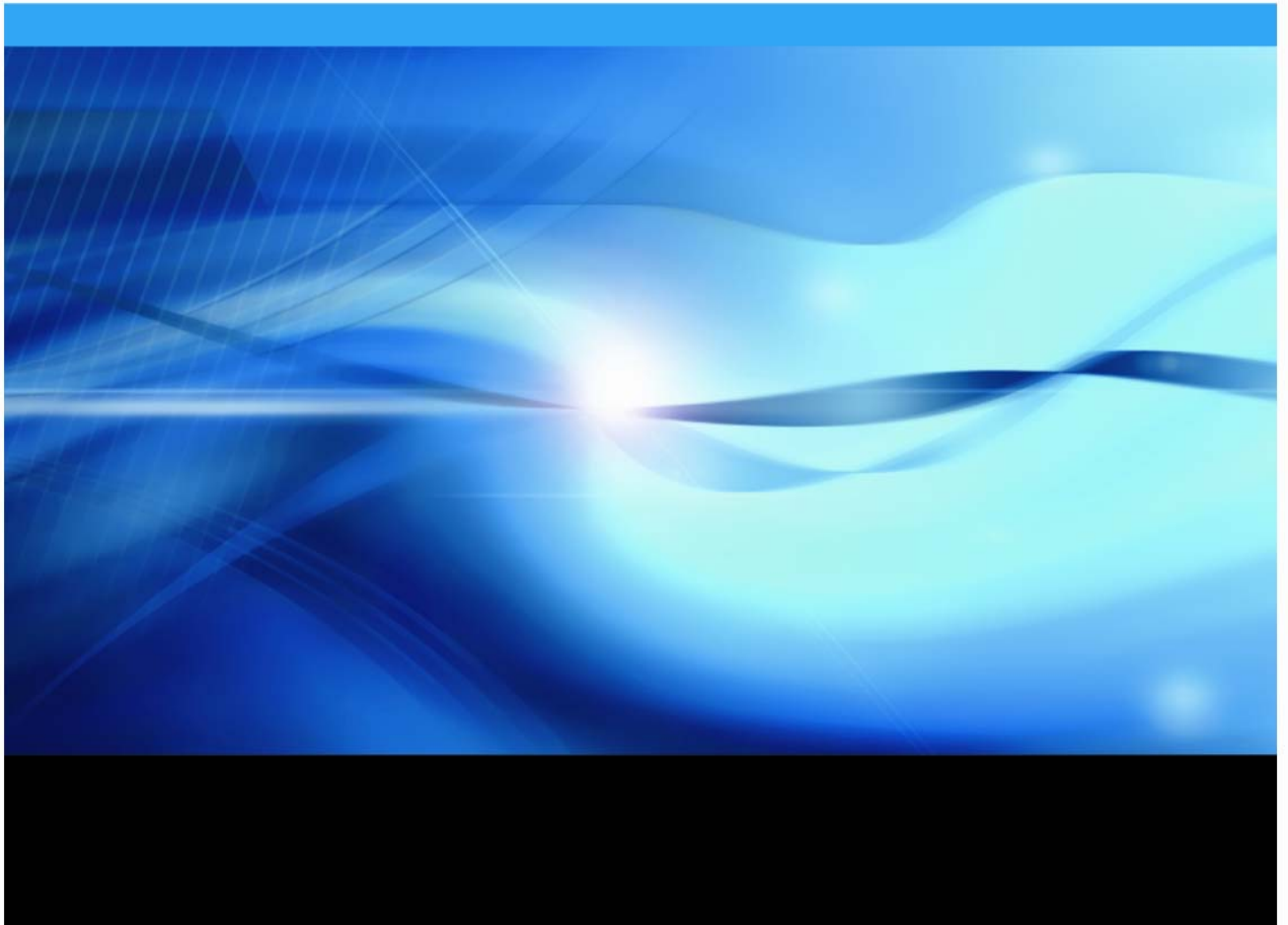




SAS[®] 9.1.3 検証ツールユーザーガイド



著作権情報

このマニュアルの正確な書籍情報は、以下のとおりです。

SAS® 9.1.3 Qualification Tools User's Guide

Copyright® 2008, SAS Institute Inc., Cary, NC, USA.

本書は、発行元であるSAS Institute, Inc.の事前の書面による承諾なく、この出版物の全部あるいは一部を、電子データ、印刷、コピー、その他のいかなる形態または方法によって、複製、転送、または検索システムに保存することは禁止されています。これらの説明書は著作権により保護されています。

著作権保護を受ける本書の使用の範囲は制限されています。許される使用の範囲とは、使用者のシステムに保存して端末に表示すること、本書が提供された目的である、SAS プログラミングおよびライセンスプログラムのインストール・サポートの責任者が使用するために、必要な部数だけコピーすること、および特定のインストール要件を満たすように内容を修正することを指します。本書の全部あるいは一部を印刷する場合、またはディスプレイ媒体に表示する場合は、SAS Instituteの著作権表示を明記する必要があります。上記の条件以外で本書を複製または配布することは一切禁止されています。

アメリカ合衆国政府の制約された権限についての通知

アメリカ合衆国政府による、本ソフトウェアおよび関連するドキュメントの使用、複製、公開は、「FAR52.227-19 Commercial Computer Software-Restricted Rights」（1987年6月）に定められた制限の対象となります。

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

SAS®およびSAS Instituteのプロダクト名またはサービス名は、米国およびその他の国におけるSAS Institute Inc.の登録商標または商標です。

®は米国で登録されていることを示します。

その他、記載されている会社名および製品名は各社の登録商標または商標です。

目次

はじめに.....	1
SAS インストール検証ツール.....	3
概要.....	3
Windows	3
インストール検証ツールの実行	3
UNIX	4
方法 1:SASインストール担当者によるインストール検証ツールの実行.....	4
方法 2:ユーザーによるインストール検証ツールの実行.....	4
SASインストール検証ツールの結果の参照.....	5
SAS 動作検証ツール.....	7
提供しているツール.....	7
ユーザー環境	7
出力レポート	10
SAS 動作検証ツールのテーブル言語.....	11
テーブル言語の一般規則.....	11
ブロック識別子.....	11
コマンド.....	12
マクロ	15
SAS 動作検証ツールのコマンド行	18
Perl Stream Editor について.....	18
IVFND コマンド.....	20

はじめに

SAS 9.1.3では、2つの検証ツール（SASインストール検証ツール - SAS Installation Qualification Tool : SAS IQと、SAS動作検証ツール - SAS Operational Qualification Tool : SAS OQ）が使用できます。これらのツールは、行政規制産業におけるSASソフトウェアの使用の適正化を支援します。さらにこれらのツールは、顧客が必要とするSASの旧バージョンから後のリリースにまたがるマイグレーション作業、統合作業、移行における検証作業の主な局面の適性をサポートします。

SAS IQは、SASの仕様に従ってSASシステムがインストールおよび維持されていることの検証を支援します。またSAS IQは、SASシステムの各ファイルの完全性を検証し、結果の詳細なレポートを提供します。SAS IQは、すべてのWindows、UNIX、OpenVMSのSAS 9.1.3をサポートしています。詳細は、3ページの「SASインストール検証ツール」を参照してください。

SAS OQは、SASシステムが使用可能な状態であることの検証を支援します。SAS OQは、SAS社のコンポーネント開発グループによって提供されているSASプログラムを使用します。そのプログラムを実行および処理し、結果をレポートします。SAS OQは、すべてのWindowsとUNIXのSAS 9.1.3をサポートしています。詳細は、7ページの「SAS動作検証ツール」を参照してください。

SASインストール検証ツール

概要

SASインストール検証ツール（SAS IQ）は、インストールされた各ファイルの正当性を検証することにより、SASシステムのインストールを確認します。すべてのファイルの結果の詳細がレポートとして生成されます。この判定では、各ファイルの値の作成にMD5アルゴリズムを使用しています。

Windows

インストール検証ツールの実行

SAS 9.1.3のインストールで、SAS IQを実行するには、2つの方法があります。

- [スタート]メニューから、次のように選択します。

[スタート] → [プログラム] → [SAS] → [sas 9.1 utilities] → [SAS Installation Qualification tool]

または

- MS-DOSプロンプトから、コマンドラインに次のコマンドを入力します。

```
c:¥><SASROOT>¥sastest¥sasiq.exe -OutputPath <dir>
```

ここで、<SASROOT>は、SASシステムをインストールした場所、<dir>は空のディレクトリです。

これらの方法は、空の出力ディレクトリが必要です。

結果レポート

SAS IQは、検証処理の結果を保存するXMLデータファイルを生成します。このXMLデータは、SASシステムによってPDF形式とHTML形式のレポートに変換されます。どちらのレポートも同じ内容です。PDF形式のレポートは、印刷に適しています。HTML形式のレポートは、画面の表示に適しています。

Windows環境では、XSL変換ファイルも用意されています。これによって、XMLデータファイルからHTML形式のレポートを生成することもできます。ただし、XSL変換は動的に実行されます。

いずれの場合も、出力データのレポートは、最初にSASシステムのコンポーネントレベル、次に各コンポーネントのファイルごとにまとめられています。

OutputPathオプション

次の表は、-OutputPathオプションで作成されるディレクトリの構造を説明しています。

ファイル名	説明
Data	各コンポーネントの生の検証データをXMLファイルで保存するディレクトリ。
Html	各コンポーネントのHTMLファイルを保存するディレクトリ。
autoexec.sas	XMLデータからPDFファイルとHTMLファイルを出力する際に使用されるSASファイル。
index.xml	すべてのコンポーネントの要約となるXMLインデックスファイル。
index.xsl	XMLスタイルシート
media.xsl	XMLスタイルシート
sasiq.htm	要約とコンポーネント情報のインデックスを含むHTMLファイル。
sasiq.log	autoexec.sasの実行によって生成されるSASログファイル。
sasiq.pdf	すべての要約とコンポーネント情報を含むPDFファイル。
SASIQindex.map	XMLファイルをPDFファイルとHTMLファイルに出力する際に使用される中間ファイル。
SASIQprod.map	XMLファイルをPDFファイルとHTMLファイルに出力する際に使用される中間ファイル。
sasiqtoc.htm	コンポーネント情報を表示する際に使用されるHTMLファイル。

UNIX

方法 1:SASインストール担当者によるインストール検証ツールの実行

SASインストール担当者は、いつでも次の手順にしたがってSAS IQを実行できます。

1. !SASROOT/sassetupで、SASセットアップを実行します。SASROOTのファイルを更新できる権限があることを確認します。
2. SASセットアップのプライマリメニューで、[Run Setup Utilities] を選択します。
3. [Perform SAS System Configuration] を選択します。
4. [Run the Installation Qualification Tool] を選択します。
5. 検証結果の要約であるインストールログファイルの内容を確認します。または画面に表示される.xmlファイルの詳細レポートを確認します。

検証レポートとログは、!SASROOT/.install/logsに保存されます。

方法 2:ユーザーによるインストール検証ツールの実行

SASのユーザーは、いつでもSAS IQを実行できます。その場合、SASセットアップに-validateと-valoutdirのコマンドフラグが渡されていれば、SASROOTディレクトリへの書き込みアクセス権は必要ありません。これらのコマンドフラグにより、ログファイルとアウトプットはすべて指定したディレクトリにリダイレクトされます。ユーザーには指定したディレクトリの書き込みアクセス権が必要です。コマンドの例を次に示します。


```
$ !SASROOT/sassetup -validate -valoutdir <directory>
```

<directory>はユーザーが書き込みアクセス権を持っているディレクトリです。

SASインストール検証ツールの結果の参照

SAS IQを実行したら、validate_date.versionディレクトリがSASROOT/install/adminディレクトリ、またはSAS IQを起動した際に指定した出力ディレクトリに作成されます（すでに説明した、SAS IQの2つの起動方法を参照してください）。SAS IQの結果を参照するには、sasiq.htmファイルをWebブラウザで開くか、Adobe Readerでsasiq.pdfファイルを開きます。

指定したディレクトリに作成されるファイルは、次のとおりです。

ファイル名	説明
Data/	各コンポーネントのXMLファイルを保存するディレクトリ。
html/	各コンポーネントのHTMLファイルを保存するディレクトリ。
autoexec.sas	XMLデータからPDFファイルとHTMLファイルを出力する際に使用されるSASファイル。
index.xml	インストールされたソフトウェアの検証の要約を保存したXMLインデックスファイル。
index.xsl	XMLスタイルシート
media.xsl	XMLスタイルシート
sasiq.htm	インストールされたソフトウェアの検証の要約を保存したHTMLファイル。
sasiq.pdf	インストールされたソフトウェアの検証の要約を保存したPDFファイル。
SASIQindex.map	XMLファイルをPDFファイルとHTMLファイルに出力する際に使用される中間ファイル。
SASIQprod.map	XMLファイルをPDFファイルとHTMLファイルに出力する際に使用される中間ファイル。
sasiq.log	autoexec.sasの実行によって生成されるSASログファイル。
sasiqtoc.htm	目次のHTML出力テーブルを保存するファイル。

SAS IQは、インストールされた各ファイルが正しいことを確認することによって、SASシステムのインストールを検証します。すべてのファイルの結果の詳細がレポートとして生成されます。この判定では、各ファイルの値の作成にMD5アルゴリズムを使用しています。

SAS動作検証ツール

ここでは、インストールされたSAS 9.1.3の動作テストを実行するツールについて説明します。この検証ツールとテストウェアはSAS 9.1.3のCD-ROMに含まれています。

提供しているツール

SAS 9.1.3には次のツールが用意されています。

- SAS動作検証ツール（SAS Operational Qualification Tool : SAS OQ） – 動作テストを呼び出すプログラムです。以下に説明するコマンド行引数で使用できます。このツールはテストを実行し、出力データを検証して、レポートの生成を行います。
- IVFND – Filter N Diffプログラムです。テストの出力データやベンチマークは、比較する前にフィルタをかけ、日時やパス名など見かけ上の違いを削除する必要があります。IVFNDは、適切なフィルタと差分ツールを実行し、出力データファイルを検証します。
- PED - IVFNDにフィルタリング機能を提供するUNIXのsedコマンドのPerl版です。
- IVDIFF (sasdiff.exe) – 差分ツールです。差分を制御するオプションがあります。

ユーザー環境

通常、ユーザーの環境にはユーザーテスト、SASが提供するテスト、テスト出力領域、テスト対象のSASイメージのディレクトリがあります。ユーザーテストのディレクトリは、SASが提供するテストと同じ形式で構成されていることが必要です。SASが提供するテストのディレクトリは、コンポーネント別のサブディレクトリに分かれています。各コンポーネントのサブディレクトリにはSASプログラムとテストテーブルが格納されています。また、テスト出力用のディレクトリを用意する必要があります。既存のテストデータのあるディレクトリに出力しようとするとうエラーが発生します。

ディレクトリ構造： UNIX

```
SASROOT - /usr/local/sas/  
SAS Operational Qualification Tool -  
/usr/local/sas/sastest/sasoq.sh  
SAS Operational Qualification Tool support files -  
/usr/local/sas/sastest  
Component content - /usr/local/sas/sastest/<component>
```

Windows

```
SASROOT - C:\Program Files\SAS\SAS 9.1\  
SAS Operational Qualification Tool and support files -  
C:\Program Files\SAS\SAS 9.1\sastest\
```

```
Component content - C:\Program Files\SAS\SAS
9.1\Sastest\<component>
```

起動

Windowsでは、次のコマンドで起動します。

```
<SAS Operational Qualification Tool path>sasoq.exe -tables
*:<component> ...
```

UNIXでは、次のコマンドで起動します。

```
<SAS Operational Qualification Tool path>sasoq -tables
*:<component> ...
```

SAS OQでこのコマンドを使用すると、指定したコンポーネントのテストがすべて実行され、テスト出力とデータが現在のディレクトリ (ftt_<YYYYMMDD>.000) の新しいサブディレクトリに書き込まれます。最後の.000は、同じ日にツールを実行した場合、1ずつ増えていきます。

次のコマンドでは、ユーザーテスト用のテストウェアの位置が指定できます。

```
<SAS Operational Qualification Tool path>sasoq.exe -testware <path
to testware> -tables *:<subdirectory> ...
```

このコマンドでは、-testwareのパスに指定したテストだけが実行されます。

次のコマンドでは、生成したデータファイルの出力データパスが指定できます。

```
<SAS Operational Qualification Tool path>sasoq.exe -tables *:base
*:ets *:or *:qc *:stat
-outdir c:\public\mydata
```

このコマンドはWindowsで使用します。このコマンドでは、base、ets、or、qc、statの各ディレクトリでテストを実行し、c:\public\mydataにデータファイルを書き込みます。mydataというサブディレクトリが存在しない場合、新しく作成されます。

verboseオプションを指定すると、画面に詳細な内容が表示されます。

```
/usr/local/sas/sasoq.sh -tables *:ets -outdir /users/guest/mydata
-verbose
```

これはETSのテストを実行するUNIXのコマンドで、データファイルはローカルのmydataディレクトリに書き込み、テストを実行するSASコマンド行、処理中のステートメント、テスト結果を表示します。

```
Running test ets:tstets:tstari01...
Running: sas -sysin "/usr/local/sas/sastest/ets/tstari01.sas" -
autoexec "/usr/local/sas/sastest/base/assert.sas" -nodate -
nostimer -ls 78 -ps 60 -noovp -nosyntaxcheck
sasoq: The command returned the proper value (0).
Processing Results:
```

quietオプションを指定すると、エラー以外は出力しません。

```
<SAS Operational Qualification Tool path>/sasq.sh -tables *:ets -
outdir /users/guest/mydata -quiet
```

このコマンドでは、すべてのテストが終了するまでは何も表示しません。テスト終了後、データファイルを保存したパスを表示します。

テストの自動化

SAS OQは、プロダクトやSAS 9.1.3のイメージに対して、動作テストを自動的に実行できます。Windowsでテストの自動化を設定する手順は、次のとおりです。

1. 下記の例のような、スクリプトを作成します。
2. タスクのスケジュールを作成します。
 - スケジュールするタスクは、[タスク] パネルから設定します。最近の Windows では、この機能はコントロールパネルにあります。
 - スケジュールしたタスクを作成する [タスクウィザード] を使用します。
 - [タスク] タブで、下記のwscriptコマンドを設定します。

```
wscript.exe "c:¥sasiq¥runoq.vbs"
```

- [スケジュール] タブで、テストを実行する適切な頻度を選択します。
 - [OK] をクリックして、スケジュールされたタスクを作成します。
3. テストが実行された後、ツールの出力を参照します。

UNIXでテストの自動化を設定する手順は、次のとおりです。

1. 下記の例のような、スクリプトを作成します。
2. crontab entry に、次のようなコマンドを入力して実行します。

```
# run SAS 9.1 SAS Operational Qualification Tool process
# at 09:05am Weekdays
05 09 * * 1,2,3,4,5 "/users/guest/bin/runoq >
/users/guest/mydata/runoq.log"
```

3. テストが実行された後、ツールの出力を参照します。

Windowsの例 :

```
'* runoq.vbs - run the SAS Operational Qualification Tool using
all the component tests
'* First edition

'* run this script only on the local machine
strComputer = "."

oqpath = "¥sasq¥mydata"

'* create the string to use for executing the tool
```

```
initoq = "c:\program files\sas\sas 9.1\sastest\sasoq.exe -tables
*:base *:dmine *:ets *:graph *:hpf *:iml *:insight *:irp *:lab
*:or *:qc *:stat -outdir " & oqpath

'* execute the tool
Set WshShell = WScript.CreateObject("WScript.Shell")
Set oExec = WshShell.Exec(initoq)
```

UNIXの例 :

```
#!/bin/sh#runoq - execute the SAS Operational Qualification Tool

/usr/local/sas/sastest/sasoq.sh -tables *:base *:dmine *:ets
*:graph *:hpf *:iml *:insight *:irp *:lab *:or *:qc *:stat -outdir
/users/guest/mydata
```

出力レポート

SAS OQは、検証処理の結果を保存するXMLデータファイルを生成します。このXMLデータは、SAS 9.1.3によってPDF形式とHTML形式のレポートに変換されます。どちらのレポートも同じ内容ですが、用途が異なります。PDF形式は出力に適し、HTML形式は画面の表示に適しています。

テストの出力レポートはコンポーネント別に、さらにテストテーブル別にまとめられています。レポートには、コマンド行、戻り値、ベンチマークと比較した出力ファイルなど、実行されたテストに関連するデータがすべて含まれています。

SAS動作検証ツールのテーブル言語

SAS OQの基本入力ファイルはテストテーブルで、ファイルの拡張子は.tabです。これらのファイルは、どのテストをどのように実行してどのように終了するかを定義します。

テーブル言語の一般規則

テストテーブルファイルの基本規則を以下に示します。

- 通常、SAS OQでは空白や改行は無視されます。
- コメントは `/*` と `*/` で区切ります。コメントは複数行にまたがることができ、どの位置に挿入することもできます。C言語のコメントと同じように使用できます。
- ステートメント引数は次のいずれかに分類できます。
 - **リスト**： 順不同に並べ、カンマで区切った項目の集まりです。たとえばテストするプログラムに渡すオプションは、リストで指定します。この場合、オプションの順序はSAS OQの動作とは関係がありません（テストするプログラムによってはオプションの並び方には一定の順序が必要ですが、SAS OQでは必要ありません）。
 - **構造**： 特定の順序で並べ、コロンで区切った項目の集まりです。たとえばファイル指定は構造です。要素の並び方には一定の順序があるので、SAS OQはファイル名やコンポーネント名を区別することができます。
- ステートメントはテストブロックに現れる順序で処理されるので、テストを実行するステートメントは結果を処理するステートメントよりも先に置く必要があります。
- テーブルはコメント、`&set`コマンド、`&unset`コマンド、テストブロックで構成されています。テストブロックはブロック識別子`&test`から始まり、次に`&run`（1回だけ）、さらに`&process`や`&rc`など一連のコマンドが続きます。コマンド内では`&infile`や`&resfile`などのマクロを使用することもできます。

ブロック識別子

有効なブロック識別子は`&test`だけです。

`&test`

使い方：

```
&test testname {test_stmt_list}
```

引数：

- `testname` は、このテストの名前です。
- `test_stmt_list`は、テストを構成するステートメントのリストです。

説明：

`&test`ステートメントは、テストブロックを定義します。テストテーブルファイルには、1つのテストに対して1つのテストブロックしか置くことはできません。

例：

この項の最後に示す例を参照してください。

コマンド

コマンドには`&process`、`&rc`、`&run`、`&set`、`&unset`があります。

&process

使い方：

```
&process (verb :(arguments) [ , verb :(arguments) ...])
```

引数：

- *verb* は、実行するコマンド名です。現在のバージョンでは、`ivfnd` だけがサポートされています。
- *arguments* は、コマンドの引数です。

説明：

`&process`ステートメントは、テストの実行結果を処理します。通常は、IVFNDプログラムを実行することによって、出力ファイルとベンチマークファイルをフィルタし、ベンチマークファイルと比較します。IVFNDの戻り値により、ベンチマークの比較が成功したか失敗したかを判断します。成功した場合の戻り値は0です。

`&process`コマンドの使用はオプションです。ただし、`&process`コマンドは各テストブロックで1度しか使用できません。`&process`を使用しない場合は、テストブロックで`&rc`コマンドを使用する必要があります。

例：

```
&process(ivfnd :(-filter &infile(prs:logfilter) -result  
&resfile(log)),  
          ivfnd :(-filter &infile(prs:lstfilter) -result  
&resfile(lst)))
```

&rc

使い方：

```
&rc (value)
```


引数 :

*value*は、このテストで期待される戻り値です。

説明 :

`&rc`ステートメントは、`&run`で指定したコマンドの戻り値をテストします。戻り値と指定した値が一致しない場合は、テストは失敗したことになります。

`&rc`コマンドは、任意で選択できます。ただし、`&rc`コマンドは、各テストブロックで1度しか使用できません。`&rc`を使用しない場合は、テストブロックで`&process`コマンドを使用する必要があります。

例 :

```
&rc(0)
```

&run

使い方 :

```
&run (verb :(arguments))
```

引数 :

- *verb* は、実行するプログラム名です。初期バージョンでは、`sas`のみサポートしています。
- *arguments* は、実行するプログラムのオプションと引数のリストです。

説明 :

`&run`ステートメントは、実際のテストコマンドを実行します。各ブロックには`&run`コマンドが1つだけあります。`&run`には`&process`や`&rc`が続きます。

例 :

```
&run(sas :(-sysin &infile() -ls 78))
```

SASシステムを実行するコマンドです。`&infile`ステートメントは、テスト名と同じベース名を持つ入力ファイルを渡します。他のオプションはそのままの形で渡されます。

&set

使い方 :

```
&set (name :value)
```

引数 :

- *name* には変数名を指定します。英数字のみ使用できます。
- *value* には、複数の連続したトークンを指定できます。トークンに括弧が含まれている場合、それら是对応がとれているか、もしくはエスケープ文字（`¥`）を直前に入力しなければなりません（例参照）。

説明：

このコマンドは、変数と割り当てる値を定義することができます。&setを使用して定義した変数名と値は、グローバル変数となり、そのタスクの処理中で使用できます。

新たに&setステートメントを使用して、新しい値を変数に割り当てることができます。

一度変数を定義すると、テーブル中のどこでもその変数を使用することができます（その前に&setステートメントを使用している必要があります）。作成した変数を使用するには、@nameを使用して変数の値を挿入します。定義されていない変数を使用した場合、空の文字列が値として使用されます。

変数値の内側のすべての括弧は、対応がとれていなければなりません。対応がとれていない括弧を使用する場合、その括弧の前にエスケープ文字（¥）を置きます。

&setステートメントは、ブロックの内側で使用してはなりません。

変数testnameと変数tablenameは、あらかじめ定義されていて、変数が展開されたときにテスト名またはテーブル名が設定されます。

変数値の中で@を使用する必要があるが、変数の展開として扱いたくない場合、@の前にエスケープ文字（¥）を置きます。

以下に例を示します。

例：

```
&set(sasopts :-ls 78 -ps 60)
```

このコマンドは、変数sasoptsに「-ls 78 -ps 60」を設定しています。

```
&set(infile :-sysin &infile( sas :@testname :base))
```

このコマンドは、変数infileに「-sysin &infile(sas : @testname : base)」を設定しています。

```
&set(infile1 : -sysin &infile¥( sas : )
&set(infile2 : : base¥))
&set(run : sas : ( @infile1 @testname @infile2 ¥@noexpand))
```

このコマンドは、変数infile1に-sysin &infile(sas :を、変数runにsas : (@infile1 @testname @infile2 @noexpand)を設定しています。テスト名をtest1とすると、変数runは、sas : (-sysin &infile(sas : test1 : base)に展開されます。

&unset

使い方：

```
&unset (name)
```

引数：

- *name*には、&setステートメントあるいは他の方法で、設定された変数を名を指定します。変数名を指定するとき、@を含めないことに注意してください。そうしないと、unset処理の前に変数とその値によって置き換えられます。

説明：

このコマンドは、変数に設定されている値を取り消します。変数を使用しても展開しません。

例：

```
&unset(testopts)
```

このコマンドは、変数testoptsの設定を取り消します。

マクロ

マクロには、&infile、&outfile、&resfileがあります。

&infile

使い方：

```
&infile(ext [: name [: component]]) または
&infile([ext]: name [: component]) または
&infile([ext] : [name] : component)
```

引数：

- *ext* はファイル拡張子です。省略した場合、デフォルトはsasになります。
- *name* はファイル名です。省略した場合、テスト名がファイル名として使用されます。
- *component* は、ファイルが置かれているコンポーネント領域です。省略した場合、テーブルファイルのコンポーネント領域が使用されます。

説明：

&infileステートメントは、入力ファイルを定義します。通常は、&runステートメントのコマンドオプションの一部として使用します。引数をすべて省略して、&infileだけで使用することもできます。

例：

```
&infile(sas : foobar : base)
```

baseテストウェア領域のファイルfoobar.sasを指定しています。

```
&infile()
```

.tabファイルがあるコンポーネントのテストウェア領域のファイルtestname.sasを指定しています。testnameは現在のテスト名です。

&outfile

使い方 :

```
&outfile(ext [: name ])
```

引数 :

- *ext*はファイル拡張子です。デフォルトの拡張子はありません。拡張子を指定する必要があります。
- *name*はファイル名です。省略した場合、テスト名がファイル名として使用されます。

説明 :

&outfileステートメントは、出力ファイルを定義します。通常は、&runステートメントのコマンドオプションの一部として使用します。

```
&outfile( lst : foo )
```

ファイルfoo.lstを指定しています。

```
&outfile(log)
```

ファイルtestname.logを指定しています。

&resfile

使い方 :

```
&resfile(ext [: name [: component]]) または  
&resfile(ext: name [: component])
```

引数 :

- *ext* はファイル拡張子です。デフォルトの拡張子はありません。拡張子を指定する必要があります。
- *name* はファイル名です。省略した場合、テスト名がファイル名として使用されます。
- *component* は、ファイルが置かれているコンポーネント領域です。省略した場合、テーブルファイルのコンポーネント領域が使用されます。

説明 :

&resfileステートメントは、テストの出力ファイルと対応するベンチマークを定義します。通常は、&processステートメントのIVFNDコマンドオプションの一部として使用します。実際には、1組のファイルを指定します。1つは、テストの出力ファイルです。テストが実行されるワークディレクトリに生成されます。もう1つは、ベンチマークファイルです。ベンチマークファイルは、指定したディレクトリにあります。ファイルの名前と拡張子は同じでなくてはなりません。

例 :

```
&resfile(log : foobar : test : base)
```

baseテストウェア領域のファイルfoobar.logを指定しています。

```
&resfile(lst)
```

.tabファイルがあるコンポーネントのテストウェア領域のファイルtestname.lstを指定しています。testnameは現在のテスト名です。

例

```
&set (sasopts : -sysin &infile -log &outfile( log : barfoo ))
&test foobar
{
&run(sas : (@sasopts))
&process(ivfnd : (-filter &infile(prs : logfilter) -result
&resfile(log, barfoo)),
        ivfnd : (-result &resfile( lst )))
&rc(0)
}
```

この例では、入力ファイルfoobar.sasを使用してSASシステムを実行します。ファイルはテストテーブルファイルと同じコンポーネント領域にあります。このテストにより、2つの出力ファイルbarfoo.logとfoobar.lstを生成します。ログファイルと対応するベンチマークは、比較される前に、フィルタスクリプトlogfilter.prs（現在のプロダクトのコンポーネント領域にあります）のpedを使用してフィルタがかけられます。lstファイルは比較される前にフィルタをかけられません。SASコマンドは戻り値0で終了します。

SAS動作検証ツールのコマンド行

SAS OQのコマンド引数は、次のように定義します。

```
-tables table_spec [table_spec ...]
```

-tablesオプションは、どのテストテーブルを実行するかを指定します。*table_spec*の形式はname:componentです。これは&infileステートメントおよび&resfileステートメントでの指定方法とよく似ていますが、拡張子は必ず.tabになります。*table_spec*は、少なくとも1つ指定する必要があります。

```
-help
```

-helpオプションを指定すると、ツールの使い方が出力されます。

```
-verbose | -quiet
```

この2つの引数は、実行中のSAS OQによるSTDOUTへの出力を規定します。-verboseオプションを使用すると、STDOUTへ詳細に出力します。これは、新しいテストを設定する場合に便利な機能です。-quietオプションを指定すると、エラーだけを出力します。これは、SAS OQコマンドを記述している場合に便利な機能です。

```
-sasroot <path>
```

この引数は、SASをインストールしている場所を指定します。デフォルトでは、SASROOTのディレクトリになります。何らかの理由によりSASROOTを使用できない場合、この引数を使用してパス名を指定します。同一マシン上に複数のSASをインストールしている場合も、この引数を使用します。

```
-testware <path>
```

この引数を使用して、テストウェアの場所を指定します。デフォルトは、SASROOT/sastestです。

```
-outdir <path>
```

この引数を使用して出力ファイルの出力先を指定します。デフォルトはツールの実行先である現在のディレクトリです。

Perl Stream Editorについて

Perl Stream Editor (ped) は、sed (元はUnix SysV Stream Editor) に代わるエディタとして設計されました。sedを基本にしていますが、pedが認識するフィルタ式はsedとは異なります。sedに有効なフィルタ式の一部はpedにも有効ですが、すべてが有効なわけではありません。pedには既存のsedフィルタファイルを変換する機能はありません。

pedを使用する最大の利点は、便利ではあっても制限があるsed正規表現を使用するのではなく、Perlの正規表現を自由に利用できることです。また、他のフィルタファイルを含める構文

が使用できる点も便利です。これは他の言語においてファイルを読み込むプロセスに似ています。

pedでは、Perlの提供する正規表現コマンドの拡張を定義しています。このコマンドを、Substringing Substitutionと呼びます。このコマンドを使用すると、入力ストリームのサブストリングを識別する正規表現を作成して、そのサブストリングだけに置換の正規表現を適用できます。この機能によりフィルタ対象を特定し、実際の検索パターン正規表現を単純化することができます。たとえば、次のようなコマンド構文が使用できます。

```
S/<substring-RE>/<substring-Options>/<search-RE>/<replacement-RE>/<substitute-Options>
```

substring-Optionsは、Perlの置換コマンドで利用できるオプションと同じです。Substringing Substitutionコマンドは、置換コマンドのペアとして実行されます。

Perlの正規表現ででき、sedの正規表現ではできないこと

Perlの正規表現では、復帰改行文字が何文字あってもパターンにマッチさせることができます。また何重にもネストさせたパターンでもマッチさせることができるので、HTML、XML、SGMLなどのマークアップ言語で使用すると便利です。空白スペース、ワード文字、8進文字、16進文字、10進数など、豊富なメタシーケンスがあります。マッチしたサブストリングをメモリバッファに取り込み、出力文字列に戻して置き換えることもできます。前方検索も後方検索もできます。限定子を簡単に表現して新しい文字クラスを定義することができます。複雑な正規表現に可読式のコメントを直接追加して、他のユーザーが理解して保守できるようにした拡張構文もあります。

また、pedで利用できるフィルタ式の数には制限がありません。sedの場合は100個までに制限されています。

フィルタファイル構文

- #で始まる行はコメント行として無視されます。これはpedのコメントですが、sedでも同じです。
- 空白スペースしかない行は無視されます。
- "%include"で始まる行はコマンドを含んでいると解釈され、正規表現とは見なされません。pedでは"%include"に続くトークンが検索され、ファイルパスと見なされます。そのファイルを読みとり、内容を追加正規表現として処理します。

pedコマンド行の構文

```
ped [-e <'|"><filter RE><'|">] [-f <filter file path>] [-n] [<file path 1> [<file path n>]]
ped <'|"><filter RE><'|"> [<file path 1> [<file path n>]]
```

Substringing Substitute ("S"コマンド) 構文

```
S/<substring-RE>/<substring-Options>/<search-RE>/<replacement-RE>/<substitute-Options>
```

substring-Optionsは、Perl正規表現の置換 (s) コマンドで利用できるオプションと同じように定義されています。substring-REとsearch-REは必須です。substring-Options、replacement-RE、substitute-Optionsはオプションです。

substring-REを入力ストリームに適用することによって、sコマンドが有効になり、通常のsコマンドが入力ストリームでマッチした部分にのみ適用されます。

Pedフィルタファイルの例

```
mylogfilter.prs:

#-----
# Purpose: To filter out the common, non-component-specific stuff
#           from SAS .log files.
# Filter: ped
#-----

# Get rid of the page headers
s/^\%f*\%d{1,6}\%s+The SAS System\%s+$/gmo
# Get rid of the copyright notice.
s/^\%s*note:\%s*copyright\%(c\%)\.*$/imo
# Get rid of the "proprietary/licensed" note.
s/^\%s*note:\%s*sas\%s*\%(r\%)\%s*proprietary.+?\%n\%s*licensed.+?site.+?$/
//imos
# Get rid of the Institute address note.
s/^\%s*note:\%s*sas\%s+institute\%s+inc\%.,\%s+sas\%s+campus\%s+drive,\%s+c
ary,\%s+nc.+?$/imo
# Get rid of the "SAS used" message.
s/^\%s*note:\%s+the\%s+sas\%s+system\%s+used:\%s*\%n\%s+real\%s+time\%s+\%d+\%
.\%d\%d\%s+seconds\%s*\%n\%s+cpu\%s+time\%s+\%d+\%.\%d\%d\%s+seconds\%s+$/imos
```

問題点

pedでは、処理が始まる前にメモリへの入力ストリームをすべて読みとるので、pedでフィルタできるファイルのサイズは使用可能な合計メモリ容量を超えることができません。しかしながら、最近のメモリは容量が大きいので、ファイルが大きすぎてフィルタできないということはありません。sedで読み取るのは一度に数行だけなので、事実上どのサイズのファイルでも処理することができます。

IVFNDコマンド

IVFND (Install Validation Filter 'N' Diff) は、SAS OQにおいて、フィルタと差分の算出を行います。通常は、テストテーブルの&processステートメントから呼び出します。

引数:

```
-filter ext:name:component
```


`-filter`オプションには、フィルタ入力ファイルのファイル名とディレクトリを指定します。このオプションを使用すると、IVFNDは比較する前に、指定したフィルタスクリプトを使用したpedを実行して、テスト出力ファイルと対応するベンチマークファイルをフィルタします。すべての項目を指定する必要があることに注意してください。

```
-result ext:name:component
```

`-result`には、結果ファイルおよび対応するベンチマークファイルのファイル名とディレクトリを指定します。結果とベンチマークのファイルをフィルタした後（`-filter`オプションを指定した場合）、IVDIFFで比較されます。

戻り値：

0：入力ファイルがすべて見つかり、正常にフィルタが終了し（指定した場合）、フィルタされたファイルが正しく比較された場合。

0以外：何らかの理由で正しく処理されていません。stdout/stderrにメッセージが出力されます。ファイルが比較されなかった場合は、現在のワークディレクトリに残っている差分ファイルを見て原因を調査することができます。IVFNDコマンドのいずれかが0以外の戻り値を返した場合は、そのテストは失敗したことになります。



support.sas.com

SAS is the world leader in providing software and services that enable customers to transform data from all areas of their business into intelligence. SAS solutions help organizations make better, more informed decisions and maximize customer, supplier, and organizational relationships. For more than 30 years, SAS has been giving customers around the world The Power to Know®. Visit us at **www.sas.com**.

英語版更新日 June 19 2007

SAS 9.1.3 検証ツール ユーザーガイド

2008年5月23日 第3版第5刷発行 (913E6)

発行元 SAS Institute Japan株式会社

〒106-6111 東京都港区六本木6丁目10番1号 六本木ヒルズ森タワー11階

本書の内容に関する技術的なお問い合わせは下記までお願い致します。

SASテクニカルサポート

TEL: 03(6434)3680 FAX: 03(6434)3681