

# ODS RTFでの出力環境の整備

## ～RTFスタイルとSASマクロの運用・管理手順～

○宮辻 淳, 麻生 美由紀, 平野 勝也

(株式会社インテリム)

Preparation for ODS RTF programming

-Operation and management procedures for Style template and SAS macros-

Atsushi Miyatsuji, Miyuki Aso, Katsunari Hirano

intellim Corporation

### 要旨

Dynamic Data Exchange を用いて Excel 形式の解析図表を作成する場合、しばしば SAS システムと Excel との間でデータ交換する際に原因不明の I/O エラーが発生する問題がある。Excel を利用せずに SAS システムだけで解析図表を作成する方法として、ODS RTF を利用して Rich Text Format (RTF) 形式の解析図表を作成する方法が、日本では SAS Version 8.2 から利用できるようになった。この方法は、プログラム実行時の安定性が高く、何より SAS 社によりサポートされている機能であり、分からないことをいつでも質問できる体制がある、という点で安心して利用できる。本稿では、RTF 形式の解析図表を作成するために、検討・準備したこととして、RTF 出力時の体裁（フォントの種類と大きさ、四隅の余白、表の位置と罫線と塗りつぶし、等）を制御している Style テンプレートの準備と、効率的に業務を進めるために必要と考えた SAS マクロとその管理方針を紹介する。

キーワード：ODS RTF, RTF 出力, Style テンプレート, TEMPLATE プロシジャ, SAS マクロ

### 1 はじめに

近年、Rich Text Format (RTF) 形式の解析図表の納品をクライアントから要望されることが増えてきている。また、Dynamic Data Exchange (DDE) を用いて Excel 形式の解析図表を作成する際に問題（原因不明の I/O エラー、PDF 化時の文字切れ）が発生することがあるため、RTF 形式の解析図表を作成するための環境を整備しようと考えた。

RTF 形式で解析図表を作成する場合、SAS プログラムだけで完結する簡潔さと、プログラム実行時の安定性、SAS 社による Q&A 体制の充実が長所である。RTF 形式の解析図表の作成法に関しては、これまでに沢山のテクニカルな事例が報告されているため、本稿では、RTF 出力時の体裁（フォントの種類と大きさ、四隅の余白、表の位置と罫線と塗りつぶし等）を制御している Style テンプレートの作成・利用法を説明する。

そして、Excel DDE プログラミングと ODS RTF プログラミングの違いを振りかえり、効率的に ODS RTF プログラミングを進める上で必要と考えた SAS マクロと、その管理方針を紹介する。

## 2 Style テンプレート

本章では Style テンプレートの使用経験がない人に向けて、Style テンプレートの役割と Style テンプレートの参照・作成方法を説明して、作成した Style テンプレートの利用法を説明する。

### 2.1 Style テンプレートの役割

ODS RTF ステートメントを使用して RTF ファイルを作成する際に、Style=オプションを指定する場合と指定しない場合のプログラム (Program code 1) と出力結果 (Figure 1) を示す。

Program code 1

<pre>/*Style テンプレートの指定なし*/ ods rtf file="出力先の Path¥OUT1.rtf" ;  proc print data = sashelp.class noobs label ; run ; ods rtf close ;</pre>	<pre>/*Style テンプレートの指定あり*/ ods rtf file="出力先の Path¥OUT1.rtf" style = Styles.roman9 ; proc print data = sashelp.class noobs label ; run ; ods rtf close ;</pre>
---	--

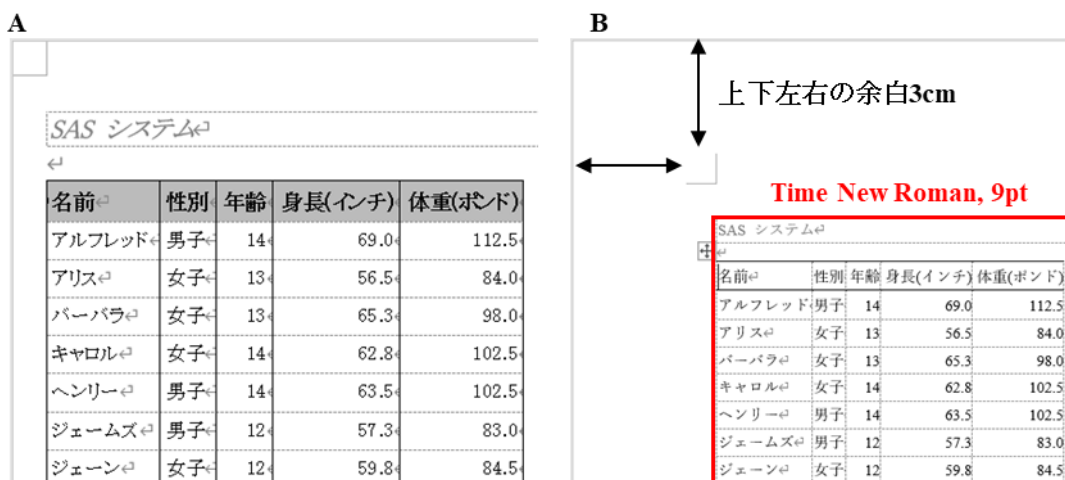


Figure 1. Style=オプションを指定する場合と指定しない場合の出力結果

Style=オプションで Style テンプレートを指定することで、出力の全体的な外観 (セルの塗りつぶし、フォント種とサイズ、余白、罫線など) が変わっていることに気付くと思う。実際、Style テンプレート (Styles.Roman9) では、セルの塗りつぶしをなくし、フォント種とサイズを Times New Roman と 9pt に、上下左右の余白を 3cm に、縦罫線をなくし横罫線を設定している。単純な設定であるが、Style テンプレートを利用することで、より解析図表らしい見た目の RTF ファイルを作成できることが理解できると思う。

Style テンプレートでは、半角/全角文字用のフォントを別々に定義できないため、「MS 明朝」に設定した Style テンプレート (Mincho9) と「Times New Roman」に設定した Style テンプレート (Roman9) を作成して、使い勝手を確認した。当たり前であるが、Style テンプレート (Mincho9) では数値を含むすべての出力が「MS

明朝」になってしまう。半角文字を「Times New Roman」で、全角文字を「MS 明朝」で表示したい場合には、Style テンプレート (Roman9) を利用して、全角文字を含む部分だけを、インラインフォーマットを利用して半角/全角フォントを指定するほうが効率的であったので報告する[1, 2]。

## 2.2 SAS 社が提供する Style テンプレートの参照方法

SAS 社が提供する Style テンプレートは SASHELP ライブラリに保存されているが、SAS データセットやカタログのようにエクスプローラでは参照できない。Style テンプレートを確認するための画面の遷移を Figure 2 に示す。まず、結果ウィンドウをアクティブ (A) な状態にして、「表示」メニューから「テンプレート」を指定 (B) することでテンプレートウィンドウ (C) を表示できる。Style テンプレートは、Sashelp.Tmplmst に保存されているため、それを選択することで主要な Style テンプレートを確認できる。テンプレートウィンドウで Styles 内の Rtf (RTF ファイルを作成する際のデフォルトの Style テンプレート) をダブルクリックすると、TEMPLATE プロシジャで記述された Style テンプレートのソースコードがテンプレートブラウザ (D) に表示される。

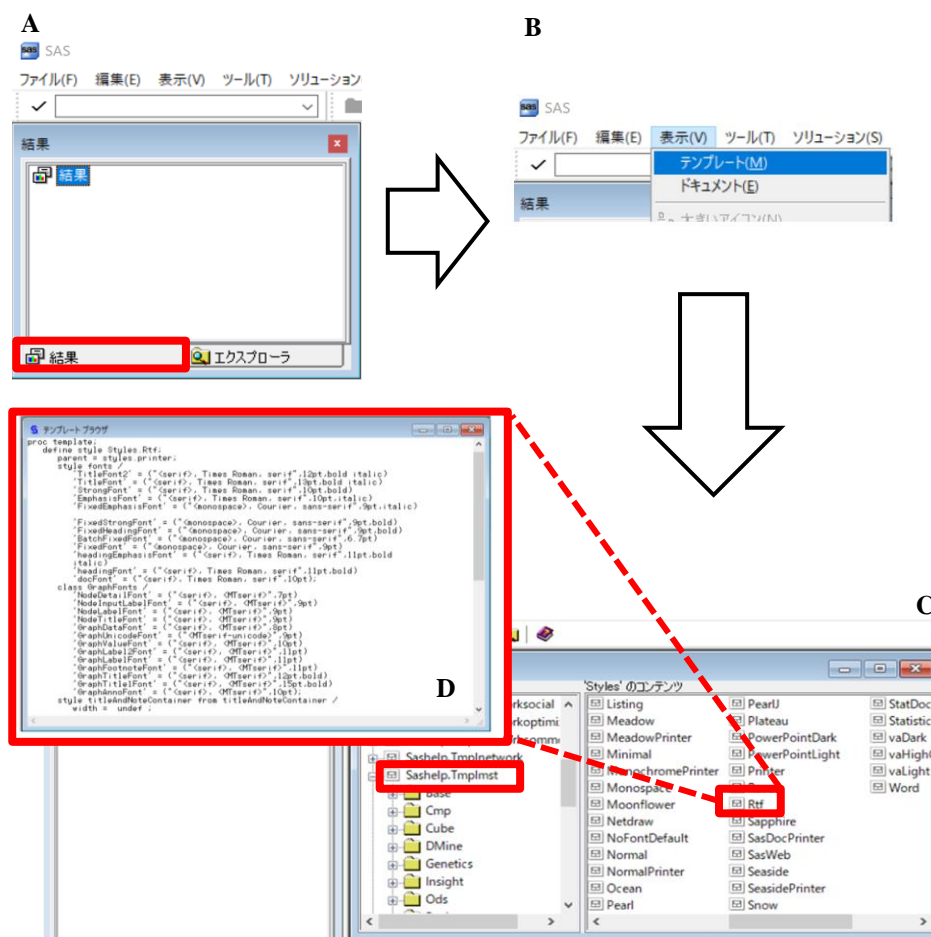


Figure 2. SAS 社が提供する Style テンプレートの参照方法

参照したい Style テンプレート (以下では、Styles.rtf) を特定できている場合には、Program code 2 を実行することで、ログ画面 (Figure 3) にソースコードを表示できる。

Program code 2

```
PROC TEMPLATE ;
    source styles.rtf ;
run ;
```



```
1 proc template ;
NOTE: HTML Bodyファイルの書き込み先: sashtml.htm
2 source styles.rtf ;
define style Styles.Rtf;
parent = styles.printer;
style fonts /
'TitleFont2' = ("<serif>, Times Roman, serif",12pt,bold italic)
'TitleFont' = ("<serif>, Times Roman, serif",13pt,bold italic)
'StrongFont' = ("<serif>, Times Roman, serif",10pt,bold)
'EmphasisFont' = ("<serif>, Times Roman, serif",10pt,italic)
'FixedEmphasisFont' = ("<monospace>, Courier, sans-serif",9pt,italic)
'FixedStrongFont' = ("<monospace>, Courier, sans-serif",9pt,bold)
'FixedHeadingFont' = ("<monospace>, Courier, sans-serif",9pt,bold)
'BatchFixedFont' = ("<monospace>, Courier, sans-serif",6.7pt)
'FixedFont' = ("<monospace>, Courier, sans-serif",9pt)
'headingEmphasisFont' = ("<serif>, Times Roman, serif",11pt,bold italic)
'headingFont' = ("<serif>, Times Roman, serif",11pt,bold)
'docFont' = ("<serif>, Times Roman, serif",10pt);
class GraphFonts /
NodeDetailFont' = ("<serif>, <MTserif>",.7pt)
```

Figure 3. ログ画面

### 2.3 Style テンプレートの作成方法

クライアントの文書規定は様々であり、これらの文書規定に柔軟に対応するためには、最低限の設定を盛り込んだ Style テンプレートを準備しておくことが得策かもしれない。Program code 3 では、Style テンプレートを保存するフォルダを作成した後に、そのフォルダに対して SAS ライブラリ「mystyle」を割り当てている。次に ODS PATH ステートメントで書き込み権を設定したうえで、PROC TEMPLATE プロシジャで Style テンプレート (NewStyle) を作成している。

Program code 3

```
libname mystyle "Styleテンプレートを保存するフォルダ" ;
ODS PATH mystyle.template(WRITE) sashelp.tmplmst(read) ;
PROC TEMPLATE
define style NewStyle ;
parent = Styles.Rtf ;
class fonts /
'TitleFont' = ("Times New Roman",9pt)
'TitleFont2' = ("Times New Roman",9pt)
'StrongFont' = ("Times New Roman",9pt)
'EmphasisFont' = ("Times New Roman",9pt)
'headingEmphasisFont' = ("Times New Roman",9pt)
'headingFont' = ("Times New Roman",9pt)
'footFont' = ("Times New Roman",9pt)
'FixedEmphasisFont' = ("Times New Roman",9pt)
'FixedStrongFont' = ("Times New Roman",9pt)
'FixedHeadingFont' = ("Times New Roman",9pt)
```

```
'BatchFixedFont' = ("Times New Roman",9pt)
'FixedFont' = ("Times New Roman",9pt)
'docFont' = ("Times New Roman",9pt);

end;

run;
```

Style テンプレート (NewStyle) は、parent ステートメントを用いて、SAS 社が提供する Styles.Rtf をベースにして、フォントを Times New Roman、サイズを 9pt で設定している。作成される Style テンプレートは、事前に作成した Style テンプレートを保存するフォルダ内の「template.sas7bitm」に保存されている。

## 2.4 Style テンプレートの利用方法

2.3 節で作成した Style テンプレート「NewStyle」の利用例を Program code 4 に示す。利用例では、SAS ライブラリ「mystyle」を割り当て、次に ODS PATH ステートメントで Style テンプレートの参照順を設定している。そして、ODS RTF ステートメントの style=オプションで「NewStyle」を指定することで、作成する RTF ファイルの出力形式 (この事例では、すべてのフォントを Times New Roman、サイズを 9pt) を制御している。

Program code 4

```
libname mystyle "Styleテンプレートを保存したフォルダ";
ODS PATH mystyle.template(read) sashelp.tmplmst(read);

ODS RTF file ='xxx.rtf' style=NewStyle ;
proc print data= sashelp.class ;
run ;
ODS RTF close ;
```

## 3 Excel DDE プログラミングと ODS RTF プログラミングの違い

Excel DDE プログラミングで解析図表を作表する場合、図表計画書として作成した出力見本を出力先のテンプレートとして利用し、Figure 4 の赤枠の部分に集計結果を出力しているケースが多いと思う。つまり、テンプレートの一部、例えば、図表タイトルや項目名やカテゴリ、等を利用して解析図表を作成している。



①の検討時に、解析図表のタイトルと脚注を RTF ファイルのどの領域に出力するか、という問題に遭遇した。RTF ファイルを Word で開くと、Figure 5 のようにヘッダー部、本文、フッター部という 3 つの領域が存在する。ODS RTF プログラミングで、ODS RTF ステートメントに `bodytitle` オプションを指定しなければ、3 つの領域を利用できることが分かった。Title/Footnote ステートメントにテキスト情報を記述することでヘッダー部/フッター部に、ODS RTF `prepage=/text=` にテキスト情報を記述することで本文に、テキスト情報を出力できることを確認した。クライアントの要望が第一優先であるが、ODS RTF プログラミングの表現力を生かすためには、`bodytitle` オプションを利用しないほうがよいと我々は結論づけた。

## 4 SAS マクロを作成し運用するにあたって

### 4.1 SAS マクロの管理方針

業務で利用する SAS マクロは一元管理し、バージョンも管理したいと考えている。各プロジェクトの業務開始時に、最新版の SAS マクロをプロジェクトフォルダに配布する。プロジェクト進行中に新バージョンの SAS マクロがリリースされた場合には、各プロジェクトで導入の可否を判断いただくことを考えている。なお、一元管理を開始する前に、SAS マクロのベリフィケーションを実施し、その記録を残す予定である。

### 4.2 ベリフィケーション記録

SAS マクロごとに、Table1 に示したベリフィケーション記録を残す予定である。

Table 1. ベリフィケーション文書一覧

文書	作成者	承認の要否
ユーザ要求仕様書	利用者	要
指名書（プログラマー、ベリファイヤー）	承認者	—
ベリフィケーション計画書	プログラマー	要
マクロ仕様書	プログラマー	要
SAS マクロ	プログラマー	—
テスト計画書	ベリファイヤー	要
テスト報告書	ベリファイヤー	要
ベリフィケーション報告書	プログラマー	要

利用者からの新規の開発要求を受け付けて、SAS マクロを開発する否かを判断し、開発することを決定した後に、担当者をアサインする。そして、ベリフィケーション計画書とマクロ仕様書を作成してから、SAS マクロを作成し、テスト計画書に従いテストして、テスト結果をテスト報告書にまとめて、最後にベリフィケーション報告書を作成する、という一連の流れで記録を残す予定である。

## 5 最後に

Excel DDE プログラミングに慣れている人が、ODS RTF プログラミングに移行するのは簡単なことではない。ODS RTF での出力環境を整備することで、ODS RTF プログラミング未経験者に対して手助けとなることを期待している。ODS RTF プログラミングに移行することで、冒頭で述べたシステム連携時の不安定さや

DDE 機能のサポート期限の不透明さから解放されて、ODS 機能を活用する機会になればと思う。ODS 機能には RTF だけでなく EXCEL、WORD、PDF、HTML 等が存在し、これらを活用する契機となることも期待している。

SAS マクロの管理に関して少し記載したが、まだまだ手探り状態であるため、いろんな角度からアドバイスや情報交換いただけると有難い。

## 6 引用文献

- [1] 麻生美由紀. 2018. ods rtf+proc report における解析帳票の日本語/英数字フォントの明示的制御. SAS ユーザー総会 2018.  
[https://www.sas.com/ja\\_jp/events/18/users-group-2018/sessions.html#m=4-2](https://www.sas.com/ja_jp/events/18/users-group-2018/sessions.html#m=4-2)
- [2] 太田裕二. 2017. ODS 機能と REPORT プロシジャを用いた 解析帳票の作成. SAS ユーザー総会 2017.  
[https://www.sas.com/content/dam/SAS/ja\\_jp/doc/event/sas-user-groups/usergroups2017-b-11-02.pdf](https://www.sas.com/content/dam/SAS/ja_jp/doc/event/sas-user-groups/usergroups2017-b-11-02.pdf)