	なし	X	X	X	X	SOR
FORMS 引数	DEFAULT	X	X	X	X	SOR
FULLSTIMER	NOFULLSTIMER	X	X	X	X	Comp
GSTYLE	GSTYLE	X	X	X	X	SOR
GWINDOW	GWINDOW	X	X	X	X	SOR
HELPADDR	なし	X	X	X	X	Web
HELpbROWSER	REMOTE	X	X	X	X	SOR
HELpENCMD	HELpENCMD	X	X			SOR
HELpHOST	NULL	X	X	X	X	SOR
HELpINDEX 引数	/help/ common.hlp/ index.txt、/help/ common.hlp/ keywords.htm、 common.hhk	X	X			Comp

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
HELPLLOC 引数	("!sasuser \classdoc" "!sasroot\core \help" "!sasroot\nls\en \help")	X	X			SOR、 Comp
HELPPORT	0	X	X	X	X	SOR
HELPPREGISTER 引数	なし	X	X			Comp
HELPTOC 引数	/help/helpnav.hlp/ config.txt /help/ common.hlp/ toc.htm common.hhc	X	X			Comp
HOSTPRINT	HOSTPRINT	X	X	X	X	Comp
HTTPSERVERPORTMAX	0	X	X			SOR
HTTPSERVERPORTMIN	0	X	X			SOR
IBUFNO	0	X	X	X	X	SOR
IBUFSIZE	0	X	X	X	X	SOR
ICON	NOICON	X	X	X	X	Comp
IMPLMAC	NOIMPLMAC	X	X	X	X	Macro
INITCMD 引数	なし	X	X			SOR
INITSTMT 引数	なし	X	X			SOR、 Comp
INSERT	なし	X	X	X	X	SOR
INTERVALDS		X	X	X	X	SOR
INVALIDDATA 引数	ピリオド(.)	X	X	X	X	SOR
IPADDRESS	NOIPADDRESS	X	X			CONN
JPEGQUALITY	75	X	X	X	X	SOR
JREOPTIONS	SASV9.CFG を参照	X	X			Comp
LABEL	LABEL	X	X	X	X	SOR
LAST 引数	_NULL_	X	X	X	X	SOR

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
LEFTMARGIN 引数	0.000IN	X	X	X	X	SOR
LINESIZE 引数	可変	X	X	X	X	SOR、Comp
LOADMEMSIZE	0	X	X			Comp
LOCALE	ENGLISH_UNIT EDSTATES	X	X	X	X	NLS
LOCALELANGCHG	NOLOCALELANGCHG	X	X			NLS
LOG 引数	バッチモードでは ファイル名.LOG	X	X			Comp
LOGAPPLNAME=	なし	X	X			ログ
LOGCONFIGLOC=	なし	X	X			ログ
LOGPARM	WRITE: BUFFERED ROLLOVER	X	X			SOR
LRECL	256	X	X	X	X	SOR
マクロ	マクロ	X	X			マクロ
MAPS 引数	!sasroot\maps	X	X	X	X	SOR、Comp
MAUTOLOCDISPLAY	NOMAUTOLOC DISPLAY	X	X	X	X	マクロ
MAUTOSOURCE	MAUTOSOURCE	X	X	X	X	マクロ
MAXMEMQUERY	0	X	X	X	X	Comp
MAXSEGRATIO	75	X	X	X	X	SPDE
MCOMPILE	MCOMPILE	X	X	X	X	マクロ
MCOMPILENOTE	なし	X	X	X	X	マクロ
MEMBLKSZ	16 MB	X	X			Comp
MEMCACHE	0	X	X	X	X	Comp
MEMLIB	NOMEMLIB	X	X			Comp
MEMMAXSZ	2G	X	X			Comp
MEMSIZE	2G	X	X			Comp

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
MERGENOBY	NOWARN	X	X	X	X	SOR
MERROR	MERROR	X	X	X	X	マクロ
METAAUTORESOURCES	なし	X	X			Meta
METACONNECT	なし	X	X	X	X	Meta
METAENCRYPTALG 引数	SASPROPRIETARY	X	X			Meta
METAENCRYPTLEVEL	CREDENTIALS	X	X			Meta
METAID		X	X			Meta
METAPASS	なし	X	X	X	X	Meta
METAPORT	SASV9.CFG を参照	X	X	X	X	Meta
METAPROFILE	なし	X	X			Meta
METAPROTOCOL	BRIDGE	X	X	X	X	Meta
METAREPOSITORY	SASV9.CFG を参照	X	X	X	X	Meta
METASERVER	SASV9.CFG を参照	X	X	X	X	Meta
METASPN=	なし	X	X	X	X	Meta
METAUSER	なし	X	X	X	X	Meta
MEEXECNOTE	NOMEEXECNOTE	X	X	X	X	マクロ
MEEXCSIZE	65536	X	X	X	X	マクロ
MFILE	NOMFILE	X	X	X	X	マクロ
MINDELIMITER	なし	X	X	X	X	マクロ
MINOPERATOR	NOMINOPERATOR	X	X	X	X	マクロ
MINPARTSIZE	16777216	X	X			SPDE
MISSING 引数	ピリオド(.)	X	X	X	X	SOR
MLOGIC	NOMLOGIC	X	X	X	X	マクロ

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
MLOGICNEST	NOMLOGICNEST	X	X	X	X	マクロ
MPRINT	NOMPRINT	X	X	X	X	マクロ
MPRINTNEST	NOMPRINTNEST	X	X	X	X	マクロ
MRECALL	NOMRECALL	X	X	X	X	マクロ
MREPLACE	MREPLACE	X	X	X	X	マクロ
MSG 引数	!sasroot\core\sasmsg	X	X			Comp
MSGCASE	NOMSGCASE	X	X			Comp
MSGLEVEL 引数	N	X	X	X	X	SOR
MSTORED	NOMSTORED	X	X	X	X	マクロ
MSYMTABMAX 引数	4,194,304 バイト	X	X	X	X	Comp、マクロ
MULTENVAPPL	NOMULTENVAPPL	X	X		X	SOR
MVARSIZE 引数	4,096 バイト	X	X	X	X	Comp、マクロ
NETENCRYPT	NONETENCRYPT	X	X	X	X	DST
NETENCRYPTALGORITHM 引数	なし	X	X	X	X	DST
NETENCRYPTKEYLENGTH 引数	0	X	X	X	X	DST
NEWS 引数	なし	X	X			SOR、Comp
NLSCOMPATMODE	NONLSCOMPAT - MODE	X	X			NLS
NOTES	NOTES	X	X	X	X	SOR
NUMBER	NUMBER	X	X	X	X	SOR
NUMKEYS 引数	可変	X	X			Comp
NUMMOUSEKEYS 引数	3	X	X			Comp

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
OBJECTSERVER	NOOBJECTSERVER	X	X			SOR
OBS 引数	MAX	X	X	X	X	SOR、Comp
ORIENTATION 引数	PORTRAIT	X	X	X	X	SOR
OVP	NOOVP	X	X	X	X	SOR
PAGEBREAKINITIAL	NOPAGEBREAKINITIAL	X	X			SOR
PAGENO 引数	1	X	X	X	X	SOR、Comp
PAGESIZE 引数	場合によって異なる	X	X	X	X	SOR、Comp
PAPERSIZE 引数	LETTER	X	X	X	X	SOR
PAPERSOURCE 引数	なし	X	X	X	X	SOR
PAPERTYPE 引数	PLAIN	X	X	X	X	SOR、Comp
PARM 引数	なし	X	X	X	X	SOR
PARMCARDS 引数	FT15F001	X	X	X	X	SOR
PATH 引数	!sasroot\core\sasexe	X	X			Comp
PDFACCESS	PDFACCESS	X	X	X	X	SOR
PDFASSEMBLY	NOPDFASSEMBLY	X	X	X	X	SOR
PDFCOMMENT	NOPDFCOMMENT	X	X	X	X	SOR
PDFCONTENT	NOPDFCONTENT	X	X	X	X	SOR
PDFCOPY	PDFCOPY	X	X	X	X	SOR
PDFFILLIN	PDFFILLIN	X	X	X	X	SOR
PFKEY 引数	WIN	X	X			Comp
PDFPAGELAYOUT	DEFAULT	X	X	X	X	SOR
PDFPAGEVIEW	DEFAULT	X	X	X	X	SOR
PDFPASSWORD	xxxxxxxx	X	X		X	SOR

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
PDFPRINT	HRES	X	X	X	X	SOR
PDFSECURITY	NONE	X	X	X	X	SOR
PRINT 引数	バッチモードでは ファイル名.LST	X	X			Comp
PRIMARYPROVIDER DOMAIN	PRIMPD=	X	X			SOR
PRINTERPATH 引数	なし	X	X	X	X	SOR
PRINTINIT	NOPRINTINIT	X	X			SOR
PRINTMSGLIST	PRINTMSGLIST	X	X	X	X	SOR
PRNGETLIST	PRNGETLIST	X	X			Comp
PRTABORTDLGS	BOTH	X	X	X	X	Comp
PRTPERSIST DEFAULT	NOPRTPERSIST DEFAULT	X	X			Comp
PRTSETFORMS	PRTSETFORMS	X	X	X	X	Comp
QUOTELENMAX	QUOTELENMAX	X	X	X	X	SOR
REALMEMSIZE	0	X	X			Comp
REGISTER 引数	なし	X	X			Comp
REPLACE	REPLACE	X	X	X	X	SOR
RESOURCESLOC 引数	SASV9.CFG を参照	X	X			Comp
REUSE 引数	NO	X	X	X	X	SOR
RIGHTMARGIN 引数	0.000 IN	X	X	X	X	SOR
RSASIOTRANSE RROR	RSASIOTRANSE RROR	X	X	X	X	NLS
RSASUSER	NORSASUSER	X	X			SOR、 Comp
RTRACE	なし	X	X			Comp
RTRACELOC 引数	なし	X	X	X	X	Comp
S 引数	0	X	X	X	X	SOR
S2 引数	0	X	X	X	X	SOR

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
S2V 引数	0	X	X	X	X	SOR
SASAUTOS 引数	SASAUTOS	X	X	X	X	Comp、Macro
SASCMD	なし	X	X	X	X	Conn
SASCONTROL	SYSTEMMENU MINMAX	X	X	X	X	Comp
SASFRSCR	#LN00003					Conn
SASHELP 引数	SASV9.CFG を参照	X	X			SOR、Comp
SASINITIALFOLDER	なし	X	X			Comp
SASMSTORE 引数	なし	X	X	X	X	マクロ
SASSCRIPT 引数	SASV9.CFG を参照	X	X	X	X	Conn
SASUSER 引数	SASV9.CFG を参照	X	X			SOR、Comp
SCROLLBARFLASH	NOSCROLLBAR FLASH	X	X	X	X	Comp
SECPACKAGE	NEGOTIATE	X	X			Comp
SECPACKAGELIST	KERBEROS、 NTLM	X	X			Comp
SEQ 引数	8	X	X	X	X	SOR
SERROR	SERROR	X	X	X	X	マクロ
SET 引数	なし	X	X	X	X	Comp
SETINIT	NOSETINIT	X	X			SOR
SGIO	NOSGIO	X	X			Comp
SHARESESSIONCNTL =	SERVER	X	X	X	X	共有
SIGNONWAIT	SIGNONWAIT	X	X	X	X	Conn
SKIP 引数	0	X	X	X	X	SOR
SLEEPWINDOW	SLEEPWINDOW	X	X			Comp
SOLUTIONS	SOLUTIONS	X	X			SOR

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
SORTANOM	なし	X	X	X	X	Comp
SORTCUT	0	X	X	X	X	Comp
SORTCUTP	0	X	X	X	X	Comp
SORTDEV	-WORK と同じ場所	X	X	X	X	Comp
SORTDUP 引数	PHYSICAL	X	X	X	X	SOR
SORTEQUALS	SORTEQUALS	X	X	X	X	SOR
SORTNAME	なし	X	X		X	Comp
SORTPARM	なし	X	X	X	X	Comp
SORTPGM	BEST	X	X	X	X	Comp
SORTSEQ 引数	なし	X	X	X	X	NLS
SORTSIZE 引数	256M	X	X	X	X	SOR、Comp
SORTVALIDATE	NOSORTVALIDATE	X	X	X	X	SOR
SOURCE	SOURCE	X	X	X	X	SOR
SOURCE2	NOSOURCE2	X	X	X	X	SOR
SPDEINDEXSORTSIZE	33554432	X	X	X	X	SPDE
SPDEMAXTHREADS	0	X	X			SPDE
SPDESORTSIZE	33554432	X	X	X	X	SPDE
SPDEUTILLOC	なし	X	X			SPDE
SPDEWHEVAL	COST	X	X			SPDE
SPLASH	SPLASH	X	X			Comp
SPLASHLOC	なし	X	X			Comp
SPOOL	NOSPOOL	X	X	X	X	SOR
SQLCONSTDATETIME	SQLCONSTDATETIME	X	X	X	X	SOR
SQLMAPPUTTO	SAS_PUT	X	X	X	X	ACCESS
SQLREDUCEPUT	DBMS	X	X	X	X	SOR

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
SQLREDUCEPUTOBS	0	X	X	X	X	SOR
SQLREDUCEPUTVALUES	0	X	X	X	X	SOR
SQLREMERGE	SQLREMERGE	X	X	X	X	SOR
SQLLUNDOPOLICY	REQUIRED	X	X		X	SOR
SSLCERTISS	なし	X	X	X	X	DST
SSLCERTSERIAL	なし	X	X	X	X	DST
SSLCERTSUBJ	なし	X	X	X	X	DST
SSLCLIENTAUTH	NOSSLCLIENTAUTH	X	X	X	X	DST
SSLCRLCHECK	NOSSLCRLCHECK	X	X	X	X	DST
SSPI	NOSSPI	X	X	X	X	AppServ
STARTLIB	STARTLIB	X	X			SOR
STEPCHKPT	NOSTEPCHKPT	X	X			SOR
STEPCHKPTLIB	WORK	X	X			SOR
STEPRESTART	NOSTEPRESTART	X	X			SOR
STIMEFMT	M	X	X	X	X	Comp
STIMER	STIMER	X	X	X	X	Comp
SUMSIZE 引数	0	X	X	X	X	SOR
SVGCONTROLBUTTONS	NOSVGCONTROLBUTTONS	X	X	X	X	SOR
SVGHEIGHT	INITIAL	X	X	X	X	SOR
SVGPRESERVEASPECTRATIO	ELEMENT	X	X	X	X	SOR
SVGTITLE	なし	X	X	X	X	SOR
SVGVIEWBOX	なし	X	X	X	X	SOR
SVGWIDTH	なし	X	X	X	X	SOR
SVGX	なし	X	X	X	X	SOR

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
SVGY	なし	X	X	X	X	SOR
SYMBOLGEN	NOSYMBOLGEN	X	X	X	X	マクロ
SYNTAXCHECK	SYNTAXCHECK	X	X	X	X	SOR
SYSGUIFONT 引数	表示設定	X	X			Comp
SYSIN 引数	なし	X	X			Comp
SYSPARM 引数	なし	X	X	X	X	Comp, マクロ
SYSPRINT 引数	システムのデフォルトプリンタ	X	X	X	X	Comp
SYSPRINTFONT 引数	なし	X	X	X	X	SOR, Comp
SYSRPUTSYNC	NO	X	X			Conn
TBUFSIZE 引数	0	X	X	X	X	Conn, CAM
TCPPORTFIRST 引数	0	X	X	X	X	Conn, CAM
TCPPORTLAST 引数	0	X	X	X	X	Conn, CAM
TERMINAL	TERMINAL	X	X			SOR
TERMSTMT	なし	X	X			SOR
TEXTURELOC	なし	X	X	X	X	SOR
THREADS	THREADS	X	X	X	X	SOR
TOOLDEF 引数	TOP RIGHT	X	X			Comp
TOOLSMENU	TOOLSMENU	X	X			SOR
TOPMARGIN 引数	0.000 IN	X	X	X	X	SOR
TRAINLOC	なし	X	X			SOR
TRANTAB 引数	なし	X	X	X	X	NLS
UNIVERSALPRINT	NOUNIVERSALPRINT	X	X			SOR
UPRINTMENUSWITCH	NOUPRINTMENU SWITCH	X	X			Comp

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
UPRINTCOMPRESSIO N	UPRINTCOMPR SSION	X	X	X	X	SOR
USER 引数	なし	X	X	X	X	SOR、 Comp
USERICON 引数	なし	X	X			Comp
UTILLOC 引数	WORK	X	X			SOR
UIDCOUNT 引数	100	X	X	X	X	SOR
UIDGENDHOST 引 数	なし	X	X			SOR
V6CREATEUPDATE 引数	ERROR	X	X			SOR
VALIDFMTNAME 引 数	LONG	X	X	X	X	SOR
VALIDMEMNAME	COMPAT	X	X	X	X	SOR、 Comp
VALIDVARNAME 引 数	V7	X	X	X	X	SOR
VARLENCHK	WARN	X	X	X	X	SOR
VERBOSE	NOVERBOSE	X	X			Comp
VIEWMENU	VIEWMENU	X	X			SOR
VNFERR	VNFERR	X	X	X	X	SOR
WEBUI	NOWEBUI	X	X			Comp
WINDOWSMENU	NOWINDOWSM ENU	X	X	X	X	Comp
WORK 引数	!TEMP\SAS Temporary Files	X	X			SOR、 Comp
WORKINIT	WORKINIT	X	X			SOR
WORKTERM	WORKTERM	X	X	X	X	SOR
XCMD	XCMD	X	X			Comp
XMIN	NOXMIN	X	X	X	X	Comp
XSYNC	XSYNC	X	X	X	X	Comp
XWAIT	XWAIT	X	X	X	X	Comp

オプションの指定	デフォルト値	SAS 起動	構成ファイル	システムオプションウィンドウ	OPTIONS ステートメント	参照
YEARCUTOFF 引数	1920	X	X	X	X	SOR

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: データプロセス

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

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のアクセス機能の有効化を指定します。

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。

ユーザー設定タブには、**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一部のプロパティダイアログボックスにアクセ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、これらの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。

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: 環境設定: 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.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は *SAS システムオプション: リファレンス* での 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:

- [“プロシジャ出力と SAS ログをファイルに送る” \(188 ページ\)](#)
- 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: 環境設定: 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:

- [“プロシジャ出力と SAS ログをファイルに送る” \(188 ページ\)](#)
- [“印刷” \(171 ページ\)](#)

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 argument*

必須引数

system-option

は次のどれかとなります。AUTOEXEC、CMPLIB、FMTSEARCH、HELPOC、MAPS、MSG、SAMPLOC、SASAUTOS、SASHELP、SASSCRIPT、SET。

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、HELPOC、MAPS、MSG、SASAUTOS、SASSCRIPT、SASHELP、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 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` となり、PROC OPTIONS 出力は `c:\app2\help` のみを表示します。

```
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"
```

HELPOC 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:

- “INSERT システムオプションと APPEND システムオプションを使用したオプション値の変更” (*SAS システムオプション: リファレンス 1 章*)
- “APPEND=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*)
- “INSERT=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*)

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:	環境設定: 初期化と動作
PROC OPTIONS GROUP=	EXECMODES
別名:	AUTHPD
参照項目:	“AUTHPROVIDERDOMAIN 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)

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:	環境設定: 初期化と動作
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' |'!>
```

必須引数

“ ”

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” \(519 ページ\)](#)

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:	環境設定: 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:	<i>file-specification</i> 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 ファイルに関する詳細な情報は *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追加の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=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)
- “INSERT=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)

AWSCONTROL システムオプション: ウィンドウ

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: 環境設定: 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>
```

必須引数

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” \(583 ページ\)](#)

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:	環境設定: 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法の例は、“[メイン SAS ウィンドウのサイズと配置の変更](#)” (67 ページ)を参照してください。

AWSMENU システムオプション: ウィンドウ

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:	環境設定: 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: ウィンドウ

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:	環境設定: 表示
PROC OPTIONS GROUP=	ENVDISPLAY
デフォルト:	AWSMENUMERGE
Windows 固有:	すべて

構文

-AWSMENUMERGE | -NOAWSMENUMERGE

AWSMENUMERGE | NOAWSMENUMERGE

必須引数

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” \(614 ページ\)](#)

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: 環境設定: 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=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)

構文

-BUFNO *n* | *nK* | *nM* | *nG* | *hexX* | MIN | MAX

BUFNO= *n* | *nK* | *nM* | *nG* | *hexX* | MIN | MAX

必須引数

n | *nK* | *nM* | *nG*

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、*nM* オプションを使用するか、MAX を指定します。

hexX

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” (526 ページ)
- “SASFILE ステートメント” (SAS ステートメント: リファレンス)
- システム性能最適化の章 参照先: SAS 言語リファレンス: 解説編

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: <i>n</i>
参照項目:	“BUFNO=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)

構文

-BUFSIZE *n* | *nK* | *nM* | *nG* | *hexX* | MAX

BUFSIZE=*n* | *nK* | *nM* | *nG* | *hexX* | MAX

必須引数

n | *nK* | *nM* | *nG*

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。

hexX

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” \(524 ページ\)](#)

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=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)

構文

-CATCACHE n | nK | MIN | MAX

必須引数

n | nK

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 (nK)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:	環境設定: 表示

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 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)

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。その代わりに、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 が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入力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:	環境設定: 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>
```

必須引数

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の設定” \(60 ページ\)](#)
- [“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:	システム管理者: 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せん。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、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: 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お問い合わせください。

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:	ログおよびプロシージャ出力設定: 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 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*)
- “SAS ログ” (*SAS 言語リファレンス: 解説編 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:	環境設定: 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 | 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。

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
カテゴリ:	環境設定: 電子メール
PROC OPTIONS GROUP=	E-MAIL
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)、VIM (Lotus Notes など)、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: “SMTP 電子メールインターフェイス” \(SAS 言語リファレンス: 解説編\)](#)

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 エンジン指定します。

BMDP

BMDP データファイル用のエンジン指定します。

OSIRIS

OSIRIS データファイル用のエンジン指定します。

SPSS

SPSS データファイル用のエンジン指定します。

V8

バージョン 8 ファイル用の SAS エンジン指定します。

V7

バージョン 7 ファイル用の SAS エンジン指定します。

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指定します。

XML

XML ファイル用のデフォルトのエンジン指定します。

XPORT

ランスポートエンジン指定します。

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” \(125 ページ\)](#)
- *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: 環境設定: 表示

PROC OPTIONS ENVDISPLAY
GROUP=

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” \(379 ページ\)](#)

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 600
オル
ト

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” \(481 ページ\)](#)

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:	環境設定: 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も指定できます。

詳細

FILTERLIST のすべてのフィルタが、開くダイアログボックスのファイルの種類ダイアログボックスと名前を付けて保存ボックスのファイルの種類を変更して保存ダイアログ

ボックス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” (354 ページ)
- “DLGSAVE コマンド: Windows” (360 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:	環境設定: 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、**実行**ダイアログボックスに `font` と入力します。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” (603 ページ)
- “SYSPRINTFONT システムオプション: Windows” (606 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: ドライバー設定
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 23.2 Default Font Aliases

SAS font	Windows font
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: 環境設定: 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=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)

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:	ログそしてプロシジャ出力設定: 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: <i>formatting-characters</i>
参照項目:	“FORMCHAR=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)

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: 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:	ログおよびプロシジャ出力設定: 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なため)、“[ALTLOG システムオプション: Windows](#)” (516 ページ)を使用して、SAS ログのコピー先を指定します。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 23.3 FULLSTIMER 統計量の説明

統計量	説明
実時間	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” (602 ページ)
- システム性能の最適化に関するセクション 参照先: SAS 言語リファレンス: 解説編
- SAS ログイン 参照先: SAS 言語リファレンス: 解説編

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、VMS_SAS_OPTIONS DCL 記号、SASV9_OPTIONS 環境変数
デフォルト:	NULL
参照項目:	“HELPHOST 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)

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せん。

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:	環境設定: 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”>
```

必須引数

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、HELPLC 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、HELPLC シ

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 となります。

関連項目:

[“HELPLOC システムオプション: Windows” \(544 ページ\)](#)

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:	環境設定: 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: <i>pathname</i>

構文

```
-HELPLOC <(> "pathname-1" <"pathname-2" "pathname-n">
```

必須引数

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カッコで囲む必要があります。

詳細

HELPPLOC の値の指定

HELPPLOC 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せん。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さらに詳細なファイル仕様を追加することができます。

例

次のコマンドには、HELPPLOC 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=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)
- “INSERT=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)

HELPPREGISTER 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:	環境設定: ヘルプ
PROC OPTIONS GROUP=	HELP
デフォルト:	なし
Windows 固有:	すべて

構文

```
-HELPPREGISTER “menu string” help file location <“help string”> <topic> <CHM | HLP | HTML>
```

必須引数

“メニュー文字列”

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、ブ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" . html
sas -helpregister "Local HTML Doc" c:\mypage.htm
"My own help" middle
```

例 2: HTML ヘルプファイル (.CHM)

```
sas -helpregister "My CHM file" \\server\share\HelpStuff.chm .
"InternalFile.htm"
sas -helpregister "SAS Windows Companion" host.chm .
"/host.hlp/chostfutil.htm"
```

例 3: WinHelp ファイル (.HLP)

```

sas -helpregister "A WinHelp File" c:\somefile.hlp
"simply an .hlp file"
sas -helpregister "WinHelp with a Topic" c:\somefile.hlp .
"My Topic"

```

関連項目:

[“ヘルプメニューにヘルプを追加” \(67 ページ\)](#)

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:	環境設定: 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”>
```

必須引数**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カッコ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、HELPTOC 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、HELPTOC 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、HELPTOC の値が `/help/common.hlp/mytoc.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/mytoc.htm` となります。

関連項目:

[“HELPLOC システムオプション: Windows” \(544 ページ\)](#)

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:	ログおよびプロシジャ出力設定: 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`

必須引数**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” \(181 ページ\)](#)

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:	環境設定: 表示
PROC OPTIONS GROUP=	ENVDISPLAY
デフォルト:	NOICON
Windows 固有:	すべて

構文

-ICON | -NOICON

ICON | NOICON

必須引数

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 は初期化が終了す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 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:	環境設定: 初期化と動作
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 <i>statement</i> を指定する必要があります。
参照項目:	“INITSTMT=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)

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 argument*

必須引数

system-option

は次のどれかとなります。AUTOEXEC、CMPLIB、FMTSEARCH、HELPLLOC、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、HELPLLOC、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 コマンドを入力した場合、SAS がヘルプファイルを検索する唯一の場所は `c:\app2\help` となり、PROC OPTIONS の出力は `c:\app2\help` のみを表示します。

```
sas -helploc "c:\app2\help"
```

もし SAS にて現在のパスと `c:\app2\help` の両方の場所を検索し、さらに `c:\app2\help` を先に検索したい場合は INSERT オプションを使って HELPLLOC の値に新しい場所を追加させます。

```
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、HELPLLOC オプションの値 PROC OPTIONS が表示されます。

```
("c:\app2\help" "!sasroot\nls\en\help")
```

関連項目:

- “INSERT システムオプションと APPEND システムオプションを使用したオプション値の変更” (*SAS システムオプション: リファレンス 1 章*)
- “APPEND=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*)
- “INSERT=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*)

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: 環境設定: 初期化と動作

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

構文

```
-JREOPTIONS (-JRE-option-1 <-JRE-option-n>)
```

```
JREOPTIONS (-JRE-option-1 <-JRE-option-n>)
```

必須引数

-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。クラスパスを定義する場合は、Djava.class.path オプションを使用してください。

例: 例

- `-jreoptions (-verbose)`
- `-jreoptions (-Djava.class.path= "c:\my java\classes \myclasses.jar";c:\java2\classes2\classes2.jar -oss600k)`

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、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:	ログおよびプロシジャ出力設定: 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=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)

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 などの 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場合、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します。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:

- “PAGESIZE システムオプション: Windows” (568 ページ)
- “ORIENTATION=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)
- “PAPERSIZE=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)

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 | nK | nM | nG | hexX | MIN | MAX
```

必須引数

n | *nK* | *nM* | *nG*

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。

hexX

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:	環境設定: ファイル
PROC OPTIONS GROUP=	ENVFILE、LOGCONTROL
デフォルト:	<i>filename</i> .LOG バッチモード上では <i>filename</i> は SAS ジョブの名前となります。
Windows 固有:	<i>destination</i> 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、ログはバッチモードで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 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せん。

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は *SAS システムオプション: リファレンス* での 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せん。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 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*)
- “LOGPARM=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*)
- “SAS ログ” (*SAS 言語リファレンス: 解説編 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、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:	グラフィックス: ドライバ設定
PROC OPTIONS GROUP=	GRAPHICS
デフォルト:	!sasroofmaps
Windows 固有:	デフォルト値および <i>location-of-maps</i> 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” (<i>SAS/GRAPH: Reference</i>)

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さらに詳細なファイル仕様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=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)
- “INSERT=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)

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 | nK | nM | nG | hexX | MIN | MAX
```

```
MAXMEMQUERY= n | nK | nM | nG | hexX | MIN | MAX
```

必須引数

n | *nK* | *nM* | *nG*

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。

hexX

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 | nK | nM | nG | nT | hexX
```

必須引数

n | *nK* | *nM* | *nG* | *nT*

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。

hexX

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せん。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、パフォーマンスに悪影響を及ぼす可能性があります。

- 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 から始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” \(208 ページ\)](#)
- [“MEMMAXSZ システムオプション: Windows” \(559 ページ\)](#)

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せん。

関連項目:

- [“メモリベースのライブラリ” \(208 ページ\)](#)
- [“MEMLIB システムオプション: Windows” \(559 ページ\)](#)

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” \(208 ページ\)](#)
- [“LIBNAME ステートメント: Windows” \(481 ページ\)](#)
- [“MEMCACHE システムオプション: Windows” \(558 ページ\)](#)
- [“パフォーマンスツール” \(234 ページ\)](#)

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 | nK | nM | nG | nT | hexX`

必須引数

`n | nK | nM | nG | nT`

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。

`hexX`

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” \(208 ページ\)](#)
- [“MEMBLKSZ システムオプション: Windows” \(557 ページ\)](#)
- [“MEMLIB システムオプション: Windows” \(559 ページ\)](#)
- [“MEMCACHE システムオプション: Windows” \(558 ページ\)](#)

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* | *nK* | *nM* | *nG* | *nT* | *hexX* | MAX

必須引数

n | *nK* | *nM* | *nG* | *nT*

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。

hexX

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、16 進値(0-9、A-F)、X を指定する必要があります。たとえば、0F00000x は、MEMSIZE オプションの値を 15,728,640 バイトに設定します。これは、0x と同じです。

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に自動的に増加されず。

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:	環境設定: ファイル
PROC OPTIONS GROUP=	ENVFILES
デフォルト:	!sasroot\core\resource
Windows 固有:	次に対する有効な値: <i>library-specification</i>

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さらに詳細なファイル仕様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=システムオプション” \(SAS システムオプション: リファレンス\)](#)
- [“INSERT=システムオプション” \(SAS システムオプション: リファレンス\)](#)

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:	ログおよびプロシジャ出力設定: SAS ログ
PROC OPTIONS GROUP=	LOGCONTROL
デフォルト:	NOMSGCASE
Windows 固有:	すべて

構文

-MSGCASE | -NOMSGCASE

必須引数

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 フォーマットに関する詳細な情報は *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:

“SAS ログ” (*SAS 言語リファレンス: 解説編* 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 参照先: SAS マクロ言語: リファレンス

構文

-MSYMTABMAX *n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

MSYMTABMAX=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

必須引数

n | *nK* | *nM* | *nG* | *nT*

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。

hexX

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 System Option in SAS マクロ言語: リファレンス

構文

-MVARSIZE *n* | *nK* | *hexX* | MIN | MAX

MVARSIZE=*n* | *nK* | *hexX* | MIN | MAX

必須引数***n* | *nK***

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。

hexX

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せん。

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:	環境設定: 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: <i>file-specification</i>
参照項目:	“NEWS=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)

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:	入力設定: データプロセス
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せん。

```
-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ポ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: 入力設定: データプロセス

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ポ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ポタンは、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ポタン(RMB)、Ctrl + 右マウスポタンおよび Shift + 右マウスポ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ポタンと中央マウスポ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=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)

構文

```
-OBS n | nK | nM | nG | nT | hexX | MIN | MAX
```

```
OBS=n | nK | nM | nG | nT | hexX | MIN | MAX
```

必須引数

n* | *nK* | *nM* | *nG* | *nT

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。

hexX

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。

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、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:	ログおよびプロシジャ出力設定: 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=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)

構文

-PAGENO n | nK | nM | nG | $hexX$ | MIN | MAX

PAGENO= n | nK | nM | nG | $hexX$ | MIN | MAX

必須引数

n | nK | nM | nG

ページ番号をそれぞれ、1 (n)、1,024 (nK)、1,048,576 (nM)、1,073,741,824 (nG)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。

$hexX$

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、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:	ログおよびプロシジャ出力設定: 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=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)

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” \(552 ページ\)](#)

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、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: ログおよびプロシジャ出力設定: ODS 印刷

**PROC OPTIONS
GROUP=** ODSPRINT

デフォルト: PLAIN

Windows 固有: 有効な値

参照項目: “PAPERTYPE=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)

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-defined-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 INSTALL

GROUP=

デフォルト: !sasroot\core\sasexe

Windows 固有: すべて

構文

-PATH <(>“*directory-specification-1*” <“*directory-specification-n*”>

必須引数

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* のリストをカッコで囲みます。

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:	入力設定: データプロセス
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 23.4 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	ズーム停止、提出
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

PC のキー	メインフレーム定義	キー	メインフレーム定義
F11	キー	Shift+F11	右
F12	取り消す	Shift+F12	ホーム
RMB	ズーム停止、提出		

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 23.5 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	ズーム停止、提出	F9	出力
F4	pgm; recall	F10	左
F5	rfind	F11	右
F6	rchange	F12	ホーム
		RMB	ズーム停止、提出

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 23.6 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	ズーム停止、提出	Shift+F3	保存
F4	ホーム	Shift+F4	選択
F5	pgm; recall	Shift+F5	非選択
F6	ズーム	Shift+F6	smark
F7	後ろへ	Shift+F7	左
F8	前へ	Shift + F8	右
F9	prevcmd	Shift+F9	rfind
F10	pmenu	Shift+F10	rchange
F11	コマンド	Shift+F11	取り消す
F12	キャンセル	Shift+F12	次
RMB	ズーム停止、提出		

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)を使用する場合には、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:	環境設定: 初期化と動作
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:	環境設定: ファイル
PROC OPTIONS GROUP=	ENVFILES
デフォルト:	<i>filename.LST</i> バッチモード上では <i>filename</i> は SAS ジョブの名前となります。
Windows 固有:	すべて

構文

`-PRINT destination | -NOPRINT`

必須引数

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。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:	LISTCONTROL
PROC OPTIONS GROUP=	LISTCONTROL
デフォルト:	PRNGETLIST
Windows 固有:	すべて

構文

PRNGETLIST | NOPRNGETLIST

必須引数

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

印刷中止の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:	環境設定: 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

ファイルおよびプリンタに出力しているときに印刷中止のダイアログボックスを表示するよう指定します。

NEITHER

ファイルおよびプリンタに出力しているときに印刷中止の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のダイアログボックスを表示するよう指定します。

PRINTER

プリンタに出力しているときにのみ印刷中止のダイアログボックスを表示するよう指定します。

詳細

印刷中止の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のみ表示されます。印刷中止のダイアログボックスを表示しない場合は、NEITHER の値を使用します。

関連項目:

[“印刷ジョブの取り消し” \(185 ページ\)](#)

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:	設定ファイル、SAS の起動
カテゴリ:	ログおよびプロシジャ出力設定:ODS 印刷
PROC OPTIONS GROUP=	ODSPRINT
デフォルト:	NOPRTPERSISTDEFAULT
Windows 固有:	すべて

構文

-PRTPERSISTDEFAULT | -NOPRTPERSISTDEFAULT

必須引数

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” \(605 ページ\)](#)
- [“印刷” \(171 ページ\)](#)

PRTSETFORMS システムオプション: Windows

印刷設定ダイアログボックスにフォームを使用するチェックボックス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:	環境設定: 表示
PROC OPTIONS GROUP=	ENVDISPLAY
デフォルト:	PRTSETFORMS
Windows 固有:	すべて

構文

-PRTSETFORMS | -NOPRTSETFORMS

PRTSETFORMS | NOPRTSETFORMS

必須引数

PRTSETFORMS

印刷設定ダイアログボックスにフォームを使用するチェックボックスを表示するように指定します。

NOPRTSETFORMS

印刷設定ダイアログボックスにフォームを使用するチェックボックスを表示しないように指定します。

詳細

印刷設定ダイアログボックスのフォームを使用するチェックボックスを非表示にするには、NOPRTSETFORMS システムオプションを使用します。

関連項目:

- [“SAS 印刷フォームの使用” \(181 ページ\)](#)

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* | *nK* | *nM* | *nG* | *hexX* | MIN | MAX

必須引数

n | *nK* | *nM* | *nG*

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。

hexX

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でサポートされます。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:	環境設定: 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'>
```

必須引数

'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。作業ディレクトリの指定が必要なアプリケーションかどうかは、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” \(68 ページ\)](#)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:	環境設定: 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')> | “.”

必須引数

'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、リストをカッコで囲みます。

“.”

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または読み込み専用モードで開くかどうかを制御します。

該当要素:	構成ファイル、SAS 起動
カテゴリ:	環境設定: 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 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)

構文

-RSASUSER | -NORSASUSER

必須引数

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せん。

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になります。しかし、SAS/ASSIST を使用する場合には、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: ログおよびプロシジャ出力設定: SAS ログ

PROC OPTIONS LOGCONTROL

GROUP=

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。

関連項目:

- [“RTRACELOC システムオプション: Windows” \(582 ページ\)](#)
- [“SAS ログ” \(SAS 言語リファレンス: 解説編 9 章\)](#)

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: 環境設定: ファイル

**PROC OPTIONS
GROUP=** ENVFILES

デフォルト: なし

Windows 固有: すべて

構文

`-RTRACELOC filename | pathname\filename`

必須引数

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” \(581 ページ\)](#)

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: 環境設定: 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")>
```

必須引数

“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さらに詳細なファイル仕様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” (629 ページ)
- “SAS 自動実行ファイル” (20 ページ)
- “APPEND=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)
- “INSERT=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)
- 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ー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: 環境設定: 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: ウィンドウ” \(521 ページ\)](#)

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: 環境設定: 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=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)

構文

-SASHELP ("*library-specification-1*"...<"*library-specification-n*">

必須引数

“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さらに詳細なファイル仕様を追加することができます。

関連項目:

- “APPEND=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)
- “INSERT=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)

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:	環境設定: 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。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として、SASUSER フォルダが自動設定されます。

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: 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: 環境設定: ファイル

PROC OPTIONS GROUP= ENVFILES

デフォルト: Windows Vista、Windows 7 そして Windows Server 2008 では c:\Users\userid\Documents\My SAS Files\9.3 となります。
Windows XP、Windows Server 2003 では c:\Documents and Settings\userid\My Documents\My SAS Files\9.3 となります。

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*”>)

必須引数

“*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格納する 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せん。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” (135 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:	環境設定: 表示
PROC OPTIONS GROUP=	ENVDISPLAY
デフォルト:	NOSCROLLBARFLASH
Windows 固有:	すべて

構文

-SCROLLBARFLASH | -NOSCROLLBARFLASH

SCROLLBARFLASH | NOSCROLLBARFLASH

必須引数

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でスクロールバーのフォーカスを無効にするを選択して、カーソルの点滅を無効にすることもできます。

関連項目:

[“セッションプリファレンスの設定” \(60 ページ\)](#)

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:	環境設定: 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"> )
```

必須引数

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さらに詳細なファイル仕様を追加することができます。

注: AUX、CON、NUL、LPT1 - LPT9、COM1 - COM9、PRN の各語は、Windows で予約語です。CON または NUL を環境変数名として使用してはいけません。

関連項目:

- “環境変数を使用した SAS ライブラリの割り当て” (130 ページ)
- “環境変数の使用” (152 ページ)
- “APPEND=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)
- “INSERT=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)

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

必須引数

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” \(524 ページ\)](#)
- [“パフォーマンスを最適化する SAS の機能” \(213 ページ\)](#)

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:	環境設定: 表示
PROC OPTIONS GROUP=	ENVDISPLAY
デフォルト:	SLEEPWINDOW
Windows 固有:	すべて

構文

-SLEEPWINDOW | -NOSLEEPWINDOW

必須引数

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” \(431 ページ\)](#)
- [“WAKEUP 関数 Windows” \(433 ページ\)](#)

SORTANOM システムオプション: Windows

SyncSort 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=*option(s)*

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” \(460 ページ\)](#)

SORTCUT システムオプション: Windows

SAS ソートプログラムではなく、SyncSort を使用する基準となるオブザベーション数を指定します。

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: 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* | *nK* | *nM* | *nG* | *hexX* | MIN | MAX

SORTCUT=*n* | *nK* | *nM* | *nG* | *hexX* | MIN | MAX

必須引数

n | *nK* | *nM* | *nG*

オブザベーション数をそれぞれ 1、1(*n*)、1,024 (*nK*)、1,048,576 (*nM*) および 1,073,741,824 (*nG*) 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。

hexX

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” (592 ページ)
- “SORTPGM システムオプション: Windows” (595 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” (460 ページ)

SORTCUTP システムオプション: Windows

SAS ソートプログラムではなく、SyncSort を使用する基準となるバイト数を指定します。

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: 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* | *nK* | *nM* | *nG* | *hexX* | MIN | MAX

SORTCUTP=*n* | *nK* | *nM* | *nG* | *hexX* | MIN | MAX

必須引数

n | *nK* | *nM* | *nG*

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。

hexX

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。

$$\text{バイト数} = ((\text{length-of-obs}) + (\text{length-of-all-keys})) * \text{number-of-obs}$$
関連項目:

- [“SORTPGM システムオプション: Windows” \(595 ページ\)](#)
- [“SORTCUT システムオプション: Windows” \(591 ページ\)](#)
- [“サイズまたはオブザベーションを基準にした並べ替え” \(460 ページ\)](#)

SORTDEV システムオプション: Windows

SyncSort 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” \(615 ページ\)](#)
- [“SyncSort へのパラメータの受け渡し” \(461 ページ\)](#)

SORTNAME システムオプション: 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せん。syncsort を指定できます。

関連項目:

[“SORTPGM システムオプション: Windows” \(595 ページ\)](#)

SORTPARM システムオプション: Windows

SyncSort 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

SORT プロシジャで使用するソートユーティリティを指定します。

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:	BEST
Windows 固有:	すべて

構文

-SORTPGM SAS | BEST | HOST

SORTPGM = SAS | BEST | HOST

必須引数**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” \(457 ページ\)](#)
- [“SORTCUT システムオプション: Windows” \(591 ページ\)](#)
- [“SORTCUTP システムオプション: Windows” \(592 ページ\)](#)

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:	256M
Windows 固有:	デフォルト値
参照項目:	“SORTSIZE=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)

構文

-SORTSIZE *n* | *nK* | *nM* | *nG* | *hexX* | MIN | MAX

SORTSIZE= *n* | *nK* | *nM* | *nG* | *hexX* | MIN | MAX

必須引数

n | *nK* | *nM* | *nG*

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。

hexX

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。

比較

SORTSIZE オプションは、REALMEMSIZE オプションと似ています。SORTSIZE は、SORT プロシジャにのみ影響を与えます。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” \(457 ページ\)](#)
- [“SORT プロシジャのパフォーマンス向上” \(214 ページ\)](#)

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:	環境設定: 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

必須引数**SPLASH or SPLASH ON**

SAS の起動時にロゴ画面を表示します。

NOSPLASH or 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” \(597 ページ\)](#)

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:	環境設定: 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`

必須引数

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*)を指定しないと、デフォルトのリソース番号は 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: ログおよびプロシージャ出力設定: 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` 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 YNNDDS);
```

- 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. (Absolute seconds since Jan. 1, 1970)

DATE. DATE9.

DDMMYY. DDMMYY10. DDMMYYB.

DDMMYYB10. DDMMYYC. DDMMYYC10.

DDMMYYD. DDMMYYD10. DDMMYYN.

DDMMYYN10. DDMMYYP. DDMMYYP10.

DDMMYYYS. DDMMYYYS10.

ISO. (ISO Standard Time)

MMDDYY. MMDDYY10. MMDDYY.
 MMDDYYB10. MMDDYYC. MMDDYYC10.
 MMDDYYD. MMDDYYD10. MMDDYYN.
 MMDDYYN8. MMDDYYP. MMDDYYP10.
 MMDDYYYS. MMDDYYYS10.

NLDATM. NLDATMAP.

YYMMDD. YYMMDD10. YYMMDDDB.
 YYMMDDDB10. YYMMDDC. YYMMDDC10.
 YYMMDDD. YYMMDDD10. YYMMDDN.
 YYMMDDN8. YYMMDDP. YYMMDDP10.
 YYMMDDS. YYMMDDS10.

TOD. (Writes time as military time.)

TIME. (Writes time as military time.)

TIMEAMP. (Writes time as AM and PM.)

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 23.1 PROC OPTIONS からのログ出力

```
SAS (r) Proprietary Software Release 9.3 TS1B0

STIMEFMT=(NLDATM2. HMS TIMEAMP 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:

“SAS ログ” (SAS 言語リファレンス: 解説編 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:	ログおよびプロシージャ出力設定: 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

必須引数

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ー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 23.7 STIMER 統計量の説明

統計量	説明
実時間	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、一部のプロシー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” \(541 ページ\)](#)
- [でのシステム性能最適化の章 参照先: SAS 言語リファレンス: 解説編。](#)

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:	環境設定: 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>
```

必須引数

“*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” (537 ページ)
- “フォントの選択” (59 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:	環境設定: 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: <i>file-specification</i>

構文

`-SYSIN file-specification | -NOSYSIN`

必須引数

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:	環境設定: 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 <“> 文字<”>

SYSPARM=<“> 文字<”>

必須引数

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: ログおよびプロシジャ出力設定: 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”>

必須引数

“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。

印刷設定ダイアログボックスで別のプリンタを選択すると、SYSPRINT システムオプションの値 (PROC OPTIONS プロシジャによって表示可能) にも変更が反映されます。

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” (171 ページ)
- “PRTPERSISTDEFAULT 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” (606 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:	ログおよびプロシジャ出力設定: 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=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)

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>
```

必須引数

"*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。*printer-name* 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。SYSPRINTFONT システムオプションを OPTIONS ステートメントに指定する場合は、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 内でプリンタにフォントが定義されておらず、プリンタに有効なフォントが SYSPRINTFONT に指定されていない場合、現在の表示フォントがスケラブルならば、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 フォントが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で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” (605 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:	環境設定: 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>

必須引数

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” \(70 ページ\)](#)
- [“ツールバーを使用してコマンドの発行” \(42 ページ\)](#)

UNIVERSALPRINT 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:	設定ファイル、SAS の起動
カテゴリ:	出力制御のログおよび手順: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回付します。

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ようと、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:

- “ユニバーサルプリント” (SAS 言語リファレンス: 解説編 15 章)

システムオプション

- “PRINTERPATH=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)
- “SYSPRINT システムオプション: Windows” (605 ページ)
- “UPRINTCOMPRESSION 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)

UPRINTMENUSWITCH システムオプション: Windows

ファイルメニューにてユニバーサルプリントコマンドを有効化します。

該当要素:	設定ファイル、SAS の起動
カテゴリ:	ログおよびプロシジャ出力設定:ODS 印刷
PROC OPTIONS GROUP=	ODSPRINT
デフォルト:	NOUPRINTMENUSWITCH
Windows 固有:	すべて

構文

-UPRINTMENUSWITCH | -NOUPRINTMENUSWITCH

必須引数

UPRINTMENUSWITCH

ファイルメニューで印刷コマンドを指摘すると、**ユニバーサル印刷**ダイアログボックスが呼び出されます。

NOUPRINTMENUSWITCH

ファイルメニューで印刷コマンドを指摘すると、Windows のダイアログボックスが呼び出されます。

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呼び出されません。

関連項目:

[“Windows 環境での SAS の印刷について” \(171 ページ\)](#)

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:	環境設定: 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: <i>library-specification</i>
参照項目:	“USER=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)

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 レベル名で指定した SAS ファイルは、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: 環境設定: 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:	ログおよびプロシジャ出力設定: 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
```

必須引数

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:

“SAS ログ” (SAS 言語リファレンス: 解説編 9 章)

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:	入力設定: データプロセス
PROC OPTIONS GROUP=	INPUTCONTROL
デフォルト:	NOWEBUI
Windows 固有:	すべて

構文

`-WEBUI | -NOWEBUI`

必須引数

WEBUI

Web 拡張機能を有効にします。

NOWEBUI

Web 拡張機能を無効にします。

詳細

Microsoft Internet Explorer 5.0 (IE)以降をインストールし、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インドウなどの特定のウインドウが 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。

WINDOWSMENU システムオプション: Windows

メニューを表示するウインドウに、Windows の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インドウ
カテゴリ:	環境設定: 表示
PROC OPTIONS GROUP=	ENVDISPLAY
デフォルト:	NOWINDOWSMENU
Windows 固有:	すべて

構文

-WINDOWSMENU | -NOWINDOWSMENU
 WINDOWSMENU | NOWINDOWSMENU

必須引数

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 Windows の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 Windows の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: ウィンドウ” \(523 ページ\)](#)

WORK システムオプション: Windows

WORK データライブラリを格納するディレクトリ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:	!TEMP\SAS Temporary Files
Windows 固有:	次に対する有効な値: <i>library-specification</i>
参照項目:	“WORK=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)

構文

-WORK “*library-specification*”

必須引数

“*library-specification*”

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。 *library-specification* は、二重引用符で囲む必要があります。

詳細

デフォルトの SAS 構成ファイルは、TEMP 環境変数が指定するシステムの一時的領域内の "SAS Temporary Files" フォルダに WORK データライブラリを作成するように設定されています。

SAS では、WORK オプションに指定したディレクトリの下に、SAS プロセスごとに TDnnnnn という名前でサブディレクトリが作成されます。nnnnn は一意の番号を示します。

library-specification でディレクトリを指定した場合、work-data ライブラリはそのディレクトリに格納されます。*library-specification* がファイルの場合、ファイルからディレクトリを選定して WORK データライブラリを格納します。プレーンテキスト形式のファイルである必要があります。次の種類の行がファイル内で有効です。

- ブランク行(無視されます)。
- C スタイルのコメント行(/*...*/) (無視されます)。
- 引用符で囲まれていないディレクトリパスを含む行(1 行あたり 1 パスのみ許可されます)。
- 引数 *method=space* や *method=random* を使用したパスの選択方法を含む行。複数の *method=* 行がある場合、最終行のみが処理されます。ファイルに *method=* 行が存在しない場合、*method=random* が処理されます。

次の場合、ライブラリ WORK は存在しないのエラーメッセージが表示されます。

- パスが存在しない
- 引数にディレクトリではなくパスが指定されている

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:	環境設定: 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

必須引数

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” \(392 ページ\)](#)

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が開くか、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: 環境設定: 表示

PROC OPTIONS GROUP= ENVDISPLAY

デフォルト: NOXMIN

Windows 固有: すべて

構文

-XMIN | -NOXMIN

XMIN | NOXMIN

必須引数**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すべて起動します。

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)ですべて起動します。

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:	環境設定: 表示
PROC OPTIONS GROUP=	ENVDISPLAY
デフォルト:	XSYNC
Windows 固有:	すべて

構文

-XSYNC | -NOXSYNC

XSYNC | NOXSYNC

必須引数

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” (619 ページ)
- “X ステートメント: Windows” (488 ページ)

- “X コマンド: Windows” (392 ページ)
- “CALL SYSTEM ルーチン: Windows” (409 ページ)
- “マクロステートメント” (628 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:	環境設定: 表示
PROC OPTIONS GROUP=	ENVDISPLAY
デフォルト:	XWAIT
Windows 固有:	すべて

構文

-XWAIT | -NOXWAIT
XWAIT | NOXWAIT

必須引数

XWAIT

SAS セッションに戻るために、EXIT の入力を必要と指定します。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が自動的に SAS セッションに戻ることを指定します。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” (618 ページ)
- “X ステートメント: Windows” (488 ページ)

- “X コマンド: Windows” (392 ページ)
- “CALL SYSTEM ルーチン: Windows” (409 ページ)
- “マクロステートメント” (628 ページ)

24 章

Windows 版での変数の長さや精度

Windows 版での変数の長さや精度	621
数値変数	621
文字変数	622

Windows 版での変数の長さや精度

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” (621 ページ)
- “文字変数” (622 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 24.1 (621 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 24.1 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
5	536,870,912	2 ²⁹	8

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
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。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 2GB、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。

1 バイト文字セットの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。最大で 1GB まで、変数名と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)を保存できます。352 バイトが入力形式、出力形式および変数名に対する 32 バイトからの結果であり、そしてラベル値に対しては 256 バイトです。

これらの制限を超えない場合、変数の最大可能数は、32 ビットホストでは 412,977,617、64 ビットホストでは 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 2GB、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。

1 バイト文字セットの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。最大で 1GB まで、変数名と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)を保存できます。352 バイトが入力形式、出力形式および変数名に対する 32 バイトからの結果であり、そしてラベル値に対しては 256 バイトです。

これらの制限を超えない場合、変数の最大可能数は、32 ビットホストでは 412,977,617、64 ビットホストでは 2GB です。

25 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	625
自動マクロ変数	625
マクロステートメント	628
マクロ関数	628
自動呼び出しライブラリ	629
自動呼び出しライブラリの概要	629
SASAUTOS システムオプション	629

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” (625 ページ)
- “マクロステートメント” (628 ページ)
- “マクロ関数” (628 ページ)
- “自動呼び出しライブラリ” (629 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、SAS システムオプション: リファレンス内の ERRORCHECK=システムオプションを参照してください。

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](#)” (530 ページ) と 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 上では常に FORE 値を含みます。

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 ログに書きだされます。
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、“The corrected Windows status is”の値は 1 になります。XWAIT オプションを指定してこの例を実行した場合、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 の入力後、ステートメント“This Windows status is”の値は 9009、“The corrected Windows status is”の値は 0 になります。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。

NET_ASRV

Microsoft Windows Server 2003 Enterprise Edition

NET_DSRV

Microsoft Windows Server 2003 Datacenter Edition

NET_SRV

Microsoft Windows Server 2003 Standard Edition

W32_DSRV08
Microsoft Windows 2008 Datacenter Server

W32_ESRV08
Microsoft Windows 2008 Enterprise Server

W32_SRV08
Microsoft Windows Server 2008

W32_VSPRO
Microsoft Windows Vista または Windows 7

X64_VSPRO
Microsoft Windows Vista x64 または Windows 7 x64

W64_ASRV
Microsoft Windows Server 2003 Enterprise 64-bit Edition for Itanium-based computers

W64_DSRV
Microsoft Windows Server 2003 Datacenter 64-bit Edition for Itanium-based computers

W64_DSRV08
Microsoft Windows Server 2008 Datacenter 64-bit edition for Itanium-based computers

W64_ESRV08
Microsoft Windows Server 2008 Enterprise 64-bit edition for Itanium-based computers

W64_SRV08
Microsoft Windows Server 2008 Standard 64-bit edition for Itanium-based computers

X64_DSRV
Microsoft Windows Server 2003 Datacenter 64-bit edition for x64 based computers

X64_ESRV
Microsoft Windows Server 2003 Enterprise 64-bit edition for x64 based computers

X64_SRV
Microsoft Windows Server 2003 Standard 64-bit edition for x64 based computers

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_PRO
Microsoft Windows XP Professional x64 edition

X64_SRV08
Microsoft Windows Server 2008 Standard 64-bit edition for x64-based computers

XP_PRO
Microsoft Windows XP Professional

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、*Windows 版 SAS* 内の「はじめに」の中の「SAS から DOS または Windows のコマンドを実行する」で説明されている X ステートメントに相当します。`%SYSEXEC` ステートメントは、マクロ内またはオープンコードで使用することができます。次に、`%SYSEXEC` ステートメントの構文を示します。

`%SYSEXEC <コマンド>;`

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](#)” (618 ページ) および “[XWAIT システムオプション: Windows](#)” (619 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;` 次の行は SAS ログに書き込まれます。`COMSPEC 環境変数は C:\winnt\system\command.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 を設定する必要があります。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](#)” (582 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。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するには、次の例のように SAS 構成ファイルを編集する必要があります

```
-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

26 章	Windows 版で OLE オブジェクトを自動化するための SCL メソッド.....	633
27 章	Windows 版に固有の SAS エラーメッセージ.....	643
28 章	Windows 版でのグラフィックに関する注意点.....	651
29 章	Windows 版での対話型 SAS セッションのデフォルトキー設定.....	653
30 章	SAS Disk Cleanup Handler.....	663
31 章	Java Runtime Environment エラーのトラブルシューティング.....	667

26 章

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	633
ディクショナリ	634
COMPUTE	634
_DISABLE_DEFAULT_ACTION_	635
DO	635
_ENABLE_DEFAULT_ACTION_	636
EXECUTE	636
_GET_EVENT_	637
_GET_PROPERTY_	637
_GET_REFERENCE_ID_	638
_GET_TYPE_	639
_IN_ERROR_	639
NEW	640
_SET_PROPERTY_	640
UPDATE	641

OLE クラスメソッドの要約

表 26.1 (633 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 26.1 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
<code>_GET_EVENT_</code>	Yes	No
<code>_GET_PROPERTY_</code>	Yes	Yes
<code>_GET_REFERENCE_ID_</code>	Yes	Yes
<code>_GET_TYPE_</code>	Yes	No
<code>_IN_ERROR_</code>	Yes	Yes
<code>_NEW_</code>	No	Yes
<code>_SET_PROPERTY_</code>	Yes	Yes
<code>_UPDATE_</code>	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 26.2 OLE 引数

引数	文字(C)または数字(N)	説明
<code>in-OLE-method</code>	C	OLE メソッド名を指定します。
<code>in-parm</code>	C または N	OLE メソッドへパラメータを与えます。
<code>out-value</code>	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り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 26.3 OLE 引数

引数	文字(C)または 数字(N)	説明
<i>in-OLE-method</i>	C	OLE メソッド名を指定します。
<i>in-parm</i>	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せん。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 26.4 OLE 引数

引数	文字(C)または数字(N)	説明
<i>in-verb</i>	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利用できる 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、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 26.5 OLE 引数

引数	文字(C)または数字(N)	説明
<i>out-event</i>	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 26.6 OLE 引数

引数	文字(C)または 数字(N)	説明
<i>in-OLE-property</i>	C	OLE プロパティの名前を指定します。
<i>out-property-value</i>	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 26.7 OLE 引数

引数	文字(C)または 数字(N)	説明
<i>out-refid</i>	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 26.8 OLE 引数

引数	文字(C)または 数字(N)	説明
<i>out-type</i>	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 26.9 OLE 引数

引数	文字(C)または 数字(N)	説明
<i>error-status</i>	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。
<i>error-msg</i>	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 26.10 OLE 引数

引数	文字(C)または数字(N)	説明
<i>in-OLE-property</i>	C	OLE プロパティ名を指定します。
<i>in-value</i>	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 26.11 OLE 引数

引数	文字(C)または数字(N)	説明
<i>in-hservice</i>	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にのみ使用され、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');
```


27 章

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	643
リターンコードと完了ステータス	643
ファイルへのアクセス	644
SAS 機能の使用	645
OLE の使用	646
ネットワークの使用	647
内部エラーの解決	648
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	649
初期化と終了のエラーメッセージ	649

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。

表 27.1 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](#)”(648 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](#)” (649 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” (493 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” (493 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。リンクダイアログボックスを起動し、リンク先を他のデータソースに変更することができます。

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。詳細は、OLE コントロールに関するマニュアル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は頻繁に表示されるものではありませんが、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たため、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する必要があります。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” \(586 ページ\)](#) 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を確認してください。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 9.2 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。

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 Vista、Windows 7、Windows Server 2008 環境下では、SAS コンソールログは通常 `c:\Users\ユーザー ID\AppData` 内に置かれます。その他すべての Windows 動作環境では、SAS コンソールログは通常 `c:\Documents and Settings\ユーザー ID\Application Data` フォルダ内に置かれます。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、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。

28 章

Windows 版でのグラフィックに関する注意点

SAS/GRAPH での TrueType フォントの使用 651

SAS/GRAPH での TrueType フォントの使用

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せん。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、**Start** ⇒ **Settings** ⇒ **Control Panel** を選択し、フォントアイコン上で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。

29 章

Windows 版での対話型 SAS セッションのデフォルトキー設定

Windows 版でのデフォルトキー定義	653
メイン SAS ウィンドウでのショートカットキー	655
拡張エディタでのショートカットキー	657
印刷プレビュー内でのショートカットキー	661

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”(657 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”(655 ページ) を参照してください)。

この表に示されているキー定義を閲覧または変更したい場合は、**ツール** ⇒ **オプション** ⇒ **キー**と選択するか、もしくは KEYS コマンドを実行します。

表 29.1 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
F8	縮小;サブミット	Ctrl + E	消去

キー	デフォルト設定	キー	デフォルト設定
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	ファイル名
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 29.2 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
Shift + F10	コンテキストメニューを開きます
ドッキングビューのサイズ調整	
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 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)	NOTEPAD 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 29.3 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
	切り取り選択	Ctrl + X Shift+Delete
	現在の文字を削除 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 Shift+Insert
	やり直し	Ctrl + Y Alt + Shift + Backspace
	取り消し	Ctrl + Z 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
	コンテキストヘルプ	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 Ctrl + Home
	カーソルをダイルの下部に移動	Ctrl + Page Down 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	左揃え
	カーソルを右へ移動	右揃え
	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カッコに移動	Ctrl + [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 + [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 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 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す	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 29.4 [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 Alt + F4

30 章

SAS Disk Cleanup Handler

The SAS Disk Cleanup Handler Utility	663
解説	663
必要条件	664
Disk Cleanup Handler の手順	664
Disk Cleanup Handler の使用	664
Microsoft タスクスケジューラを使用したクリーンアップの自動化	664

The SAS Disk Cleanup Handler Utility

解説

SAS Disk Cleanup Handler は、SAS が生成する一時ファイルのほか、SAS 作業ディレクトリや SAS ユーティリティディレクトリを検索して削除します。SAS Disk Cleanup Handler は、Microsoft ディスククリーンアップハンドラ(cleanmgr.exe)に連携する COM です。

SAS Work directories は *TD#####_hostname_* という形式をとり、そこでは、#####が SASWORK ディレクトリを生成した SAS.exe のプロセス ID となります。

SAS Disk Cleanup Handler 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_utilNNNNPPPPPPPP_nodename

場所

- *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。

SAS Disk Cleanup Handler は、次の条件を満たす場合のみ、ユーティリティディレクトリとコンテンツを削除します。

- SAS Disk Cleanup Handler の実行マシンが、*nodename* とプロセス ID を示しています。
- *PPPPPPPP* は使用されていません。

また、SAS Disk Cleanup Handler を使用して、自動生成されたログファイルを削除できます。ログファイル名の形式は、*module name.PID.DATE.log* です。PID は、ログファイ

ルを作成したプロセスのプロセス ID を表します。実行中ではないプロセスのログファイルは削除されます。

必要条件

SAS Disk Cleanup Handler の動作には 32 ビット版および 64 ビット版 Windows とともに Microsoft .Net Framework 2.0 以降を必要とします。

Disk Cleanup Handler の手順

Disk Cleanup Handler を実行するには、次の操作を行います。

1. `cleanmgr` コマンドプロンプトからタイプします。

注: Windows エクスプローラから SAS Disk Cleanup Handler を起動することもできます。マイコンピュータにアクセスして、ローカルの物理ハードディスクを右クリックします。プロパティを選択し、ディスククリーンアップをダブルクリックします。

注: Disk Cleanup Manager が起動します。このプロセスには数分かかる場合があります。
2. **SAS Temporary Files** を選択します。

注: もし **SAS Temporary Files** が表示されない場合、SAS Disk Cleanup Handler はインストールされていません。
3. 検出した SAS 一時ファイルとディレクトリを表示するには、**ファイルの表示**をクリックします。アクティブなプロセスのファイルとディレクトリは表示されません。保存するファイルやディレクトリはすべて削除する必要があります。
4. 変更を保存するには、**OK** をクリックします。破棄するには、**キャンセル** をクリックします。
5. クリーンアップを起動するには **OK**(、**ディスクのクリーンアップ**。ウィンドウ)をクリックします。
6. **EventViewer** でエラーを確認します。

注: Windows Server 2008 では、Microsoft Disk Cleanup Manager (cleanmgr.exe)は自動的にインストールされません。Microsoft Disk Cleanup Manager をデフォルトのアプリケーションとしてインストールするには、**デスクトップエクスペリエンス**をインストールする必要があります。デスクトップエクスペリエンスをインストールするには、次の操作を行います。

- **サーバー マネージャ**を開きます。
- 操作メニューから**機能の追加**を選択します。
- **デスクトップ エクスペリエンス**を選択します。

Disk Cleanup Handler の使用

Microsoft タスクスケジューラを使用したクリーンアップの自動化

cleanmgr.exe で特定のタスクを設定し、クリーンアップを実行します。

Disk Cleanup Manager を構成するには、次の操作を行います。

1. **スタート** ⇨ **ファイル名を指定して実行**と選択します。
2. **ファイル名を指定して実行**ダイアログボックスに以下のように入力します:
cleanmgr /sageset:n ここで n は 0 から 65535 までの任意の数字を指します。
3. クリーンアップする項目を選択します。
4. OK を選択します。
注: また、Windows フォルダ cleanmgr /sagerun:n (ここで n はステップ2の数値です)へのショートカットを作成することもできます。そのショートカットへの名前をタイプします。(例えば、cleanuptask)
5. 設定を確認するには、ステップ 1 から 3 を繰り返します。

Windows XP Professional と Windows Server 2003 では、次の操作を行います。

1. **スタート** ⇨ **すべてのプログラム** ⇨ **アクセサリ** ⇨ **システムツール** ⇨ **スケジューラ**と選択します。
2. スケジュールされたタスクを追加します。
3. **次へ**を選択します。
4. **ディスク クリーンアップ**またはステップ 2 で作成したショートカットを選択します。ショートカットを使用する場合、**次へ**を選択します。
5. [次へ]を選択して設定します。
注: このタスクを実行するには、ユーザー ID とパスワードを入力する必要があります。
6. ショートカットを使用しない場合、[完了]をクリックしたときに**詳細プロパティを開く**を選択します。
7. 操作が完了したら、**完了**を選択します。
8. ショートカットを使用しない場合、**プロパティ**ダイアログボックスを表示します。**タスク**タブを選択します。
9. **実行するファイル名**テキストボックスに、/sagerun:＃(＃はステップ 2 で指定した数字)を追加します。
10. **イベントビューア**で、**アプリケーション**にエラーがないかを確認します。

Windows Vista と Windows Server 7 では、次の操作を行います。

1. **スタート** ⇨ **すべてのプログラム** ⇨ **アクセサリ** ⇨ **システムツール** ⇨ **タスクスケジューラ**を順に選択します。
2. **基本タスクの作成**を選択します。
3. **名前**フィールドにタスク名を入力します。
4. **次へ**を選択します。
5. 適切なトリガーとオプションを選択します。
注: **次へ**を選択し、トリガーオプションへと進みます。
6. **アクション** ⇨ **プログラムを開始**を選択します。
7. **次へ**を選択します。
8. cleanmgr と**プログラム/スクリプト**テキストボックスに入力します。
9. :/sagerun:＃, (＃はステップ 2 で指定した数字)の適切な引数を入力します。

10. **次へ**を選択します。
11. **完了**をクリックします。

31 章

Java Runtime Environment エラー のトラブルシューティング

Java Runtime Environment エラーのトラブルシューティング 667

Java Runtime Environment エラーのトラブルシューテ ィング

SAS 9.3 で SAS のウィンドウ環境(双方向 SAS)を使う場合、プログラム(例:Proc javainfo; run;)を実行するときに SAS ログウィンドウ内で次のようなエラーメッセージに遭遇する場合があります。

```
ERROR: Unable to attach current thread"
ERROR: Unable to load the java virtual machine"
ERROR: Proc javainfo did not run correctly
ERROR: The Java proxy is not responding.
ERROR: The Java proxy's JNI call to start the VM failed.
ERROR: Java failed to start during the SAS startup.
```

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。

1. SAS 内の次のコードをサブミットして JREOPTIONS 設定を確認します。

```
proc options option=jreoptions;
run;
```

出力されたログは次のようなものになるはずですが。

```
proc options option=jreoptions; run;

JREOPTIONS=( -DPFS_TEMPLATE=C:\Program
Files\SASHome\SASFoundation\9.3\tkjava\sasmisc\qrpfstpt.xml
-Djava.class.path=C:\PROGRA~1\SASHome\SASVER~1\eclipse\plugins\
SASLAU~1.JAR
-Djava.security.auth.login.config=C:\Program
Files\SASHome\SASFoundation\9.3\tkjava\sasmisc\sas.login.config
-Djava.security.policy=C:\Program Files\SASHome\SASFoundation\
9.3\tkjava\sasmisc\sas.policy
-Djava.system.class.loader=com.sas.app.AppClassLoader
-Dlog4j.configuration=file:/C:/Program%20Files/SASHome/
SASFoundation/9.3/tkjava/sasmisc/sas.log4j.properties
-Dsas.app.class.path=C:\PROGRA~1\SASHome\SASVER~1\eclipse\
plugins\tkjava.jar
```

```
-Dsas.ext.config=C:\Program Files\SASHome\SASFoundation\9.3\
tkjava\sasmisc\sas.java.ext.config
-Dsas.jre.libjvm=C:\PROGRA~2\Java\JRE16~1.0_2\bin\client\jvm.dll
-Dtkj.app.launch.config=C:\PROGRA~1\SASHome\SASVER~1\picklist -Xms128m )
Java Runtime Environment options
NOTE: PROCEDURE OPTIONS used (Total process time):
real time 0.01 seconds
cpu time 0.00 seconds
```

出力されたログの最後で次のようなものに似た値が見つかるはずですが。

```
-Dsas.jre.libjvm=C:\PROGRA~2\Java\JRE16~1.0_2\bin\client\jvm.dll
```

2. -Dsas.jre.libjvm が指定する場所へと移動し、jvm.dll ファイルの位置を確認します。
3. コンピュータにインストールされた JRE のバージョンを表示させるには次の手順を行います。

スタート ⇨ **ファイルを指定して実行と選択し、cmd を入力します。**

cd コマンドを使って bin ディレクトリへと移動します。例:C:\PROGRA~2\Java\JRE16~1.0_2\bin\次は bin の場所へ移動し Java のバージョンを確認するための DOS コマンドの例です。

```
C:\>cd program files (x*
```

```
C:\Program Files (x86)\Java>cd jre1.6.0_24
```

```
C:\Program Files (x86)\Java\jre1.6.0_24>java -fullversion
java full version "1.6.0_24-b07"
```

入力 java -version

これは **java -version** コマンドをサブミットすることで得られる出力の例です。

```
java version "1.6.0_24"
Java(TM) SE Runtime Environment (build 1.6.0_24-b07)
Java HotSpot(TM) 64-bit Server VM (build 19.1-b02, mixed mode)
```

4. sasv9.cfg ファイル内の Windows の略称を確認します。例: Progra~1DOS コマンドで入力します。**dir /x c:\prog***

出力されるものは次のようなものになるはずですが。

```
Directory of c:\
<DIR> PROGRA~1 Program Files
<DIR> PROGRA~2 Program Files (x86)
```

5. **C:\Program Files\SASHome\SASFoundation\9.3\nls\en\sasv9.cfg** から Notepad のようなテキストエディタで SAS 設定ファイルを開き、-jreoptions がある行を特定します。このファイルの 1 行にテキスト全体が書かれているのを確認します。各オプションの先頭には空白が挿入されており、それぞれはハイフン(-)で始まります。また、パスも確認してください。次は sasv9.cfg ファイル内の JREOPTIONS の例です。

```
/* Options used when SAS is accessing a JVM for JNI processing */
-JREOPTIONS=(

-DPFS_TEMPLATE=!SASROOT\tkjava\sasmisc\qrpfstpt.xml
-Djava.class.path=C:\PROGRA~1\SASHome\SASVER~1\eclipse
\plugins\SASLAU~1.JAR
-Djava.security.auth.login.config=!SASROOT\tkjava\sasmisc\
```

```

sas.login.config -Djava.security.policy=!SASROOT\tkjava\sasmisc\sas.policy
-Djava.system.class.loader=com.sas.app.AppClassLoader
-Dlog4j.configuration=file:/C:/Program%20Files/SASHome/SASFoundation
/9.3/tkjava/sasmisc/sas.log4j.properties
-Dsas.app.class.path=C:\PROGRA~1\SASHome\SASVER~1\eclipse
\plugins\tkjava.jar
-Dsas.ext.config=!SASROOT\tkjava\sasmisc\sas.java.ext.config
-Dsas.jre.libjvm=C:\PROGRA~2\Java\JRE16~1.0_2\bin\client\jvm.dll
-Dtkj.app.launch.config=C:\PROGRA~1\SASHome\SASVER~1\picklist
-Xms128m
-Xmx128m
)

```

6. C:\Documents and Settings\\ApplicationData\SAS\LOGS にある Java ログ内でエラーメッセージを探します。
7. コントロールパネルから JRE をアンインストールしてから再起動し、JRE ディレクトリを削除してください。そののち、SAS Software デポから JRE を再インストールします。その場合、次の手順を実行してください。
 - コントロールパネルからプログラムの追加と削除を選択し、⇒ Java(TM) 6 Update24 を削除します。この操作はプログラムと機能から実行することもできます。
 - PC を再起動します。
 - ウィンドウズエクスプローラーは、スタートボタンを右クリックしてウィンドウズエクスプローラーまたはエクスプローラーを開きます。JRE ディレクトリを削除します。例:C:\Program Files (x86)\Java\jre1.6.0_24)
 - SAS の製品ページから Third Party Software Requirements for use をクリックして、必要な JRE をダウンロードします。次の URL の <http://support.sas.com/resources/thirdpartysupport/index.html> を参照してください。また、SAS Software デポ内にある jre*.exe ファイルをクリックしてお買い求めた製品に必要な JRE をインストールすることもできます。いくつかの例を挙げます。

```
<SAS Software Depot>\products\javaruntime__99170__win__xx__sp0__1\jre-6u24-windows-i5
```

```
<SAS Software Depot>\products\javaruntime__99170__wx6__xx__sp0__1\jre-6u24-windows-i5
```

注: オペレーティングシステムが 64 ビットの場合でも、これらの 32 ビット版 JRE のインストールが必要です。

8. SASHOME ディレクトリ(C:\Program Files\SASHome\sassw.config)内にある sassw.config ファイルの JREHOME へのパスが正しいことを確認してください。必要なバージョンかどうかは、SAS Third Party Software Requirements を確認してください。<http://support.sas.com/resources/thirdpartysupport/index.html> を参照します。このファイルは Java ベースの製品群を起動します。内容例:

```

[properties]
JREHOME=C:\Program Files (x86)\Java\jre1.6.0_24\bin\java.exe
SASHOME=C:\Program Files\SASHome
DPLMTREGLOC=C:\Program Files\SASHome\deploymntreg
VJRHOME=C:\Program Files\SASHome\SASVersionedJarRepository

```

9. Java の設定を確認して、SAS の次のコードをサブミットすることによりエラーメッセージを受け取るかどうかを決定します。

```
proc javainfo; run;
```

ログウィンドウ内に表示される出力の例:

```
PFS_TEMPLATE = C:\Program Files\SASHome\SASFoundation\9.3\tkjava\sasmisc
\qrpfstpt.xml
java.class.path = C:\PROGRA~1\SASHome\SASVER~1\eclipse\plugins\SASLAU~1.JAR
java.class.version = 50.0
java.runtime.name = Java(TM) SE Runtime Environment
java.runtime.version = 1.6.0_24-b07
java.security.auth.login.config = C:\Program
Files\SASHome\SASFoundation\9.3\tkjava\sasmisc\sas.login.config
java.security.policy = C:\Program Files\SASHome\SASFoundation\
9.3\tkjava\sasmisc\sas.policy
java.specification.version = 1.6
java.system.class.loader = com.sas.app.AppClassLoader
java.vendor = Sun Microsystems Inc.
java.version = 1.6.0_24
java.vm.name = Java HotSpot(TM) Client VM
java.vm.specification.version = 1.0
java.vm.version = 19.1-b02
sas.app.class.path = C:\PROGRA~1\SASHome\SASVER~1\eclipse\plugins\tkjava.jar
sas.ext.config = C:\Program Files\SASHome\SASFoundation\9.3\
tkjava\sasmisc\sas.java.ext.config
sas.jre.libjvm = C:\PROGRA~2\Java\JRE16~1.0_2\bin\client\jvm.dll
tkj.app.launch.config = C:\PROGRA~1\SASHome\SASVER~1\picklist
user.country = US
user.language = en
```

```
NOTE: PROCEDURE JAVAINFO used (Total process time):
real time 0.01 seconds
cpu time 0.01 seconds
```

*** 注記: JRE の問題は他の SAS のアプリケーションが開かない問題をも引き起こす場合があります。(例: SAS Install

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

COM

関連項目: コンポーネントオブジェクトモデル

COM/DCOM クライアント

Microsoft Component Object Model (COM) または Distributed Component Object Model (DCOM) を使用してサーバーに要求を送るプログラム。COM/DCOM クライアントは、Windows 環境で Visual Basic、C++、Perl、またはその他のプログラミング言語を使用して作成されます。

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

EBCDIC 照合順序

特定の EBCDIC 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、EBCDIC 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。例えば、ドイツ語の EBCDIC コードページでは、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。ドイツ語の大文字(コードポイント 4A)は、小文字 g (コードポイント 87)より優先されるため、g の前に並べ替えられます。

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

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

I/O 時間

入力/出力時間の省略形。ストレージ領域(ディスクやテープなど)からメモリへのデータの移動に要する時間が入力時間、メモリからストレージや表示デバイス(端末やプリンタなど)への結果の移動に要する時間が出力時間です。

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

関連項目: Open Database Connectivity

ODBC Driver

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、SAS ODBC Driver 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

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

Sasuser.Profile カタログを参照してください。

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

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。たとえば、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せん。

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。

Secure Sockets Layer(SSL)

インターネットを介した安全な通信のための暗号化プロトコル。SSL は次の暗号化アルゴリズムを使用します。RC2、RC4、DES、TripleDES、AES

SMP

関連項目: 対称型マルチプロセッシング

SSL

Secure Sockets Layer の項を参照してください。

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。

生データ

SAS データセットに読み込まれていないデータ。

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。

エンコーディングメソッド

文字セット中の文字に数値表記を割り当てるのに使用されるルールのセット。たとえば、文字の数値表記を保存するのに使用されるビット数、また、文字が表示されるコードページの範囲は、これらのルールによって指定されます。エンコーディングメソッドはコンピューティング産業によって開発された標準規格です。特定のエンコーディングメソッドは、特定のコンピュータハードウェアベンダーに特化しています。業界共通のエンコーディングメソッドは、ASCII、EBCDIC、ISO 646 ファミリ、ISO 8859 ファミリ、Unicode です。

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、または他のベンダー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

ラテン文字(A から Z および a から z)ではなく、文字入力システムに依存する文字(文字、表意文字、象形文字)。

環境変数

ある文字列を他の文字列と同等に扱いた、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

完全修飾ファイル名

C:\SAS\SASUSER\PROFILE.SC2 のように、パス名とファイル名の両方を含むファイル指定。

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。

クライアント

サーバーから、あるいはネットワークを介して、リソースまたはサービスを要求するアプリケーション

クラス

オブジェクトに対するテンプレート 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。

グローバルオプション

指定されてから変更されるまで、SAS プログラム全体もしくは対話型 SAS セッションの処理に影響するオプション。SAS システムオプションに制御されるアイテムの例には、SAS の出力の表示、SAS が使ういくつかのファイルの扱い、システム変数の使用、SAS データセットの監視の処理、SAS 初期化の機能、SAS のユーザーのホスト動作環境に対する相互作用方法を含みます。

現在のフォルダ

アプリケーションの実行時に、コマンドや操作を適用するフォルダ。

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。

コンポーネントオブジェクトモデル

コンポーネントを基盤としたアプリケーションを作成するために使われるオブジェクト指向のプログラミングモデル。このプログラミングモデルは、どのようにソフトウェアコンポーネント同士が相互作用するかを定義し、他のアプリケーションとプロセスがコンポーネントを基盤としたアプリケーションにアクセスできるようにします。短縮形: COM

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。

最小化

グラフィカルユーザーインターフェイスでは、アクティブウィンドウの[最小化]ボタンをクリックして、デスクトップ上のアイコンにウィンドウを置き換える動作。アイコンをクリックまたは選択すると、ウィンドウのサイズと位置は元に戻り、ステータスもアクティブに戻ります。

最大化

グラフィカルユーザーインターフェイスでは、アイコンやアクティブウィンドウの最大化ボタンをクリックまたは選択し、サイズの変更が可能なウィンドウを最大化して表示する動作。

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。

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。

ステップの境界

DATA ステップまたは PROC ステップの終わりとして SAS が認識する SAS プログラムの位置。

スレッド

CPU 上の 1 つのコアで実行されるプロセス処理のシングルパス。

スレッドに対応したオペレーティングシステム

複数の CPU による、共有メインメモリ領域への対称アクセスを調整できるオペレーティングシステム。アクセスが調整されると、同一プロセスのスレッド間で、非常に効率的にデータを共有することができます。

スレディング

1 つ以上の CPU 上の複数コアで同時に処理されるスレッドへと分割されるタスクがある、データ処理またはデータ入出力のためのハイパフォーマンス技術。

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。

ディレクトリ

コンピュータディスク上にある名前付きの区画で、ファイルとファイルに関するメタデータの整理に使われます。

テキスト文字列

文字列の項を参照してください。

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

名前の付いていないパイプを参照してください。

パス名

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)ものいずれかになります。

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。

ファイル名

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。

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

プログラムまたはタスクの機能単位。

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。

ポータビリティ

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機能。

ポータブル

関連項目: ポータビリティ

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。

メモリア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

文字列の項を参照してください。

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。

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

ライブラリ参照を参照します。

ライブラリ参照

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。

ライブラリメンバー

SAS ライブラリの SAS ファイル形式。SAS をファイルの形式は、データセット、ビュー、カタログ、保存されたプログラム、アクセス記述子を含みます。

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 484
 ANY オプション
 WAITFOR ステートメント 487
 COMPUTE メソッド 634
 _DISABLE_DEFAULT_ACTION_ メソ
 ド 635
 DO メソッド 635
 _ENABLE_DEFAULT_ACTION_ メソ
 ド 636
 EXECUTE メソッド 636
 _GET_EVENT_ メソッド 637
 _GET_PROPERTY_ メソッド 637
 _GET_REFERENCE_ID_ メソッド 638
 _GET_TYPE_ メソッド 639
 _IN_ERROR_ メソッド 639
 NEW メソッド 640
 _SET_PROPERTY_ メソッド 640
 UPDATE メソッド 641

!
 !DDE_FLUSH 文字列 293

.
 .INI ファイル 240
 ASCII エディタを使用して作成する
 244
 SSCU を使用して作成する 241

\$
 \$BYVALw.出力形式
 DLL 322
 \$CSTRw.出力形式
 DLL 322
 \$HEXw.出力形式 398
 \$HEXw.入力形式 438

%
 %INCLUDE ステートメント 476
 外部ファイルの読み込み 163
 %SYSEXEC ステートメント 628
 %SYSGET 関数 628

1
 16 進値
 固定小数点表現のバイナリ値への変
 換 438
 バイナリ実数値の変換 397
 浮動小数点表現のバイナリ値への変
 換 438
 文字値の変換 398
 文字データへの変換 438

A
 ABEND オプション
 ABORT ステートメント 465
 ABORT ステートメント 465
 ACCESSIBILITY システムオプション
 515
 ACCESS メンバタイプ 124
 ALTLOG システムオプション 516
 ALTPRINT システムオプション 517
 APPEND システムオプション 518
 ARG ステートメント
 DLL 313
 ASCII
 代替文字 59
 ASCII 照合順序
 1 文字返す 408
 文字位置 431
 文字列を返す 411
 ATTRIB ステートメント 466
 AUTHPROVIDERDOMAIN システムオ
 プション 519
 AUTHSERVER システムオプション 519
 AutoComplete 43

- AUTOEXEC システムオプション 520
 autoexec ファイル
 代替ファイル 520
 AutoScroll 56
 AUTOSCROLL コマンド 345
 AWSCONTROL システムオプション 521
 AWSDEF システムオプション 522
 AWSMAXIMIZE コマンド 346
 AWSMENUMERGE システムオプション 523
 AWSMENU システムオプション 523
 AWSMINIMIZE コマンド 346
 AWSRESTORE コマンド 347
 AWSTITLE システムオプション 524
- B**
 BMDP Engine 143
 BMDP ファイル
 SAS データセットへの変換 449
 読み取り 143
 BUFNO システムオプション 524
 BUFSIZE システムオプション 526
 BYTE 関数 408
- C**
 CALL RECONNECT ルーチン 307
 CALL ルーチン 407
 CAPS コマンド 347
 CATALOG プロシジャ 445
 CATCACHE システムオプション 527
 パフォーマンス 213
 CGM ドライバ 199
 CGM ファイル
 SAS/GRAPH での作成 199
 他のアプリケーションにエクスポート 199
 CIMPORT プロシジャ 446
 SAS 9.3 のファイルを以前のリリースで使用する 142
 CLEANUP システムオプション 527
 COBOL
 DLL 形式 321
 COLLATE 関数 411
 COLOR コマンド 348
 COMDEF システムオプション 528
 COMMAND オプション
 SYSTASK ステートメント 484
 COMMAND コマンド 348
 COMPRESS データセットオプション
 パフォーマンス 214
 CONFIG システムオプション 529
 CONTENTS プロシジャ 447
 CONVERT プロシジャ 449
 CPORT プロシジャ 451
 SAS 9.3 のファイルを以前のリリースで使用する 142
 CUT コマンド 350
 C 言語
 DLL 形式 319
- D**
 DATASETS プロシジャ 452
 DATA ステップ
 開始時間のスケジュール 433
 出力の表示 183
 中断 431
 電子メールの送信 48, 51
 DDE (動的データ交換) 285
 !DDE_FLUSH 文字列 293
 DDE トリプレット 287
 Excel からの読み込み 289
 Excel への書き出し 289
 HOTLINK オプション 292
 NOTAB オプション 291
 Word からの読み込み 290
 Word への書き出し 289
 アプリケーションコマンドの起動 290
 アプリケーションの制御 288
 外部ファイルの参照 287
 欠損値 286, 294
 構文 286
 タブ文字 291
 動的データ転送 293
 例 288
 DDE サーバー
 X コマンドを使用して開く 288
 DDE トリプレット 287
 DEVICE システムオプション 530
 DINFO 関数 412
 DLGABOUT コマンド 350
 DLGCDIR コマンド 351
 DLGCOLUMNSIZE コマンド 351
 DLGCOLUMNSORT コマンド 351
 DLGCONVERT コマンド 352
 DLGENDR コマンド 352
 DLGFIND コマンド 353
 DLGFONT コマンド 353
 DLGLIB コマンド 353
 DLGLINKS コマンド 354
 DLGOPEN コマンド 354
 DLGPAGESETUP コマンド 356
 DLGPREF コマンド 356
 DLGPRTPREVIEW コマンド 358
 DLGPRTSETUP コマンド 358
 DLGPRT コマンド 357
 DLGREPLACE コマンド 359
 DLGRUN コマンド 359
 DLGSAVE コマンド 360
 DLGSMAIL コマンド 361

- DLL 309
 IML プロシジャからルーチンを起動する 328
 MODULE 引数の出力形式 319
 MODULE 引数の入力形式 319
 MODULE 関数 428
 MODULE の引数として使用する式 318
 MODULE の引数として使用する定数 318
 MODULE ログメッセージ 323
 PEEK 関数 315
 SASCBTBL 属性テーブル 310
 SAS からアクセスする 336
 引数を値として渡す 326
 構造 327
 効率的にアクセスする 316
 変数を構造引数としてグループ化する 317
 文字列引数 315
 文字列引数を更新する 325
 呼び出し 428
 戻されたポインタにアクセスする 326
 DOPEN 関数 413
 DOPTNAME 関数 414
 DOPTNUM 関数 415
 DOS シェル
 EXIT コマンドを使用可能にする 619
 終了する 30
- E**
 ECHO システムオプション 531
 EMAILDLG システムオプション 532
 EMAILSYS システムオプション 532
 ENCODING=オプション
 %INCLUDE ステートメント 476
 FILENAME ステートメント 469
 FILE ステートメント 467
 INFILE ステートメント 478
 ENGINE システムオプション 533
 ENHANCEDEDITOR システムオプション 534
 ERRORLEVEL バッチ変数 643
 Excel アプリケーション
 DDE を使用した読み込み 289
 Excel スプレッドシート
 OLE 266
- F**
 FDELETE 関数 415
 FEXIST 関数 416
 FILEEXIST 関数 416
 FILELOCKWAITMAX=システムオプション 535
 FILENAME 関数 417
 FILENAME ステートメント 469
 DDE 構文 286
 電子メールの送信 48
 名前付きパイプ 300
 名前の付いていないパイプ 298
 FILEOPEN コマンド 364
 FILEREF 関数 418
 FILE コマンド 164, 362
 FILE ステートメント 467
 外部ファイルへの書き込み 161
 FILL コマンド 364
 FILTERLIST システムオプション 536
 FINFO 関数 419
 FONTALIAS システムオプション 538
 FONTSLOC システムオプション 539
 FONT システムオプション 537
 FOOTNOTE ステートメント 475
 FOPTNAME 関数 422
 FOPTNUM 関数 424
 FORMCHAR システムオプション 540
 FORM ウィンドウ 182
 FORTRAN
 DLL 形式 320
 FRAME エントリ
 OLE オブジェクトの挿入 252
 OLE オブジェクトの編集 256
 FULLSTIMER システムオプション 541
- G**
 GSUBMIT 365
 GSUBMIT コマンド 165, 365
 GUI 5
- H**
 HELPHOST システムオプション 79, 542, 543
 HELPINDEX システムオプション 543
 HELPLOC システムオプション 544
 HELPPORT システムオプション 79
 HELPREGISTER システムオプション 545
 HELPTOC システムオプション 547
 HEXw.出力形式 397
 HEXw.入力形式 438
 HOME コマンド 365
 HOSTPRINT システムオプション 548
 HOTLINK オプション
 DDE 292
 HSERVICE エントリ
 OLE オブジェクトを読み込む 254
 HTML 出力 62
 ODS 187
 プリファレンス 186

HTML ヘルプ 75

データベースを取り込む 222

I

I/O のディスクペース 205
 IBw.d 出力形式 399
 IBw.d 入力形式 439
 ICON コマンド 366
 ICON システムオプション 549
 INCLUDE コマンド 164, 366
 INFILE ステートメント 478
 外部ファイルの読み込み 162
 INITSTMT システムオプション 549
 INSERT システムオプション 550
 IntelliMouse 56

J

Java Runtime Environment (JRE)オプション 551
 Java Runtime Environment エラー 667
 JREOPTIONS システムオプション 551

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 369
 マウスボタン 566
 KILL オプション
 SYSTASK ステートメント 484

L

LENGTH ステートメント 480
 LIBNAME ウィンドウ
 ライブラリ参照名のクリア 133
 LIBNAME 関数 425
 ライブラリ参照名のクリア 133
 LIBNAME ステートメント 481
 ライブラリ参照名のクリア 133
 ライブラリ参照名の割り当て 128
 LINESIZE システムオプション 552
 LIST オプション
 SYSTASK ステートメント 484
 LOADMEMSIZE システムオプション 553
 LOGEVENT.EXE ユーティリティ 233
 LOG システムオプション 554
 Lotus Notes 221
 NOTESDB Engine 221
 NotesSQL ODBC Driver 227
 Notes ドキュメントの作成 222
 SAS/GRAPH 出力 226
 クライアントの必要条件 221
 情報を取り込む 228

M

MAPS システムオプション 555
 MAXMEMQUERY システムオプション 556
 MCIPISLP 関数 426
 MCIPISTR 関数 427
 member-name 構文 153
 解釈 155
 MEMBLKSZ システムオプション 557
 MEMCACHE システムオプション 558
 64 ビット Windows 環境 210
 MEMLIB オプション
 LIBNAME ステートメント 481
 MEMLIB システムオプション 559
 64 ビット Windows 環境 210
 MEMMAXSZ システムオプション 559
 MEMSIZE システムオプション 560
 Microsoft HTML ヘルプ 75, 544
 Microsoft IntelliMouse 56
 Microsoft Task Scheduler
 ディスククリーンアップの自動化 664
 MNAME=オプション
 SYSTASK ステートメント 484
 MODULE 関数 428
 引数として使用する式 318
 引数として使用する定数 318
 出力形式 319
 入力形式 319
 ログメッセージ 323
 MS-DOS コマンド
 SAS から実行する 28
 MSGCASE システムオプション 562
 MSG システムオプション 562
 MSYMTABMAX システムオプション 563
 MVARSIZE システムオプション 564

N

NEWS システムオプション 565
 NOBLOCK オプション
 名前付きパイプ 305
 NOTAB オプション
 DDE 291
 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 369
 NOTEPAD テキストエディタ
 コードをサブミットする 25
 NOTESDB Engine 221
 NotesSQL ODBC Driver 227
 NUMKEYS システムオプション 565

NUMMOUSEKEYS システムオプション
566

O

OBS システムオプション 567

OCX

参照項目: OLE カスタムコントロール
(OCX)

ODBC Driver 147

ODS HTML 出力 187

ODS 出力

リモートブラウジング 78

OLE 252

FRAME エントリ内でのオブジェクトの
編集 256

OLE Verb の起動 257

SAS/AF アプリケーションでの OCX
268

SAS/AF カタログの互換性 252

エラーメッセージ 646

オブジェクトとアプリケーションの自動
化 261

オブジェクトの変換 259

オブジェクトを FRAME エントリに挿入
する 252

リンクされたオブジェクト 258

OLE Verb

起動 257

OLE オブジェクト

FRAME エントリ内での編集 256

FRAME エントリへの挿入 252, 255

HSERVICE エントリからの読み込み
254

オートメーションメソッドとプロパティ
280

クリップボードから貼り付ける 254

自動化 261

変換 259

OLE カスタムコントロール(OCX) 268

FRAME エントリへの挿入 269

SCL コードをイベントに割り当てる 271

SCL メソッド 271

イベントから引数値を取り込む 272

イベント処理 271, 273

イベントマップダイアログボックス 271

コンボボックスにアイテムを追加する
274

コンボボックスのアイテムを検索する
274

サブクラス化 273

テキスト値を取り込む 274

登録 269

プロパティ 269

OLE クラスメソッド 633

OLE 自動化 261, 277

Excel スプレッドシートを取り込む 266

SAS セッションからのフィードバック
278

SAS のインスタンスの作成 278

Visual Basic のコードと対応する SCL
267

値のプロパティ 263

オブジェクトのメソッドとプロパティ 280

外部インスタンスの作成 264

サーバーが返した配列値 262

サーバーメソッドのオプションパラメー
タ 264

例 279

OPTIONS ステートメント

システムオプション 496

OPTIONS プロシジャ 453

OSIRIS Engine 144

OSIRIS ファイル

データセットへの変換 449

読み取り 143

P

PAGENO システムオプション 568

PAGESIZE システムオプション 568

PAPERTYPE システムオプション 569

PATH システムオプション 570

PCLEAR コマンド 345

PDw.d 出力形式 400

PDw.d 入力形式 440

PEEKLONG 関数 430

文字列引数にアクセスする 315

PFKEY システムオプション 571

PIBw.d 出力形式 402

PIBw.d 入力形式 441

PL/I

DLL 形式 320

PLIST コマンド 345

PMENU コマンド 368

PMENU プロシジャ 454

PRIMARYPROVIDERDOMAIN=システ
ムオプション: 573

PRINTTO プロシジャ 455

プロシジャ出力の送信 189

PRINT システムオプション 574

PRNGETLIST システムオプション 574

PROC SQL クエリ 238

Profile カタログ 22

削除 23

デフォルト 22

場所の変更 22

PRTABORTDLGS システムオプション

575

P RTPERSISTDEFAULT システムオプション 576

- PRTSETFORMS システムオプション
577
- Q**
QueryWindow メソッド
メソッド 281
Quit メソッド
メソッド 282
- R**
RANK 関数 431
RBw.d 出力形式 403
RBw.d 入力形式 443
RBwd 403
REALMEMSIZE システムオプション
578
REGISTER システムオプション 579
Remote Browser Server 77
インストール 79
ポート番号 79
RESOURCESLOC システムオプション
579
RETURN オプション
ABORT ステートメント 465
ROUTINE ステートメント
DLL 311
RSASUSER システムオプション 580
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 387
RTF(リッチテキストフォーマット) 116
RTRACELOC システムオプション 582
RTRACE システムオプション 581
- S**
SAS
SAS プロダクトのヘルプ 76
バッチモード 26
SAS_EXECFILENAME 環境変数 96
SAS_EXECFILEPATH 環境変数 96
SAS, 起動 6
SAS が起動しない場合 14
SAS ファイル 10
Windows のサービスとして 240
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 6
バッチモード 6
プログラムアイテムから 8
SAS 9.3 ファイル
以前のリリース 142
リモートホストファイル 142
SAS Disk Cleanup Handler 663
SAS Institute の Web サイト 77
SAS NOTEPAD テキストエディタ
コードをサブミットする 25
SAS/AF アプリケーション
OCX 268
アイコン 337
自動起動 337
ロゴ 337
SAS/AF カタログ
OLE の互換性 252
SAS/GRAPH
CGM ファイルを作成 199
Notes ドキュメント向けの出力を用意す
る 226
ネイティブドライバ 193, 194
汎用ドライバ 193
SASAUTOS システムオプション 582,
629
SASCBTBL 属性テーブル 310
ARG ステートメント 313
ROUTINE ステートメント 311
構文 311
重要性 314
SASCONTROL システムオプション 583
Sashelp カタログ
エントリの検索 584
SASHELP システムオプション 584
SASINITIALFOLDER システムオプシ
ョン 585
SASUSER システムオプション 586
Sasuser データライブラリ 135
エンジンの変更 135
Sasuser ライブラリ
名前の指定 586
読み込み/書き込みアクセス制御 580
Sasuser ライブラリ参照名
フォルダ 135
SAS オートメーションオブジェクト 279,
283
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 247
起動 249
構成ユーティリティ(SSCU) 240
削除 249
SAS システムダイアログボックスについ
て 77, 350
SAS セッション 39
ウィンドウタイトルの設定 280
カスタマイズ 59
最小化 67
サンプル 11
終了 33, 280

- 終了時に設定を保存する 61
- 終了の確認 61
- セッションプリファレンスの設定 60
- タイトル 67
- 中断 28
- ビジーステータスの確認 279
- ビットマップの貼り付け 57
- 表示/非表示の切り替え 280
- フォントの選択 59
- SAS セッションプリファレンス
 - Web 63
 - 一般 61
 - 詳細 63
 - ビュー 61
 - 編集 62
- SAS セッションを中断する 28
- SAS 入門ガイド 76
- SAS の起動
 - 参照項目: SAS, 起動
- SAS の反応性 206
- SAS ファイル 14
 - 関連項目: ファイル
 - Profile カタログ 22
 - SAS の起動 10
 - Work データライブラリ 23
 - 構成ファイル 15
 - 自動実行ファイル 20
 - レジストリファイル 24
- SAS プログラミングの学習 77
- SAS プロセス
 - 終了 32
- SAS ヘルプとドキュメント 76
- SAS リソース
 - ディレクトリの場所 579
- SAS ログ機能 190
- SAVE コマンド 369
- Scatter-read/Gather-write 214, 588
 - データセットのアクティブ化 341
- SCL
 - 電子メールの送信 48, 53
- SCL メソッド
 - OLE 633
 - OLE コントロール 271
- SCROLLBARFLASH システムオプション 587
- SCROLLBAR コマンド 345
- SET Windows コマンド 132
- SET コマンド
 - 外部ファイルのファイル参照名 153
- SET システムオプション 587
 - 外部ファイルのファイル参照名 153
- SGIO システムオプション 588
- SGIO データセットオプション 341
- SHELL=オプション
 - SYSTASK ステートメント 484
- SLEEPWINDOW システムオプション 590
- SLEEP ウィンドウ
 - 有効化/無効化 590
- SLEEP 関数 431
- SMARK コマンド 345
- SMP (対称型マルチプロセッシング) 207
- SMTP
 - 電子メールの送信 54
- SORTANOM システムオプション 590
- SORTCUTP システムオプション 592
- SORTCUT システムオプション 591
- SORTDEV システムオプション 593
- SORTNAME システムオプション 594
- SORTPARM システムオプション 594
- SORTPGM システムオプション 595
- SORTSEQ=オプション
 - PROC SORT ステートメント 461
- SORTSIZE=システムオプション
 - パフォーマンス 215
- SORTSIZE システムオプション 596
- SORT プロシジャ 457
 - SyncSort 459
 - ソートユーティリティ 595
 - パフォーマンス 214, 236
 - メモリの制限 596
- SOUND Call ルーチン 408
- SPLASHLOC システムオプション 597
- SPLASH システムオプション 597
- SPSS Engine 145
- SPSS ファイル
 - 再フォーマット 146
 - データセットへの変換 449
 - 読み取り 143
- SSCU (SAS Service Configuration Utility) 240
- STATE オプション
 - SYSTASK ステートメント 484
- STATUS=オプション
 - SYSTASK ステートメント 484
- STATVAR オプション
 - SYSTASK ステートメント 484
- STIMEFMT システムオプション 598
- STIMER システムオプション 602
- STORE コマンド 369
- Submit メソッド
 - メソッド 282
- SUBTOP コマンド 95, 370
- SyncSort 459
 - アクティブにする 591, 592
 - 一時ファイルの場所 460
 - 一時ファイルのパス名 593
 - オプションの指定 590
 - オプションを渡す 460
 - 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 460

- ソートアルゴリズムとして設定する 459
 - パラメータの指定 594
 - パラメータを渡す 461
 - SYSCC 自動マクロ変数 625
 - SYSDEVIC 自動マクロ変数 626
 - SYSENV 自動マクロ変数 626
 - SYSGUIFONT システムオプション 603
 - SYSIN システムオプション 604
 - SYSJOBID 自動マクロ変数 626
 - SYSMAXLONG 自動マクロ変数 626
 - SYSPARM システムオプション 604
 - SYSPRINTFONT システムオプション 606
 - SYSPRINT システムオプション 605
 - SYSRC 自動マクロ変数 626
 - SYSSCPL 自動マクロ変数 626
 - SYSSCP 自動マクロ変数 626
 - SYSTASK ステートメント 484
 - SYSTEM Call ルーチン 409
- T**
- TAGSORT オプション
 - PROC SORT ステートメント 458
 - パフォーマンス 215
 - TASKNAME=オプション
 - SYSTASK ステートメント 484
 - TIMEOUT=オプション
 - WAITFOR ステートメント 487
 - TITLE ステートメント 487
 - TOOLCLOSE コマンド 371
 - TOOLDEF システムオプション 609
 - TOOLEDIT コマンド 371
 - TOOLLARGE コマンド 371
 - TOOLLOAD コマンド 372
 - TOOLSWITCH コマンド 373
 - TOOLTIPS コマンド 373
 - Top メソッド
 - メソッド 282
 - TRANSLATE 関数 432
 - TrueType フォント 651
- U**
- UNIVERSALPRINT システムオプション 171, 609
 - UPRINTMENUSWITCH システムオプション 171, 611
 - USERICON システムオプション 612
 - USER システムオプション 611
- V**
- VERBOSE システムオプション 613
 - Visual Basic のコード
 - 対応する SCL 267
- W**
- WAITFOR ステートメント 487
 - WAIT オプション
 - SYSTASK ステートメント 484
 - WAKEUP 関数 433
 - WATTACH コマンド 374
 - WATTENTION コマンド 375
 - WAUTOSAVE コマンド 375
 - WBROWSE 376
 - WBROWSE コマンド 376
 - WCOPY コマンド 376
 - WCUT コマンド 377
 - WDOCKVIEWMINIMIZE コマンド 378
 - WDOCKVIEWRESIZE コマンド 378, 379
 - WDOCKVIEWRESTORE コマンド 379
 - WDOCKVIEW コマンド 377
 - WDRAG コマンド 345
 - WEBUI システムオプション 614
 - Web 上の SAS 77
 - Web 拡張 614
 - 使用可能にする 69
 - Web サイト
 - ヘルプ 77
 - Web ブラウザ
 - 開始ページ 63
 - 起動 376
 - 選択 63
 - プロシジャ出力を送る 186
 - Web プリファレンス 63
 - WEDIT コマンド 379
 - WEMAILFMT コマンド 380
 - WEXITSAVE コマンド 381
 - WFILE コマンド 381
 - WGROW コマンド 345
 - WHIDECURSOR コマンド 382
 - WHSBAR コマンド 382
 - Windows 4
 - エラーメッセージ 649
 - 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 229
 - メモリのブロックサイズ 557
 - メモリベースのライブラリ 559
 - WINDOWS
 - 印刷 174
 - 印刷する 172
 - Windows 2003 Server
 - メモリベースのライブラリ 208
 - Windows Server 2003
 - アプリケーションのパフォーマンス 207
 - アプリケーションログ 230
 - サポートされるエディション 4
 - Windows Vista
 - アプリケーションのパフォーマンス 206
 - アプリケーションログ 230

サポートされるエディション 4
 Windows XP
 アプリケーションのパフォーマンス 207
 アプリケーションログ 231
 サポートされるエディション 4
 Windows XP Professional
 メモリベースのライブラリ 209
 WINDOWSMENU システムオプション
 614
 Windows アプリケーション
 SAS セッションから実行する 488
 サブミット 409
 Windows イベントビューア
 ログメッセージの書き込み 190
 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 240
 Windows フォント 538
 Windows プリンタマネージャ 548
 WINPxxx ドライバ 193, 194
 WINSERT コマンド 383
 WMENUPOP コマンド 383
 WMF ファイル
 グラフィックのエクスポート 201
 WMOVE コマンド 345
 WMRU コマンド 384
 WNAVKEYUNMARK コマンド 385
 WNEWTITLE コマンド 385
 WNEXTEDIT コマンド 386
 Word
 DDE を使用した書き込み 289
 DDE を使用した読み込み 290
 WORK システムオプション 615
 Work データライブラリ 23, 135
 位置 23
 一時サブフォルダ 23
 エンジンの変更 136
 環境変数を使用して指定する 136
 デフォルトのフォルダ 23
 パス名 615
 フォルダの削除 23
 メモリベースのライブラリ 559
 WORK フォルダ 63
 WPASTE コマンド 386
 WPGM コマンド 387
 WPOPUP コマンド 387
 WRTFSAVE コマンド 387
 WSCREENTIPS コマンド 388

WSHRINK コマンド 345
 WSTATUSLN コマンド 389
 WUNDO コマンド 390
 WVSBAR コマンド 390
 WWINDOWBAR コマンド 391

X
 XCMD システムオプション 616
 XMIN システムオプション 617
 XSYNC システムオプション 618
 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 619
 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 392
 DDE サーバーを開く 288
 Windows のコマンドを実行する 29
 現在の SAS セッションで有効 616
 最小化したアプリケーションを開く 617
 同期と非同期 618
 X ステートメント 488
 Windows のコマンドを実行する 29
 同期と非同期 618

Z
 ZDw.d 出力形式 404
 ZDw.d 入力形式 443
 ZOOM コマンド 392

あ
 アイコン
 SAS/AF アプリケーション 337
 ユーザー定義 69, 612
 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 369
 行サイズ 552
 スクロール頻度 345
 値のプロパティ 263
 アプリケーション
 最小化 617
 ツールバー 73
 ツールメニューに追加する 68, 579
 アプリケーションのパフォーマンス
 Windows Server 2003 207
 Windows Vista 206
 Windows XP 207
 アプリケーションログ
 Windows Server 2003 230
 Windows Vista 230
 Windows XP 231

- メッセージ, LOGEVENT.EXE 233
- メッセージ, ユーザー定義関数 231
- 安全なサーバーログイン
 - 検索 519
- 以前のリリースからの移行 138
- 一般プリファレンス 61
 - ツールバーのカスタマイズ 70
- イベントビューア
 - SAS イベントの表示 231
 - アプリケーションログ 230
 - ログメッセージの書き込み 190
- イベントマップダイアログボックス 271
- イメージ使用状況の統計量 541
- 色
 - ウィンドウの色の変更 65
 - カラー印刷 179
 - 選択したウィンドウ部分 348
 - テキストの色属性 58
- 印刷 171
 - DATA ステップ出力 183
 - FORM ウィンドウ 182
 - GRAPH ウィンドウ 194
 - GRAPH ウィンドウでのプレビュー 194
 - SAS/GRAPH ネイティブドライバ 193, 194
 - SAS/GRAPH 汎用ドライバ 193
 - WINDOWS 174
 - Windows プリンタマネージャ 548
 - WINPxxx ドライバ 193, 194
 - 印刷可能なウィンドウ 174
 - 印刷形式 181
 - 印刷ダイアログボックス 174
 - ウィンドウ内 172
 - ウィンドウのプレビュー 180
 - オプション 172
 - カラー印刷 179
 - 行サイズ 178
 - 行番号 179
 - グラフィック 192
 - コマンド 183
 - システムに割り当てられたプリンタの認識 574
 - 出力先プリンタ 576, 605
 - ジョブの取り消し 185
 - 中止 575
 - デフォルトプリンタ 185
 - バッチモード 184
 - バッチモードのプロシジャ出力ファイル 574
 - ビットマップ 179
 - ファイル 184
 - フォーム 577
 - フォント 606
 - フォントの指定 175
 - 部数 180
 - プリンタ設定 358
 - プリンタの指定 174
 - プリンタの変更 174
 - ページサイズ 178
 - ページ設定 176
 - ページ範囲 180
 - ページ番号 179
 - 問題 14
 - 用紙の種類 569
- 印刷オプション 172
- 印刷形式 181
- 印刷するページ範囲 180
- 印刷設定ダイアログボックス
 - 開く 358
 - フォームを使用するチェックボックス 577
- 印刷ダイアログボックス 174, 357
- 印刷中止 575
- 印刷中止ダイアログボックス 575
- 印刷プレビューウィンドウ 180
 - 起動 358
 - ショートカットキー 181, 661
- インターフェイス 36
- インターフェイスデータファイル 123
- インターフェイスライブラリエンジン 126
- インデックス 123
- インデックスファイル
 - SAS ヘルプとドキュメント 543
- インデント
 - 自動 93
- 引用符で囲まれた Windows ファイル名 159
- ウィンドウ
 - RTF ファイルに保存する 387
 - 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 387
 - 位置 64
 - 色 58, 348
 - 外部ファイルの行をコピーする 366
 - 外部ファイルへの保存 55
 - 切り取ってクリップボードに移動する 377
 - クリア 55, 385
 - クリップボードから貼り付ける 386
 - クリップボードにコピーする 369, 376
 - 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 61
 - 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 362, 381
 - 最小化 366
 - 最大化 392
 - ツールバー 73
 - テキストの強調表示 58
 - テキストを切り取る 350
 - 電子メールにコンテンツを添付する 374

特殊文字の属性 58
 フォント 537, 603
 ウィンドウ環境 36, 37
 色 65
 コマンドを使用してカスタマイズする 64
 システムオプションを使用してカスタマイズする 66
 メイン SAS ウィンドウを制御するコマンド 333
 ウィンドウの最小化 366
 ウィンドウバー 37, 40, 62
 切り替え 391
 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 614
 埋め込み文字 364
 上書きモード 62
 エクスプローラウィンドウ
 ライブラリ参照名のクリア 133
 エディタ
 拡張エディタ 81
 プラグラムエディタ 110
 エラーチェック
 拡張エディタ 92
 エラーメッセージ 643
 OLE 646
 SAS 機能 645
 大文字への変換 562
 解釈 649
 完了ステータス 643
 終了 649
 初期化 649
 内部エラー 648
 ネットワーク 647
 ファイルへのアクセス 644
 ライブラリ 562
 リターンコード 643
 エンジン 124
 BMDP 143
 NOTESDB 221
 OSIRIS 144
 SPSS 145
 環境変数への割り当て 131
 決定規則 126
 デフォルトのエンジン 126
 ビューエンジン 125
 複数の割り当て 130
 変更, Sasuser データライブラリ 135
 変更, Work データライブラリ 136
 ライブラリエンジン 125
 割り当て 129, 130
 大文字
 警告とエラーメッセージ 562
 変換 347
 オブザベーション

処理の終点 567
 オンラインドキュメント 75
 オンラインヘルプ 75

か
 カーソル
 スクロールバーの点滅 587
 非表示 64
 非表示にする 382
 ホームポジション 365
 改行
 プラグラムエディタ 112
 外部 DLL
 参照項目: DLL
 外部ファイル 149
 %INCLUDE ステートメント 163
 %INCLUDE ステートメントを使用して読み込む 163
 COM ポートから読み込む 159, 168
 FILE コマンド 164
 FILE ステートメント 161
 FILE ステートメントを使用して書き出す 161
 GSUBMIT コマンド 165
 I/O の操作法 165
 INCLUDE コマンド 164
 INFILE ステートメント 162
 INFILE ステートメントを使用して読み込む 162
 LPT ポートへ書き出す 159
 UNC パス 156
 引用符で囲まれた Windows ファイル名 159
 ウィンドウのコンテンツを書き出す 164
 ウィンドウのコンテンツを保存する 362
 ウィンドウの保存 55
 ウィンドウへの行のコピー 366
 各国語文字 167
 コマンドを使用してアクセスする 163
 作業ディレクトリ 160
 削除 415
 参照 150, 475, 480
 情報アイテムの値 419
 情報アイテムの名前 422
 ステートメントを使用してアクセスする 161
 追加 166
 通信ポートのタイムアウト 168, 169
 テキストエディタウィンドウへのコピー 164
 デフォルトのファイル拡張子 154
 長いファイル名 156
 ファイル参照名 151
 ファイル参照名による検証 416
 物理名による検証 416

- 予約された物理名 159
- レコード形式の変更 165
- 連結 152
- 連結ファイル 158
- ワイルドカード 152
- 外部ファイルの存在 416
- カウンタ 235
- 拡張エディタ 81
 - ウィンドウ間の切り替え 386
 - キーワードの作成 102
 - 機能 81
 - コードをサブミットする 25
 - 自動保存ファイル 85
 - ショートカットキー 657
 - 全般オプション 105
 - デザインタブ 106, 107
 - 配色 107
 - 開くダイアログボックス 364
 - ファイルタイプへのファイル拡張子の割り当て 103
 - ファイルを開く 364
 - プリファレンスの設定 62
 - プログラムエディタに切り替える 110
 - 無効化する 10
 - 有効/無効 109
 - 有効化/無効化 379, 534
- 拡張エディタウィンドウ
 - 概要 82
 - カタログエントリにコンテンツを保存する 369
 - キーボードマクロ 98
 - 行のブックマーキング 93
 - 行番号コマンド 86
 - コーディングエラー 92
 - コードセクションの折りたたみ 101
 - 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 96
 - 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 96
 - 自動インデント 93
 - ショートカットキー 97
 - 省略形 93
 - スクロールコマンド 86
 - 挿入位置 87
 - タブ 93
 - テキストの検索と置換 90
 - テキストの選択 88
 - テキストのドラッグ 90
 - テキストの編集 88
 - 同一ファイルの複数ビュー 85
 - 開く 379
 - ファイルの保存 84
 - ファイルを開く 83
 - プログラムのサブミット 94
 - ワードチップ 94
- 拡張エディタオプションウィンドウ 104
- 格納されたプログラムファイル 123
- 仮想メモリ 578
- カタログ 123
 - 9.3 形式への変換 138
 - 9.3 でのバージョン 6 の変換 140
 - Profile カタログ 22
 - 以前のリリースからの移行 138
 - バージョン 7 と 8 140
 - リリース 6.03 および 6.04 から 9.3 への変換 142
 - リリース 6.08 から 9.3 への変換 141
 - 開いておける数 527
- カタログエントリ
 - ウィンドウのコンテンツを保存する 369
- カタログの変換
 - 9.3 形式へ 138
 - バージョン 6 から 9.3 へ 140
 - リリース 6.03 および 6.04 から 9.3 へ 142
 - リリース 6.08 から 9.3 へ 141
- カタログメンバタイプ 123
- 各国語文字
 - 外部ファイル 167
- 環境変数
 - SET Windows コマンドを使用した定義 132
 - SET システムオプションを使用した定義 131
 - Work データライブラリ 136
 - 外部ファイルのファイル参照名 152
 - 拡張エディタ 96
 - 定義 587
 - ライブラリ参照名の割り当て 130
- 関数 407
- 完了ステータス 643
- キー定義 56, 653
- キーボードマクロ 98
- キーワード
 - ユーザー定義 102
- 起動ファイル
 - システムオプション 495
- キャリッジコントロール文字 189
- 行サイズ 178
- 強調表示 58
- 行のブックマーキング 93
- 行番号 179
- 行番号の使用
 - 拡張エディタ 86
 - プログラムエディタ 111
- 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 199
 - CGM ファイルのエクスポート 199
 - SAS/GRAPH からのインポート 196
 - SAS/GRAPH へのエクスポート 197

- WMF ファイルとしてエクスポートする 201
 - 印刷 192
 - クリップボードからの貼り付け 196
 - クリップボードから貼り付け 198
 - 作成 201
 - 他のアプリケーションからのインポート 195
 - 他のアプリケーションで使用するためのエクスポート 197
 - ディスプレイ上での作成 191
 - ファイル形式のインポート 196
 - グラフィックアダプタ 205
 - グラフィックのインポート 195
 - グラフィックのエクスポート 197
 - グラフ出力
 - デバイスドライバ 530
 - クリップボード
 - OLE オブジェクトの貼り付け 254
 - WPASTE コマンドを使用した貼り付け 386
 - ウィンドウのコンテンツのコピー 369, 376, 377
 - グラフィックの貼り付け 196, 198
 - コードをサブミットする 26, 58, 365
 - ステートメントをサブミットする 165
 - テキストの選択とコピー 57
 - ビットマップを SAS セッションに貼り付ける 57
 - 非テキストウィンドウでの選択とコピー 57
 - 警告
 - 大文字への変換 562
 - 結果タブ 62
 - 結果ビューアウィンドウ 187
 - 欠損値
 - DDE 286, 294
 - 現在のフォルダ 11, 41
 - 初期パスの設定 68
 - 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 353
 - 検索と置換
 - 拡張エディタ 90
 - プログラムエディタ 113
 - 検索パス
 - 実行可能ファイル 570
 - コーディングエラー 92
 - 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, 58, 365
 - 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 101
 - 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 9, 15
 - カスタム 16
 - 検索順序 17
 - システムオプション 495
 - 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, 529
 - デフォルト 15
 - 編集 16
 - 命名規則 16
 - 用途 15
 - 互換性 5, 138
 - 固定小数点表現のデータ
 - 書き込み 399
 - 読み取り 439, 441
 - 固定小数点表現のバイナリ値
 - 16 進値の変換 438
 - このウィンドウの使用 76
 - コマンド
 - AutoComplete 機能 43
 - Windows 環境 344
 - Windows 版ではサポート対象外 345
 - 印刷 183
 - ウィンドウ環境のカスタマイズ 64
 - 外部ファイルにアクセスする 163
 - 拡張エディタウィンドウ 86
 - 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 528
 - コマンド行 5
 - 切り替え 368
 - コマンドの発行 44
 - コマンドバー 37
 - オプションの設定 348
 - コマンドの発行 43
 - ヘルプ 75
 - 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 281
 - コンボボックス
 - アイテムの検索 274
 - アイテムの追加 274
- さ
- サーバーログイン 519

- 最近使用したファイルのリスト 61, 384
- サウンドの生成 408
- サブクラス化 273
- 時間の表示 598
- シグネチャ行 101
- システムオプション 493
 - OPTIONS ステートメント 496
 - 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 66
 - 起動ファイル 495
 - 構成ファイル 495
 - 構成ファイルでの指定 15
 - 制限 494
 - 設定の表示 494
 - 設定の変更 495
 - ターミナルに書き出す 613
 - 電子メールの送信 44
 - パス名の挿入 550
 - パス名の追加 518
 - 複数の場所に設定 497
 - まとめ 497
 - メイン SAS ウィンドウの制御 332
 - ライブラリの連結 496
 - リモートブラウジング 79
 - ログに書き出す 613
- 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 78
- システムパフォーマンスの統計量 602
- 実行可能ファイル
 - 検索パス 570
- 実行ダイアログボックス
 - SAS の起動 9
 - 開く 359
- 自動インデント 93
- 自動実行ファイル 20
 - 作成に使用するテキストエディタ 20
 - 使用 21
 - 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
- 自動的に保存する 375
- 自動保存 62, 117, 375
- 自動保存ファイル 85
- 自動マクロ変数 625
- 自動呼び出しライブラリ 629
 - SASAUTOS システムオプション 629
 - 指定 582
- 集計構文 153
- 終了ダイアログボックス 352
- 終了のエラーメッセージ 649
- 終了プリファレンス 61
- 出力
 - プレビュー 180
 - ページサイズ 568
 - 文字のフォーマット 540
 - リモートブラウザで表示する 77
 - 出力形式 395
 - バイナリデータの書き込み 395
 - 出力のプレビュー 180
- 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 661
 - 拡張エディタ 97, 657
 - キーボードマクロ 98
 - 行のブックマーキング 93
 - 削除 109
 - テキストの選択と編集 88, 89
 - メイン SAS ウィンドウ 655
 - リセット 109
 - 割り当て 108
- 照合順序
 - 作成 458
- 詳細プリファレンス 63
- 省略形 93
- 初期化のエラーメッセージ 649
- 新規ライブラリダイアログボックス 128
- 数値変数
 - 長さと精度 621
- スクリーンチップ 38
 - 切り替え 388
- スクロール
 - 拡張エディタ 86
 - ドラッグ時のスクロール 116
 - プリファレンス 63
- スクロールバー
 - オン/オフの切り替え 382, 390
 - カーソルの点滅 587
 - フォーカスを無効にする 64
 - 横方向 382
- 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, 62
 - オン/オフの切り替え 389
- 領域の割合 389
- ステートメント 465
 - 外部ファイルにアクセスする 161
 - クリップボードからサブミットする 165
 - 実行命令 549
- スプラッシュ画面 68
 - 起動時に表示する 597
 - ビットマップ位置 597
- 制限オプション 494
- 正のバイナリ整数データ
 - 書き込み 402
 - 読み取り 441
- セキュリティ設定のロード 210
- セクション 508 515
- セッションプリファレンス 60
- ソースファイル
 - バッチモード 604

- ソートアルゴリズム
 - SyncSort 459
- ソートユーティリティ 595
 - ホストソートユーティリティの名前 594
- ゾーン 10 進データ
 - 書き込み 404
 - 読み取り 443
- 挿入位置
 - 拡張エディタ 87
 - プラグラムエディタ 112
- 挿入モード
 - オン/オフの切り替え 383
- 速度 204

- た
- ダイアログボックス
 - ヘルプ 76
- 対称型マルチプロセッシング(SMP) 207
- タイトルバー 524
- 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 218
- 対話型セッション 6
 - ファイルのドラッグ 27
- タスクマネージャウィンドウ 375
- タブ
 - 拡張エディタ 93
 - プラグラムエディタ 112
- タブ文字
 - DDE 291
- ダミー変数
 - 長さと精度 622
- 置換ダイアログボックス 359
- ツールチップ
 - オン/オフの切り替え 373
- ツールのカスタマイズダイアログボックス
 - 70
 - 開く 371
 - ユーザー補助機能 515
- ツールバー 37, 38
 - カスタマイズ 69
 - カスタムコントロール 336
 - コマンドの発行 42
 - 全般的なプリファレンスの設定 70
 - ツールの削除 73
 - ツールの作成例 74
 - ツールの追加 72
 - デフォルトに戻す 74
 - 閉じる 371
 - 保存 73
 - ボタンサイズ 371
- ツールボックス
 - ツール切り替え 373
 - 閉じる 371
 - 表示場所 609
- ボタンサイズ 371
- ロード 372
- ツールメニュー
 - アプリケーションの追加 68, 579
- 通信ポート
 - タイムアウト 168, 169
 - 読み込み 159, 168
- データ
 - 他のアプリケーションからアクセスする 147
- データセット 122
 - 9.3 形式への変換 138
 - Scatter-read/Gather-write 341
 - 以前のリリースからの読み込み/書き込み 5
 - サイズの計算 216
 - 作成, リリース 6.08 から 6.12 まで 142
 - バッファ数 524
 - バッファサイズ 526
 - パフォーマンス 216
 - リリース 6.03 と 6.04 140
 - リリース 6.08 から 6.12 までの変換 140
 - リリース 6.08 から 8.2 まで 139
- データセットオプション 341
- データセットの変換
 - リリース 6.08 から 6.12 まで 140
- データディレクティブ
 - Lotus Notes データベースに入力する 222
 - 電子メール 50
- データの追加
 - 外部ファイル 166
- データビュー 123
- データファイル 123
 - インターフェイス 123
 - サイズの上限を超える 137
 - ネイティブ 123
- データベースファイル
 - SAS/ACCESS を使用してアクセスする 146
- データライブラリ 127
 - LIBNAME ステートメントを使用して割り当てる 128
 - OLE を使用して割り当てる 280
 - Sasuser 135
 - Work 23, 135
 - 新規ライブラリダイアログボックス 128
 - 属性のリスト表示 483
 - デフォルトの永久ライブラリ 611
 - マルチエンジンアーキテクチャ 124
 - ライブラリ参照名の指定 127
 - 連結 134
- テーブルメンバタイプ 122
- ディスククリーンアップ 663
 - 自動化 664

- ディスクコントローラ 205
 - ディスクスペース 205
 - ディスクリプタファイル 124
 - ディレクトリ
 - ID 値 413
 - 空の場合は削除する 415
 - 情報 412
 - 情報アイテムの数 415
 - 情報アイテムの名前 414
 - 存在 416
 - ディレクトリの存在 416
 - ディレクトリリスト 410
 - テキスト
 - マーク解除 385
 - 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 110
 - 上書きモード(デフォルト) 62
 - 自動保存の間隔 62
 - テキストの削除 113
 - プリファレンス 62
 - テキストのコピー 57
 - テキストの選択
 - 拡張エディタ 88
 - プラグラムエディタ 112
 - テキストの選択とコピー 57
 - テキストのマーク解除 62, 385
 - デバイスドライバ 530
 - デフォルトのエンジン 126
 - 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 374
 - 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 61
 - 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, 53
 - 送信に使用する SMTP 54
 - 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 380
 - メールの送信ダイアログボックス 46, 361
 - 電子メールインターフェイス 532
 - 電子メールダイアログボックス 532
 - 添付の形式 380
 - 統計量
 - システムパフォーマンス 602
 - 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 75
 - インデックスファイル 543
 - ヘルプファイルの場所 544
 - 目次ファイル 547
 - 特殊文字の属性 58
 - ドッキングビュー 39, 62
 - 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 377
 - 最小化 378
 - タスクバーから復元する 379
 - 分割バーのサイズ変更モード 379
 - ドメインサフィックス
 - 認証プロバイダを割り当てる 519
 - ドメイン認証サーバー
 - 安全なサーバーログインの検索 519
 - ドラッグアンドドロップ
 - 拡張エディタウィンドウ 90
 - 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 115
 - デフォルトでない操作 256
 - プラグラムエディタ 114
 - ドラッグアンドドロップ(デフォルト以外) 115
 - ドラッグ時のスクロール 116
 - ドラッグ修飾子 255
 - 取り消し 390
- な**
- 内部エラー 648
 - 長いファイル名 156
 - 名前付きパイプ 300
 - 1 クライアント, 1 サーバー 302
 - 1 サーバー, 複数クライアント 304
 - CALL RECONNECT ルーチン 302
 - NOBLOCK オプション 305
 - SCL 302
 - 構文 300
 - 次のクライアントに接続する 302, 307
 - データを待つ 305
 - 定義 297
 - 例 302
 - 名前の付いていないパイプ 298
 - 構文 298
 - 定義 297
 - リダイレクトの順序 299
 - 例 299
 - 名前を付けて保存ダイアログボックス
 - 開く 360
 - ファイルフィルタ 536

- フォルダ 585
- 並べ替え
 - SyncSort 459
- 入力形式 435
 - 9.1 への変換 437
 - バイナリデータの読み取り 436
- 認証プロバイダ
 - ドメインサフィックスを割り当てる 519
- ネイティブデータファイル 123
- ネイティブライブラリエンジン 125
- ネットワーク
 - エラーメッセージ 647
 - ネットワークのパフォーマンス 214

- は**
- バージョン 6 のカタログ
 - 9.3 での変換 140
- バージョン 6 のファイル 138
- バージョン 7 と 8 のカタログ 140
- バージョン 7 と 8 のファイル 138
- ハードウェア 203
 - I/O に必要なディスクスペース 205
 - グラフィックアダプタ 205
 - プロセッサの速度 204
 - メモリ 204
- ハードディスクドライブ 205
 - 構成 205
 - マッピング 166
- 配色 107
- バイナリ実数データ
 - 16 進数への変換 397
 - 書き込み 403
 - 読み取り 443
- バイナリ整数データ
 - 書き込み 399
 - 読み取り 439
- バイナリデータ
 - 書き込み 395
 - 読み取り 436
- パイプ 297
 - 名前が付いていない 298
 - 名前付き 300
 - ファイル属性 423
- パス名
 - システムオプション値に挿入する 550
 - システムオプションに追加する 518
- バスの速度 205
- パック 10 進データ
 - 書き込み 400
 - 読み取り 440
- 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 6, 26
 - 印刷 184
 - ソースファイル 604
 - プロシジャ出力ファイル 574
 - ログファイル 554
- バッファ
 - データセットの処理に使用するバッファ数 524
 - ページサイズ 526
- パフォーマンス
 - DLL 316
 - SAS の機能 213
 - SORT プロシジャ 214, 236
 - Windows Vista アプリケーション 206
 - Windows XP アプリケーション 207
 - 対話型処理 218
 - 調整メソッド 214
 - データセットのサイズ, 計算 216
 - ネットワーク 214
 - ハードウェア 203
 - 反応性 206
 - 複数プロセッサに対応した I/O 拡張 207
 - メモリベースのライブラリ 208
- パフォーマンスオブジェクト 235
- パフォーマンスカウンタ 235
- パフォーマンスモニタ 234
 - PROC SQL クエリ 238
 - 起動 234
 - 構成する 236
- 反応性 206
- ビットマップ
 - SAS セッションへの貼り付け 57
 - 印刷 179
- ビューエンジン 125
- ビューメンバタイプ 122
- 表示プリファレンス 61
- 開くダイアログボックス
 - 拡張エディタで開く 364
 - 指定されたパス 68
 - デフォルトエディタで開く 354
 - ファイルフィルタ 536
 - フォルダ 585
- ファイアウォール
 - リモートブラウジング 78
- ファイル 120
 - SAS 9.3 でリモートホストの SAS ファイルを使用する 142
 - SAS 9.3 のファイルを以前のリリースで使用する 142
 - 移行 138
 - 印刷 184
 - 開始時にサブミットする 61
 - 拡張エディタでの保存 84
 - 拡張エディタで開く 83
 - 共有 137

- 互換性 138
- 情報アイテムの数 424
- タイプ 120
- 転送 146
- ファイル拡張子 120
- 複数の SAS セッション 137
- プログラムエディタでの保存 117
- プログラムエディタを開く 110
- リモートホスト 142
- ロックされたファイル 535
- ファイルアクセス
 - エラーメッセージ 644
- ファイル拡張子 120
 - 変更 121
- ファイルキャッシュ
 - メモリベースのライブラリ 212, 558
- ファイル形式
 - リリース間の互換性 138
- ファイル参照名 151
 - FILENAME ステートメント 151
 - FILENAME ステートメントを使用した
割り当て 417
 - SET コマンド 153
 - SET システムオプション 153
 - 環境変数 152
 - クリア 157
 - 検証 418
 - ディレクトリへの割り当て 153
 - ファイルショートカットの割り当て 151
 - リスト 157
 - 連結ディレクトリへの割り当て 155
 - 連結ファイルへの割り当て 156
 - 割り当ての取り消し 417
- ファイル名
 - クリア 55
- ファイルの回帰
 - SAS 9.3 のファイルを以前のリリースで
使用する 142
- ファイルの拡張子
 - ファイルタイプへの割り当て 103
- ファイルの種類
 - ファイル拡張子の割り当て 103
- ファイルのショートカット 151
- ファイルフィルタ 536
- ファイルメニュー
 - 最近使用したファイルのリスト 384
 - ユニバーサル印刷のコマンド 611
- ファイルリソース
 - リスト 581
- ファイルリソーストラッキングシステム
582
- ファンクションキー
 - 数 565
 - マッピング 571
- ファンクションキーのマッピング 571
- フォルダ
 - HTML 出力ファイル 63
 - Sasuser ライブラリ参照名 135
 - WORK フォルダ 63
 - 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 129
 - デフォルトの構造 24
 - ライブラリ参照名の割り当て 129
 - フォルダの変更ダイアログボックス 351
- フォント 59
 - TrueType 651
 - Windows フォントを SAS フォントに割り
当てる 538
- 印刷 606
- ウィンドウ 537
- ウィンドウのコンテンツ 603
- ディレクトリの場所 539
- 変更 175
- フォントダイアログボックス 353
- 複数プロセッサに対応した I/O 拡張 207
- 浮動小数点表現のデータ
 - 16 進数への変換 397
 - 書き込み 403
 - 読み取り 443
- 浮動小数点表現のバイナリ値
 - 16 進値の変換 438
- ブラウザ
 - 開始ページ 63
 - 起動 376
 - 選択 63
- プログラムエディタ 110
 - RTF テキスト 116
 - 改行 112
 - 拡張エディタに切り替える 110
 - 機能 117
 - 行番号の使用 111
 - 検索と置換 113
 - 自動保存 117
 - 挿入位置 112
 - タブ 112
 - テキストの削除 113
 - テキストの選択 112
 - ドラッグ/ドロップアクションより優先する
115
 - ドラッグアンドドロップ 114
 - ドラッグ時のスクロール 116
 - ファイルの保存 117
 - ファイルを開く 110
- フラットファイル 213
- プリファレンス
 - HTML 出力 186
 - Web 63
 - 一般 61
 - 結果タブ 62
 - 終了時に保存する 381

- 詳細 63
- セッションプリファレンス 60
- ビュー 61
- 編集 62
- プリファレンスダイアログボックス 60, 356
- プリントマネージャ 548
- プログラム, サブミット 370
- プログラムアイテム
 - SAS の起動 8
- プログラムエディタ
 - SAS 起動時に開始する 10
 - コードをサブミットする 25
- 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 369
- プログラムのサブミット 370
 - 拡張エディタ 94
- プログラムメンバタイプ 123
- プロシジャ 445
 - 最大メモリ量 556
- プロシジャ出力
 - PRINTTO プロシジャを使用した送信 189
 - Web ブラウザに表示する 186
 - キャリッジコントロール文字 189
 - システムオプションを使用した送信 190
 - 代替ファイル 517
 - 名前を付けて保存ダイアログボックスを使用して送る 188
 - バッチモード 574
 - ファイルに送る 188
- プロセスの終了 32, 375
- プロセッサ
 - 複数に対応した I/O 拡張 207
- プロセッサの速度 204
- ページサイズ 178, 568
- ページ設定 176
- ページ設定ダイアログボックス 356
- ページ番号 179
 - リセット 568
- ヘルプ
 - Microsoft HTML ヘルプ 75
 - SAS プロダクト 76
 - Web サイト 77
 - インデックスファイル 543
 - オンライン 75
 - コマンドバー 75
 - ダイアログボックス 76
 - ヘルプメニュー 76
 - ヘルプメニューに追加する 67
 - メイン SAS ウィンドウ 38
 - 目次ファイル 547
 - リモートブラウザで表示する 77
- ヘルプファイル
 - Microsoft HTML ヘルプの場所 544
 - 登録 545
 - メイン SAS ウィンドウのヘルプメニューに表示する 545
- ヘルプメニュー 76
 - ヘルプの追加 67
- 変換ダイアログボックス 352
- 編集プリファレンス 62
- 変数
 - 構造引数としてグループ化する 317
 - 長さ精度 621
 - 変数の精度 621
 - 変数の長さ 621
- ポート番号
 - Remote Browser Server 79
- ポインタ
 - 返す 326
 - 構造に渡す 318
- ホストコマンド
 - SAS セッションからサブミットする 392
 - SAS セッションから実行する 488
 - サブミット 409
 - 条件付きで実行する 410
- ホストコンピュータ
 - 名前の指定 79, 543
- ホストソートユーティリティ
 - 名前 594
- 保存ダイアログボックス
 - 指定されたパス 68
- ポップアップメニュー
 - オン/オフの切り替え 383
 - 表示 387
- ま
- マウス
 - IntelliMouse 56
- マウスボタン 566
- マクロ, キーボード 98
- マクロ関数 628
- マクロ機能 625
 - 自動マクロ変数 625
 - 自動呼び出しライブラリ 629
- マクロ関数 628
- マクロステートメント 628
- マクロステートメント 628
- マクロ変数
 - メモリ内変数のサイズ 564
- マクロ変数シンボルテーブル 563
- マップデータセット
 - ライブラリの名前 555
- マルチエンジンアーキテクチャ 124
- 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 427
 - アクティブになるまで待機する 426

- メールの送信ダイアログボックス 46, 361
- 命名規則
 - 構成ファイル 16
- メイン SAS ウィンドウ 36
 - 位置 522
 - ウィンドウ環境コマンド 333
 - ウィンドウの制御 583
 - 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 55
 - カスタマイズ 521
 - 画面のヒント 38
 - コンポーネント 36
 - 最小化 346, 549
 - サイズと配置 67
 - 最大化 346
 - ショートカットキー 655
 - ステータス行 38
 - 寸法 522
 - 制御 332
 - 制御のためのシステムオプション 332
 - タイトル 524
 - ツールメニューにアプリケーションを追加する 579
 - ドッキングビュー 39
 - 表示画面と機能 331
 - ヘルプ 38
 - ヘルプファイル 545
 - メニュー 41
 - メニュー項目を組み込む 523
 - メニューバー 523
 - 元の状態に戻す 347
- 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 190
- 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 583
 - 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 523
- メニューバー 37
 - 表示 523
- メモ
- 大文字への変換 562
- メモリ
 - SAS がロードする実行プログラム 553
 - SORT プロシジャの上限 596
 - 仮想メモリ 578
 - 使用状況の統計量 541
 - 総量の上限 560
 - パフォーマンス 204
 - 必要条件 204
 - プロシジャでの最大量 556
 - マクロ変数シンボルテーブル 563
 - メモリ内マクロ変数 564
 - メモリベースのライブラリのブロックサイズ 557
- メモリアドレス
 - 格納 430
- メモリ内マクロ変数
 - サイズ 564
- メモリベースのライブラリ 208
 - SAS ライブラリの処理 211
 - Windows 2003 Server 208
 - Windows XP Professional 209
 - Work ライブラリ 559
 - ファイルキャッシュ 212, 558
 - メモリの最大量 559
- メンバ 120
- メンバタイプ 120
 - アクセス 124
 - カタログ 123
 - テーブル 122
 - ビュー 122
 - プログラム 123
- 目次ファイル 547
- 文字値
 - 16 進数への変換 398
- 文字式
 - 特定の文字を変換する 432
- 文字データ
 - 16 進データの変換 438
- 文字のフォーマット
 - デフォルト 540
- 文字変数
 - 長さ精度 622
- 文字列
 - プログラムに渡す 604
- や
- ユーザー定義アイコン 69, 612
- ユーザー定義のキーワード 102
- ユーザー補助機能 515
- ユーザーライブラリ参照名 136
- ユニバーサル印刷 171, 172
 - コマンドを使用可能にする 611
 - 使用可能にする 609
- 用紙の種類 569
- ら
- ライブラリ
 - システムオプションで連結する 496
 - デフォルトのアクセス方式 533
 - メモリベース 208, 211
- ライブラリエンジン 125
- ライブラリ参照名 124, 127
 - 関連付ける 483
 - 起動時に使用可能にする 130
 - クリア 132, 425

- ユーザー 136
 - リスト 132
 - ライブラリ参照名, 割り当て
 - 1つのディレクトリへ 128
 - GUI 128
 - LIBNAME 関数 425
 - 環境変数 130
 - 作業フォルダへ 129
 - 複数のフォルダへ 129
 - 複数を1つのディレクトリへ 130
 - ライブラリ参照名のクリア 132
 - ライブラリダイアログボックス 353
 - リソースディレクトリ 579
 - リソース不足 527
 - リターンコード 643
 - 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 387
 - リッチテキストフォーマット(RTF) 116
 - リモートブラウジング 77
 - ODS 出力 78
 - コンピュータ名 543
 - システムオプション 79
 - 出力とヘルプの表示 77
 - ファイアウォール 78
 - リモートホスト
 - SAS 9.3 の SAS ファイル 142
 - リリース 6.03 と 6.04 のデータセット 140
 - リリース 6.08 から 8.2 までのデータセット 139
 - リリース 6.08 のカタログ
 - 9.3 への変換 141
 - リンクされた OLE オブジェクト 258
 - 自動更新する 259
 - リンクダイアログボックスを使用して更新する 258
 - リンクダイアログボックス
 - 開く 354
 - リンクされた OLE オブジェクトを更新する 258
 - レジストリファイル 24
 - 列の設定ダイアログボックス 351
 - 列の並べ替えダイアログボックス 351
 - 連結データライブラリ 134
 - 同じ名前 134
 - 更新アクセス権 134
 - 出力アクセス権 134
 - 入力アクセス権 134
 - 連結ディレクトリ
 - アクセス 158
 - ファイル参照名の割り当て 155
 - 連結ファイル
 - アクセス 158
 - ファイル参照名の割り当て 156
 - ログ
 - PRINTTO プロシジャを使用した送信 189
 - Windows イベントビューアへのメッセージの書き込み 190
 - システムオプションを使用した送信 190
 - 代替ログ 516
 - 名前を付けて保存ダイアログボックスを使用して送る 188
 - バッチモード 554
 - ファイルに送る 188
 - メッセージの表示 531
 - メッセージ用のニュースファイル 565
 - ログイベント
 - Windows イベントビューアへの書き込み 190
 - 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 369
 - 行サイズ 552
 - スクロール頻度 345
 - ログメッセージの書き込み
 - Windows イベントビューア 190
 - ロゴ
 - SAS/AF アプリケーション 337
 - ロゴ画面 68
 - 起動時に表示する 597
 - ビットマップ位置 597
 - ロックされたファイル 535
- わ**
- ワードチップ 94
 - ワイルドカード 152

