SAS Technical News

Spring 2000

CONTENTS

特集

SAS/AFソフトウェアによるGUIアプリケーション作成方法

はじめに

SAS/AFソフトウェアを使用してさまざまなGUIアプリケーションを作成することができます。ここでは、その際に必要となる、便利なテクニックを紹介します。

- 1ではフレーム間で情報を受け渡す方法を4種類紹介します。
- 2では一般的なアプリケーションであるマスター／ディテイル アプリケーションの作成方法を2通り紹介します。
- 3では2の例を応用し、トラフィックライティング（例外強調）するアプリケーションの作成方法を紹介します。

1. フレーム間での情報の受け渡し

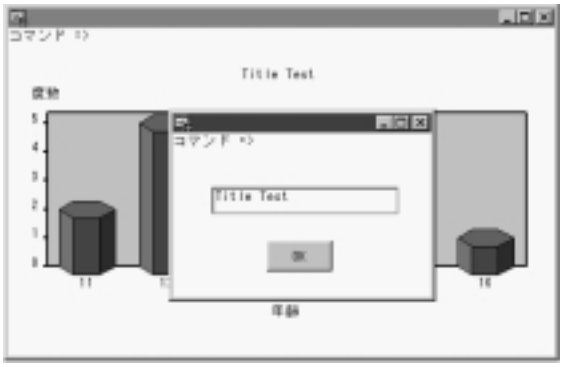
複数のフレームで構成するアプリケーションを作成する時、フレーム間で情報の受け渡しが必要となる場合があります。ここでは簡単な例を挙げながら、4通りの方法を紹介します。

1.1 DISPLAYルーチンのパラメータを利用する方法

一番基本的で、簡単な方法です。DISPLAYルーチンで受け渡されたパラメータは、呼び出されたフレーム内で値が変更された場合、呼び出したフレーム内でも値が自動的に変更されます。その仕組みを利用した方法です。

例題）

次ページの図のようなシンプルなアプリケーションを作成します。一つ目のフレームにはグラフがあります。そのグラフをクリックすると二つ目のフレーム（ダイアログ）が表示され、一つ目のフレームにあるグラフのタイトルを変更できます。



GRAPH.FRAMEのSCL

```
init:
   /* グラフオブジェクトのオブジェクトIDを取得します */
   call notify('.', '_get_widget_', 'graph1', graphid);
return;

graph1:
   /* 現在のタイトルを取得します */
   title1=getnitemc(graphid, 'title1');

   /* タイトルダイアログを呼び出します */
   call display('title.frame', title1);

   /* 新しいタイトルをセットします */
   call notify('graph1', '_set_title_', 1, title1);
return;
```

1.2 SENDルーチンを利用する方法

●CALL NOTIFYとCALL SENDの違い

SENDルーチンとNOTIFYルーチンはほぼ同じように使用できますが、オブジェクトを指定する方法が異なります。NOTIFYルーチンではオブジェクト名で指定するのに対し、SENDルーチンではオブジェクトIDでオブジェクトを指定します。オブジェクトIDは一つのSASセッションで一意のものなので、SENDルーチンはどのフレーム上にあるオブジェクトのメソッドでも実行できます。

●SENDルーチンの使用方法

SENDルーチンを使用するためには、まずオブジェクトIDを調べる必要があります。フレームオブジェクトの_GET_WIDGET_メソッドを実行することにより、オブジェクトIDを取得できます。下記の二通りの方法があります。

```
CALL NOTIFY('.', '_GET_WIDGET_',widget-name, widget-id);
```

```
CALL SEND(_SELF_, '_GET_WIDGET_',widget-name, widget-id);
```

例題）

1.1と同様に、シンプルなアプリケーションを作成します。しかし、今回は先の例題とは少し動作が異なります。タイトルダイアログでタイトルを変更してAPPLYボタンを押すたびに変更がグラフに反映されます。これはSENDルーチンによって、違うフレーム上にあるオブジェクトのメソッドを実行できるために可能になったものです。

GRAPH.FRAMEのSCL

```
init:
   /* グラフのオブジェクトIDを取得します */
   call send(_self_, '_get_widget_', 'graph1', graphid);
return;

graph1:
   /* ダイアログを呼び出します */
   call display('send2.frame', graphid, _self_);
return;
```

TITLE.FRAMEのSCL

```
entry graphid 8 frameid 8;

init:
   /* 入力フィールドに現在のタイトルを表示します */
   field1=getnitemc(graphid, 'title1');
return;

apply:
   /* 新しいタイトルを設定します */
   call send(graphid, '_set_title_', 1, field1);

   /* タイトルの変更を反映させます */
   call send(frameid, '_refresh_');
return;
```

1.3 環境リストを利用する方法

●環境リストとは

環境リストは複数のフレームから参照できる特別なSCLリストです。環境リストには、グローバル環境リストとローカル環境リストの2種類があります。ローカル環境リストは一つのAFタスク（1回のAFAまたはTESTAFコマンド）につき一つ、自動的に作成されます。グローバル環境リストは一つのSASセッションにつき一つ作成されます。

●ENVLIST関数

環境リストのリストIDは、ENVLIST関数を使って取得できます。

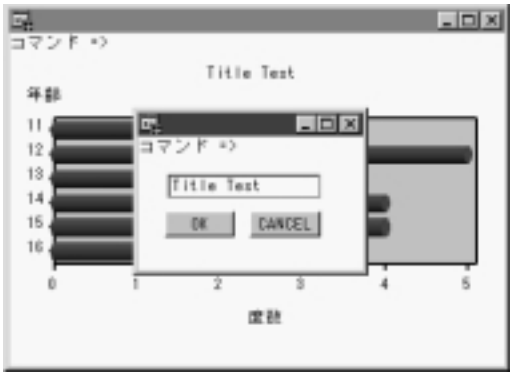
ENVLIST関数の構文

```
list-id=ENVLIST(class);
```

※ *class* には'G'（global）または'L'（local）が入ります。

例題）

ここで作成するアプリケーションでは、CANCELコマンドでタイトルダイアログを終了することにより、タイトルの変更のキャンセルが可能です。これはタイトルダイアログの終了時に_STATUS_自動変数をチェックして、_STATUS_='E' 以外の場合は環境リストに変更を加えないためです。



GRAPH.FRAMEのSCL

```
init:
   /* ローカル環境リストのIDを取得します */
   localid=envlist('L');
   put '(envlist1)' localid=;

   /* グラフオブジェクトのIDを取得します */
   call send(_self_, '_get_widget_', 'graph1', graphid);
return;

graph1:
   /* 現在のタイトルを取得します */
   title1=getnitemc(graphid, 'title1');

   /* ローカル環境リストに現在のタイトルを保存します */
   rc=setnitemc(localid, title1, 'title');

   /* タイトルダイアログを呼び出します */
   call display('title.frame');

   /* ローカル環境リストから変更後のタイトルを読み出します */
   title1=getnitemc(localid, 'title');

   /* 新しいタイトルをグラフにセットします */
   call notify('graph1', '_set_title_', 1, title1);
return;
```

TITLE.FRAMEのSCL

```
init:
   /* local listのリストIDを取得します */
   localid=envlist('L');

   /* 入力フィールドに現在のタイトルを表示します */
   field1=getnitemc(localid, 'title');
return;

term:
   if _status_='E' then
      /* local listに変更後のタイトルを保存します */
      rc=setnitemc(localid, field1, 'title');
return;
```

1.4 SAVELIST/FILLIST関数を利用する方法

●SAVELIST関数とFILLIST関数

SAVELIST関数とFILLIST関数を使用することにより、SCLリストをファイルへ保存したり、ファイルから読み込むことができます。ファイルに保存されたSCLリストを利用して、フレーム間で情報の受け渡しができます。

SAVELIST関数の構文

```
rc=SAVELIST(type, source, list-id<,attr-list-id<,description>>);
```

- *rc* **リターンコード。成功ならば0、失敗ならば0以外の値となります。**
- *type* **保存先ファイルの種類。'CATALOG'又は'FILE'、'FILEREF'。**
- *source* **保存先の名前。**
- *list-id* **保存されるリストのID。**
- *att-list-id* **テキスト属性が入っているSCLリストのID（*type*が'CATALOG'の場合のみ）。**
- *description* **カタログエントリの説明。**

FILLIST関数の構文

```
rc=FILLIST(type, source, list-id<, attr-list-id<, description>>);
```

- *rc* **リターンコード。成功ならば0、失敗ならば0以外の値となります。**
- *type* **SCLリストが保存されるファイルの種類。**
- *source* **読み込み元の名前。**
- *list-id* **値が入るリストのID。**

例題）

今回のアプリケーションではグラフのタイトルをファイルに保存するため、アプリケーションを終了しても次回起動した時にユーザーが設定したタイトルが復元します。

GRAPH.FRAMEのSCL

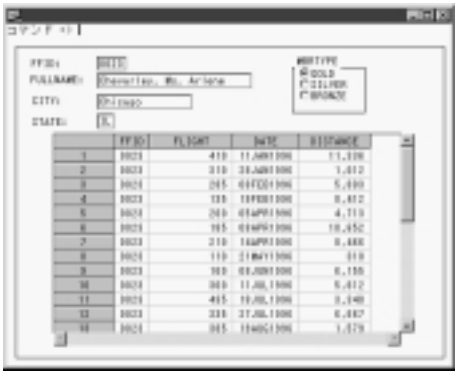
```
init:
   /* グラフオブジェクトのIDを取得します */
   call send(_self_, '_get_widget_', 'graph1', graphid);

   entry='sasuser.profile.titles.slist';

   /* タイトルを保持するためのリストを作成します */
   titles=makelist();

   /* ファイルからタイトルを読み込みます */
   link loadttl;
return;

/* 保存されているタイトルを読み込み、表示します */
loadttl:
   if cexist(entry) then do;

      /* ファイルからリストを読み込みます */
      rc=fillist('catalog', entry, titles);

      /* リストからタイトルを取り出します */
      title1=getnitemc(titles, 'title1');

      /* グラフにタイトルをセットします */
      call notify('graph1', '_set_title_', 1, title1);

   end;
return;

graph1:
   /* タイトルダイアログを呼び出します */
   call display('title.frame');

   /* ファイルからタイトルを読み込みます */
   link loadttl;
return;
```

TITLE.FRAMEのSCL

```
init:
   entry='sasuser.profile.titles.slist';

   /* タイトルを保持するリストを作成します */
   titles=makelist();

   if cexist(entry) then do;

      /* ファイルからリストを読み出します */
      rc=fillist('catalog', entry, titles);

      /* リストからタイトルを取り出します */
      field1=getnitemc(titles, 'title1');

   end;
return;
```

2. マスター／ディテイル アプリケーション

マスター／ディテイル アプリケーションとは、マスターデータセットで選択されたオブザベーションに関するデータだけ詳細データセットから抽出し、表示するアプリケーションです。
下の画面はマスター／ディテイル アプリケーションの一例です。データフォームには一人分の顧客情報が、データフォーム上のデータテーブルには、その顧客の搭乗履歴が表示されています。



このアプリケーションでは、二つのデータセットを使用しています。マスターデータセット（顧客情報）と、詳細データセット（全顧客の搭乗履歴）です。

マスターデータセット（顧客情報）の一部

FFID	FULLNAME	CITY	STATE	MBRTYPE
0023	Chevarley, Ms. Arlene	Chicago	IL	GOLD
0231	Chavez, Ms. Louise	Fort Lauderdale	FL	GOLD
0301	Licht, Mr. Bryan	Edwardsville	IL	SILVER
0386	Keever, Ms. Linda	Oakland	CA	BRONZE
0427	Wells, Mr. Roy	Cleveland	OH	GOLD

詳細データセット（搭乗履歴）の一部

FFID	FLIGHT	DATE	DISTANCE
1955	110	01JAN1996	3,923
2704	290	01JAN1996	9,178
3177	340	01JAN1996	3,731
7711	340	01JAN1996	7,738
6080	510	01JAN1996	7,145

このアプリケーションを実現するためには、データフォームで表示される顧客情報と、データテーブルで表示される搭乗履歴をリンクさせなくてはなりません。リンクをさせる方法には2通りあります。以下にそれぞれの方法をご説明いたします。

2.1 キー列を利用する方法

キー列の利用は容易ですが、制約があります。以下のいずれかの条件の時にキー列が使用できます。

・同期をとるための変数が一つだけ存在する
・同期をとるための条件が「=（イコール）」である

●作成手順

I. データフォームオブジェクトを作成する
1. テーブルに、マスターデータ（顧客情報）のデータセット名を設定する
2. データフォーム エントリには「FLYERS1.DATAFORM」（例）と設定する
3. OKボタンを押してデータフォームの属性ウィンドウを閉じる

II. データフォームオブジェクトの上に、データテーブルオブジェクトを作成する
1. テーブルに、詳細データ（搭乗履歴）のデータセット名を設定する
2. 列のカスタマイズを選択する
3. テーブルのカスタマイズウィンドウで、オプションボタンを押す
4. キー列に「FFID」（例）と設定する
5. 了解ボタンを2回押して、フレームに戻る

III. リンクを完成させる
1. データフォームオブジェクトの上で右クリックをして、ポップアップメニューから、[フォーム ->] を選択する
2. ポップアップメニューから、[列ウィンドウ] を選択する
3. FFIDを選択し、ドラッグして、データテーブルオブジェクトの上にドロップする

IV. 動作確認
1. TESTAFコマンドを使用して、プログラムを実行する
2. データフォームを選択した状態で、「Page Up」、「Page Down」キーを使用して、マスターデータと詳細データがリンクされていることを確認する

2.2 モデルSCLを利用する方法

●モデルSCLとは

モデルSCLは、（フレームではなく）データテーブル、またはデータフォームに関連付けられた特別なSCLエントリです。

モデルSCLの応用例
- ◆計算列の値の計算
- ◆データテーブル（又はデータフォーム）に入力された値のチェック
- ◆マスター／ディテイル アプリケーション
- ◆トラフィックライティング（例外強調）

●モデルSCLのラベル

モデルSCLにおいて、特別な意味を持つラベルを紹介します。

・DFINIT
全ての列が表示される前に、1度だけ実行されます。
・INIT
1行毎に、行が表示される前に実行されます。
・MAIN
フィールドの値が更新される度に実行されます。
・TERM
違う行へ移る前に実行されます。
・DFTERM
（アプリケーション終了時など）オブジェクトが終了する時に実行されます。
・列名
列の値が更新される度に、更新された列名と同名のセクションが実行されます。

●フレームSCLとモデルSCLの実行順序

データテーブルまたはデータフォームのフィールドが更新された際のSCL実行順序は、下記のようになります。
1. モデルSCLの更新されたフィールド名（列名）と同名のセクション
2. モデルSCLのMAIN
3. フレームSCLのデータテーブルまたはデータフォームオブジェクト名と同名のセクション
4. フレームSCLのMAIN

●モデルSCL内のSCL変数

モデルSCL内で使用できるSCL変数は、次の3種類です。

・列変数
テーブルにある列と同じ名前のSCL変数は、テーブルの列と同じ値を持ちます。テーブルの列の値が更新されると、同じ名前を持つSCL変数の値も自動的に更新されます。逆に、SCL変数の値を更新すると、テーブルの列の値も自動的に更新されます。

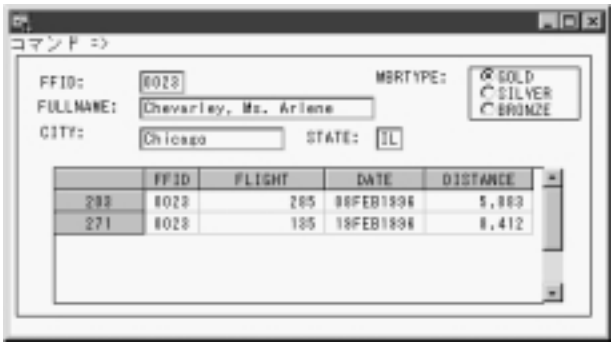
・列変数ではないSCL変数
テーブルの列名と関連のないSCL変数は普通のSCL変数として使用できます。モデルSCL内のSCL変数はローカルのSCL変数として扱われ、フレームSCL内のSCL変数から独立しています。

・システム変数
システムから情報を受け取ったり、渡したりするための特別な変数です。システム変数の一つである_VIEWER_は、値にデータテーブルまたはデータフォームのオブジェクトIDを持っています。

2.3 モデルSCLを利用したマスター／ディテイル アプリケーション

モデルSCLを利用すると、キー列では実現できないマスター/ディテイル アプリケーションを作成できます。
例えば次のような場合でも、モデルSCLを使用するとマスター/ディテイル アプリケーションを作成できます。
・サブセット化のキーとなる変数が複数の場合
・サブセット化の条件がイコール以外の場合

2.1で作成したマスター/ディテイル アプリケーションとほとんど同じですが、2月の搭乗履歴のみが表示されています。



サブセット化のキーとなる変数が複数（FFIDとDATE）のため、キー列だけではこのようなアプリケーションを作成できません。

●作成手順

I. データフォームオブジェクトを作成する
1. テーブルに、マスターデータ（顧客情報）のデータセット名を設定する
2. データフォーム エントリには「FLYERS2.DATAFORM」（例）と設定する
3. SCLエントリには「FLYERS_M.SCL」（例）と設定する
4. OKボタンを押してデータフォームの属性ウィンドウを閉じる

II. データフォームオブジェクトの上に、データテーブルオブジェクトを作成する
1. 名前を「TABLE」（例）と設定する
2. テーブルに、詳細データ（搭乗履歴）のデータセット名を設定する
3. 了解 ボタンを押して、フレームに戻る

III. モデルSCLを編集、コンパイルする
1. データフォームオブジェクトの上で右クリックをして、ポップアップメニューから、[フォーム ->]を選択する
2. ポップアップメニューから、[SCLの編集...]を選択する
3. 下記のプログラムを入力する

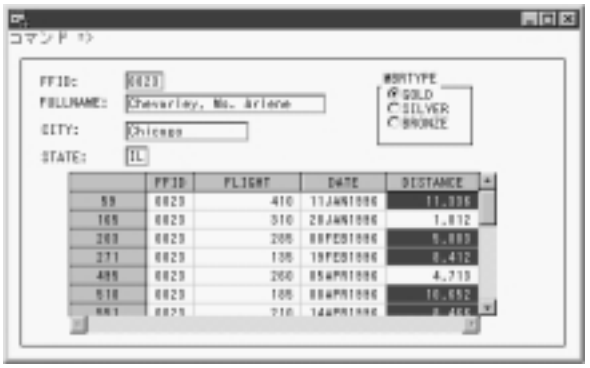
```
/* DFINIT 全ての行が表示される前に、一度だけ実行されます */
DFINIT:
   /* _SET_WHERE_メソッドで使用するリストを作成します */
   whereclist=makelist();
   /* データテーブルのオブジェクトIDを調べます */
   call send(_viewer_, '_get_widget_', 'OBJ2', tabid);
return;

/* INIT ユーザがスクロールして、各行が表示される前に実行されます */
INIT:
   /* WHERE条件を初期化します */
   rc=clearlist(whereclist);
   /* WHERE条件を作成します */
   rc=insertc(whereclist, 'ffid=' || quote(ffid), -1);
   rc=insertc(whereclist, 'and month(date)=' || month(today()), -1);

   /* WHERE条件をセットします */
   call send(tabid, '_set_where_', whereclist);
return;

/* DFTERM データフォームがなくなる時に、一度だけ実行されます */
DFTERM:
   /* リストを消去します */
   rc=dellist(whereclist);
return;
```

4. COMPILEコマンドを実行して、モデルSCLをコンパイルする
5. ENDコマンドを実行して、SCLの編集を終了する（同時にモデルSCLが保存されます）

IV. 動作確認
1. TESTAFコマンドを使用して、プログラムを実行する
2. データフォームを選択した状態で、「Page Up」、「Page Down」キーを使用して、次の項目を確認する
   ・マスターデータと詳細データがリンクされていること
   ・ 2月（当月と同じ月）の履歴のみが表示されていること

3．トラフィック ライティング（例外強調）

データテーブルのモデルSCL内で_SET_VIEWER_ATTRIBUTE_メソッドを使用すると、下の図のようなトラフィック ライティング（例外強調）が実現できます。



このアプリケーションは、2.1のキー列を使用したマスター/ディテイル アプリケーションに変更を加えたものです。DISTANCEが5,000マイル以上の場合はセルの背景が青色に、1,000マイル未満の場合はセルの背景が黄色になっています。

●作成手順

I. 先に作成したマスター/ディテイル アプリケーションのフレーム編集画面を開く
II. データテーブルのモデルSCLを編集、コンパイルする
1. データテーブルの上で右クリックをして、ポップアップメニューから[テーブル ->]を選択する
2. [SCLの編集...]を選択する
3. 下記のようにプログラムを入力する

```
/* INIT ー 各行が表示される前に、1度ずつ実行されます */
init:
   if distance >= 5000 then do;
      /* 背景を青色に、文字を白に設定します */
      call send(_viewer_, '_set_viewer_attribute_',
                'distance', 'bcolor', 'blue');
      call send(_viewer_, '_set_viewer_attribute_',
                'distance', 'fcolor', 'white');
   end;
   else if distance < 1000 then do;
      /* 背景を黄色に設定します */
      call send(_viewer_, '_set_viewer_attribute_',
                'distance', 'bcolor', 'yellow');
   end;
return;
```

4. COMPILEコマンドを実行して、モデルSCLをコンパイルする
5. ENDコマンドを実行して、SCLの編集を終了する（同時にモデルSCLが保存される）

III. 動作確認

1. TESTAFコマンドを使用して、プログラムを実行する
2. データフォームを選択した状態で、「Page Up」、「Page Down」キーを使用して、次の項目を確認する
   - DISTANCEが5,000マイル以上の場合、セルの背景が青色になっていること
   - DISTANCEが1,000マイル未満の場合、セルの背景が黄色になっていること

参考マニュアル

●SAS/AFソフトウェア：FRAMEエントリ 使用法およびリファレンス
Version 6, First Edition

●SAS/AFソフトウェア：FRAMEエントリ入門ガイド
Version 6, First Edition

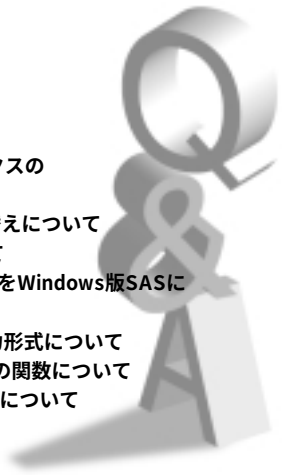
●SAS/AFスクリーンコントロール言語：リファレンス
Version 6, Second Edition

●SASスクリーンコントロール言語：使用法ガイド
Version 6, First Edition

●SAS/AFソフトウェア：使用法およびリファレンス
Version 6, Second Edition

●SAS/AFソフトウェア：FRAME Class Dictionary

Q&A



●Windows版SASでのダイアログボックスの
フォルダ変更について
●SAS/CONNECTのポート番号の切り替えについて
●UNIX版SASでのディスク分割について
●メインフレーム上のSASデータセットをWindows版SASに
取り込む方法について
●SAS/INSIGHTにおける棒グラフの出力形式について
●DISCRIMプロシジャでの2群判別分析の関数について
●MIXEDプロシジャでの反復測定の出力について

Q プルダウンメニューの［ファイル］から［オープン］を選択し、［ファイルを開く］ダイアログボックスを表示します。SASシステムをインストールしたフォルダ内のファイルが最初に表示されますが、他のフォルダに変更できますか。

A **SASシステムの作業フォルダ内のファイルが、最初に表示されます。デフォルトでは、SASセッションを起動したディレクトリが作業フォルダになりますので、SASプログラムショートカットのプロパティページで［作業フォルダ］フィールドの値を変更します。なお、作業フォルダの変更により、SASUSERライブラリ用とWORKライブラリ用のフォルダが変更されますが、次のように環境設定ファイル（CONFIG.SAS）を設定することで回避できます。**

例）
SASUSERシステムオプションとWORKシステムオプションに、SASルートディレクトリを指定します。

変更前			変更後	
-SASUSER	sasuser･	⟶	-SASUSER	c:･sas･sasuser
-WORK	saswork･	⟶	-WORK	c:･sas･saswork

Q SAS/CONNECTソフトウェアでTCP/IPを使用してリモートホストに接続をする際、リモートホスト側のTELNETのポートを'23'以外で定義しているため、下記のようなエラーとなります。使用ポートを変更する方法はありますか。

ログ

```
ERROR: TCP ソケットを接続できません.
       システムメッセージ '10061 - WSAECONNREFUSED'
ERROR: RHOST へのリモート接続を取り消しました.
```

A **SAS/CONNECTソフトウェアではリモートホストとの接続・切断時に、デフォルトでTELNETのポート'23'を使用します。このポートを変更する場合は、マクロ変数にそのポート番号を明示的に格納します。**

例）
リモートホストのIPアドレス　　：111.222.333.444
リモートホストのtelnet ポート番号：1234

```
%let rhost = 111.222.333.444 1234 ;
filename rlink 'スクリプトファイル名' ;
options comamid=tcp remote=rhost ;
```

Q UNIX版SASシステムで作成したSASデータセットを分割してディレクトリに保存するには、どのようにすればよいでしょうか。

A **PARTSIZEシステムオプションで、指定のディレクトリ内で1回あたりに使用可能なファイルサイズを指定することにより、SASデータセットを分割して保存することができます。下記に、ユーザ定義領域とWORK領域に対する指定方法を紹介します。**

［ユーザ定義領域に対する指定方法］
/disk1と/disk2にそれぞれ100GBずつ割り振る場合は、次のように指定します。

```
libname libref('/disk1','/disk2') type=partition partsize=100G;
                                    ※1           ※2
```

※1　type=partitionは省略できます。
※2　partsize=のデフォルト値は500MBです。変更する場合には、単位にG（ギガバイト）、M（メガバイト）、K（キロバイト）を使用して値を設定します。ただし単位を省略すると、K（キロバイト）と判断します。

また、以下のようにも指定できます。

```
libname lib1 '/disk1';
libname lib2 '/disk2';
libname libcon ('lib1','lib2') type=partition partsize=100G;
                                ※3
```

※3　type=partitionは必須です。

［WORK領域に対する指定方法］
SASシステムの起動時に使用する環境設定ファイル（config.sas612）に、次のように指定します。

```
-work (/disk3,/disk4,/disk5)
-partsize 200M
```

例）
物理的に/disk1～/disk10まで各1GB、合計10GB の一時作業領域が使えるディスクがあるとします。ここで、PARTSIZEにおいて200MBを指定したとします。

・ケース1：1GBのデータセットをWORK領域に作成する場合
PARTSIZEが200MBであることから、1GB/200MB=5となり、/disk1から順番にPARTSIZEで指定したサイズまでファイルを作成して、次の領域に作りにいきます。その結果、/disk1～/disk5まで作成することになります。

・ケース2：300MB のデータセットをWORK領域に作成する場合
/disk1に200MB、/disk2に100MBのファイルを作成します。

・ケース3：3GBのデータセットをWORK領域に作成する場合
3GB/200MB=15 となり、格納域が10を超えているため作成できません。全体として10GBの容量があるわけですから、PARTSIZEを調整して300MB以上にする必要があります。作成するデータセットの大きさが毎回同じ場合は、次のように計算して設定することが望ましくなります。

```
作成するデータセットの大きさ / 指定する物理パス数 = PARTSIZE
```

メインフレーム上のシーケンシャルデータを、Windows上のSASシステムで利用する方法を教えてください。

メインフレーム上のデータをWindows版SASシステムで利用するには、SASインフォーマットを使用してデータを変換します。メインフレーム上のデータをPCに転送する際、EBCDICコードからASCIIコードへの変換はせずにバイナリ転送してください。代表的なSASインフォーマットと変換方法を下記に紹介します。

[英数字データ]
メインフレーム上のEBCDICコード体系のデータ（英数字）に関しては、基本的に$EBCDICw.インフォーマットにてASCIIコードに変換が行われます。

例）

```
ABCDE ⟶ input @1 name $EBCDIC5.;
```

[数値データ]
例1）ゾーン10進データ
S370FZDw.d（符号あり）あるいはS370FZDUw.d（符号なし）インフォーマットを使用する場合

符号ありの数値

```
'F1F2F3F4C5'x ⟶ input @1 numvar S370FZD5.2; ⟶ 123.45
'F1F2F3F4D5'x ⟶ input @1 numvar S370FZD5.2; ⟶ 123.45
```

符号なしの数値

```
'F1F2F3F4F5'x ⟶ input @1 numvar S370FZDU5.2; ⟶ 123.45
```

例2) パック10進データ
S370FPDw.d（符号あり）あるいはS370FPDUw.d（符号なし）インフォーマットを使用する場合

符号ありの数値

```
'000012345C'x ⟶ input @1 numvar S370FPD5.2; ⟶ 123.45
'000012345D'x ⟶ input @1 numvar S370FPD5.2; ⟶ 123.45
```

符号なしの数値

```
'000012345F'x ⟶ input @1 numvar S370FPDU5.2; ⟶ 123.45
```

例3) バイナリ整数データ
S370FIBw.d（符号あり）あるいはS370FIBUw.d（符号なし）インフォーマットを使用する場合

符号ありの数値

```
'0000003039'x ⟶ input @1 numvar S370FIB5.2; ⟶ 123.45
'FFFFFFCFC7'x ⟶ input @1 numvar S370FIB5.2; ⟶ 123.45
```

符号なしの数値

```
'0000003039'x ⟶ input @1 numvar S370FIBU5.2 ; ⟶ 123.45
```

[漢字（IBM漢字）、半角カタカナ]
漢字データおよび半角カタカナについては、DBCS機能として提供されているKCVT関数を使用して目的のコード体系の漢字/カタカナに変換します。

構文

```
KCVT(text,intype,outtype<options,...>)
```

text ：変換処理を行いたい文字変数の指定
intype ：変換したい文字の漢字コード体系
outtype ：変換後の漢字コード体系
options ：必要に応じて指定

例1）漢字データにシフト文字が含まれている場合
下記の内容で、1桁目から6バイトの長さで'漢字'というデータが含まれているファイル(IBM漢字)をシフトJIS漢字コードに変換します。シフトコードが削除されるため、変換後の漢字を出力する変数は2バイト短くなり、4バイト長となります。

```
ファイルの漢字部分（プログラムのkanji1.bin）
'0E4F5848F20F'（IBM漢字コード、0E・0Fがシフトコード）
```

プログラム

```
data _null_ ;
    infile  'c:¥work¥kanji1.bin'  ;
   input  @1  knj  $char6. ;
   length knj2  $4 ;
    knj2  =  kcvt(knj,'ibm','sjis') ;
   put knj2 ;
   put knj2 hex10. ;
run ;
```

ログ

```
漢字       (put knj2)
8ABF8E9A  (put knj2 hex10.)
```

例2) 漢字データにシフト文字が含まれていない場合
下記の内容で、1桁目より4バイトの長さで'漢字'というデータが含まれているファイル(IBM漢字)を'シフトJIS漢字コード'に変換します。NOSHIFTオプションを追加すれば、シフトコードなしの漢字も変換できます。

```
ファイルの漢字部分（プログラムのkanji2.bin)
'4F5848F2'（IBM漢字コード)
```

プログラム

```
data _null_ ;
    infile  'c:¥work¥kanji2.bin'  ;
   input  @1  knj  $char4. ;
   length knj2  $4 ;
    knj2  =  kcvt(knj,'ibm','sjis','noshift') ;
   put knj2 ;
   put knj2 hex10. ;
run ;
```

ログ

```
漢字       (put knj2)
8ABF8E9A  (put knj2 hex10.)
```

例3) 半角カタカナの場合
下記の内容で、1桁目より4バイトの長さで'カタカナ'というデータが含まれているファイル(EBCDIK)をシフトJISコードに変換します。半角カタカナには、KANAオプションを追加します。

```
ファイルの半角カタカナ部分（プログラムのkana.bin)
'86918696'（EBCDIK)
```

プログラム

```
data _null_ ;
    infile  'c:¥work¥kana.bin' ;
   input  @1  kana  $char4. ;
   length kana2  $4 ;
    kana2  =  kcvt(kana,'ibm','sjis','kana') ;
   put kana2 ;
   put kana2 hex10. ;
run ;
```

ログ

```
カタカナ   (put kana2)
B6C0B6C5  (put kana2 hex10.)
```

Q SAS/INSIGHTソフトウェアで分類ごとに棒グラフやモザイク図を出力すると、'その他'として扱われてしまうグループがあります。'その他'ではなく、全グループを対象に出力するにはどのようにすればよいのでしょうか。

A **SAS/INSIGHTで棒グラフやモザイク図を作成すると、デフォルトで全体の割合から4%未満のグループが'その他'として扱われます。グラフ作成の閾値を小さくするには、下記の方法で変更します。**

[変更方法]
棒グラフの場合
1. **プルダウンメニューの[解析]から[ヒストグラム/棒グラフ]を選択します。**
2. **[ヒストグラム/棒グラフ]のダイアログボックスが開きます。棒グラフを作成したい変数を[Y]に選択し、[手法]ボタンをクリックします。**
3. **[手法]のダイアログウィンドウが開きます。[その他,閾値]の値を0%として下さい。[了解]を選択し、手法のダイアログウィンドウを閉じます。**
4. **[ヒストグラム/棒グラフ]のダイアログボックスの[了解]を選択します。**

Q DISCRIMプロシジャで2群の判別分析を実行すると、判別関数が2つ出力されます。一般的に知られる判別関数はどのように求められますか。

A **DISCRIMプロシジャで2群の判別分析を実行した場合、判別関数は下記のように2つ出力されます。**

DISCRIMプロシジャ実行時のアウトプット

```
Discriminant Analysis Linear Discriminant Function
                Y1      Y2
  CONSTANT   -13.9   -10.1
        X1    0.12    0.14
        X2    0.03    0.06
        X3    0.05    0.02
```

判別関数

```
Y1=-13.9+0.12*x1+0.03*x2+0.05*x3  （Y1に対する判別関数）
Y2=-10.1+0.14*x1+0.06*x2+0.02*x3  （Y2に対する判別関数）
```

2つの判別関数にデータを当てはめ、Y1＞Y2ならY1に、Y2＞Y1 ならY2に判別します。例えば、x1=100 x2=22 x3=44 の場合、上記の判別関数ではY1=0.96 Y2=6.1となり、Y2＞Y1よりY2に判別します。また、一般的な判別関数（Y>0のときは1群、Y<0のときは2群に判別される判別関数）を求めるには、2つの判別関数を引き算します。下記の判別式を利用すると、Y>0のときはY1，Y<0のときはY2に判別されます。

```
   Y1=-13.9+0.12*x1+0.03*x2+0.05*x3
-) Y2=-10.1+0.14*x1+0.06*x2+0.02*x3
    Y= -3.9-0.02*x1-0.03*x2+0.03*x3
```

MIXEDプロシジャによって反復測定データを解析した時に、REPEATEDステートメントで誤差にTYPE=CS（Coumpound Symmetry）を指定した時の結果と、RANDOMステートメントで被験者の変量効果を指定した時の結果が異なる時があります。2モデルの分散共分散構造は同じになると予想されますが、どうして異なった結果になるのでしょうか。

サンプルデータ

```
data data1;
  do group=1 to 2;
    do id=1 to 4;
      do time=1 to 4;
        input resp@@;
        output;
      end;
    end;
  end;
cards;
11.1 12.5 14.1  18.5
10.4 12.4 10.2  19.0
11.2 14.1 10.9  12.5
10.4 10.7 21.2  14.5
11.2 15.1 14.4  18.1
10.4 12.3 11.1  14.8
11.2 14.2 17.9  12.0
12.4 10.3 22.1  14.2
;
run;
```

サンプルプログラム1

```
proc mixed data=data1;
  class id group time;
  model resp=group time group*time;
  repeated time /subject=id(group) type=cs;
run;
```

サンプルプログラム1の結果

```
          Tests of Fixed Effects
Source       NDF    DDF   Type III  F  Pr > F
GROUP          1      6       0.42     0.5408
TIME           3     18       3.50     0.0370
GROUP*TIME     3     18       0.43     0.7362
```

サンプルプログラム2

```
proc mixed data=data1;
  class id group time;
  model resp=group time group*time;
  random id(group);
run;
```

サンプルプログラム2の結果

```
          Tests of Fixed Effects
Source       NDF    DDF   Type III  F  Pr > F
GROUP          1      6       0.22     0.6521
TIME           3     18       4.04     0.0232
GROUP*TIME     3     18       0.49     0.6917
```

REPEATEDステートメントにおいてTYPE=CSを指定した時、誤差の共分散は負になることが許されています。一方、RANDOMステートメントを用いた時、デフォルトでは変量効果の分散推定値は0以上の値をとる制約が課せられています。そのため、変量効果の分散が負に推定されるようなデータの時には、両者の結果には違いが生じます。なお、PARMSステートメントにNOBOUNDオプションを指定すると、分散推定値に対する負の推定値を許すことになりますが、0以上の制約はなくなります。

例）

```
proc mixed data=data1;
  parms /nobound;
   class id group time;
   model resp=group time group*time;
  random id(group);
run;
```

New Publications
新刊マニュアルのご紹介

SASインスティチュートジャパンおよび米国SAS Instituteより発売された新刊マニュアルをご紹介します。なお、こちらに記載する価格はすべて税抜きです。ご購入については、弊社「マニュアル販売係」までお問い合わせください。
TEL　：03-3533-3835
FAX　：03-3533-3781
E-mail：booksale@jpn.sas.com

はじめようSASシステム - 実習データFD付き -
●注文番号：10081（日本語版）
●価格　　：5,000円
本書は、SASシステムをはじめてお使いになるユーザを対象として書かれています。SASシステムのDMS (Display Manager System) を利用して、対話的に処理を進める方法を中心に解説しています。本書ではプラットフォームとして、Windows版およびUNIX版SASシステムリリース6.12を使用して書かれていますが、SASシステムの基本的な操作方法やプログラムは、すべてのプラットフォームで共通です。したがって、本書に掲載されているウィンドウの形状を、お使いのプラットフォームに置き換えていただければ、どのプラットフォームをお使いの方でも本書の内容を実行できます。本書で使用するデータもあわせてFDにて提供しております（マニュアルに添付）ので、簡単に自習できます。

Applied Multivariate Statistics with SAS Software, Second Edition
●注文番号：56903（英語版）
●価格　　：8,800円
本書は統計手法、データ解析やアプリケーションをまとめる独特なアプローチで記載されています。今回大幅に改定され、有効なMIXED効果モデルの新情報、MIXEDプロシジャのアプリケーション、IMLプロシジャに対応する回帰診断、共分散構造についても記載されています。多数のSASプログラムとそれに対応する出力結果には例題が掲載されており、いろいろなSASプロシジャの説明が含まれています。理論・実務の両面から実験データや反復測定データの多変量解析に関するトピックやバイプロットを含むデータのグラフィックの表現や多変量回帰が含まれています。さらに付録に多変量データの特別なリファレンスとして、IMLプロシジャの簡単な概要が記載されています。

PROC TABULATE by Example
●注文番号：56514（英語版）
●価格　　：7,700円
本書は主として初級～中級のSASユーザー向けに、TABULATEプロシジャで表を作成する方法について、ステップ・バイ・ステップ形式で説明しています。また、上級ユーザーにも役立つ多くのヒントやテクニックが記載されています。本書には300以上のサンプルプログラムと出力結果が紹介されており、表作成の基礎からカスタマイズされた複雑な出力の構築まで学ぶことができます。このマニュアルを通して、マクロを用いて表を作成する方法、SAS/ASSISTソフトウェアを用いて表を作成する方法、さらにTABULATEプロシジャの出力を他のアプリケーションに書き出す方法も学ぶことができます。

SAS Training
SASトレーニングのお知らせ

臨時トレーニング開催！
回帰分析2コース
～SASによる回帰分析7.8.9章の解説～
3月30日(木) 10：30～17：00
¥40,000（テキスト持参の方は¥36,000）

回帰分析コース受講済、単回帰分析・重回帰分析の知識がある方を対象に、回帰分析2コースを開催します。
・説明変数に質的変数を含む回帰分析
・非線型回帰分析
・重みつき回帰分析を中心に東大出版会発行、「SASによる回帰分析」の7～9章を解説します。
開催会場は追ってご連絡します。

お問合せ先：トレーニング担当
- TEL　03-3533-3835
- FAX　03-3533-3781
- E-mail　training@jpn.sas.com
- URL　http://www.sas.com/japan/

SAS Technical News Spring 2000

発行
株式会社SASインスティチュート ジャパン

本　社
〒104-0054 東京都中央区勝どき1-13-1 イヌイビル・カチドキ 8F
TEL: 03-3533-6921 FAX: 03-3533-6927

大阪支店
〒530-0004 大阪市北区堂島浜1-4-16 アクア堂島西館 12F
TEL: 06-6345-5700　FAX: 06-6345-5655

九州営業所
〒802-0001 北九州市小倉北区浅野2-14-1 小倉興産KMMビル3F
TEL: 093-512-5014　FAX: 093-512-5016

URL http://www.sas.com/japan/

テクニカルニュースに関するお問い合わせ先
テクニカルサポートグループ
TEL:03-3533-3877
FAX:03-3533-3781
E-mail:technews@jpn.sas.com