

SASの“あるある”をスマートに解決

実務に効く3つのマクロをパッケージ化

2025/9/25

SAS ユーザー総会 2025

鈴木 慎吾、西井 雅之、毎田 豊



要旨

- SAS Institute Japanには、お客様の業務課題に対する解決策を検討・実現する部門としてコンサルティングサービス事業本部・カスタマーアドバイザー事業本部、SASユーザー様のご利用促進を支援する部門としてカスタマーサクセス事業本部がございます。
- これら部門の有志メンバーが、これまでSASプログラムを用いた実案件において、頻繁に遭遇してきた“SAS固有の煩わしさ”とその解決法について議論し、今後の案件推進の加速、成果物の均質化に役立つと思われるマクロのパッケージ化を進めており、本日はその“煩わしさ”を解決するための関数を提供するパッケージを御紹介致します。
- なお本資料に記載された取り組みは、発表者メンバーの見解であり、所属する会社の公式な見解を示すものではありません。（全員、業務時間外にて手弁当で作りました。またお客様案件との兼ね合いもあり、10以上の案から3つに絞り作成しました。）

パッケージ紹介

SASの“あるある”をスマートに解決にする“SASToolbox”
パッケージ

SAS Toolbox パッケージ

<https://github.com/Pharmaforest/SASToolbox>



日々の業務を加速する便利 sas マクロ関数を提供します

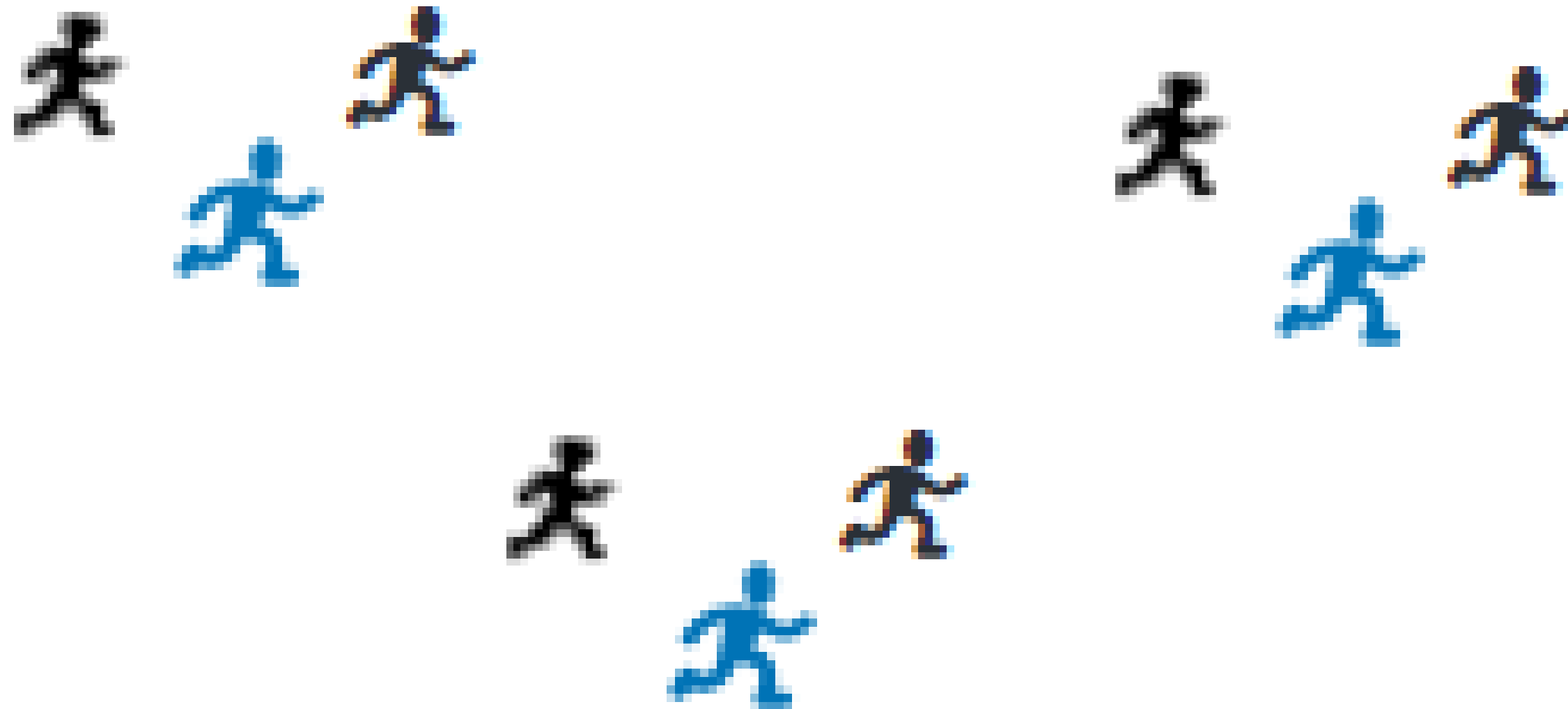
- データセットのエンコードを一括変更！
 - `%trancd2u8_d`
- フォルダ内のファイルを一括データセット化！
 - `%Read_csv_in_folder`
 - `%Read_excel_in_folder`
- フォルダ内の .sas ファイルを一括インクルード！
 - `%includeAll`

trancd2u8_dマクロ

データセットのエンコードをUTF8に一括変更

こんな時の利用を想定しています

- 日々海外やDBからの様々なエンコーディングのデータが集まってくるけど、利用時にはUTF-8に統一したい
- UTF-8に変換するとファイルサイズが大きくなってしまふのでどの程度変わったのか一覧で確認したい
- 無駄にlengthが長くなってしまった変数は最適な長さに調整したい・・・テーブル定義と実際のデータ長を比較したい



trancd2u8_d マクロ

指定ディレクトリ内の .sas7bdat ファイルを別に指定のディレクトリ配下にUTF8にエンコードを変換してコピーする。
コピー・変換したデータセットはcsvファイルに一覧出力

構文

```
%trancd2u8_d (directory_path ----- 対象ディレクトリ  
  <, transcode_dir_path => ----- 変換データセット保存先ディレクトリ  
    <, compress_yn => ----- Compressを使用するか否か  
    <, contents_yn => ----- 指定ディレクトリのみか、配下ディレクトリも含むか  
    <, length_obs =>) ----- テーブル情報にデータの最大lengthを含めるか
```

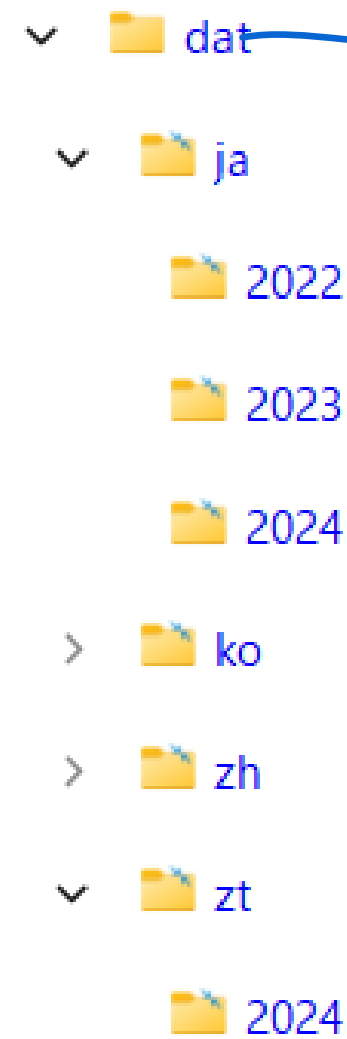
trancd2u8_d マクロ

引数

引数	内容	引数指定例	デフォルト値	必須
directory_path	エンコード変換対象の.sas7bdat ファイルを保存しているディレクトリのフルパス	D:¥SAS19875A¥Meta-Analysis¥SASDATA	-	✓
transcode_dir_path	変換した.sas7bdatファイルを保存するトップディレクトリのフルパス	D:¥SAS19875A¥Meta-Analysis¥SASDATA_u8	directory_path え指定されたフォルダ名の最後に「_u8」を付加	
compress_yn	コピーするSASデータセットにcompressを適用するかどうか	y または n を指定	y	
contents_yn	変換したデータセット情報をCSVに出力する場合はそのファイル名 n を指定した場合はデータセットの作成のみ	ファイル名 または n を指定	n	
length_obs	データの最大lengthリストの取得時に読み込む行数 nを指定した場合はデータの最大lengthを取得しない MAX を指定した場合はデータのすべてのオブザベーションから最大lengthを求める	n 、 数値 または MAX を指定	n	

trancd2u8_d マクロ

動作イメージ



directory_path で指定したフォルダ

transcode_dir_path で指定したフォルダ
(変換対象のデータセット情報csvもここに出力)

フォルダ階層構造をコピーし、
含まれる.sas7bdatファイルの
文字コードをUTF-8に変換



trancd2u8_d マクロ

変換後テーブル情報に出力される内容

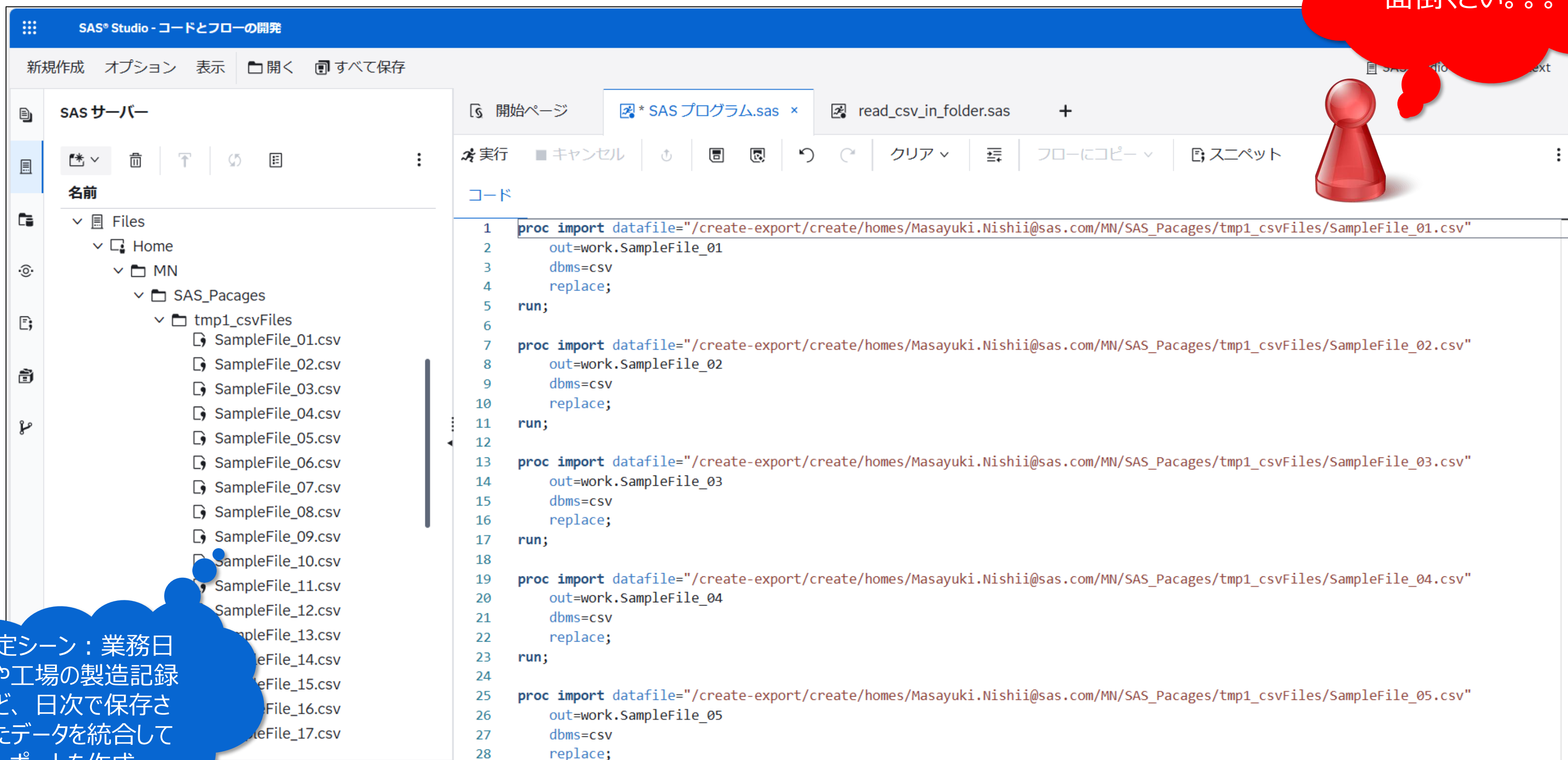
オリジナルディレクトリパス	変数名
変換後のディレクトリパス	変数タイプ
変換前ファイルサイズ	変数長
変換後ファイルサイズ	変数番号
作成したホスト	変数ラベル
エンジン	フォーマット
テーブル名	インフォーマット
テーブルラベル	変換前変数長
テーブル行数	データの最大長(文字変数のみ)
作成日	

Read_csv_in_folder/Read_excel_in_folder マクロ

フォルダ内のcsv/Excelファイルを一括でデータセット化

こんな不満ありませんか？

一個一個ファイルを
インポートするのは
面倒くさい。。。。



The screenshot shows the SAS Studio interface. On the left, a file explorer displays a directory structure: SASサーバー > Files > Home > MN > SAS_Packages > tmp1_csvFiles. This folder contains 17 CSV files named SampleFile_01.csv through SampleFile_17.csv. The main editor window shows a SAS program with five repetitive PROC IMPORT statements, each importing a different CSV file from the tmp1_csvFiles directory into a corresponding work dataset (e.g., SampleFile_01, SampleFile_02, etc.). The code is as follows:

```
1 proc import datafile="/create-export/create/homes/Masayuki.Nishii@sas.com/MN/SAS_Packages/tmp1_csvFiles/SampleFile_01.csv"
2   out=work.SampleFile_01
3   dbms=csv
4   replace;
5 run;
6
7 proc import datafile="/create-export/create/homes/Masayuki.Nishii@sas.com/MN/SAS_Packages/tmp1_csvFiles/SampleFile_02.csv"
8   out=work.SampleFile_02
9   dbms=csv
10  replace;
11 run;
12
13 proc import datafile="/create-export/create/homes/Masayuki.Nishii@sas.com/MN/SAS_Packages/tmp1_csvFiles/SampleFile_03.csv"
14  out=work.SampleFile_03
15  dbms=csv
16  replace;
17 run;
18
19 proc import datafile="/create-export/create/homes/Masayuki.Nishii@sas.com/MN/SAS_Packages/tmp1_csvFiles/SampleFile_04.csv"
20  out=work.SampleFile_04
21  dbms=csv
22  replace;
23 run;
24
25 proc import datafile="/create-export/create/homes/Masayuki.Nishii@sas.com/MN/SAS_Packages/tmp1_csvFiles/SampleFile_05.csv"
26  out=work.SampleFile_05
27  dbms=csv
28  replace;
```

想定シーン：業務日報や工場の製造記録など、日次で保存されたデータを統合してレポートを作成。

%Read_csv_in_folder マクロ

指定ディレクトリ内のすべての csv ファイルを一括で SAS データセットとしてそれぞれインポートします。

構文

`% Read_csv_in_folder(path)` ----- csv を保存したディレクトリのフルパス

%Read_excel_in_folder マクロ

指定ディレクトリ内のすべての excel ファイルを一括で SAS データセットとしてそれぞれインポートします。

構文

`% Read_excel_in_folder(path)` ----- excel を保存したディレクトリのフルパス

%Read_csv_in_folder 関数

引数

引数	内容	引数指定例	デフォルト値	必須
path	csv を保存しているフォルダのフルパス (中身は csvファイルだけである必要があります)	例： /example/homes/SampleUser/SASPac/tmp1_csvFiles	-	✓

%Read_excel_in_folder 関数

引数

引数	内容	引数指定例	デフォルト値	必須
path	excel を保存しているフォルダのフルパス (中身は excel ファイルだけである必要があります)	例： /example/homes/SampleUser/SASPac/tmp2_excelFiles	-	✓

実行例 (csvファイル)

The screenshot shows the SAS Studio interface. On the left, the file explorer shows a directory structure: SAS サーバー > Files > Home > MN > SAS_Packages > tmp1_csvFiles. The code editor shows the following code:

```
1  
2 %Read_csv_in_folder(path=  
3 "/create-export/create/homes/Masayuki.Nishii@sas.com/MN/SAS_Packages/tmp1_csvFiles")  
4
```

A blue callout box points to the code with the text: "一括でインポート可能 (path= は省略可)".

The Results pane shows a list of data sets: SAMPLEFILE_01 through SAMPLEFILE_06, all with library WORK. A blue callout box points to the first data set with the text: "csvファイルと同じ名前のSASデータセットが work に保存されます。(32文字を超えた場合、そのファイルだけ実行されません。他は実行されます。)".

The output data for WORK.SAMPLEFILE_01 is shown in a table:

#	File	Val1	Val2
1	SampleFile_01	a	10
2	SampleFile_01	b	20
3	SampleFile_01	c	30
4	SampleFile_01	d	40
5	SampleFile_01	e	50
6	SampleFile_01	f	60
7	SampleFile_01	g	70

excelファイルの場合も同様

The screenshot shows the SAS Studio interface. On the left, a file explorer shows a directory structure: SASサーバー > Files > Home > MN > SAS_Pacages > tmp2_excelFiles. This folder contains 17 Excel files named SampleFile_01.xlsx through SampleFile_17.xlsx. The main editor shows a SAS script with the following code:

```
1  
2 %Read_excel_in_folder(path=  
3 "/create-export/create/homes/Masayuki.Nishii@sas.com/MN/SAS_Pacages/tmp2_excelFiles")  
4
```

The code is highlighted with a red box. Below the code, the '出力データ (31)' (Output Data) tab is active, showing a list of data sets: SAMPLEFILE_01 through SAMPLEFILE_06, all with library WORK. The first data set, WORK.SAMPLEFILE_01, is selected, and its data is displayed in a table with 7 rows and 4 columns: #, File, Val1, and Val2.

#	File	Val1	Val2
1	SampleFile_01	a	10
2	SampleFile_01	b	20
3	SampleFile_01	c	30
4	SampleFile_01	d	40
5	SampleFile_01	e	50
6	SampleFile_01	f	60
7	SampleFile_01	g	70

A blue callout box on the right contains the following text:

Excelファイルと同じ名前のSASデータセットが work に保存されます。
(32文字を超えた場合、そのファイルだけ実行されません。他は実行されます。)

留意事項
- エクセルファイルの最初のシートだけ読み込まれます。

IncludeAllマクロ

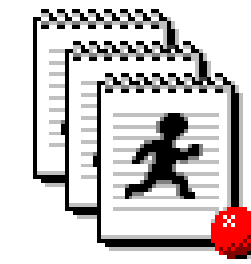
指定ディレクトリ内の .sas ファイルを一括インクルード

こんな不満ありませんか？

.sas ファイルを分割したいが、、、

機能別に .sas ファイル
を分けて、開発を手分
けしたい

機能別に .sas ファ
イルを分けて再利用
性を高めたい



DataHandling.sas
FileHandling.sas
Configuration.sas
ModelManager.sas
...



```
%include "DataHandling.sas";  
%include "FileHandling.sas";  
%include "Configuration.sas";  
%include "ModelManager.sas";  
...  
/* 処理 */  
%main()
```

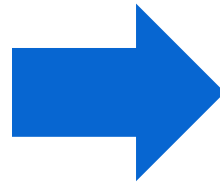
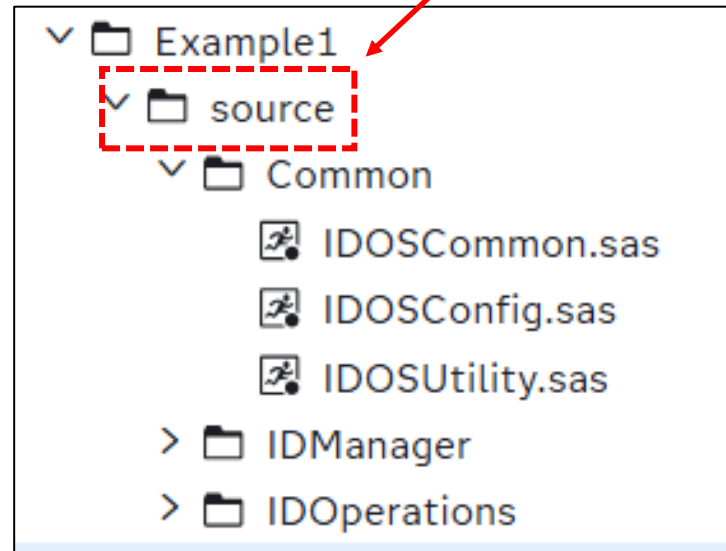
ファイルの変更（追加、
削除、移動、名称変
更）の度にインクルード
コードを修正...

実行例1

シンプルに解決！

注1) デフォルトで、サブディレクトリもインクルード対象です
注2) 拡張子部分 (.sas) は cas-insensitive です
(.SAS も .Sas もインクルード対象)。

`%IncludeAll (/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source)`



```
実行  キャンセル  コード  ログ  結果  出力データ
(0) (0) (0)
includeing mode          : RELEASE
Excluding Dirname filter regex : <NONE>
Excluding Filename filter regex: <NONE>
29 .sas file(s) found in the directory.
[1/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDCommon.sas...Included(19 lines)
[2/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDConfig.sas...Included(19 lines)
[3/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDUtility.sas...Included(19 lines)
[4/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDManager.sas...Included(19 lines)
[5/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations.sas...Included(19 lines)
[6/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDModule.sas...Included(470 lines)
[7/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDModuleExporter.sas...Included(337 lines)
[8/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDModuleImporter.sas...Included(277 lines)
[9/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDObjectViewer.sas...Included(139 lines)
[10/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDPublishedCodeGenerator.sas...Included(412 lines)
[11/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDTransferJobCleaner.sas...Included(150 lines)
[12/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDVersionFixer.sas...Included(89 lines)
[13/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDVersionFixer.sas...Included(434 lines)
[14/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDVersionFixer.sas...Included(80 lines)
[15/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDVersionFixer.sas...Included(302 lines)
[16/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDVersionFixer.sas...Included(430 lines)
[17/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDVersionFixer.sas...Included(136 lines)
[18/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDVersionFixer.sas...Included(81 lines)
[19/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDVersionFixer.sas...Included(433 lines)
[20/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDVersionFixer.sas...Included(233 lines)
[21/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDVersionFixer.sas...Included(117 lines)
[22/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDVersionFixer.sas...Included(138 lines)
[23/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDVersionFixer.sas...Included(720 lines)
[24/29] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example1/source/IDOperations/IDVersionFixer.sas...Included(720 lines)
29 .sas file(s) included.
7873
7874
7875
```



%IncludeAll マクロ

指定ディレクトリ内の .sas ファイルを一括インクルードします

構文

```
%IncludeAll(directory_path ----- インクルード対象ディレクトリ
  <, i_is_recursive => ----- 再帰インクルード読込フラグ
  <, i_exc_dirname_regex => ----- 除外ディレクトリフィルタ (正規表現)
  <, i_exd_filename_regex => ----- 除外ファイルフィルタ (正規表現)
  <, i_leading_files => ----- 優先インクルードファイル
  <, i_trailing_files => ----- 劣後インクルードファイル
  <, i_is_verbose => ----- 冗長ログフラグ
  <, i_is_debug_mode =>) ----- デバッグモードフラグ
```

%IncludeAll関数

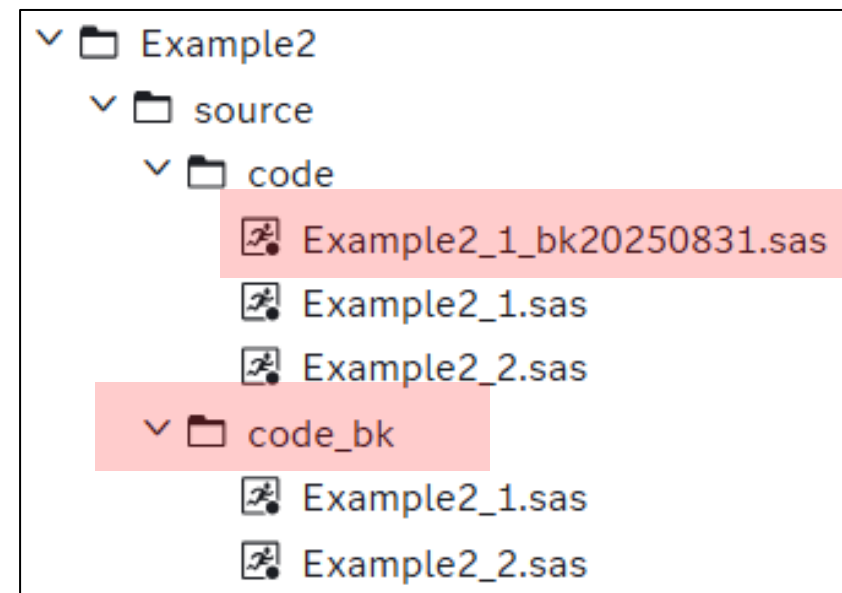
引数

引数	内容	引数指定例	デフォルト値	必須
directory_path	インクルード対象の.sas ファイルを保持しているディレクトリのフルパス	例：/tmp/programs下の .sasファイルをインクルードする /tmp/programs	-	✓
i_is_recursive	再帰インクルードフラグ '1'とすると、サブディレクトリ内の .sasもインクルードします	例：サブディレクトリ内のファイルもインクルードする i_is_recursive = 1	1	
i_exc_dirname_regex	除外ディレクトリフィルタ（正規表現） この正規表現にマッチするディレクトリは除外されます	例：“_bk” で終わるディレクトリを除外する i_exc_dirname_regex = /_bk\$/		
i_exc_filename_regex	除外ファイル名フィルタ（正規表現） ファイル名のうち、拡張子（“.sas”）を覗いた部分の文字列が正規表現にマッチするものは除外されます	例：8桁の数字で終わるファイルを除外する i_exc_filename_regex = /%d{8}\$/		
i_leading_files	優先インクルードファイル 読み込み対象のファイルのうち、優先的にインクルードするファイルを指定します。 ' ' で区切ることで、複数指定可能です。	例：/tmp/programs/common.sas を最初にインクルードする i_leading_files = /tmp/programs/common.sas		
i_trailing_files	劣後インクルードファイル 読み込み対象のファイルのうち、劣後的にインクルードするファイルを指定します。 ' ' で区切ることで、複数指定可能です。	例：/tmp/programs/x.sas、y.sas、z.sasを最後にインクルードする i_trailing_files = /tmp/programs/file_x.sas /tmp/programs/file_y.sas /tmp/programs/file_z.sas		
i_including_mode	読込モード “RELEASE” または “DEBUG”を指定します 👉 詳細は「例4」	例：デバッグモードでインクルードする i_including_mode = DEBUG	RELEASE	
i_is_verbose	冗長ログ出力フラグ '1'とすると、冗長ログを出力します。	例：冗長ログを出力する i_is_verbose = 1	0	

すると、こんな心配が生まれますね？

バックアップの残骸

このファイルもこの
ディレクトリも念のため
バックアップを取っ
ておこう！



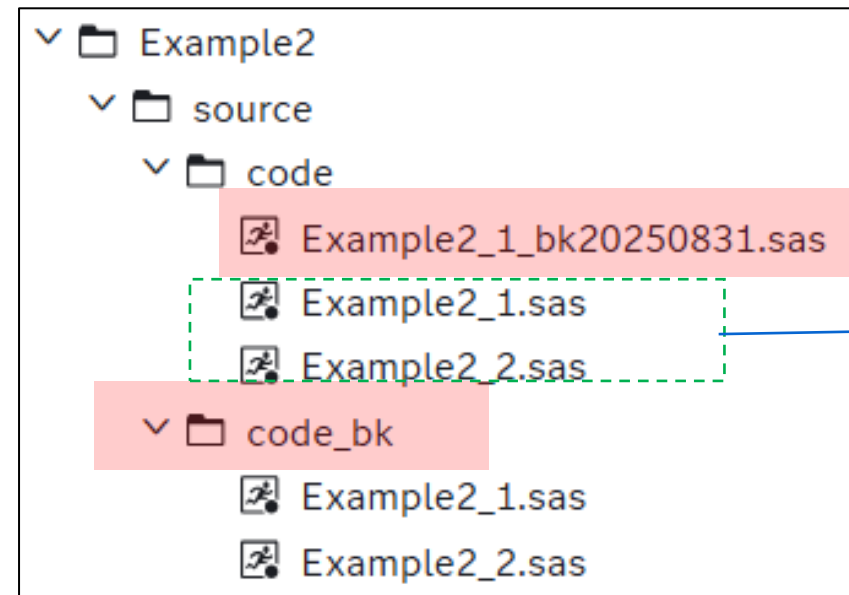
バックアップファイルまで
インクルードされてしま
んですが、、

```
%IncludeAll(/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example2/source)
```

```
Including all .sas files in "/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example2/source" and sub-directories
Includeing Mode          : RELEASE
Excluding Dirname filter regex : <NONE>
Excluding Filename filter regex: <NONE>
5 .sas file(s) found in the directory.
[1/5] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example2/source/code/Example2_1.sas...Included(2 lines)
[2/5] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example2/source/code/Example2_1_bk20250831.sas...Included(2 lines)
[3/5] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example2/source/code/Example2_2.sas...Included(2 lines)
[4/5] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example2/source/code_bk/Example2_1.sas...Included(2 lines)
[5/5] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example2/source/code_bk/Example2_2.sas...Included(2 lines)
5 .sas file(s) included.
```

実行例2

スマートに解決！～フィルタリング～



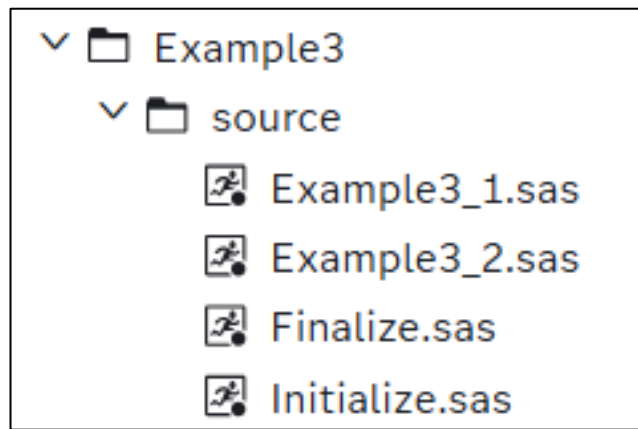
この2ファイルがインクルード対象

```
%IncludeAll(/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example2/source  
  , i_exc_dirname_regex = /_bk/  
  , i_exc_filename_regex = /%d{8}$/)
```

```
Including all .sas files in "/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example2/source" and sub-directories  
Including Mode : RELEASE  
Excluding Dirname filter regex : /_bk/  
Excluding Filename filter regex: /\d{8}$/  
2 .sas file(s) found in the directory.  
[1/2] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example2/source/code/Example2_1.sas...Included(2 lines)  
[2/2] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example2/source/code/Example2_2.sas...Included(2 lines)  
2 .sas file(s) included.
```

こんな状況も考えられますよね？

インクルード時に即時実行



順序依存あり

- グローバル設定
- 即時実行

Initialize.sas

```
%global g_authors;
```

Example3_1.sas

```
%let g_authors = &g_authors. Shingo;
```

Example3_2.sas

```
%let g_authors = &g_authors. Suzuki
```

Finalize.sas

```
%put Authors: &g_authors.;
```

Initialize.sas では
グローバル設定

Finalize.sas で読
み込み状態を確認
しよう

ファイルの名前順に読
み込まれるので、順序
依存があるとマズいの
ですが、

```
%IncludeAll(/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example3/source)
```

読み込み順は ファイルのフルパスの昇順

```
4 .sas file(s) found in the directory.
```

```
WARNING: G_AUTHORSのシンボリック参照を解決できません。
```

```
ERROR: テキスト式&G_AUTHORS. SHINGOはマクロ変数G_AUTHORSへの再帰的参照を含んでいます。マクロ変数にnull値が割り当てられます。
```

```
[1/4] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example3/source/Example3_1.sas...Included(3 lines)
```

実行例3（続き）

こんな仕組みで解決！ ～優先/劣後インクルード～

```
%IncludeAllIn(/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example3/source
, i_leading_files = /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example3/source/Initialize.sas
, i_trailing_files = /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example3/source/Finalize.sas)
```

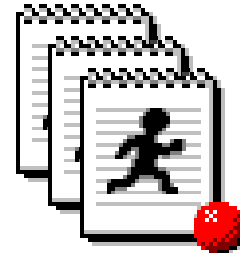
```
4 .sas file(s) found in the directory.
[1/4] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example3/source/Initialize.sas...Included(1 lines)
[2/4] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example3/source/Example3_1.sas...Included(3 lines)
[3/4] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example3/source/Example3_2.sas...Included(3 lines)
Authors: Shingo Suzuki
[4/4] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example3/source/Finalize.sas...Included(1 lines)
4 .sas file(s) included.
```

こんな経験ありませんか？

マクロの重複定義



```
%include "SASFile1.sas";  
%include "SASFile2.sas";  
%include "SASFile3.sas";  
  
%TestMacro1()
```



SASFile1.sas
SASFile2.sas
SASFile3.sas

SASFile1.sas

```
%macro TestMacro1();  
    %put Hello SAS;  
%mend;
```

SASFile1.sasのマクロ
"TestMacro1"を修正
しました！

```
%macro TestMacro1();  
    %put Good by SAS;  
%mend;
```

おかしい、いくら
直しても挙動変
が変わらない…

ハッ！もしかして…
他のファイルにも同名
マクロがいるのでは？

実行例4

こうすれば解決！ ～冗長ログ～

```
%IncludeAll(/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/Example4  
            , i_is_verbose = 1)
```

```
Including all .sas files in "/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/Example4" and sub-directories  
Including Mode           : RELEASE  
Excluding Dirname filter regex : <NONE>  
Excluding Filename filter regex: <NONE>  
3 .sas file(s) found in the directory.  
[1/3] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Example4/Example5_3.sas  
[Macro(s) defined in "/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Example4/Example5_3.sas"]  
    SAMPLEMACRO1  
    SAMPLEMACRO2  
[1/3] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Example4/Example5_2B.sas  
[2/3] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Example4/Example5_2A.sas  
[Macro(s) defined in "/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Example4/Example5_2A.sas"]  
    EXAMPLE5_2A  
    EXAMPLE5_2B  
[2/3] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Example4/Example5_3.sas  
[3/3] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Example4/Example5_3.sas  
[Macro(s) defined in "/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Example4/Example5_3.sas"]  
    EXAMPLE5_3  
    (*) SAMPLEMACRO1  
[3/3] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Example4/Example5_3.sas  
3 .sas file(s) included.
```

'(*)' はマクロの重複定義を示す

こんな要望ありませんか？

デバッグ/リリースモード

開発中

バグの除去のために追跡コードを埋め込みたい
(あとでコメントアウトすりゃいい)

確認のため一時テーブルを消さずにおきたい
(あとでコメントアウトすりゃいい)

コメントアウトするよりもデバッグフラグを用意するか？

リリース時

リリース時にもれなくデバッグコードを除去できる？

デバッグモード/リリースモードコンパイルという概念があればなあ...

%if だらけでパフォーマンスに影響あるんじゃないの？

実行例5

ついに解決！ ～ デバッグ/リリースモード ～

サンプルコード

```
%macro Example4();  
  %put This is DEBUG Message; /* #DEBUG# */  
  %put This is RELEASE Message; /* #RELEASE# */  
  /* #DEBUG- */  
  data WORK.example4;  
    set SASHELP.class;  
run;  
quit;  
/* -DEBUG# */  
/* #RELEASE- */  
data WORK.example4;  
  set SASHELP.cars;  
run;  
quit;  
/* -RELEASE# */  
%mend Example4;
```

リリースモード
(通常)

デバッグモード

```
%macro Example4();  
  /*%put This is DEBUG Message; /* #DEBUG# */  
  %put This is RELEASE Message; /* #RELEASE# */  
  /* #DEBUG- */  
  /*data WORK.example4;  
  /*   set SASHELP.class;  
  /*run;  
  /*quit;  
  /* -DEBUG# */  
  /* #RELEASE- */  
  data WORK.example4;  
    set SASHELP.cars;  
run;  
quit;  
/* -RELEASE# */  
%mend Example4;
```

```
%macro Example4();  
  %put This is DEBUG Message; /* #DEBUG# */  
  %put This is RELEASE Message; /* #RELEASE# */  
  /* #DEBUG- */  
  data WORK.example4;  
    set SASHELP.class;  
run;  
quit;  
/* -DEBUG# */  
/* #RELEASE- */  
/*data WORK.example4;  
/*   set SASHELP.cars;  
/*run;  
/*quit;  
/* -RELEASE# */  
%mend Example4;
```

実行例5 (続き)

ついに解決！ ～ デバッグ/リリースモード ～

サンプルコード (RELEASEモード)

```
%IncludeAll(/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example4/source)  
%Example4()
```

i_is_debug_mode 無指定の場合、RELEASEモード

```
80 %IncludeAllIn(/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example4/source)  
Including all .sas files in "/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example4/source" and sub-directories  
Including Mode : RELEASE  
Filename filter regex: <NONE>  
NOTE: データセットWORK._TEMP_OPTIONS_DS (memtype=DATA) 作成  
NOTE: PROCEDURE OPTSAVE処理(合計処理時間):  
処理時間 0.00 秒  
CPU時間 0.00 秒  
  
1 .sas file(s) found in the directory  
[1/1] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example4/source  
1 .sas file(s) included.  
NOTE: PROCEDURE OPTLOAD処理(合計処理時間):  
処理時間 0.00 秒  
CPU時間 0.00 秒  
  
NOTE: WORK._TEMP_OPTIONS_DS (memtype=DATA) 解放  
NOTE: PROCEDURE DELETE処理(合計処理時間):  
処理時間 0.00 秒  
CPU時間 0.00 秒  
  
102 %Example4()  
SASPackage "IncludeAllIn" Test (Example4)  
This is RELEASE Message
```

PROCEDURE DELETE
処理時間
CPU時間

102 %Example4()
SASPackage "IncludeAllIn" Test (Example4)
This is RELEASE Message

WORK.EXAMPLE4

列: 15 行: 428

式の入力

#	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	Engi
1	Acura	MDX	SUV	Asia	All	\$36,945	\$33,337	
2	Acura	RSX Type S 2dr	Sedan	Asia	Front	\$23,820	\$21,761	
3	Acura	TSX 4dr	Sedan	Asia	Front	\$26,990	\$24,647	
4	Acura	TL 4dr	Sedan	Asia	Front	\$33,195	\$30,299	
5	Acura	3.5 RL 4dr	Sedan	Asia	Front	\$43,755	\$39,014	
6	Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	\$46,100	\$41,100	
7	Acura	NSX coupe 2dr manual S	Sports	Asia	Rear	\$89,765	\$79,978	
8	Audi	A8	Sedan	Europe	Front	\$25,000	\$22,000	

実行例5 (続き)

ついに解決！ ～ デバッグ/リリースモード ～

サンプルコード (DEBUGモード)

```
%IncludeAll(/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example4/source  
            , i_is_debug_mode = 1)  
%Example4()
```

```
Including all .sas files in "/innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example4/source" and sub-directories  
Including Mode : DEBUG  
Filename filter regex: <NONE>  
NOTE: データセットWORK._TEMP_OPTIONS_DSは3  
NOTE: PROCEDURE OPTSAVE処理(合計処理時間)  
処理時間 0.00 秒  
CPU時間 0.00 秒  
  
1 .sas file(s) found in the directory  
[1/1] /innovationlab-export/innovationlab/homes/Shingo.Suzuki@sas.com/SASPackage/Examples/Example4/source/Example4  
1 .sas file(s) included.  
NOTE: PROCEDURE OPTLOAD処理(合計処理時間)  
処理時間 0.00 秒  
CPU時間 0.00 秒  
  
NOTE: WORK._TEMP_OPTIONS_DS (memtyp  
NOTE: PROCEDURE DELETE処理(合計処理時間)  
処理時間 0.00 秒  
CPU時間 0.00 秒  
  
107 %Example4()  
SASPackage "IncludeAllIn" Test (Example4)  
This is DEBUG Message
```

PROCEDURE
処理時間
CPU時間

107 %Example4()
SASPackage "IncludeAllIn"
This is DEBUG Message

WORK.EXAMPLE4 ⓘ

列: 5 行: 19

式の入力

#	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83

SAS Toolbox パッケージ

<https://github.com/Pharmaforest/SASToolbox>

- データセットのエンコードを一括変更！
 - `%trancd2u8_d`
- フォルダ内のファイルを一括データセット化！
 - `%Read_csv_in_folder`
 - `%Read_excel_in_folder`
- フォルダ内の `.sas` ファイルを一括インクルード！
 - `%includeAll`



是非、ご活用ください！

Fin.

