PharmaSUG China 2024 - Paper CC-10015

# Automatically Setting CSR Listing Column Widths - A SAS Macro-based Solution

Shuchun (Coral) Wu, Johnson & Johnson

## ABSTRACT

In our daily work, adjusting column widths to create visually appealing listings can be a tedious and time-consuming task, especially when dealing with numerous columns in a single output. This task is not a one-time effort; as data refreshes, the need for re-adjustment arises, leading to higher maintenance costs and lower efficiency. While an effective Python-based solution exists for automatically setting column widths during the generation of a PDF output directly from a SAS dataset, this approach is not widely adopted in current practices. This article introduces a SAS-based method that could be seamlessly integrated into the process of generating RTF outputs from datasets, a more commonly used workflow in our daily tasks. A suite of macros has been developed based on this method, offering SAS programmers a streamlined solution. By incorporating these macros into their programs, SAS programmers can significantly reduce the need for manual adjustments, saving time and effort. This not only enhances efficiency but also contributes to a better work-life balance.

## INTRODUCTION

The combination of the ODS RTF destination and PROC REPORT in SAS equips programmers with a robust set of tools for creating diverse and high-quality tables and listings. Despite these capabilities, generating an output that is both visually appealing and informative often involves a laborious process of trial and error to determine the optimal column width allocation. This iterative adjustment is necessitated by a lack of precise knowledge regarding the required width for displaying text in a particular style, leading to a time-consuming endeavor. Typically, we resort to setting column widths as fixed values in our code, a practice that proves insufficient when dealing with dynamic data updates. As new data is introduced and our outputs require refreshing, the static width values may no longer be suitable due to potential changes in the database, which may increase the maintenance costs of the program. By accurately calculating the necessary width for displaying text, we can leverage this information to dynamically adjust column widths, facilitating a fully data-driven layout of the report. This approach saves valuable time and effort that can be reallocated towards addressing other critical tasks.

## HOW TO REALIZE THE AUTOMATION OF COLUMN WIDTH SETTING

In this section, we will present a SAS macro-based approach for automating the setting of column widths in a listing. This method involves the following four steps:

1. Calculate the necessary cell width values for all displayed text.
2. Determine the optimal column width value.
3. Convert the width values into a format appropriate for configuration in PROC REPORT.
4. Transfer the values to reporting macro for RTF output and provide a printout for user verification.

### STEP 1: CALCULATE THE NECESSARY CELL WIDTH VALUES FOR ALL DISPLAYED TEXT.

To realize the automation, the measurement with high precision of the width required for displaying a text is the most basic and critical thing. In SAS programming, only length function is available for programmers to count the number of characters of each string, however, this method only works well when a monospaced font is used, as within this font each character occupies the same amount of space and no kern between each character, which is easier for users to identify the special characters. Due to this attribute, monospace fonts take the lead in coding environments while for most outputs, font setting is often not that case. Take Times New Roman - a commonly used font setting - as an example, different characters may take up different amounts of horizontal space, which making them more aesthetically

pleasing for readers while leading to more complexity for precise estimation on each string width. Though there is no ready-made function available in SAS to handle this, this value could still be obtained by calculation based on the following key elements with the comprehensive knowledge of the font attributes and RTF language:

- Font, Font size, Font TTF
- Template style setting for RTF

First, each font has its TTF file which is a font file that utilizes the TrueType font format and defines the shape and appearance of individual characters within a font. Therefore, in theory, the width of a string could be calculated out based on the attribute of individual character defined in this file. To compute the width of each string, at least two attributes are required and should be extracted from this file. One is advance width value, which could be found in the Horizontal Metrics (<htmx>) Table that defines the advance width in font design units for each character. Another is kern value, which could be retrieved in the Kerning (<kern>) Table that contains the values that control the inter-character spacing for each paired characters (see an example of pair characters in figure 1). With them, a basic formula could be derived (listed as below with an example) to calculate the horizontal width of a single word or string, even a paragraph.

The basic formula:

The horizontal width of a text in design unit = sum of the advanced width of each character + any kern values for every two paired characters (if applicable)

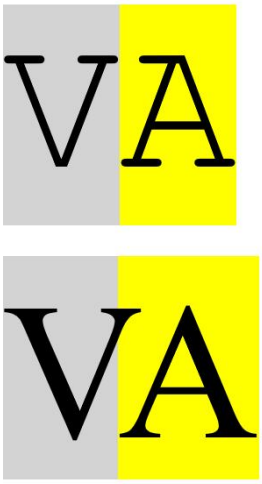


**Figure 1. no kern (font: Courier New) vs. kern (font: Times New Roman) of paired VA**

An example: VALUE in Times New Roman font

| char | aw |
|---|---|
| V | 1522 |
| A | 1518 |
| L | 1336 |
| U | 1512 |
| E | 1340 |

| | |
|---|---|
| Pair | kern |
| VA | -264 |
| ---------- | |
| | 6964 units |

Second, considering to each string will be contained within a table cell in a RTF output, formatted according to a specified template style, based on the knowledge of RTF language, we could get the left/right cell margin value by opening one RTF file as a plain text and searching the following keywords: \clpadl or \clpadr. Per Rich Text Format (RTF) Specification, \clpadl signifies left cell margin and \clpadr signifies right cell margin. For instance, in the following RTF code snippet:

\cltxlrtb\clvertalt \clpadt67\clpadft3\clpadr67\clpadfr3\cellx1780

This snippet specifies text in a top-aligned cell flows from left to right and top to bottom, with left and right cell margin set as 67 twips per the current template style, \cellx defines the right boundary of a table cell, encompassing its half of the space between cells. With this information, we can derive the following formula (listed as below) of the cell width of each paragraph required for printing in a single line without any wrapping.

The secondary formula:

cell width of each paragraph required in RTF in twips (no wrapping) = left cell margin width + the horizontal width of a text + right cell margin width + half of the space between cells

The margin and spacing elements in this formula are highly determined on the template style setting. This implies that in certain configurations, there may be no cell margin or no space between cells. Note here the horizontal width of a text should be a value in twip unit, which involves a unit conversion from font design unit to twip. Two factors are required for this conversion: UPM value and font size. In a font, each character is fitted into its own space container. The advanced width value is measured in design unit relative to the width of whole container, known as the 'UPM' or 'em size'. In TrueType fonts, the UPM is generally set to 1024 or 2048, a value that can be identified by examining the font file.

Fonts are often measured in points (pt). Font size is a number that measures the size of each character space container relative to 1 inch (equivalent to 72 points) on a screen or printed page. For example, with a font size of 8pt, each space container is measured as 8 / 72, or 1/9th of an inch. And measurements in RTF are generally in twips. A twip is a twentieth of a point, or a 1440th of an inch. Therefore, the formula for conversion from design unit to twip should be calculated as advanced width of paragraph in design unit / UPM x font size in points / 72 x 1440.

Let's consider the example of the word 'VALUE' in 8 pt font size with a UPM setting of 2048. The advanced width of 'VALUE' in twips can be calculated as 6964 / 2048 x 8 / 72 x 1440, which is approximately 544 twips. Using this calculated value, the cell width of 'VALUE' required in our RTF template without wrapping can be determined as 544 + 67 + 67 + 40 / 2 = 698 twips.

## STEP 2: DETERMINE THE OPTIMAL COLUMN WIDTH VALUE.

Assume we're creating an RTF file on an 11-inch-wide paper. After setting both the left and right margins to 1 inch, we're left with 9 inches of paper width to fill with columns. Here take 8 records from a test data as an example (see Figure 2). By utilizing all the formulas outlined in step 1, we can determine the necessary cell width values for display. These width values will be organized into different percentiles, ranging from the minimum to the maximum in increments of 10% (refer to Table 1). Additionally, the width of each column header, along with the maximum wrap width of the header or body with a delimiter of space, will be detailed in Table 1. In this step, we will basically refer to the strategy with three scenarios introduced by Tao Zhang, 2023 for determination. Furthermore, a maximum wrap width value will serve as a threshold for each column, ensuring that width values below this threshold are excluded. As per the default setting in MS Word, text will automatically wrap by space if the column width value is set not less than this threshold, avoiding word breaks when there isn't sufficient space to display content in a single line. Another width limit value, derived as twice the maximum page width divided by the number of columns, is also used to filter out excessively large values in specific cases. In situations where there is no space in each value for a column, that column will be fixed by default with the maximum width value. Users also have the option to fix any column by specifying it through a macro argument.

| TRT01A | SUBJID | AGESEXR | SOCPRTERM | ONSETDAY | STDATE | ENDATE | ACTION | RELATION | OUTSERSEV |
|---|---|---|---|---|---|---|---|---|---|
| Placebo | 200001 | 80/F/~nMultiple | Psychiatric Disorders/~nAnxiety/~nAnxiety | 17 | 2003-10-16 | 2003-10-20/ 5 | Dose Not Changed | Possibly Related | Recovered/Resolved/~nSevere/~nNo |
| Placebo | 200001 | 80/F/~nMultiple | Musculoskeletal And Connective Tissue Disorders/~nArthralgia/~nLeft Knee Pain Worsening | 126 | 2004-02-02 | Ongoing | Drug Withdrawn | Not Related | Not Recovered/Not Resolved/~nSevere/~nNo |
| Placebo | 200001 | 80/F/~nMultiple | Gastrointestinal Disorders/~nConstipation/~nConstipation | 87 | 2003-12-25 | Ongoing | Dose Not Changed | Not Related | Not Recovered/Not Resolved/~nModerate/~nNo |
| Placebo | 200001 | 80/F/~nMultiple | General Disorders And Administration Site Conditions/~nFatigue/~nTiredness | 87 | 2003-12-25 | Ongoing | Dose Not Changed | Possibly Related | Not Recovered/Not Resolved/~nSevere/~n |
| Placebo | 200001 | 80/F/~nMultiple | Gastrointestinal Disorders/~nNausea/~nNausea Intermittent | 17 | 2003-10-16 | 2003-10-20/ 5 | Dose Not Changed | Possibly Related | Recovered/Resolved/~nSevere/~nNo |
| Miracle Drug 10 mg | 200002 | 70/F/~nBlack or African American | Nervous System Disorders/~nDizziness/~nLightheadedness | 140 | 2004-02-26 | 2004-02-26/ 1 | Dose Not Changed | Not Related | Recovered/Resolved/~nMild/~nNo |
| Miracle Drug 10 mg | 200002 | 70/F/~nBlack or African American | Musculoskeletal And Connective Tissue Disorders/~nMuscle Spasms/~nMuscle Spasms | 88 | 2004-01-05 | Ongoing | Dose Not Changed | Not Related | Not Recovered/Not Resolved/~nMild/~nNo |
| Miracle Drug 10 mg | 200002 | 70/F/~nBlack or African American | Cardiac Disorders/~nPalpitations/~nPalpitations Intermittent | 88 | 2004-01-05 | Ongoing | Dose Not Changed | Not Related | Not Recovered/Not Resolved/~nMild/~nNo |

**Figure 2 shows some records from test data (from open-source library)**

| | COL1 | COL2 | COL3 | COL4 | COL5 | COL6 | COL7 | COL8 | COL9 | COL10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **BODYMIN** | 660 | 634 | 492 | 669 | 314 | 901 | 714 | 1233 | 923 | 350 |
| **BODY10%** | 660 | 634 | 492 | 678 | 314 | 901 | 714 | 1233 | 923 | 350 |
| **BODY20%** | 660 | 634 | 492 | 770 | 314 | 901 | 714 | 1372 | 923 | 350 |
| **BODY30%** | 660 | 634 | 492 | 954 | 314 | 901 | 714 | 1372 | 923 | 350 |
| **BODY40%** | 660 | 634 | 492 | 1016 | 314 | 901 | 714 | 1372 | 923 | 510 |
| **BODY50%** | 1062.5 | 634 | 598.5 | 1213.5 | 314 | 901 | 714 | 1372 | 923 | 634 |
| **BODY60%** | 1465 | 634 | 705 | 1420 | 314 | 901 | 714 | 1372 | 923 | 634 |
| **BODY70%** | 1465 | 634 | 705 | 1705 | 314 | 901 | 1065 | 1372 | 1225 | 1522 |
| **BODY80%** | 1465 | 634 | 1882 | 1874 | 394 | 901 | 1065 | 1372 | 1225 | 2082 |
| **BODY90%** | 1465 | 634 | 1882 | 3434 | 394 | 901 | 1065 | 1372 | 1225 | 2082 |
| **BODYMAX** | 1465 | 634 | 1882 | 3705 | 394 | 901 | 1065 | 1372 | 1225 | 2082 |
| **HEADMAX** | 1275 | 863 | 1274 | 1493 | 1321 | 818 | 1867 | 2488 | 2306 | 785 |
| **BODYWRAPMAX** | 660 | 634 | 816 | 1229 | 394 | 901 | 945 | 904 | 696 | 1522 |
| **HEADWRAPMAX** | 806 | 634 | 683 | 749 | 637 | 516 | 763 | 863 | 963 | 785 |

**Table 1 summary of the width values**

| COL1 | COL2 | COL3 * | COL4 | COL5 | COL6 | COL7 | COL8 | COL9 | COL10 |
|---|---|---|---|---|---|---|---|---|---|
| 1465 | 634 | 1595 | 1874 | 637 | 901 | 1065 | 1372 | 1225 | 2082 |

**Table 2 Width allocation for test data**

In our example, the maximum page width allowable is 12850 twips after accounting for width loss. Following the above rules, col6 has been set as fixed. Values falling below the threshold (indicated in red in Table 1) have been highlighted in orange within the table. The column widths that have been determined are highlighted in green. The ultimate width allocation is presented in Table 2.

## STEP 3: CONVERT THE WIDTH VALUES INTO A FORMAT APPROPRIATE FOR CONFIGURATION IN PROC REPORT.

It is important to note that in the proc report define statement, the value setting for cell width is in inch units. However, it may not always be straightforward to obtain this value by simply dividing 1440 by the cell width value determined in step 2. There is a known issue, confirmed by SAS technical support, where proc report is unable to utilize the full width of the page even when the sum of cell width settings reaches 100% of the maximum page width. Unfortunately, this issue has not yet been resolved. Several solutions have been proposed to address this problem, such as adjusting template settings or implementing post-programming modifications. However, if we choose not to address this issue in our example, we must factor in this width loss during our calculations. To mitigate any width loss resulting from this bug during conversion, we must consider this width loss when using the maximum page width in step 2 and converting the width value from twips to inches in this step.

Upon examining the RTF code, it appears that the width loss is likely correlated with the number of columns. In scenarios where there is no cell padding or cell spacing specified in the RTF template style, the width loss is approximately 10 twips for each column, with an additional 10 twips for the leftmost column. Therefore, the maximum page width eligible for use in step 2 should be adjusted by subtracting the total width loss.

Consequently, in this step, when converting the width value from twips to inches, 10 twips should be added back to the width for each column, along with an extra 10 twips for the first column. The conversion formula is as follows: Each column width in inches = (column width in twips + 10 twips + 10 twips for the first column) / 1440.

In our example, the column width values (inch) are shown in Table 3.

| COL1 | COL2 | COL3 | COL4 | COL5 | COL6 | COL7 | COL8 | COL9 | COL10 |
|---|---|---|---|---|---|---|---|---|---|
| 1.03125 | 0.447222 | 1.114583 | 1.308333 | 0.449306 | 0.632639 | 0.746528 | 0.959722 | 0.857639 | 1.452778 |

**Table 3 Width setting after conversion for test data**

## STEP 4: TRANSFER THE VALUES TO REPORTING MACRO FOR RTF OUTPUT AND PROVIDE A PRINTOUT FOR USER VERIFICATION.

During this step, a series of width setting values will be displayed in the result window for the user's reference. By incorporating this macro into other output macros, these values can be seamlessly passed on to the final proc report process for configuring the display settings (refer to Figure 3).

**LSFAE01TEST: Listing of Adverse Events; Safety Analysis Set (OPEN-SOURCE DATA)**

| Treatment Group | Subject ID | Age (years)/ Sex/ Race | System Organ Class/ Preferred Term/ Reported Term | Day of AE Onset[a] | Start Date | End Date/ Duration (days) | Action Taken With Study Treatment | Relationship to Study Treatment[b] | Outcome/ Severity/ Serious |
|---|---|---|---|---|---|---|---|---|---|
| Miracle Drug 10 mg | 200002 | 70/F/ Black or African American | Musculoskeletal And Connective Tissue Disorders/ Muscle Spasms/ Muscle Spasms | 88 | 2004-01-05 | Ongoing | Dose Not Changed | Not Related | Not Recovered/Not Resolved/ Mild/ No |
| | | | Cardiac Disorders/ Palpitations/ Palpitations Intermittent | 88 | 2004-01-05 | Ongoing | Dose Not Changed | Not Related | Not Recovered/Not Resolved/ Mild/ No |
| | | | Nervous System Disorders/ Dizziness/ Lightheadedness | 140 | 2004-02-26 | 2004-02-26/ 1 | Dose Not Changed | Not Related | Recovered/Resolved/ Mild/ No |
| Placebo | 200001 | 80/F/ Multiple | Psychiatric Disorders/ Anxiety/ Anxiety | 17 | 2003-10-16 | 2003-10-20/ 5 | Dose Not Changed | Possibly Related | Recovered/Resolved/ Severe/ No |
| | | | Gastrointestinal Disorders/ Nausea/ Nausea Intermittent | 17 | 2003-10-16 | 2003-10-20/ 5 | Dose Not Changed | Possibly Related | Recovered/Resolved/ Severe/ No |
| | | | Gastrointestinal Disorders/ Constipation/ Constipation | 87 | 2003-12-25 | Ongoing | Dose Not Changed | Not Related | Not Recovered/Not Resolved/ Moderate/ No |
| | | | General Disorders And Administration Site Conditions/ Fatigue/ Tiredness | 87 | 2003-12-25 | Ongoing | Dose Not Changed | Possibly Related | Not Recovered/Not Resolved/ Severe/ No |
| | | | Musculoskeletal And Connective Tissue Disorders/ Arthralgia/ Left Knee Pain Worsening | 126 | 2004-02-02 | Ongoing | Drug Withdrawn | Not Related | Not Recovered/Not Resolved/ Severe/ No |

[a] Day of last administration of study treatment (prior to the AE) and Day of AE Onset are in reference to the date of first dosing on Day 1.
[b] Relationship to study agent is assessed by the investigator.

[lsfae01test.rtf] [path/lsfae01test.sas] 15MAY2024, 04:15

**Figure 3** **A screenshot of the final report based on open-source dummy data**

## ALTERNATIVE SOLUTION FOR SPECIFIC NEED

While automation can streamline processes, it's important to acknowledge that not every scenario fits a one-size-fits-all approach. In instances where the automated macro falls short of meeting user demands, a supplementary macro has been developed to address such special cases. This additional tool allows users to manually adjust column widths to their specific requirements.

To utilize this macro, users are prompted to adjust column widths in the RTF file by interactively dragging the adjustment tool to the desired width, saving the changes, and then closing the file. Upon running the supplemental macro, users can extract, convert, and retrieve the customized width settings specified in the RTF file. Unlike the primary automation macro, this tool serves as an assistant, intended for temporary use during program development rather than integration into production code.

It's crucial to note that the values obtained through this process are fixed based on the immediate special requirements and should not be viewed as an ultimate solution. Nevertheless, this macro offers an efficient alternative for users to quickly obtain tailored width settings without the need for repetitive adjustments. It serves as a practical tool to cater to specific needs, enhancing user flexibility and efficiency in the development process.

## CONCLUSION

In conclusion, this article presents a SAS macro-based solution to automatically set the column widths for CSR listings in four steps.

The effectiveness of this method depends on accurately estimating and converting text width within table cells. To facilitate understanding, the article introduces and explains the key components of the calculation formulas in detail. While the specific numbers in the formulas may vary, grasping these essential elements empowers users to tailor formulas to suit their unique work contexts. This solution

could be seamlessly integrated into the established workflow from a SAS dataset to a RTF output, simplifying the generation of listing outputs. Programmers can easily incorporate the provided macros into their existing workflows or utilize them as a resource for determining column width settings based on their requirements. For scenarios where users need to customize width settings in special cases, a dedicated macro is available to meet their specific needs.

## REFERENCES

Rich Text Format (RTF) Specification,version 1.6 https://latex2rtf.sourceforge.net/rtfspec.html.

Tao Zhang, 2023. “Automatically Setting Column Width and Pagination using Python” *Proceedings of PharmaSUG China 2023 Conference, Nanjing*. Available at https://www.lexjansen.com/pharmasug-cn/2023/AP/Pharmasug-China-2023-AP144.pdf

## ACKNOWLEDGMENTS

The author would like to thank Yubo (David) Fan, Yufan Chen, Tingting Wu for their great advice.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Shuchun (Coral) Wu
Johnson & Johnson
cwu40@its.jnj.com