

# **TIBCO Spotfire® Clinical Graphics**

## **Connector for SAS Macro Reference**

*Software Release 2.2*  
*August 2012*

## Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

EJB, J2EE, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 1999-2012 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

The correct bibliographical reference for this document is as follows:

*TIBCO Spotfire® Clinical Graphics 2.2 Connector for SAS Macro Reference*, TIBCO Software Inc.

For technical support, please visit <http://spotfire.tibco.com/support> and register for a support account.

# CONTENTS

---

<b>Chapter 1</b>	<b>TIBCO Spotfire Clinical Graphics Connector for SAS</b>	<b>1</b>
	Overview	2
	Introduction to TSCG Connector for SAS	3
<b>Chapter 2</b>	<b>Global and Layout Macros</b>	<b>5</b>
	Introduction	6
	tscg_Axis	7
	tscg_CreateGraph	11
	tscg_GraphTable	13
	tscg_Legend	14
	tscg_Page	16
	tscg_PageBy	17
	tscg_Theme	18
	tscg_Titles	19
<b>Chapter 3</b>	<b>Plot Types</b>	<b>21</b>
	Introduction	23
	tscg_AreaBarChart	24
	tscg_AreaPlot	27
	tscg_BarChart	30
	tscg_BoxPlot	33

<b>tscg_BubblePlot</b>	<b>35</b>
<b>tscg_CDFPlot</b>	<b>38</b>
<b>tscg_ContourPlot</b>	<b>40</b>
<b>tscg_CumulativeIncidence</b>	<b>42</b>
<b>tscg_DeltaPlot</b>	<b>44</b>
<b>tscg_DotPlot</b>	<b>47</b>
<b>tscg_DoubleDotPlot</b>	<b>51</b>
<b>tscg_DurationPlot</b>	<b>55</b>
<b>tscg_ForestPlot</b>	<b>58</b>
<b>tscg_Histogram</b>	<b>61</b>
<b>tscg_ImagePlot</b>	<b>64</b>
<b>tscg_KaplanMeierPlot</b>	<b>66</b>
<b>tscg_LinePlot</b>	<b>68</b>
<b>tscg_PopulationPyramid</b>	<b>72</b>
<b>tscg_QQMathPlot</b>	<b>75</b>
<b>tscg_QQPlot</b>	<b>77</b>
<b>tscg_ScatterPlot</b>	<b>79</b>
<b>tscg_ScatterPlotMatrix</b>	<b>81</b>
<b>tscg_SurvivalCurve</b>	<b>83</b>
<b>tscg_TreeMap</b>	<b>85</b>
 <b>Chapter 4 Reference Elements</b>	 <b>89</b>
<b>Introduction</b>	<b>90</b>
<b>tscg_LSFIt</b>	<b>92</b>
<b>tscg_LTSRobFit</b>	<b>93</b>
<b>tscg_MMRobFit</b>	<b>94</b>
<b>tscg_PolyLSFit</b>	<b>95</b>
<b>tscg_RefABLines</b>	<b>96</b>
<b>tscg_RefArrow</b>	<b>98</b>

<b>tscg_RefCurve</b>	<b>101</b>
<b>tscg_RefPoints</b>	<b>103</b>
<b>tscg_RefText</b>	<b>105</b>
<b>tscg_RefXLines</b>	<b>107</b>
<b>tscg_RefYLines</b>	<b>110</b>
 <b>Chapter 5 Utility Macros</b>	 <b>113</b>
<b>Introduction</b>	<b>115</b>
<b>tscg_AddGraphParm</b>	<b>116</b>
<b>tscg_Configuration</b>	<b>117</b>
<b>tscg_DeleteGraphParm</b>	<b>118</b>
<b>tscg_DeleteParmSet</b>	<b>119</b>
<b>tscg_FindPath</b>	<b>120</b>
<b>tscg_GetProgName</b>	<b>121</b>
<b>tscg_GetSourcePath</b>	<b>122</b>
<b>tscg_GraphOutput</b>	<b>123</b>
<b>tscg_HttpPost</b>	<b>124</b>
<b>tscg_HttpRqst</b>	<b>125</b>
<b>tscg_InsertGraphODS</b>	<b>126</b>
<b>tscg_InsertGraphParm</b>	<b>127</b>
<b>tscg_InsertSPJName</b>	<b>128</b>
<b>tscg_InsertSPLUSLog</b>	<b>129</b>
<b>tscg_InsertSPLUSScript</b>	<b>130</b>
<b>tscg_MakeList</b>	<b>131</b>
<b>tscg_Parse</b>	<b>132</b>
<b>tscg_ParseBody</b>	<b>133</b>
<b>tscg_Sort</b>	<b>134</b>
<b>tscg_StartSubPlot</b>	<b>136</b>
<b>tscg_StripTag</b>	<b>137</b>

<b>tscg_SubPlot</b>	<b>138</b>
<b>tscg_TestBoolean</b>	<b>140</b>
<b>tscg_Validate</b>	<b>141</b>
<b>tscg_Wait</b>	<b>142</b>
<b>tscg_WriteGraphJob</b>	<b>143</b>
 <b>Chapter 6 Troubleshooting</b>	 <b>145</b>
<b>Introduction</b>	<b>146</b>
<b>SAS Macro Errors</b>	<b>147</b>
<b>Job Processing Errors</b>	<b>148</b>
 <b>Index</b>	 <b>149</b>

# TIBCO SPOTFIRE CLINICAL GRAPHICS CONNECTOR FOR SAS

# 1

---

Overview	2
Introduction to TSCG Connector for SAS	3
Macro Overview	3

## **OVERVIEW**

TIBCO Spotfire<sup>®</sup> Clinical Graphics (TSCG) includes a local desktop client with a rich, interactive user interface that works in conjunction with the TSCG Server, which in turn sits on the TIBCO Spotfire Statistics Services. In some instances, such as production or batch execution of graphics, the interactive interface is not required. The TIBCO Spotfire Clinical Graphics Connector for SAS provides a library of SAS macros that can communicate with the TSCG server to generate graphics without an interactive interface.



# INTRODUCTION TO TSCG CONNECTOR FOR SAS

TIBCO Spotfire Clinical Graphics Connector for SAS (TSCG Connector) is a library of SAS macros that directs a SAS program to create TSCG graphics. You can create these graphics by invoking the macro specific to a particular plot type or by referencing an IGD file that you created and saved using the rich Windows client. The SAS macros provide a familiar mechanism for specifying the details of the desired plots and insulate you from the specifics of the interactions with the TSCG server.

## Macro Overview

The Connector macros described in this manual are grouped into four categories:

**Table 1.1:** *Connector macro general categories.*

Category	Description
Global and Layout Macros	Used to specify details that are not specific to a particular graph type. These details include features such as the page size and layout, titles, legends and others.
Plot Types Macros	Includes a macro for each type of plot available in TSCG (for example, area bar charts, delta plots, scatter plots, and so on).
Reference Elements Macros	Includes macros for features to add to different plot types, including reference lines, reference arrows, reference text, and so on.
Utility Macros	Includes macros that handle the low-level details of making requests of the TSCG server and processing the results of those requests. Typically, users do not call these macros directly; rather, they are called by the other macros as needed.



# GLOBAL AND LAYOUT MACROS

# 2

---

Introduction	6
tscg_Axis	7
tscg_CreateGraph	11
tscg_GraphTable	13
tscg_Legend	14
tscg_Page	16
tscg_PageBy	17
tscg_Theme	18
tscg_Titles	19

## INTRODUCTION

This chapter contains descriptions for general use macros.

- Use the **global macros** to control page layout, themes, and graph output.
- Use the **layout macros** to control the titles, the graph table's number of columns and rows, axes, and how character and categorical labels should be ordered according to a numeric column.

In addition, you can use the `tscg_CreateGraph` macro to render any graphic defined in an IGD file.

# TSCG\_AXIS

The Axis macro `tscg_Axis` provides control over the axis display, including the label, the minimum and maximum values, the tick marks, the grid, and the scale for both X and Y values.

## Note

TSCG provides support for axis groups (that is, axis names with a numeric suffix), but only for custom templates that support them. No graph types shipped with TSCG 2.0 support axis groups.

## Usage

```
tscg_Axis(name=,
  labelEnabled=,
  label=,
  from=,
  to=,
  by=,
  at=,
  majorTickLabels=,
  enableMinorTicks=,
  numberOfMinorTicks=,
  enableMajorGrid=,
  logTransform=,
  alternating=BLF,
  symmetric=FALSE,
  symmetricBase=,
  relation=);
```

## Required Arguments

None

## Optional Arguments

### name

The character string containing the name of the X or Y axis. For templates that support axis groups, the name can be suffixed with an integer so Y, Y2, Y3, and Y4 refer to axis 1, 2, 3, and 4, respectively.

### labelEnabled

TRUE to print the axis label; otherwise FALSE.

**label**

Available only if name is X or Y. For templates that support axis groups or matrix plots, you can set axis labels using line breaks. For example,

```
1[label1] 2[label2]
```

**from**

The numeric value for the minimum of the current axis.

**to**

The numeric value for the maximum of the current axis.

**by**

The numeric value for the intervals between major tick marks (from to by).

**at**

The numeric values to place major tick marks.

**Note**

You must separate multiple at positions with commas, and you must enclose the comma-separated list in a SAS %str call. (Note that labels can not contain commas.)

**majorTickLabels**

The labels for major tick marks. (This option requires at. Defaults to those values).

**Note**

You must separate multiple tick labels with commas, and you must enclose the comma-separated list in a SAS %str call. (Note that labels can not contain commas.)

**enableMinorTicks**

Indicates whether minor ticks should be shown (TRUE) or not (FALSE).

**numberOfMinorTicks**

The number of minor ticks between major tick marks

**enableMajorGrid**

Logical value indicating whether to set grid to off (FALSE) or on (TRUE).

**logTransform**

Logical value for linear (FALSE) or log scale (TRUE).

**alternating**

Sets the sides for alternating axes. Specify one of the following:

- BLF (Bottom/Left First; The default).
- BLO (Bottom/Left Only).
- TRF (Top/Right First).
- TRO (Top/Right Only).

**symmetric**

Specifies whether to draw the axis symmetrically about a specified point. If TRUE, enables symmetric axis about the value specified by `symmetricBase`. If FALSE (the default), a standard axis is displayed.

**symmetricBase**

Numeric specifying the center of a symmetric axis.

**relation**

Sets the relation of scales. Specify `free`, `same`, or `identical`.

**Example**

```
%tscg_Axis(name=X,
    label=x-axis,
    from=10,
    to=50,
    at=%str(10, 12, 16, 24, 40),
    majorTickLabels=%str(+0, +2, +4, +8, +16),
    enableMajorGrid=T);

%tscg_Axis(name=Y,
    label=y-axis,
    from=1000,
```

```
to=10000,  
logTransform=T,  
enableMajorGrid=T,  
enableMinorTicks=T);  
  
%tscg_scatterplot(dataLibrary = examples,  
dataSet = fuel,  
outputFile=&outputPath.\xyaxis,  
outputFormat=png,  
xVariable = Mileage,  
yVariable = Weight,  
groupVariable = Type);
```



# TSCG\_CREATEGRAPH

The graph creation macro `tscg_CreateGraph` creates a TIBCO Spotfire Clinical Graphics plot using the indicated IGD file as a source and writing the output to the named target.

## Usage

```
tscg_CreateGraph(outputFile=&tscg_progName.,
graphType=&tscg_outputFormat.,
graphDoc=&tscg_progName..igd,
keepSScript=&tscg_keepSScript.,
keepJobSpec=&tscg_keepJobSpec.,
parmSet=&tscg_ParmSet.);
```

## Required Arguments

None.

## Optional Arguments

### outputFile

Fully-qualified name of the output graphic to be created. (It can be a path with a file name or a URL.)

### Note

You must ensure the specified output target location exists before you try to write output to a location. The macros do not automatically create a destination folder.

### graphType

Graph file format for output.

**NOTE:** If an extension is specified on the `outputFile` parameter, this argument is ignored.

(It can be one of EPS, JPG, PDF, PNG, WMF; the default is determined by the `tscg_outputFormat` variable, which is defined in the `TSCG_CONFIGURATION` macro. See the section `tscg_Configuration` on page 117.)

### graphDoc

Fully-qualified name of the graph document to be used as the basis of the new graph. (Can be a path and file name or a URL.)

### **keepSScript**

A logical flag indicating if the Spotfire S+ script used to generate the graph should be retained. Specify either T or F (the default).

### **parmSet**

The name of dataset to hold run-time parameter names and values prior to writing them to the **GraphJobSpec** file. The default name is provided by the global macro variable `tscg_ParmSet` that is set in the `tscg_configuration` macro.

### **Example**

To use this macro, call the `tscg_AddGraphParm` macro as many times as needed to create the run-time parameters (that is, the data source, the columns, the title, and so on), and then call `tscg_CreateGraph` with the name of an existing IGD file:

```
%tscg_AddGraphParm(name=DataSource,  
    value=C:\TSCG\Sample\Data\fuel.sas7bdat);  
%tscg_AddGraphParm(name=XColumns, value=baseline);  
%tscg_AddGraphParm(name=YColumns, value=endpoint);  
  
%tscg_CreateGraph(outputFile=C:\TSCG\Sample\MyGraph.png,  
    graphDoc=C:\TSCG\Sample\IGDFiles\MyGraph.igd);
```

## TSCG\_GRAPHTABLE

The Graph Table macro `tscg_GraphTable` provides the ability to specify the number of columns or number of rows for the Trellis layout. You can also specify the plot aspect ratio Graph Table macro.

### Usage

```
tscg_GraphTable(numberOfRows=,  
                numberOfColumns=,  
                plotAspectRatio =);
```

### Required Arguments

None.

### Optional Arguments

#### **numberOfRows**

The number of rows in Trellis layout.

#### **numberOfColumns**

The number of columns in Trellis layout.

#### **plotAspectRatio**

The plot aspect ratio. Must be strictly positive. The default is unspecified for max fill.

### Example

```
%tscg_GraphTable(numberOfRows=2, numberOfColumns=1);
```

## TSCG\_LEGEND

The Legend macro, `tscg_Legend`, provides the ability to show the legend or hide it, to specify the position and style. Note that legend labels and order is best obtained by labeling the groupingVariable and use of the `tscg_sort` macro to specify the order.

### Usage

```
tscg_Legend(visible=,  
            legendLocation=,  
            numberOfColumns=,  
            insideLegend=False,  
            includeFitLines=True,  
            includeIntervalLines=True,  
            includeAllGroups=True,  
            title=);
```

### Required Arguments

None

### Optional Arguments

#### **visible**

TRUE to show the legend (the default); otherwise FALSE.

#### **legendLocation**

Specifies the location of the legend in the plot area. Can be one of the following:

- TopLeft
- TopCenter
- TopRight
- BottomLeft
- BottomCenter
- BottomRight
- RightTop
- RightCenter
- RightBottom
- LeftTop
- LeftCenter
- LeftBottom

#### **numberOfColumns**

Specifies the number of columns in legend.

**includeFitLines**

TRUE to include fit lines in the legend (the default), if they are included in the plot; otherwise FALSE.

**includeIntervalLines**

TRUE to include interval lines in the legend (the default), if they are included in the plot; otherwise FALSE.

**includeAllGroups**

TRUE to include all group symbols on all pages of a multi-page plot in the legend (the default); otherwise FALSE.

**title**

Contains the legend title.

**Example**

```
%tscg_Legend(visible=T,  
  legendLocation=leftbottom,  
  numberOfColumns=3,  
  insideLegend=T,  
  title=KM legend);  
  
%tscg_KaplanMeierPlot(dataLibrary=examples,  
  dataSet=kmsubjid,  
  outputFile=&outputPath.\legendKM,  
  outputFormat=png,  
  timeVariable=ttp,  
  statusVariable=ttpcnsr,  
  groupVariable=armcd);
```

## TSCG\_PAGE

Use the Page Layout macro, `tscg_Page`, to specify your graph's margin settings and the page format.

### Usage

```
tscg_Page(units=,  
width=,  
height=,  
outerMarginTop=,  
outerMarginBottom=,  
outerMarginLeft=,  
outerMarginRight=);
```

### Required Arguments

None

### Optional Arguments

#### **units**

The units for page measurements (cm, mm, in, ...).

#### **width**

The numeric value specifying page width.

#### **height**

The numeric value specifying page height.

#### **outerMarginTop**

The numeric value specifying page top margin.

#### **outerMarginBottom**

The numeric value specifying page bottom margin.

#### **outerMarginLeft**

The numeric value specifying page left margin.

#### **outerMarginRight**

The numeric value specifying page right margin.

### Example

```
%tscg_page(units=cm, width=8, height=12,  
outerMarginTop=0.3, outerMarginBottom=1,  
outerMarginLeft=0.7, outerMarginRight=0.4);
```

## TSCG\_PAGEBY

Use the PageBy Layout macro, `tscg_PageBy`, to specify page breaks according to values, levels, or categories in the specified column(s).

**Usage** `tscg_PageBy(pageVariable=);`

**Required** **pageVariable**

**Arguments** A space-delimited list of one or more variable names to be used for page breaking.

**Example** `%tscg_pageby(pageVariable=site);`

## TSCG\_THEME

Use the Themes macro, `tscg_Theme`, to apply a specific theme (colors and styles) to the plot.

**NOTE:** Only one of `userThemeName` or `sharedThemeName` should be specified. If both are specified, the shared theme is ignored.

### Usage

```
tscg_Theme(theme,  
            userThemeName=,  
            sharedThemeName=);
```

### Required Arguments

#### **theme**

The character name of the theme to apply to the graph. Options are:

- `BW`
- `Color`
- `Custom`
- `Default`
- `None`

Using `Default` or `None` ensures that the styling information contained in the IGD file being used to create the graph is applied.

### Optional Arguments

#### **userThemeName**

The name of an IGT file in the users default theme location on the server. This parameter is only applied if theme is set to `Custom`.

#### **sharedThemeName**

The name of an IGT file in the default Shared theme location on the server. This parameter is applied only if theme is set to `Custom`.



# TSCG\_TITLES

The Titles macro, `tscg_Titles`, sets the Title, Subtitle and Header. The footer is protected for auto generated TFL meta data content.

Each of the parameters can include special encoding to specify multiple lines of output. The only limit to the number of lines you can use is the practical limit of the amount of page real estate consumed. The following parameter value produces a title with four lines: text on lines 1, 2 and 4, with a blank third line:

```
%tscg_titles(mainTitle=1[Compound XYZ] 2[Study ABC] 4[Lab  
Data]);
```

## Usage

```
tscg_Titles(mainTitle=,  
            subTitle=,  
            header=,  
            footer = );
```

## Required Arguments

None.

## Optional Arguments

### mainTitle

A text string to use for the main title of the plot. See notes above about multi-line contents.

If this string contains special characters, the user will need to surround it with an appropriate SAS quoting function such as `%quote` or `%nrstr`.

### subTitle

A text string to use for the subtitle of the plot. See notes above about multi-line contents.

If this string contains special characters, the user will need to surround it with an appropriate SAS quoting function such as `%quote` or `%nrstr`.

### header

A text string to use for the plot header. See notes above about multi-line contents.

If this string contains special characters, the user will need to surround it with an appropriate SAS quoting function such as %quote or %nrstr.

**footer**

A text string to use for the plot footer. See notes above about multi-line contents.

If this string contains special characters, the user will need to surround it with an appropriate SAS quoting function such as %quote or %nrstr.

**Example**

```
%tscg_titles(mainTitle=1[main 1] 2[main 2] 4[main 4],
  subTitle=1[sub 1] 3[sub 3],
  header=2[head 2] 4[head 4],
  footer=1[foot 1] 3[foot 3] 5[foot 5]);

%tscg_PopulationPyramid(dataLibrary = Examples,
  dataSet = subjlist,
  outputFile=&outputPath.\TitlesOnPopPyr,
  outputFormat=png,
  valuesVariable = age,
  groupVariable = sex);
```

---

Introduction	23
tscg_AreaBarChart	24
tscg_AreaPlot	27
tscg_BarChart	30
tscg_BoxPlot	33
tscg_BubblePlot	35
tscg_CDFPlot	38
tscg_ContourPlot	40
tscg_CumulativeIncidence	42
tscg_DeltaPlot	44
tscg_DotPlot	47
tscg_DoubleDotPlot	51
tscg_DurationPlot	55
tscg_ForestPlot	58
tscg_Histogram	61
tscg_ImagePlot	64
tscg_KaplanMeierPlot	66
tscg_LinePlot	68
tscg_PopulationPyramid	72
tscg_QQMathPlot	75
tscg_QQPlot	77

<b>tscg_ScatterPlot</b>	<b>79</b>
<b>tscg_ScatterPlotMatrix</b>	<b>81</b>
<b>tscg_SurvivalCurve</b>	<b>83</b>
<b>tscg_TreeMap</b>	<b>85</b>

# INTRODUCTION

Each of the macros described in this section produces a specific type of plot. You can use their parameters to control the details of each plot.

**Note**

The parameters for these macros cannot provide as fine a level of control over the plot details as the TSCG client user interface. For instances where the plot macros do not allow the control you need, we recommend that you create the graph using the TSCG client and save the IGD file to use with the `tscg_CreateGraph` macro described in Chapter 2, Global and Layout Macros.

For a list of all the graph types and a brief description of their use, see the TSCG client Help. For the specific macros and their parameters, see the topics in this chapter.

**Note**

When you create a graph or a subplot, you must ensure the specified output target location exists before you try to write output to a specified location. The macros do not automatically create a destination folder.

# TSCG\_AREABARCHART

Creates a TIBCO Spotfire Clinical Graphics area bar chart using the specified IGD file as a source and writing the output to the named target.

**Usage**

```
tscg_AreaBarChart(dataLibrary =,  
  dataSet =,  
  outputFile =  
    &tscg_sourcePath.&tscg_dirSeparator.&tscg_progName.,  
  outputFormat=&tscg_outputFormat.,  
  xVariable =,  
  yVariable =,  
  groupVariable =,  
  trellisVariable =,  
  styleX=stacked,  
  styleY=stacked,  
  basegroupvalue=,  
  labelsVisibleFlag=False,  
  subgroupVariable=,  
  subgroupLabelsVisibleFlag=True,  
  varyGroup=Color);
```

**Required  
Arguments**

- dataLibrary**  
The name of SAS Library that includes the data.
- dataSet**  
The name of the dataset to be plotted.
- outputFile**  
Fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)
- outputFormat**  
The graph file format(s) for output.

Note
If an extension is specified on the outputFile parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the tscg_outputFormat global variable.

**Optional  
Arguments****xVariable**

The name of the variable to be plotted on the x axis.

**yVariable**

The name of the variable to be plotted on the y axis

**groupVariable**

The name of the group variable.

**trellisVariable**

The names of the Trellis variables, separated by spaces.

**styleX**

The name of the bar type: stacked (the default) or stackedpct bars.

**styleY**

The name of the bar type: stacked (the default) or stackedpct bars.

**baseGroupValue**

For grouped plots, allows the specification of a value to be used as the base. Can be numeric or character string.

**labelsVisibleFlag**

Specifies whether to include bar labels. TRUE to include bar labels; otherwise FALSE (the default).

**subgroupVariable**

A character string containing the name of the subgroup variable.

**subgroupLabelsVisibleFlag**

TRUE (the default) to include subgroup labels, if subgroups are provided; otherwise, FALSE.

**varyGroup**

Specifies the method for identifying groups in the plot through color and fill patterns. Choices include `none`, `color` (the default), `fill` or `both`.

**Example**

```
%tscg_areabarchart(dataLibrary = examples,  
  dataSet = quinidine_race_heart,  
  outputFile=&outputPath.\areabarchart,  
  outputFormat=png,  
  xVariable = SevNObs,  
  yVariable = RaceNObs,  
  groupVariable = Heart,  
  subgroupVariable = Race,  
  subgroupLabelsVisibleFlag=True,  
  varyGroup=Color)
```



# TSCG\_AREAPLOT

Creates a TIBCO Spotfire Clinical Graphics area plot and writes the output to the named target.

## Usage

```
tscg_AreaPlot(dataLibrary=,  
              dataSet=,  
              outputFile=&jobPath.\&tscg_progName.,  
              outputFormat=&tscg_outputFormat.,  
              xVariable=,  
              yVariable=,  
              groupVariable=,  
              trellisVariable=,  
              style=Stacked,  
              base=,  
              align=axis,  
              labelsVisibleFlag=False,  
              labelsVariable=,  
              labelsValues=,  
              varyGroup=Color);
```

## Required Arguments

### dataLibrary

The name of SAS Library that includes the data.

### dataSet

The name of the dataset to be plotted.

### outputFile

The fully-qualified name of the output graphic to be created.  
(Can be a path and file name or a URL.)

### outputFormat

The graph file format(s) for output.

## Note

If an extension is specified on the outputFile parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the tscg\_outputFormat global variable.

**xVariable**

The name of the variable to plot on the x axis.

**yVariable**

The name of the variable to plot on the y axis.

**Optional  
Arguments**

**groupVariable**

The name of the group variable.

**trellisVariable**

The name of the Trellis variables, blank separated.

**style**

The name of the type: stacked (the default) or stackedpct.

**base**

The data value to use as the alignment base of the graph.

**align**

Specify whether to align the area plot along the `axis` or a `level`.

**labelsVisibleFlag**

TRUE to include labels; otherwise FALSE (the default).

**labelsVariable**

The name of the data column to be used as the label.

**labelsValues**

The space-delimited list of label values to be used for the plot.

**varyGroup**

Method for identifying groups in the plot through color and fill patterns. Choices include `none`, `color` (the default), `fill` or `both`.

**Example**

```
%tscg_areaplot(dataLibrary = examples,  
  dataSet = areaplot,  
  outputFile=&outputPath.\areaplot,
```

```
outputFormat=png,  
xVariable = Visit,  
yVariable = Number_of_Patients,  
groupVariable = Dose_Group);
```

# TSCG\_BARCHART

Creates a TIBCO Spotfire Clinical Graphics bar chart and writes the output to the named target.

Usage

```
tscg_BarChart(dataLibrary=,
              dataSet=,
              outputFile=,
              outputFormat=,
              xVariable=,
              yVariable=,
              groupVariable=,
              trellisVariable=,
              style=clustered,
              base=,
              align=,
              labelsVisibleFlag=F,
              labelsVariable=,
              labelsValues=,
              varyGroup=color);
```

Required Arguments

- dataLibrary**  
The name of SAS Library that includes the data.
- dataSet**  
The name of the dataset to be plotted.
- outputFile**  
Fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)
- outputFormat**  
The graph file format(s) for output.

Note
If an extension is specified on the outputFile parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the tscg_outputFormat global variable.

## Optional Arguments

### **xVariable**

The name of the variable to plot on the x axis.

### **yVariable**

The name of the variable to plot on the y axis.

### **groupVariable**

The name of the group variable.

### **trellisVariable**

The names of the Trellis variables, blank separated.

### **style**

The name of the bar type: stacked (the default) or stackedpct bars.

### **base**

The data value use as the alignment base of the graph.

### **align**

Specify whether to align the area plot along the axis or a level.

### **labelsVisibleFlag**

TRUE to include bar labels; otherwise FALSE (the default).

### **labelsVariable**

The name of the data column to be used as the label.

### **labelsValues**

The space-delimited list of label values to be used for the plot.

### **varyGroup**

Method for identifying groups in the plot through color and fill patterns. Choices include none, color (the default), fill or both.

## Example

```
%tscg_barchart(dataLibrary = examples,  
  dataSet = likert_rspns,  
  outputFile=&outputPath.\barchart,
```

```
outputFormat=png,  
xVariable = count,  
yVariable = armed,  
groupVariable = rspns,  
style=stacked,  
base=Good,  
align=level);
```

# TSCG\_BOXPLOT

Creates a TIBCO Spotfire Clinical Graphics box plot and writes the output to the named target.

## Usage

```
tscg_BoxPlot(dataLibrary=,
  dataSet =,
  outputFile=,
  outputFormat =,
  xVariable =,
  yVariable =,
  groupVariable=,
  trellisVariable =,
  style=schematic,
  outliers=points,
  method=,
  varyGroup=color
```

## Required Arguments

### dataLibrary

The name of SAS Library that includes the data.

### dataSet

The name of the dataset to be plotted.

### outputFile

The fully-qualified name of the output graphic to be created.  
(Can be a path and file name or a URL.)

### outputFormat

The graph file format(s) for output.

## Note

If an extension is specified on the `outputFile` parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the `tscg_outputFormat` global variable.)

### xVariable

The name of the variable to plot on the x axis.

<b>Optional Arguments</b>	<b>yVariable</b>	The name of the variable to plot on the y axis.
	<b>groupVariable</b>	The name of the group variable.
	<b>method</b>	Specifies the quantile method to use. Choices include <code>sas1</code> , <code>sas2</code> , <code>sas3</code> , <code>sas4</code> , <code>sas5</code> and <code>splus</code> . The first five methods correspond to the five <code>PCTLDEF</code> methods of SAS and the last method is the default <code>S-PLUS</code> method.
	<b>outliers</b>	Method for displaying outlier observations. Can be <code>none</code> , <code>segments</code> , <code>points</code> (default) or <code>both</code> .
	<b>style</b>	The name of the box plot: <code>schematic</code> (the default) or <code>simple</code> .
	<b>trellisVariable</b>	The names of the Trellis variables, blank separated.
	<b>varyGroup</b>	Method for identifying groups in the plot through color and fill patterns. Choices include <code>none</code> , <code>color</code> (the default), <code>fill</code> or <code>both</code> .

<b>Example</b>	<pre>%tscg_BoxPlot(dataLibrary = examples,   dataSet = boxplot,   outputFile= &amp;outputPath.\boxplot,   outputFormat=png,   style=schematic,   xVariable = alb_norm,   yVariable = visit);</pre>
----------------	--



# TSCG\_BUBBLEPLOT

Creates a TIBCO Spotfire Clinical Graphics bubble plot using the specified IGD file as a source and writing the output to the named target.

Usage

```
tscg_BubblePlot(dataLibrary=,
  dataSet=,
  outputFile=,
  outputFormat =,
  xVariable=,
  yVariable=,
  groupVariable=,
  trellisVariable=,
  sizeVariable=,
  style=,
  minRadius=0.05,
  maxRadius=0.2);
labelsVisibleFlag=F,
labelsVariable=,
jitter=none
varyGroup=color);
```

Required  
Arguments

- dataLibrary**  
The name of the SAS Library containing the data.
- dataSet**  
The name of the dataset to be plotted.
- outputFile**  
The fully-qualified name of the output graphic to be created.  
(Can be a path and file name or a URL.)
- outputFormat**  
The graph file format(s) for output.

Note
If an extension is specified on the outputFile parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the tscg_outputFormat global variable.

**xVariable**

The name of the variable to be plotted on the x axis.

**yVariable**

The name of the variable to be plotted on the y axis

**sizeVariable**

The name of the data column which should be used to control the size of the bubbles.

**Optional  
Arguments**

**groupVariable**

The name of the group variable.

**trellisVariable**

The names of the Trellis variables. Separate with blank spaces.

**style**

The method for sizing bubbles. Can be area (the default) or radius.

**minRadius**

The minimum radius of a bubble using a positive real number. (The default value is 0.05; smaller numbers produce smaller bubbles).

**maxRadius**

The maximum radius of the bubble, specified as a positive real number. (The default value is 0.2; larger numbers produce larger bubbles but can cause bubbles to extend beyond plot axes. Values less than 1.0 are recommended.)

**labelsVisibleFlag**

Specifies whether the bubble labels are shown. TRUE to show; FALSE to hide (the default).

**labelsVariable**

The name of the data column to use to label the bubbles.

**jitter**

Used to indicate if bubbles should be plotted with slight, random offsets. Acceptable values are: none (default), x, y, or both

**varyGroup**

Method for identifying groups in the plot through color and fill patterns. Choices include none, color (the default), fill or both.

**Example**

```
%tscg_bubbleplot(dataLibrary = examples,  
  dataSet = ae_aedecode,  
  outputFile=&outputPath.\bubbleplot,  
  outputFormat=png,  
  xVariable=aersk_tx,  
  yVariable=aedecod,  
  sizeVariable=rr);
```

# TSCG\_CDFPLOT

Creates a TIBCO Spotfire Clinical Graphics empirical cumulative distribution function (CDF) plot and writes the output to the named target.

**Usage**

```
tscg_CDFPlot(dataLibrary=,  
             dataSet =,  
             outputFile = &jobPath.\&tscg_progName.,  
             outputFormat = &tscg_outputFormat.,  
             valuesVariable =,  
             groupVariable =,  
             trellisVariable =
```

**Required Arguments**

- dataLibrary**  
The name of the SAS Library containing the data.
- dataSet**  
The name of the dataset to be plotted.
- outputFile**  
The fully-qualified name of the output graphic to be created.  
(Can be a path and file name or a URL.)
- outputFormat**  
The graph file format(s) for output.

Note
If an extension is specified on the <code>outputFile</code> parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the <code>tscg_outputFormat</code> global variable.

- valuesVariable**  
The name of the variable containing the values whose distribution is plotted.

**Optional Arguments**

- groupVariable**  
The name of the group variable.

**trellisVariable**

The names of the Trellis variables. Separate with blank spaces.

**Example**

```
%tscg_CDFPlot(dataLibrary = examples,  
  dataSet = subjlist,  
  outputFile=&outputPath.\empiricalcdf,  
  outputFormat=png,  
  valuesVariable = age,  
  groupVariable = sex,  
  trellisVariable = armcd);
```

# TSCG\_CONTOURPLOT

Creates a TIBCO Spotfire Clinical Graphics contour plot and writes the output to the named target.

Usage

```
tscg_ContourPlot(dataLibrary=,  
  dataSet=,  
  outputFile=,  
  outputFormat=,  
  xVariable=,  
  yVariable=,  
  contourVariable=,  
  trellisVariable=,  
  contourBins=,  
  contourCutpoints=,  
  labelsVisibleFlag=True
```

Required Arguments

**dataLibrary**  
The name of SAS Library that includes the data.

**dataSet**  
The name of the dataset to be plotted.

**outputFile**  
The fully-qualified name of the output graphic to be created.  
(Can be a path and file name or a URL.)

**outputFormat**  
The graph file format(s) for output.

<b>Note</b>
If an extension is specified on the <code>outputFile</code> parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the <code>tscg_outputFormat</code> global variable.)

**xVariable**  
The name of the variable to plot on the x axis.

**yVariable**

The name of the variable to plot on the y axis.

**contourVariable**

The name of the data column containing the heights data (contour levels).

**Optional  
Arguments****trellisVariable**

The names of the Trellis variables. Separate with blank spaces.

**contourBins**

The number of levels into which the contourVariable should be divided for plotting.

**contourCutpoints**

A space-delimited list of cutpoints to be used for creating levels for plotting the contourVariable.

**labelsVisibleFlag**

Specifies if labels are shown. TRUE to show; FALSE to hide (default).

**Example**

```
%tscg_contourplot(dataLibrary = examples,  
  dataSet = prepostn,  
  outputFile=&outputPath.\contour,  
  outputFormat=png,  
  xVariable=ast_postmpre,  
  yVariable=bili_postmpre,  
  contourVariable=alt_postmpre,  
  trellisVariable=armcd);
```

# TSCG\_CUMULATIVEINCIDENCE

Creates a TIBCO Spotfire Clinical Graphics cumulative incidence plot and writes the output to the named target.

Usage

```
tscg_CumulativeIncidence(dataLibrary=,
  dataSet =,
  outputFile = &jobPath.\&tscg_progName., */
  outputFormat = &tscg_outputFormat., */
  xVariable =,
  groupVariable =,
  trellisVariable =,
  confidenceMethod=log,
  confidenceStyle=envelopes,
  confidenceValue=0.95,
  censorVariable=,
  censorMarksFlag=False);
```

Required Arguments

- dataLibrary**  
The name of SAS Library that includes the data.
- dataSet**  
The name of the dataset to be plotted.
- outputFile**  
The fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)
- outputFormat**  
The graph file format(s) for output.

Note

If an extension is specified on the `outputFile` parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the `tscg_outputFormat` global variable.)

- xVariable**  
The name of the variable to plot on the x axis.



**Optional  
Arguments****groupVariable**

The name of the group variable.

**trellisVariable**

The names of the Trellis variables. Separate with blank spaces.

**confidenceMethod**

The method for calculating confidence intervals. Valid choices are `none`, `plain`, `log` (the default), `loglog` and `se`.

**confidenceStyle**

Method for display of confidence intervals. Valid choices are `bands`, `bars`, `envelopes` (the default), `sticks` and `whiskers`.

**confidenceValue**

Size of confidence interval to compute (default is 0.95).

**censorVariable**

Variable indicating censoring versus events, where 1 represents an event and 0 represents censoring.

**censorMarksFlag**

TRUE to determine draw censor marks on the curve; otherwise FALSE. If no censor variable is provided, this setting is ignored.

**Examples**

```
%tscg_cumulativeincidence(dataLibrary = examples,  
  dataSet = kmsubjid,  
  outputFile=&outputPath.\cuminc_nocensor,  
  outputFormat=png,  
  xVariable = ttp);  
  
%tscg_cumulativeincidence(dataLibrary = examples,  
  dataSet = kmsubjid,  
  outputFile=&outputPath.\cuminc_censor,  
  outputFormat=png,  
  xVariable = ttp,  
  censorVariable = ttpcnsr);
```

# TSCG\_DELTAPLOT

Creates a TIBCO Spotfire Clinical Graphics delta plot and writes the output to the named target.

Usage

```
tscg_DeltaPlot(dataLibrary=,
               dataSet=,
               outputFile=&jobPath.\&tscg_progName.,
               outputFormat=&tscg_outputFormat.,
               valuesVariable=,
               groupVariable=,
               trellisVariable=,
               horizontalFlag=True,
               interceptReferenceFlag=True,
               baseValue=0,
               varyGroupFlag=True,
               sortAscendingFlag=True,
               labelsVisibleFlag=False);
```

Required Arguments

- dataLibrary**  
The name of SAS Library that includes the data.
- dataSet**  
The name of the dataset to be plotted.
- outputFile**  
The fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)
- outputFormat**  
The graph file format(s) for output.

Note
If an extension is specified on the <code>outputFile</code> parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the <code>tscg_outputFormat</code> global variable.)

- valuesVariable**  
The name of the variable to be plotted on the x axis.

## Optional Arguments

### **groupVariable**

The name of the group variable.

### **trellisVariable**

The name of the Trellis variables. Separate with blank spaces.

### **horizontalFlag**

TRUE (the default) sets a horizontal plot orientation; FALSE sets a vertical orientation.

### **interceptReferenceFlag**

TRUE (the default) to draw a reference line at the appropriate axis intercept. FALSE omits the reference line.

### **baseValue**

Data value to use as the alignment base of the graph.

### **varyGroupFlag**

TRUE (the default) to vary the color and fill by group. TRUE uses different styles for each group; FALSE does not.

### **sortAscendingFlag**

Specifies the sorted order for the plot. TRUE (the default) results in ascending order sort; FALSE results in descending order sort.

### **labelsVisibleFlag**

TRUE to include bar labels; otherwise FALSE.

## Examples

```
%tscg_deltaplot(dataLibrary = examples,
  dataSet = prepostn,
  outputFile=&outputPath.\delta_horiz,
  outputFormat=png,
  valuesVariable =ast_postmpre,
  groupVariable = armcd);

%tscg_deltaplot(dataLibrary = examples,
  dataSet = prepostn,
  outputFile=&outputPath.\delta_vert,
  outputFormat=png,
  valuesVariable =ast_postmpre,
```

```
groupVariable = armcd,  
horizontalFlag=False,  
interceptReferenceFlag=False);
```

# TSCG\_DOTPLOT

Creates a TIBCO Spotfire Clinical Graphics dot plot and writes the output to the named target.

## Usage

```
tscg_DotPlot(dataLibrary=,
  dataSet =,
  outputFile = &jobPath.\&tscg_progName.,
  outputFormat = &tscg_outputFormat.,
  valuesVariable=,
  categoriesVariable=,
  groupVariable=,
  trellisVariable=,
  dotsJitter=none,
  varyGroupFlag=True,
  intervalStyle=none,
  intervalXLowerVariable=,
  intervalXUpperVariable=,
  intervalYLowerVariable=,
  intervalYUpperVariable=,
  intervalJitter=none,
  confidenceMethod=none,
  confidenceStyle=whiskers,
  confidenceValue=0.95,
  confidenceJitterFlag=False);
```

## Required Arguments

### dataLibrary

The name of SAS Library that includes the data.

### dataSet

The name of the dataset to be plotted.

### outputFile

The fully-qualified name of the output graphic to be created.  
(Can be a path and file name or a URL.)

**outputFormat**

The graph file format(s) for output.

Note
If an extension is specified on the <code>outputFile</code> parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the <code>tscg_outputFormat</code> global variable.

**valuesVariable**

Name of column containing values to plot on horizontal axis.

**categoriesVariable**

Name of column containing categories to plot on vertical axis.

**Optional  
Arguments**

**groupVariable**

The name of the group variable.

**trellisVariable**

The name of the Trellis variables. Separate with blank spaces.

**dotsJitter**

Used to indicate if the dots should be plotted with slight, random offsets. Acceptable values are: none (default), x, y or both.

**varyGroupFlag**

Specifies whether to vary color and fill by group. TRUE (the default) uses different styles for each group; FALSE does not.

**intervalStyle**

Specifies the method for displaying interval information (from specified data data columns; not confidence intervals calculated by TSCG).

Valid choices are none (the default), bands, bars, envelopes, sticks, or whiskers.

**intervalXLowerVariable**

The name of the column containing the lower limit value in a horizontal direction.

**intervalXUpperVariable**

The name of the column containing the upper limit value in a horizontal direction.

**intervalYLowerVariable**

The name of the column containing the lower limit value in a vertical direction.

**intervalYUpperVariable**

The name of the column containing the upper limit value in a vertical direction.

**intervalJitter**

Indicates if the intervals should be plotted with slight, random offsets. Acceptable values are none (the default), x, y or both.

**confidenceMethod**

The method for calculating confidence intervals. Valid choices are none (the default), meanttest, meansd, meanboot, meanquantiles, or medianquantiles.

**confidenceStyle**

The method for displaying confidence intervals. Valid choices are bands, bars, envelopes, sticks, or whiskers (the default).

**confidenceValue**

The size of the confidence interval to compute. The default is 0.95.

**confidenceJitterFlag**

Indicates if confidence intervals should be plotted with slight, random offsets. TRUE jitters the intervals. FALSE (the default) does not jitter.

**Example**

```
%tscg_DotPlot(dataLibrary = examples,  
  dataSet = patientprofile,  
  outputFile=&outputPath.\dotplot,  
  outputFormat=png,  
  valuesVariable=%nrstr(or),  
  categoriesVariable=aedecod,  
  intervalStyle=whiskers,  
  intervalXLowerVariable=or95l1,  
  intervalXUpperVariable=or95u1);
```



# TSCG\_DOUBLEDOTPLOT

Creates a TIBCO Spotfire Clinical Graphics double dot plot and writes the output to the named target.

## Usage

```
tscg_DoubleDotPlot(dataLibrary=,  
  dataSet =,  
  outputFile = &jobPath.\&tscg_progName.,  
  outputFormat = &tscg_outputFormat.,  
  xVariable1=,  
  xVariable2=,  
  yVariable=,  
  groupVariable=,  
  pointsJitter=None,  
  pointsVaryFlag=True,  
  pointsSubsetVariable=,  
  intervalStyle=Whiskers,  
  intervalXLowerVariable=,  
  intervalXUpperVariable=,  
  intervalYLowerVariable=,  
  intervalYUpperVariable=,  
  intervalJitter=None,  
  intervalVaryFlag=True,  
  groupReferenceFlag=True,  
  categoryReferenceFlag=True);
```

## Required Arguments

### **dataLibrary**

The name of SAS Library that includes the data.

### **dataSet**

The name of the dataset to be plotted.

### **outputFile**

The fully-qualified name of the output graphic to be created.  
(Can be a path and file name or a URL.)

**outputFormat**

The graph file format(s) for output.

Note
If an extension is specified on the <code>outputFile</code> parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the <code>tscg_outputFormat</code> global variable.

**xVariable1**

Variable to plot on the X-axis in the first (left) panel.

**xVariable2**

Variable to plot on the X-axis in the second (right) panel.

**yVariable**

Variable to plot on the Y-axis in both panels.

**Optional  
Arguments**

**groupVariable**

The name of the group variable.

**pointsJitter**

Indicates if points are plotted with slight, random offsets.  
Acceptable values are: none (the default), x, y or both.

**pointsVaryFlag**

TRUE (the default) indicates that the style for points should vary based on the corresponding group value.

**pointsSubsetVariable**

Contains 0s and 1s to indicating which points should be omitted and which should be plotted.

**intervalStyle**

The method for displaying interval information (from specified data data columns; not confidence intervals calculated by TSCG).

Valid choices are none, bands, bars, envelopes, sticks or whiskers (the default).

**intervalXLowerVariable**

The name of the column containing the lower limit value in a horizontal direction.

**intervalXUpperVariable**

The name of the column containing the upper limit value in a horizontal direction.

**intervalYLowerVariable**

The name of the column containing the lower limit value in a vertical direction.

**intervalYUpperVariable**

The name of the column containing the upper limit value in a vertical direction.

**intervalJitter**

Indicates if the intervals are plotted with slight, random offsets. Acceptable values are none (the default) x, y or both.

**intervalVaryFlag**

TRUE varies the interval style by group using different styles for each group; FALSE does not.

**groupReferenceFlag**

TRUE (the default) indicates that the reference lines should be drawn at the group levels.

**categoryReferenceFlag**

TRUE (the default) indicates that the reference lines should be drawn between categories.

**Example**

```
%tscg_DoubleDotPlot(dataLibrary = examples,  
  dataSet = doubledot,  
  outputFile=&outputPath.\doubledot,  
  outputFormat=png,  
  xVariable1=pct,
```

```
xVariable2=rr,  
yVariable=pt,  
groupVariable=trt,  
intervalStyle=Whiskers,  
intervalXLowerVariable=rrlower,  
intervalXUpperVariable=rrupper);
```

# TSCG\_DURATIONPLOT

Creates a TIBCO Spotfire Clinical Graphics duration plot and writes the output to the named target.

Usage

```
tscg_DurationPlot(dataLibrary=,
  dataSet =,
  outputFile = &jobPath.\&tscg_progName.,
  outputFormat = &tscg_outputFormat.,
  xVariable=,
  yVariable=,
  groupVariable=,
  trellisVariable=,
  durationVariable=,
  durationType=start,
  eventTypeVariable=,
  periodMinimumVariable=,
  periodMaximumVariable=,
  shadowLinesFlag=True,
  unknownStart=Both,
  unknownEnd=Both,
  durationVaryFlag=True);
```

Required Arguments

- dataLibrary**  
The name of SAS Library that includes the data.
- dataSet**  
The name of the dataset to be plotted.
- outputFile**  
The fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)
- outputFormat**  
The graph file format(s) for output.

Note
If an extension is specified on the outputFile parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the tscg_outputFormat global variable.

**xVariable**

Variable to be plotted on the X-axis.

**yVariable**

Categorical variable to be plotted on the Y-axis.

**Optional  
Arguments**

**groupVariable**

The name of the group variable.

**trellisVariable**

The name of the Trellis variables. Separate with blank spaces.

**durationVariable**

Name of a numeric duration column to specify the length of the duration lines.

**durationType**

If set to `Start` (the default), the X values represent the start or left-hand side of a duration; if set to `End`, the X values represent the end or right-hand end of a duration.

**eventTypeVariable**

The name of a column that specifies whether a given point is a "point" (that is, an event), or a "line" (that is, the duration).

**periodMinimumVariable**

The name of a column containing values determining the period start time for each level of y.

**periodMaximumVariable**

The name of a column containing values determining the period end time for each level of y.

**shadowLinesFlag**

`TRUE` (the default) displays shadow lines. A shadow line is a segment for range of the duration for each level of y.

**unknownStart**

Determines the action when the start value is unknown. Valid choices include:

- none (displays no symbol at the start of the duration line).
- extend (extends the start to the period start).
- arrow (displays an arrow).
- both (The default; both extends and includes an arrow).

**unknownEnd**

Determines action when the end value is unknown. Valid choices include:

- none (displays no symbol at the end of the duration line).
- extend (extend the end to the period end).
- arrow (displays an arrow).
- both (The default; both extends and includes an arrow).

**durationVaryFlag**

TRUE (the default) varies the plot style by group; FALSE does not.

**Example**

```
%tscg_DurationPlot(dataLibrary = examples,
  dataSet = Ae_aedecod_duration,
  outputFile=&outputPath.\duration,
  outputFormat=png,
  xVariable=start,
  yVariable=aedecod,
  groupVariable=aebodsys,
  durationVariable=end,
  eventTypeVariable=type,
  unknownStart=both,
  unknownEnd=both);
```

# TSCG\_FORESTPLOT

Creates a TIBCO Spotfire Clinical Graphics forest plot and writes the output to the named target.

Usage

```
tscg_ForestPlot(dataLibrary=,
  dataSet =,
  outputFile = &jobPath.\&tscg_progName.,
  outputFormat = &tscg_outputFormat.,
  valuesVariable =,
  categoriesVariable =,
  groupVariable =,
  combinedValue =,
  weighttype = none,
  weightVariable =,
  sampleSizeVariable =,
  lowerBoundVariable =,
  upperBoundVariable =,
  varyGroupFlag = True);
```

Required Arguments

- dataLibrary**  
The name of SAS Library that includes the data.
- dataSet**  
The name of the dataset to be plotted.
- outputFile**  
The fully-qualified name of the output graphic to be created.  
(Can be a path and file name or a URL.)
- outputFormat**  
The graph file format(s) for output.

Note
If an extension is specified on the outputFile parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the tscg_outputFormat global variable.



**Optional  
Arguments****valuesVariable**

The name of the column containing values to plot on the horizontal axis.

**categoriesVariable**

The name of the column containing categories to plot on the vertical axis.

**groupVariable**

The name of the group variable.

**combinedValue**

The value of the categories column representing a combined or summary value.

**weighttype**

Type of weighting to use for scaling the relative sizes of plot symbols. Valid choices are none (default), standarderror or sample size.

**weightVariable**

The name of the column containing weight values when standard error or sample size are selected as the weight type.

**sampleSizeVariable**

The name of the column containing sample size information. Providing a value for this parameter includes the sample size data on the plot. Without this parameter, no sample size information is displayed.

**lowerBoundVariable**

The name of the column containing the lower limit value used to draw whiskers for individual categories and diamond width for the combined estimate.

**upperBoundVariable**

The name of the column containing the upper limit value used to draw whiskers for individual categories and diamond width for the combined estimate.

**varyGroupFlag**

Specifies whether to vary color and fill by group. TRUE (the default) uses different styles for each group; FALSE does not.

**Example**

```
%tscg_ForestPlot(dataLibrary = examples,  
  dataSet = forest_src,  
  outputFile=&outputPath.\forest,  
  outputFormat=png,  
  valuesVariable = %nrstr(or),  
  categoriesVariable = study,  
  combinedValue = Summary,  
  weighttype = SampleSize,  
  weightVariable = Weight,  
  lowerBoundVariable = CILower,  
  upperBoundVariable = CIUpper);
```

# TSCG\_HISTOGRAM

Creates a TIBCO Spotfire Clinical Graphics histogram and writes the output to the named target.

## Usage

```
tscg_Histogram(dataLibrary = ,
  dataSet =,
  outputFile =
    &tscg_sourcePath.&tscg_dirSeparator.&tscg_progName.,
  outputFormat = &tscg_outputFormat.,
  xVariable =,
  groupVariable =,
  trellisVariable =,
  histogramStyle = Frequency,
  dataType = Continuous,
  numberOfBarsMethod = Specified,
  numberOfBars = 20,
  barLabelsVisibleFlag = False,
  densityEstimateMethod = Kernel,
  kernelType = Gaussian,
  gaussianBandwidth = Sheatherjones,
  varyGroup = Color,
  includeNormDist = );
```

## Required Arguments

### **dataLibrary**

The name of SAS Library that includes the data.

### **dataSet**

The name of the dataset to be plotted.

### **outputFile**

The fully-qualified name of the output graphic to be created.  
(Can be a path and file name or a URL.)

**outputFormat**

The graph file format(s) for output.

**Note**

If an extension is specified on the `outputFile` parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the `tscg_outputFormat` global variable.

**xVariable**

The name of the variable to be plotted on the x-axis.

**Optional  
Arguments**

**groupVariable**

The name of the group variable.

**trellisVariable**

The name of the Trellis variables. Separate with blank spaces.

**histogramStyle**

Specifies whether to generate a counts, density, frequency (the default), or percent histogram.

**dataType**

Specifies whether values should be treated as `continuous` (the default) or `categorical`. With `categorical` values, a separate bar is displayed for each unique value, with the bar height determined by the number of occurrences of the value.

**numberOfBarsMethod**

With continuous data, specify the number of bars or estimate the number of bars to use, based on the data. When number of bars are estimated, choose `FreedmanDiaconis`, `Scott` or `Sturges` rule to determine the number of bins. The Default is `Specified`.

**numberOfBars**

The number of bars to use when the `numberOfBarsMethod` is set to `Specified`. The default value is 20.

**barLabelsVisibleFlag**

If TRUE, each bar is labeled with its size. The default is FALSE.

**densityEstimateMethod**

The method for calculating a density estimate. Valid choices are Estimate (the default), Normal, or Poisson.

**kernelType**

Specifies the kernel type. Valid choices are Cosine, Gaussian (the default), Rectangular, or Triangular.

**gaussianBandwidth**

For a kernel density estimate, valid choices are sheatherjones (the default), histogrambin, normalrefdensity, biasedcrossref or unbiasedcrossref.

**varyGroup**

The method for identifying groups in the plot through color and fill patterns. Choices include none, color, fill, or both.

**includeNormDist**

If TRUE, indicate if normal distribution line should be overlaid on the plot. The default is FALSE.

**Example**

```
%tscg_Histogram(dataLibrary=examples,  
  dataSet = subjlist,  
  outputFile=&outputPath.\histogram,  
  outputFormat=png,  
  xVariable = agegrp,  
  histogramStyle = Density,  
  dataType = Categorical);
```

# TSCG\_IMAGEPLOT

Creates a TIBCO Spotfire Clinical Graphics image plot and writes the output to the named target.

Usage

```
tscg_imagePlot(dataLibrary=,  
               dataSet=,  
               outputFile=,  
               outputFormat=,  
               xVariable=,  
               yVariable=,  
               contourVariable=,  
               trellisVariable=,  
               contourBins=,  
               contourCutpoints=,  
               labelsVisibleFlag=True);
```

Required Arguments

dataLibrary

The name of SAS Library that includes the data.

dataSet

The name of the dataset to be plotted.

outputFile

The fully-qualified name of the output graphic to be created.  
(Can be a path and file name or a URL.)

outputFormat

The graph file format(s) for output.

Note

If an extension is specified on the `outputFile` parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the `tscg_outputFormat` global variable.)

xVariable

The name of the variable to be plotted on the x axis.

**yVariable**

The name of the variable to be plotted on the y axis.

**contourVariable**

The name of the data column containing the heights data (contour levels).

**Optional  
Arguments****trellisVariable**

The name of the Trellis variables. Separate with blank spaces.

**contourBins**

The number of levels into which the `contourVariable` should be divided for plotting.

**contourCutpoints**

A space-delimited list of cutpoints used for creating levels for plotting the `contourVariable`.

**labelsVisibleFlag**

TRUE shows contour labels; the default is FALSE.

**Example**

```
%tscg_ImagePlot(dataLibrary = examples,  
  dataSet = prepostn,  
  outputFile=&outputPath.\imageplot,  
  outputFormat=png,  
  xVariable=ast_postmpre,  
  yVariable=bili_postmpre,  
  contourVariable=alt_postmpre,  
  trellisVariable=armcd);
```

# TSCG\_KAPLANMEIERPLOT

Creates a TIBCO Spotfire Clinical Graphics Kaplan-Meier plot and writes the output to the named target.

Usage

```
tscg_KaplanMeierPlot(dataLibrary=,
  dataSet=,
  outputFile=,
  outputFormat=,
  timeVariable=,
  statusVariable=,
  groupVariable=,
  trellisVariable=,
  confidenceMethod=Log,
  confidenceStyle=Envelopes,
  confidenceValue=0.95,
  censorFlag=True,
  varyGroupFlag=True);
```

Required Arguments

- dataLibrary**  
The name of SAS Library that includes the data.
- dataSet**  
The name of the dataset to be plotted.
- outputFile**  
The fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)
- outputFormat**  
The graph file format(s) for output.

Note
If an extension is specified on the <code>outputFile</code> parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the <code>tscg_outputFormat</code> global variable.)

- timeVariable**  
The name of the variable to plot on the x axis.



**Optional  
Arguments****statusVariable**

The name of the variable which to plot on the y axis.

**groupVariable**

The name of the group variable.

**trellisVariable**

The name of the Trellis variables. Separate with blank spaces.

**confidenceMethod**

The method for calculation of confidence intervals. Valid choices are none, plain, log (the default), log-log and stderr.

**confidenceStyle**

The method for displaying confidence intervals. Valid choices are bands, bars, envelopes (default), sticks and whiskers.

**confidenceValue**

The size of the confidence interval to compute. (The default is 0.95.)

**censorFlag**

Determines if censor marks are drawn on the curve. Valid choices are True (default) or False.

**varyGroupFlag**

TRUE (the default) to vary the style by group. TRUE uses different styles for each group; FALSE does not.

**Example**

```
%tscg_KaplanMeierPlot(dataLibrary=examples,  
  dataSet=kmsubjid,  
  outputFile=&outputPath.\kaplanmeier,  
  outputFormat=png,  
  timeVariable=ttp,  
  statusVariable=ttpcnsr);
```

## TSCG\_LINEPLOT

Creates a TIBCO Spotfire Clinical Graphics line plot and writes the output to the named target.

### Usage

```
tscg_LinePlot(dataLibrary=,  
  dataSet =,  
  outputFile = &jobPath.\&tscg_progName.,  
  outputFormat = &tscg_outputFormat.,  
  xVariable =,  
  yVariable =,  
  groupVariable =,  
  trellisVariable =,  
  style=Lines,  
  summaryType=None,  
  sortFlag=False,  
  jitter=None,  
  subsetVariable=,  
  labelsVisibleFlag=True,  
  labelsValues=,  
  varyGroupFlag=True,  
  intervalStyle=None,  
  intervalLowerVariable=,  
  intervalUpperVariable=,  
  intervalJitter=None,  
  confidenceStyle=,  
  confidenceMethod=,  
  confidenceValue=,  
  confidenceJitterFlag=);
```

### Required Arguments

#### **dataLibrary**

The name of SAS Library that includes the data.

#### **dataSet**

The name of the dataset to be plotted.

#### **outputFile**

The fully-qualified name of the output graphic to be created.  
(Can be a path and file name or a URL.)

**outputFormat**

The graph file format(s) for output.

**Note**

If an extension is specified on the `outputFile` parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the `tscg_outputFormat` global variable.

**xVariable**

The name of the variable to plot on the x axis.

**yVariable**

The name of the variable to plot on the y axis.

**Optional  
Arguments****groupVariable**

The name of the group variable.

**trellisVariable**

The name of the Trellis variables. Separate with blank spaces.

**style**

Specify how to draw the line plot. Valid choices are `lines` (default), `points`, `both`, `overlaid`, `isolated`, `stepleft`, `stepright`, `xcandle` or `ycandle`.

**summaryType**

Specify the type of data summary to use when plotting. Valid choices are `none` (default), `mean`, `median`, `meanXatY` or `medianXatY`.

**sortFlag**

Specify `TRUE` to sort the values on X. The default is `FALSE`.

**jitter**

Specify whether to jitter the lines. Valid values are `none` (the default), `x`, `y` or `both`.

**subsetVariable**

Variable containing 0s and 1s to indicate which points should be omitted and which should be plotted.

**labelsVisibleFlag**

TRUE to include labels; otherwise FALSE (the default).

**labelsValues**

A space-delimited list of values to be used as labels.

**varyGroupFlag**

TRUE (the default) to vary the color and fill by group. TRUE uses different styles for each group; FALSE does not.

**intervalStyle**

The method for displaying interval information (from specified data data columns; not confidence intervals calculated by TSCG). Valid choices are none (the default), band, bar, envelope, stick, or whisker.

**intervalLowerVariable**

The name of the column containing the lower limit value in the vertical direction.

**intervalUpperVariable**

The name of the column containing the upper limit value in the vertical direction.

**intervalJitter**

Used to indicate if the intervals should be plotted with slight, random offsets. Acceptable values are none (the default), x, y, or both.

**confidenceStyle**

The method for displaying confidence intervals. Valid choices are bands, bars, envelopes, sticks, or whiskers (the default).

**confidenceMethod**

The method for calculating confidence intervals. Valid choices are none (the default), meanttest, meansd, meanboot, meanquantiles or medianquantiles.

**confidenceValue**

The size of the confidence interval to compute. (The default is 0.95.)

**confidenceJitterFlag**

Used to indicate if confidence intervals should be plotted with slight, random offsets. TRUE jitters the intervals. FALSE (the default) does not jitter.

**Example**

```
%tscg_LinePlot(dataLibrary=examples,  
  dataSet = bilivisit_visitnum,  
  outputFile=&outputPath.\lineplot,  
  outputFormat=png,  
  xVariable = visit,  
  yVariable = biliavg,  
  intervalStyle=Envelope,  
  intervalLowerVariable=BILILO,  
  intervalUpperVariable=BILIHl,  
  intervalJitter=None);
```

# TSCG\_POPULATIONPYRAMID

Creates a TIBCO Spotfire Clinical Graphics population pyramid plot and writes the output to the named target.

Usage

```
tscg_PopulationPyramid(dataLibrary=,
  dataSet =,
  outputFile =
    &tscg_sourcePath.&tscg_dirSeparator.&tscg_progName.,*/
  outputFormat = &tscg_outputFormat.,
  valuesVariable =,
  groupVariable =,
  trellisVariable =,
  horizontalFlag = True,
  histogramStyle = Frequency,
  dataType = Continuous,
  numberOfBarsMethod = Specified,
  numberOfBars = 20,
  barLabelsVisibleFlag = False,
  varyGroup = Color,
  includeNormDist = );
```

Required Arguments

- dataLibrary**  
The name of SAS Library that includes the data.
- dataSet**  
The name of the dataset to be plotted.
- outputFile**  
The fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)
- outputFormat**  
The graph file format(s) for output.

Note
If an extension is specified on the outputFile parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the tscg_outputFormat global variable.

**Optional  
Arguments****valuesVariable**

The name of the variable containing the values whose distribution is plotted.

**groupVariable**

The name of the group variable.

**trellisVariable**

The name of the Trellis variables. Separate with blank spaces.

**horizontalFlag**

Determines the orientation of the resulting plot. `TRUE` (the default) results in a horizontal orientation. `FALSE` results in a vertical orientation.

**histogramStyle**

Specifies whether to generate a counts, density, frequency (the default), or percent histogram.

**dataType**

Specifies whether values should be treated as continuous (the default) or categorical. With categorical values, a separate bar is displayed for each unique value, with the bar height determined by the number of occurrences of the value. The default is continuous.

**numberOfBarsMethod**

With continuous data, specify the number of bars or estimate the number of bars to use, based on the data. When the number of bars is estimated, choose the `FreedmanDiaconis`, `Scott` or `Sturges` rule to determine the number of bins. The default is `Specified`.

**numberOfBars**

The number of bars to use when the `numberOfBarsMethod` is set to `Specified`. The default value is 20.

**barLabelsVisibleFlag**

Determines if each bar is labeled with its size. The default is FALSE.

**varyGroup**

The method for identifying groups in the plot through color and fill patterns. Choices include none, color, fill, or both.

**includeNormDist**

If TRUE, indicate if normal distribution line should be overlaid on the plot. The default is FALSE.

**Examples**

```
%tscg_PopulationPyramid(dataLibrary = Examples,  
  dataSet = subjlist,  
  outputFile=&outputPath.\populationpyramid_horiz,  
  outputFormat=png,  
  valuesVariable = age,  
  groupVariable = sex);  
  
%tscg_PopulationPyramid(dataLibrary = Examples,  
  dataSet = subjlist,  
  outputFile=&outputPath.\populationpyramid_vert,  
  outputFormat=png,  
  valuesVariable = age,  
  groupVariable = sex,  
  horizontalFlag = false);
```



# TSCG\_QQMATHPLOT

Creates a TIBCO Spotfire Clinical Graphics qqmath plot and writes the output to the named target.

## Usage

```
tscg_QQMathPlot(dataLibrary=,
  dataSet =,
  outputFile =
    &tscg_sourcePath.&tscg_dirSeparator.&tscg_progName.,
  outputFormat=&tscg_outputFormat.,
  valuesVariable =,
  groupVariable =,
  trellisVariable =,
  distribution=Normal,
  refLinesVisible=,
  refLinesXSpan=);
```

## Required Arguments

### dataLibrary

The name of SAS Library that includes the data.

### dataSet

The name of the dataset to be plotted.

### outputFile

The fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)

### outputFormat

The graph file format(s) for output.

## Note

If an extension is specified on the `outputFile` parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the `tscg_outputFormat` global variable.)

### valuesVariable

The name of the variable to plot on the x axis.

**Optional  
Arguments**

**distribution**

The name of the distribution to be plotted on the Y-axis. Valid choices are `normal` (the default) and `poisson`.

**groupVariable**

The name of the group variable.

**refLinesVisible**

A flag to indicate if reference lines should be visible. `True` results in reference lines being displayed; `False` results reference lines being omitted.

**refLinesXSpan**

Flag that determines the prediction base. If `True`, the span of `X` within each group is used. If `False`, the whole X-axis range is used.

**trellisVariable**

The name of the Trellis variables. Separate with blank spaces.

**Example**

```
%tscg_QQMathPlot(dataLibrary = Examples,  
  dataSet = subjlist,  
  outputFile=&outputPath.\qqmathplot,  
  outputFormat=png,  
  valuesVariable = age,  
  groupVariable= sex,  
  trellisVariable = armcd,  
  distribution=Normal);
```

# TSCG\_QQPLOT

Creates a TIBCO Spotfire Clinical Graphics qq comparison plot and writes the output to the named target.

## Usage

```
tscg_QQPlot(dataLibrary=,
  dataSet =,
  outputFile =
    &tscg_sourcePath.&tscg_dirSeparator.&tscg_progName.,
  outputFormat = &tscg_outputFormat.,
  xVariable =,
  yVariable =,
  groupVariable =,
  trellisVariable =);
```

## Required Arguments

### dataLibrary

The name of SAS Library that includes the data.

### dataSet

The name of the dataset to be plotted.

### outputFile

The fully-qualified name of the output graphic to be created.  
(Can be a path and file name or a URL.)

### outputFormat

The graph file format(s) for output.

## Note

If an extension is specified on the `outputFile` parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the `tscg_outputFormat` global variable.)

### xVariable

The name of the variable to plot on the x axis.

### yVariable

The name of the variable to plot on the y axis.

**Optional  
Arguments**

**groupVariable**

The name of the group variable.

**trellisVariable**

The name of the Trellis variables. Separate with blank spaces.

**Example**

```
%tscg_QQPlot(dataLibrary = Examples,  
  dataSet = prepostn,  
  outputFile=&outputPath.\qqplot,  
  outputFormat=png,  
  xVariable = ALB_PRE,  
  yVariable = ALB_POST,  
  trellisVariable = ARMCD);
```

# TSCG\_SCATTERPLOT

Creates a TIBCO Spotfire Clinical Graphics scatter plot and writes the output to the named target.

Usage

```
tscg_ScatterPlot(dataLibrary=,
  dataSet=,
  outputFile=,
  outputFormat=,
  xVariable=,
  yVariable=,
  groupVariable=,
  trellisVariable=,
  jitter=None,
  subsetVariable=,
  varyGroupFlag=True);
```

Required Arguments

- dataLibrary**  
The name of SAS Library that includes the data.
- dataSet**  
The name of the dataset to be plotted.
- outputFile**  
The fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)
- outputFormat**  
The graph file format(s) for output.

Note

If an extension is specified on the `outputFile` parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the `tscg_outputFormat` global variable.

- xVariable**  
The name of the variable to use plot on the x-axis.

**yVariable**

The name of the variable to use plot on the y-axis.

**Optional  
Arguments**

**groupVariable**

The name of the group variable.

**trellisVariable**

The name of the trellis variables, blank separated.

**jitter**

Used to indicate if plot symbols should be plotted with slight, random offsets. Acceptable values are: none (default), x, y, or both.

**subsetVariable**

Variable containing 0s and 1s to indicate which points should be omitted and which should be plotted.

**varyFlag**

Specifies whether to vary style by group. TRUE (the default) uses different styles for each group; FALSE does not.

**Example**

```
%tscg_ScatterPlot(dataLibrary = Examples,  
  dataSet = prepostlabs,  
  outputFile=&outputPath.\scatterplot,  
  outputFormat=png,  
  xVariable = labpre,  
  yVariable = labpost,  
  groupVariable = armcd,  
  trellisVariable = lbtestcd);
```

# TSCG\_SCATTERPLOTMATRIX

Creates a TIBCO Spotfire Clinical Graphics matrix of scatter plots and writes the output to the named target.

## Usage

```
tscg_ScatterPlotMatrix(dataLibrary=,
  dataSet=,
  outputFile=,
  outputFormat=,
  variables=,
  groupVariable=,
  jitter=None,
  subsetVariable=,
  varyFlag=True);
```

## Required Arguments

### dataLibrary

The name of SAS Library that includes the data.

### dataSet

The name of the dataset to be plotted.

### outputFile

The fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)

### outputFormat

The graph file format(s) for output.

## Note

If an extension is specified on the `outputFile` parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the `tscg_outputFormat` global variable.)

### variables

Space-delimited list of variables to use for creating the matrix of scatter plots.

**Optional  
Arguments**

**groupVariable**

The name of the group variable.

**jitter**

Used to indicate if plot symbols should be plotted with slight, random offsets. Acceptable values are: none (default), x, y, or both.

**subsetVariable**

Variable containing 0s and 1s to indicate which points should be omitted and which should be plotted.

**varyFlag**

Specifies whether to vary style by group. TRUE (the default) uses different styles for each group; FALSE does not.

**Example**

```
%tscg_ScatterPlotMatrix(dataLibrary=examples,  
    dataSet = prepostn,  
    outputFile=&outputPath.\splom,  
    outputFormat=png,  
    variables=ALP_POST ALT_POST AST_POST BILI_POST,  
    groupVariable=,  
    jitter=None,  
    subsetVariable=,  
    varyFlag=True);
```



# TSCG\_SURVIVALCURVE

Creates a TIBCO Spotfire Clinical Graphics survival curve and writes the output to the named target.

Usage

```
tscg_SurvivalCurve(dataLibrary=,
  dataSet=,
  outputFile=,
  outputFormat=,
  xVariable=,
  yVariable=,
  groupVariable=,
  trellisVariable=,
  intervalStyle=none,
  intervalLowerVariable=,
  intervalUpperVariable=,
  censorVariable=);
```

Required Arguments

- dataLibrary**  
The name of SAS Library that includes the data.
- dataSet**  
The name of the dataset to be plotted.
- outputFile**  
The fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)
- outputFormat**  
The graph file format(s) for output.

Note
If an extension is specified on the <code>outputFile</code> parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the <code>tscg_outputFormat</code> global variable.

- xVariable**  
The name of the variable to plot on the x axis.

<b>Optional Arguments</b>	<b>yVariable</b>	The name of the variable to plot on the y axis.
	<b>groupVariable</b>	The name of the group variable.
	<b>trellisVariable</b>	The name of the Trellis variables. Separate with blank spaces.
	<b>intervalStyle</b>	The method for displaying interval information (from specified data data columns; not confidence intervals calculated by TSCG). Valid choices are none, bands, bars, envelopes (default), sticks, or whiskers.
	<b>intervalLowerVariable</b>	The name of the column containing the lower limit value of the horizontal interval.
	<b>intervalUpperVariable</b>	The name of the column containing the upper limit value of the horizontal interval.
	<b>censorVariable</b>	The name of the column containing the censoring status.

<b>Example</b>	<pre>%tscg_SurvivalCurve(dataLibrary=examples,   dataSet=kkest_subjid,   outputFile=&amp;outputPath.\survival,   outputFormat=png,   xVariable=ttp,   yVariable=survival,   intervalLowerVariable=sdf_lcl,   intervalUpperVariable=sdf_ucl);</pre>
----------------	--

# TSCG\_TREEMAP

Creates a TIBCO Spotfire Clinical Graphics tree map plot and writes the output to the named target.

## Usage

```
tscg_TreeMap(dataLibrary=,
  dataSet =,
  outputFile =
    &tscg_sourcePath.&tscg_dirSeparator.&tscg_progName.,
  outputFormat = &tscg_outputFormat.,
  xVariable =,
  groupVariable =,
  trellisVariable =,
  aspectThreshold=1,
  colorLevelsVariable=,
  colorLevels=,
  groupLabelsVisibleFlag=T,
  subgroupVariable=,
  subgroupLabelsVisibleFlag=T,
  sortSubgroupFlag=T,
  subgroupLabelVariable=);
```

## Required Arguments

### dataLibrary

The name of SAS Library that includes the data.

### dataSet

The name of the dataset to be plotted.

### outputFile

The fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)

### outputFormat

The graph file format(s) for output.

## Note

If an extension is specified on the `outputFile` parameter, unique elements of this argument are added. (Can be one of EPS, JPG, PDF, PNG, WMF; the default format is determined by the `tscg_outputFormat` global variable.)

**Optional  
Arguments**

**xVariable**

The name of the variable to plot on the x axis.

**groupVariable**

The name of the group variable.

**trellisVariable**

The name of the Trellis variables. Separate with blank spaces.

**aspectThreshold**

The aspect-ratio threshold for the tree map algorithm. The default is 1. Numeric values must be positive.

**colorLevelsVariable**

The column specifying color shades. Should be a numeric or factor column. For Trellis, the color relation is the same across panels.

**colorLevels**

Sets the number of colors used for computing the color scale.

**groupLabelsVisibleFlag**

Include group labels, if groups are specified.

**subgroupVariable**

Specify the subgroups column name in the data set. The data should contain a factor variable nested within the groups; however, every level is used and a numeric column is automatically cast to a factor.

**subgroupLabelsVisibleFlag**

Include subgroup labels, if subgroups are specified.

**sortSubgroupFlag**

If TRUE, subgroups are sorted by area. Default is FALSE.

**subgroupLabelVariable**

Specifies a data column to use as subgroup labels.

**Example**

```
%tscg_TreeMap(dataLibrary = Examples,  
  dataSet = barley,  
  outputFile=&outputPath.\treemap,  
  outputFormat=png,  
  xVariable = yield,  
  groupVariable = variety,  
  trellisVariable = year,  
  colorLevelsVariable=yield,  
  subgroupVariable=site);
```



# REFERENCE ELEMENTS

# 4

---

Introduction	90
tscg_LSFit	92
tscg_LTSRobFit	93
tscg_MMRobFit	94
tscg_PolyLSFit	95
tscg_RefABLines	96
tscg_RefArrow	98
tscg_RefCurve	101
tscg_RefPoints	103
tscg_RefText	105
tscg_RefXLines	107
tscg_RefYLines	110

## INTRODUCTION

Use reference elements macros to overlay various items onto an existing graph. For example, you can add arrows and text to explain something displayed in a plot.

You can use reference elements to add items that are not part of a particular graph type. For example, you can display a curve on a bar graph, which either adds additional information to the underlying plot or represents some summary of the plotted data. The reference element types include the following:

**Table 4.1:** *Reference elements.*

Element	Macro name	Description
Least-Squares Fit Line	tscg_LSFit	Adds a least-squares fit line to the subsequent plot.
Robust Least-Trimmed Squares Fit	tscg_LTSRobFit	Adds a robust least-trimmed squares fit to the subsequent plot.
Robust MM Fit	tscg_MMRobFit	Adds a robust MM fit to the subsequent plot.
Polynomial Least-Squares Fit Curve	tscg_PolyLSFit	Adds a polynomial least-squares fit curve to the subsequent plot.
AB lines	tscg_RefABLines	A reference line using intercept and slope values. The default is the identity line ( $y = x$ ).



**Table 4.1:** *Reference elements. (Continued)*

Element	Macro name	Description
Reference arrows	tscg_RefArrow	An arrow for specific values. The arrows can be parallel to x or y (in which case you must provide a start and end x or y value) or in any direction, which requires two x and two y values.
Reference curves	tscg_RefCurve	A fitting curve for specific values using a linear model.
Reference points	tscg_RefPoints	Points at specific positions.
Reference text	tscg_RefText	Text-Labels at specific positions.
X-axis reference line	tscg_RefXLines	A vertical line at a specific x-value.
Y-axis reference line	tscg_RefYLines	A horizontal line at a specific y-value.

## TSCG\_LSFIT

The `tscg_LSFit` macro adds a least-squares fit line to the subsequent plot.

### Usage

```
tscg_LSFit(groupFlag=True,  
           xSpanFlag=True);
```

### Required Arguments

None

### Optional Arguments

#### **groupFlag**

Specifies whether to include a separate line for each level of grouping variables specified in the plot. The default is `TRUE`.

#### **xSpanFlag**

Specifies whether the fit line should be limited to the data range (`TRUE`; the default) or span the entire X axis (`FALSE`).

### Example

```
%tscg_LSFit(groupFlag=False, xSpanFlag=False);  
  
%tscg_ScatterPlot(dataLibrary=examples,  
                  dataSet = fuel,  
                  outputFile=&outputPath.\lsfitXSpanF,  
                  outputFormat=png,  
                  xVariable = mileage,  
                  yVariable = disp_,  
                  groupVariable = type);
```

# TSCG\_LTSROBFIT

The `tscg_LTSRobFit` macro adds a robust least-trimmed squares fit to the subsequent plot.

## Usage

```
tscg_LTSRobFit(groupFlag=True,  
               xSpanFlag=True);
```

## Required Arguments

None

## Optional Arguments

### groupFlag

Specifies whether to include a separate line for each level of grouping variables specified in the plot. The default is `TRUE`.

### xSpanFlag

Specifies whether the fit line should be limited to the data range (`TRUE`; the default) or span the entire X axis (`FALSE`).

## Example

```
%tscg_LTSRobFit(groupFlag=False, xSpanFlag=False);  
  
%tscg_ScatterPlot(dataLibrary=examples,  
                  dataSet = fuel,  
                  outputFile=&outputPath.\lsfitXSpanF,  
                  outputFormat=png,  
                  xVariable = mileage,  
                  yVariable = disp_,  
                  groupVariable = type);
```

## TSCG\_MMROBFIT

The `tscg_MMRobFit` macro adds a robust MM fit to the subsequent plot.

### Usage

```
tscg_MMRobFit(groupFlag=True,  
              xSpanFlag=True);
```

### Required Arguments

None

### Optional Arguments

#### **groupFlag**

Specifies whether to include a separate line for each level of grouping variables specified in the plot. The default is `TRUE`.

#### **xSpanFlag**

Specifies whether the fit line should be limited to the data range (`TRUE`; the default) or span the entire X axis (`FALSE`).

### Example

```
%tscg_MMRobFit(groupFlag=False, xSpanFlag=False);  
  
%tscg_ScatterPlot(dataLibrary=examples,  
                  dataSet = fuel,  
                  outputFile=&outputPath.\lsfitXSpanF,  
                  outputFormat=png,  
                  xVariable = mileage,  
                  yVariable = disp_,  
                  groupVariable = type);
```

# TSCG\_POLYLSFIT

The `tscg_PolyLSFit` macro adds a polynomial least-squares fit curve to the subsequent plot.

## Usage

```
tscg_PolyLSFit(degree=2,  
groupFlag=True,  
xSpanFlag=True);
```

## Required Arguments

None

## Optional Arguments

### degree

Specifies the degree of the polynomial fit line. The default is 2.

### groupFlag

Specifies whether to include a separate line for each level of grouping variables specified in the plot. The default is `TRUE`.

### xSpanFlag

Specifies whether the fit line should be limited to the data range (`TRUE`; the default) or span the entire X axis (`FALSE`).

## Example

```
%tscg_PolyLSFit(degree=2, groupFlag=False,  
xSpanFlag=False);  
  
%tscg_ScatterPlot(dataLibrary=examples,  
dataSet = fuel,  
outputFile=&outputPath.\lsfitXSpanF,  
outputFormat=png,  
xVariable = mileage,  
yVariable = disp_,  
groupVariable = type);
```

## TSCG\_REFABLINES

The `tscg_RefABLines` macro adds a reference line using intercept and slope values. The default is the identity line ( $y = x$ ).

### Usage

```
tscg_RefABLines(A=0,  
                B=1,  
                Label=,  
                LabelPos = 0.5,  
                LabelSide=Bottom,  
                LabelRotation = 0);
```

### Required Arguments

None

### Optional Arguments

#### A

Numeric value for intercept (A) as in  $y = A + B x$ . Default is 0.

#### B

Numeric value for slope (B) as in  $y = A + B x$ . Default is 1.

#### Label

Text string to be used to label the reference line. Use string fields `[intercept]` and `[slope]` for replacement by A and B, respectively.

#### LabelPos

Numeric relative position along the line between 0 and 1, where 0 is start, 0.5 (default) is center and 1 is end.

#### LabelSide

Options for label position (relative position, then position on line). Specify TOP, BOTTOM (the default), and CENTER.

#### LabelRotation

Degrees of label string rotation measured counter clockwise relative to the line. The default of 0 corresponds to text aligned with the reference line)

**Example**

```
%tscg_RefAbLines(A=-1, B=110,  
  Label=1[refABline] 3[line 3], LabelPos = 1,  
  LabelSide=Center, LabelRotation = 0);  
  
%tscg_LinePlot(dataLibrary=examples,  
  dataSet = fuel,  
  outputFile=&outputPath.\refABline,  
  outputFormat=png,  
  xVariable = mileage,  
  yVariable = weight,  
  groupVariable = type);
```

## TSCG\_REFARROW

Reference Arrows macro `tscg_RefArrow` plots an arrow for the specific values. The arrows can be parallel to x- or y (in which case a start and end x or y value has to be provided) or in any direction, which requires two x and two y values. It is possible label the arrow. Note that a reference segment can be obtained by disabling both the start and the end arrow heads.

### Usage

```
tscg_RefArrow(XStartPositions=.25,  
             YStartPositions=.90,  
             XEndPositions=.75,  
             YEndPositions=.90,  
  
             UseXAxisUnits=FALSE,  
             UseYAxisUnits=FALSE,  
  
             StartHeadVisible=FALSE,  
  
             StartHeadSize = 14,  
  
             StartHeadOpen=FALSE,  
  
             EndHeadVisible=TRUE,  
             EndHeadSize = 14,  
             EndHeadOpen=FALSE,  
  
             Label=,  
             LabelSide = CENTER,  
             LabelOuter = TRUE);
```

### Required Arguments

None

### Optional Arguments

#### **XStartPositions and XEndPositions**

Numeric values giving the x-axis start and end position(s) respectively. Note that single coordinates will be replicated to double, meaning that giving `XStartPositions` only will result in a vertical arrow with `start=end`.



**YStartPositions and YEndPositions**

Numeric values giving the y-axis start and end position(s) respectively. Note that single coordinates will be replicated to double, meaning that giving YStartPositions only will result in a horizontal arrow with start=end.

**UseXAxisUnits and UseYAxisUnits**

Sets the units of which the start and end position refers to. The default is FALSE. We refer to relative units in terms of the figure proportion. For example, a number between 0 to 1, where 0 is lower and 1 is upper.

**StartHeadVisible or EndHeadVisible**

Sets visibility of start and end head (<- and ->) at. The default (->) start is off (FALSE) and end is on (TRUE).

**StartHeadSize or EndHeadSize**

A numeric value indicating the size of the start or end head symbol in units of font size (1 pt = 1/72 inch). The default is 14.

**StartHeadOpen or EndHeadOpen**

Sets whether the start or end arrow heads should be open or filled. The default is FALSE.

**Label**

A string containing the description of the reference arrow.

**LabelSide**

The options for the label position (relative position, then position on line). Specify START, END, CENTER (the default).

**LabelOuter**

A logical value to indicate if the label is positioned over the arrow or outside of the arrow. The default of TRUE places the label beyond the ends of the arrow.

**Example**

```
%tscg_RefArrow(XStartPositions=.7,  
  YStartPositions=.70,  
  XEndPositions=.85,
```

```
YEndPositions=.90,  
UseXAxisUnits=FALSE,  
UseYAxisUnits=FALSE,  
StartHeadVisible=FALSE,  
StartHeadSize = 14,  
StartHeadOpen=FALSE,  
EndHeadVisible=TRUE,  
EndHeadSize = 18,  
EndHeadOpen=false,  
Label=arrow1,  
LabelSide = CENTER,  
LabelOuter = TRUE);  
  
%tscg_scatterplot(dataLibrary = examples,  
    dataSet = fuel,  
    outputFile=&outputPath.\refArrow1,  
    outputFormat=png,  
    xVariable = Mileage,  
    yVariable = Weight,  
    groupVariable = Type);
```

# TSCG\_REFCURVE

The Reference Curve macro `tscg_RefCurve` plots a fitting curve for specific values using a linear model. The x- and y- values are data columns.

## Usage

```
tscg_RefCurve(XVariable=,
              YVariable=,
              Label=,
              LabelPositionMethod = end,
              LabelSide=top,
              LabelRotation=0);
```

## Required Arguments

None.

## Optional Arguments

### XVariable

The variable name giving the x-axis position(s)

### YVariable

The variable name giving the y-axis position(s)

### Label

A string giving the description of the reference curve.

### LabelPositionMethod

The label position. Specify end (the default), start, min, max, XHalfRange, XMean, YHalfRange, or YMean.

### LabelSide

The options for label position (relative position, then position on line). Specify left, right, top, or bottom.

### LabelRotation

A numeric specifying the option to set the rotation. (The default 90 corresponds to a vertical text along the line.)

## Example

```
%tscg_RefCurve(XVariable=mileage, YVariable=weight,
               Label=reference curve, LabelPositionMethod = ymean,
               LabelSide=bottom, LabelRotation=315);
```

```
%tscg_LinePlot(dataLibrary=examples,  
    dataSet = fuel,  
    outputFile=&outputPath.\refCurve,  
    outputFormat=png,  
    xVariable = mileage,  
    yVariable = weight,  
    groupVariable = type);
```

## TSCG\_REFPOINTS

The Reference Points Element macro `tscg_RefPoints` plots points at specific positions. The x- and y- values can be numeric values or data columns. It is also possible to label the points.

### Usage

```
tscg_RefPoints(XPositions=,
               YPositions=,
               XVariable=,
               YVariable=,
               Labels=,
               LabelsVariable=,
               LabelSide=Center,
               LabelRotation=0);
```

### Required Arguments

#### XPositions

Numeric values specifying the x-axis position(s).

#### YPositions

Numeric values specifying the y-axis position(s).

#### XVariable

A numeric variable name giving the x-axis position(s).

#### YVariable

A numeric variable name giving the y-axis position(s).

### Optional Arguments

#### Labels

The description of the reference line.

#### Note

You must separate multiple labels with commas, and you must enclose the comma-separated list in a SAS %str call. (Note that labels can not contain commas.)

#### LabelsVariable

The name of a variable containing text to be used for labels.

### **LabelSide**

Sets the horizontal adjustment relative to the point. Specify one of the following:

- Center (Text is centered on the point).
- Top (Text is top center of the point).
- Bottom (Text is below centered on the point location).
- Left (Text is left of the point).
- Right (Text is right of the point).

### **LabelRotation**

The numeric option to set the rotation. (The default 0 corresponds to a horizontal text parallel to the x-axis).

### **Example**

```
%tscg_RefPoints(XPositions=19 24 31,  
  YPositions=189 100 281,  
  Labels=%str(one, two, three),  
  LabelSide=Center,  
  LabelRotation=290);  
  
%tscg_LinePlot(dataLibrary=examples,  
  dataSet = fuel,  
  outputFile=&outputPath.\refpoints,  
  outputFormat=png,  
  xVariable = mileage,  
  yVariable = disp_,  
  groupVariable = type);
```

## TSCG\_REFTEXT

The Reference Text Element macro `tscg_RefText` plots Text-Labels at specific positions. The x- and y- values can be numeric values or data columns.

### Usage

```
tscg_RefText(XPositions=,
             YPositions=,
             Labels=,
             XVariable=,
             YVariable=,
             LabelVariable=,
             LabelSide=Center,
             LabelRotation=0);
```

### Required Arguments

#### XPositions

The numeric values giving the x-axis position.

**NOTE:** The x-axis position *must* be supplied by either the `XPositions` or the `XVariable` parameter.

#### YPositions

The numeric values giving the y-axis position.

**NOTE:** The y-axis position *must* be supplied by either the `YPositions` or the `YVariable` parameter.

#### Labels

A text string for labels.

#### Note

You must separate multiple labels with commas, and you must enclose the comma-separated list in a SAS %str call. (Note that labels can not contain commas.)

#### XVariable

The numeric variable name giving the x-axis position.

**NOTE:** The x-axis position *must* be supplied by either the `XPositions` or the `XVariable` parameter.

### **YVariable**

The numeric variable name giving the y-axis position.

**NOTE:** The y-axis position *must* be supplied by either the YPositions or the YVariable parameter.

### **LabelsVariable**

The character or numeric variable name giving the text label *must* be supplied when using YVariable or the XVariable parameter.

### **Optional Arguments**

#### **LabelSide**

Character string that sets the horizontal adjustment relative to the x,y position. Options include Center, Top, Bottom, Left and Right.

#### **LabelRotation**

The option to set the text rotation. (The default 90 corresponds to horizontal text.)

### **Example**

```
%tscg_RefText(XVariable=mileage,  
             YVariable=disp_,  
             LabelVariable=mileage,  
             LabelSide=top,  
             LabelRotation=340);  
  
%tscg_LinePlot(dataLibrary=examples,  
              dataSet = fuel,  
              outputFile=&outputPath.\reftext,  
              outputFormat=png,  
              xVariable = mileage,  
              yVariable = disp_,  
              groupVariable = type);
```



## TSCG\_REFXLINES

The X-Reference Line macro `tscg_RefXLines` plots a vertical line at a specific x-value. The x-value can be a single numeric value, a unique data column or a calculated data column value (that is, mean or max). This line can be labeled and the position of the label can be specified.

### Usage

```
tscg_RefXLines(Positions=,
  Variable= ,
  PositionsLabel= ,
  PositionsLabelSide= right,
  PositionsLabelPositon= 1,
  PositionsLabelRotation= 90,
  VariableStats = none,
  VariableStatsFactor = 2,
  VariableStatsLabels = ,
  VariableStatsLabelSide = right,
  VariableStatsLabelPosition= 1,
  VariableStatsLabelRotation = 90,
  XSpan=True);
```

### Required Arguments

#### Positions

The numeric value giving the x-axis position of the vertical line.

#### Variable

Numeric variable name giving the x-axis position of the vertical line or to compute positions.

### Optional Arguments

#### PositionsLabel

The text string describing the positioned lines. You can use the string parameter `[Value]` to insert the position value dynamically.

#### PositionsLabelSide / VariableStatsLabelSide

The options for the label side, relative to the line: LEFT, CENTER, RIGHT.

#### PositionsLabelPositon / VariableStatsLabelPosition

The vertical relative position between 0 and 1, where 0 is start, 0.5 is center, and 1 is end. The default is 1.

### **PositionsLabelRotation / VariableStatsLabelRotation**

Option to set the text rotation (The default 90 corresponds to a vertical text direction parallel to the line).

### **VariableStats**

The methods to use. Options include:

- NONE (the default), which corresponds to no action.
- UNIQUE, which inserts at unique values of the variable.
- MIN inserts at the minimum value of the variable.
- MAX inserts at the maximum value of the variable.
- MEAN inserts at the mean value of the variable.
- MEDIAN inserts at the median value of the variable.
- SD inserts at the  $\text{MEAN} \pm \text{VariableStatsFactor} * \text{SD}$ , where SD is the Standard Deviation.
- SEM inserts at the  $\text{MEAN} \pm \text{VariableStatsFactor} * \text{SEM}$ , where SEM is the Standard Error of Mean.
- QUANTILE inserts at the minimum, lower quartile, median, upper quartile, and maximum of the variable.
- LSFIT inserts least squares.
- POLYLSFIT inserts a polynomial fit of  $\text{VariableStatsFactor}$  degrees.
- LTSROBFIT inserts a robust LTS fit.
- MMROBFIT inserts a robust MM fit.

### **VariableStatsFactor**

Factor to be passed to `variableStat` method be used with SD and SEM methods. Default is 2 and should be strictly positive.

### **VariableStatsLabels**

Text string describing the computed lines. The string parameter `[Value]` can be used to insert the position value dynamically For SD and SEM the string parameters, you can use `[sign]` and `[factor]` to insert the sign and factor dynamically (as `[sign][factor]SD` will give  $-2SD$  and  $+2SD$  for the two lines).

### **XSpan**

Can be used when `VariableStats` is set to LSFIT, POLYLSFIT, LTSROBFIT, or MMROBFIT.

Determines what is used as the prediction base. If TRUE (the default), the span of X within each group is used. If FALSE, the whole X-axis range is used.

**Example**

```
%tscg_RefXLines(Variable= mileage,  
  VariableStats = QUANTILE,  
  VariableStatsFactor = 2,  
  VariableStatsLabels = [Value],  
  VariableStatsLabelSide = right,  
  VariableStatsLabelPosition= 1,  
  VariableStatsLabelRotation = 90);  
  
%tscg_LinePlot(dataLibrary=examples,  
  dataSet = fuel,  
  outputFile=&outputPath.\refXline,  
  outputFormat=png,  
  xVariable = mileage,  
  yVariable = disp_,  
  groupVariable = type);
```

## TSCG\_REFYLINES

The Y-Reference Line macro `tscg_RefYLines` plots a horizontal line at a specific y-value. The y-value can be a single numeric value, a unique data column, or a calculated data column value (that is, mean or max). This line can be labeled and the position of the label can be specified.

### Usage

```
tscg_RefYLines(Positions=,
  Variable= ,
  PositionsLabel= ,
  PositionsLabelSide= top,
  PositionsLabelPosition= 1,
  PositionsLabelRotation= 0,
  VariableStats = none,
  VariableStatsFactor = 2,
  VariableStatsLabels = ,
  VariableStatsLabelSide = top,
  VariableStatsLabelPosition= 1,
  VariableStatsLabelRotation = 0);
```

### Required Arguments

#### Positions

The numeric value giving the y-axis position of the horizontal line.

#### Variable

The numeric variable name giving the y-axis position of the horizontal line or to compute positions.

### Optional Arguments

#### PositionsLabel

Text string describing the positioned lines. The string parameter `[Value]` can be used to insert the position value dynamically.

#### PositionsLabelSide / VariableStatsLabelSide

The options for the label side, relative to the line. Specify TOP, CENTER, or BOTTOM.

**PositionsLabelPosition / VariableStatsLabelPosition**

The numeric horizontal relative position between 0 and 1, where 0 is the start, 0.5 is center, and 1 is end. The default is 1.

**PositionsLabelRotation / VariableStatsLabelRotation**

The numeric option to set the text rotation. (The default of 0 corresponds to a horizontal text direction parallel to the line).

**VariableStats**

A character string specifying the name of the methods to use. Options include:

- NONE (the default) corresponding to no action.
- UNIQUE inserts at unique values of the variable.
- MIN inserts at the minimum value of the variable.
- MAX inserts at the maximum value of the variable.
- MEAN inserts at the mean value of the variable.
- MEDIAN inserts at the median value of the variable.
- SD inserts at the  $\text{MEAN} \pm \text{VariableStatsFactor} * \text{SD}$ , where SD is the Standard Deviation.
- SEM inserts at the  $\text{MEAN} \pm \text{VariableStatsFactor} * \text{SEM}$ , where SEM is the Standard Error of Mean.
- QUANTILE inserts at the minimum, lower quartile, median, upper quartile, and maximum of the variable.

**VariableStatsFactor**

Factor to be passed to `variableStat` method be used with SD and SEM methods. The default is 2 and should be strictly positive.

**VariableStatsLabels**

Text string giving the description of the computed lines. The string parameter `[Value]` can be used to insert the position value dynamically. For SD and SEM, you can use the string parameters `[sign]` and `[factor]` to insert the sign and factor dynamically (as `[sign][factor]SD` will give  $-2SD$  and  $+2SD$  for the two lines)

**Example**

```
%tscg_RefYLines(Positions=2000 3000 4000,  
    PositionsLabel= %str([value],[value],[value]),  
    PositionsLabelSide= top,  
    PositionsLabelPosition= 0.5,  
    PositionsLabelRotation= 10);  
  
%tscg_LinePlot(dataLibrary=examples,  
    dataSet = fuel,  
    outputFile=&outputPath.\refYline,  
    outputFormat=png,  
    xVariable = mileage,  
    yVariable = weight,  
    groupVariable = type);
```

---

Introduction	115
tscg_AddGraphParm	116
tscg_Configuration	117
tscg_DeleteGraphParm	118
tscg_DeleteParmSet	119
tscg_FindPath	120
tscg_GetProgName	121
tscg_GetSourcePath	122
tscg_GraphOutput	123
tscg_HttpPost	124
tscg_HttpRqst	125
tscg_InsertGraphODS	126
tscg_InsertGraphParm	127
tscg_InsertSPJName	128
tscg_InsertSPLUSLog	129
tscg_InsertSPLUSScript	130
tscg_MakeList	131
tscg_Parse	132
tscg_ParseBody	133
tscg_Sort	134
tscg_StartSubPlot	136

<b>tscg_StripTag</b>	<b>137</b>
<b>tscg_SubPlot</b>	<b>138</b>
<b>tscg_TestBoolean</b>	<b>140</b>
<b>tscg_Validate</b>	<b>141</b>
<b>tscg_Wait</b>	<b>142</b>
<b>tscg_WriteGraphJob</b>	<b>143</b>



## **INTRODUCTION**

The utility macros are not intended to be called directly by user code. Rather, these macros are called by the other macros to accomplish their tasks. They are documented as a reference to assist you in debugging or extending the macro library.

The utility macros handle the tasks needed to prepare a request for the TSCG server, submit the request to the server, and then process the items returned.

## TSCG\_ADDGRAPHPARAM

The macro `tscg_AddGraphParm` is used to request a run-time parameter override. This macro works by calling `tscg_InsertGraphParm` to insert a row into a temporary dataset that is used to create the graph job specification file.

### Usage

```
tscg_AddGraphParm(name,  
                  value,  
                  parmSet=&tscg_ParmSet.);
```

### Required Arguments

#### **name**

The name of the run-time parameter value.

#### **value**

The run-time parameter value.

### Optional Arguments

#### **parmSet**

The name of the dataset to hold the run-time parameter names and values. The default value is provided by the global macro variable `tscg_ParmSet`, which is set in the `tscg_Configuration` macro.

### Example

For an example of using `tscg_AddGraphParm`, see `tscg_CreateGraph`.

## TSCG\_CONFIGURATION

The `tscg_Configuration` macro configures the default values needed for the TIBCO Spotfire Clinical Graphics SAS integration package.

### Usage

```
tscg_Configuration(configXML=tscg_configuration.xml);
```

### Required Arguments

None.

### Optional Arguments

#### **configXML**

An optional parameter to use when the the name of the configuration file is changed from the default of **tscg\_configuration.xml**. This argument should contain the file name with the extension.

## TSCG\_DELETEGRAPHPARM

The parameter deletion macro, `tscg_DeleteGraphParm`, is used to request a run-time parameter deletion. This macro works by calling `tscg_insertgraphparm` to insert a row into a temporary SAS dataset that is then used to create the graph job specification file.

### Usage

```
tscg_DeleteGraphParm(name,  
    parmSet=&tscg_ParmSet.);
```

### Required Arguments

`name`

Character string containing the name of run-time parameter value.

### Optional Arguments

`parmSet`

Character string containing the name of dataset to hold the run-time parameter names and values. The default value is provided by the global macro variable `tscg_ParmSet` that is set in the `tscg_Configuration` macro.

## TSCG\_DELETEPARMSET

Use the `tscg_DeleteParmSet` to configure the default values needed for the TIBCO Spotfire Clinical Graphics SAS integration package.

**NOTE:** This macro is called by `tscg_WriteGraphJob`. Users do not need to invoke this macro directly.

### Usage

```
tscg_DeleteParmSet(parmSet=&tscg_ParmSet.);
```

### Required Arguments

#### **parmSet**

Character string containing the name of dataset to hold run-time parameter names and values. Default value is provided by the global macro variable `tscg_ParmSet` that is set in the `tscg_configuration` macro.

### Optional Arguments

None.

## TSCG\_FINDPATH

Finds the path to the specified target.

### Usage

```
tscg_FindPath(target);
```

### Required

**target**

### Arguments

The target item to find the path for.

## TSCG\_GETPROGNAME

The utility macro `tscg_GetProgName` Returns the current SAS program or process name based on the value of the SAS automatic macro variable `SYSPROCESSNAME`.

**Usage** `tscg_GetProgName;`

**Required Arguments** None

**Optional Arguments** None.

## TSCG\_GETSOURCEPATH

The source path macro `tscg_GetSourcePath` uses the automatic SAS macro variable `SYSIN` to return the path of the currently executing job.

This macro is called by `tscg_CreateGraph`. Users do not need to invoke this macro directly.

<b>Usage</b>	<code>tscg_GetSourcePath;</code>
--------------	----------------------------------

<b>Required Arguments</b>	None.
---------------------------	-------

<b>Optional Arguments</b>	None.
---------------------------	-------



## TSCG\_GRAPHOUTPUT

The Graph Output macro `tscg_GraphOutput` provides the ability disable or enable all graphics output generation.

### Usage

```
tscg_GraphOutput(generateGraphs=T);
```

### Required Arguments

None.

### Optional Arguments

#### **generateGraphs**

Default TRUE. Sets a global parameter that determines if graphics should be generated (TRUE) or not (FALSE).

## TSCG\_HTTPPOST

The `tscg_HttpPost` macro makes a generic HTTP POST request of a server using the SAS `FILENAME SOCKET` method.

**NOTE:** This macro is called by `tscg_interact`. Users do not need to invoke this macro directly.

### Usage

```
tscg_HttpPost(server,  
              service,  
              content,  
              lengthOffset=7,  
              dsname=&tscg_http.);
```

### Required Arguments

#### **server**

Address of the server where the request is submitted.

#### **service**

The "service" employ. This is expressed as a path to the appropriate "handler" on the server. For TSCG, typically this is the function or expression API.

#### **content**

The content of the HTTP POST request sent to the server.

### Optional Arguments

#### **lengthOffset**

Many requests require the POST content length to be shorter than the results of the SAS `length` function. This parameter allows for that adjustment to be made easily. The default of 7 works for all of the macro library calls *except* for calling the FileService API `restlet`, where no offset is needed.

#### **dsname**

The name of the dataset containing the HTTP response. (The default is the value of the global macro variable `tscg_http`.)

## TSCG\_HTTPRQST

The `tscg_HttpRqst` macro makes an HTTP GET request using the SAS FILENAME URL method.

**NOTE:** Calls to this macro typically are given in the application macros. Users do not need to invoke this macro directly.

### Usage

```
tscg_HttpRqst(httpstring=,  
              dsname=&tscg_http.);
```

### Required Arguments

#### **httpstring**

An HTTP request to be submitted using HTTP GET.

### Optional Arguments

#### **dsname**

A character string containing the name of the dataset containing the HTTP response. (The default is the value of the global macro variable `tscg_http`.)

## TSCG\_INSERTGRAPHODS

The `tscg_InsertGraphODS` macro inserts a Spotfire S+ Graph into SAS's ODS.

### Usage

```
tscg_InsertGraphODS(splusGraphFileName =,  
text = 'The S-PLUS System');
```

### Required Arguments

#### **splusGraphFileName**

The path and name of the Spotfire S+ graphics file.

### Optional Arguments

#### **text**

Must be 'The S-PLUS System'.

# TSCG\_INSERTGRAPHPARM

The `tscg_InsertGraphParm` macro is used to request a run-time parameter override. This macro works by inserting a row into a temporary SAS dataset that is then used to create the graph job specification file.

## Usage

```
tscg_InsertGraphParm(action,
    name,
    value,
    parmSet=&tscg_ParmSet.);
```

## Required Arguments

### action

A character string containing the action to be applied. Can be either update (add/replace) or delete.

### name

The name of run-time parameter value.

### value

A run-time parameter value.

## Optional Arguments

### parmSet

The name of dataset to hold run-time parameter names and values. The default value is provided by the global macro variable `tscg_ParmSet` that is set in the `tscg_Configuration` macro.

## TSCG\_INSERTSPJNAME

The `tscg_InsertSPJName` macro is used to insert the name of an SPJ file in a subplot.

### Usage

```
tscg_InsertSPJName(SPJName=,  
SPJList=&tscg_SPJList.);
```

### Required Arguments

#### SPJName

The fully-qualified name of an SPJ file (including extension) to be included in a subplot.

### Optional Arguments

#### SPJList

The name of a SAS dataset to hold the list of the SPJ files to include in the subplot. The default value is defined in the configuration macro.

## TSCG\_INSERTSPLUSLOG

The `tscg_InsertSPLUSLog` macro adds a Spotfire S+ verbose log to the SAS log file.

**NOTE:** Calls to this macro are typically given in the application macros. Users do not need to invoke this macro directly.

### Usage

```
tscg_InsertSPLUSLog(logFile);
```

### Required Arguments

#### **logFile**

The file path and name of the Spotfire S+ verbose log file. (The default is set by concatenating several global SAS macro variables.)

### Optional Arguments

None.

## TSCG\_INSERTSPLUSSCRIPT

The `tscg_InsertSPLUSScript` macro adds a Spotfire S+ verbose Script to the SAS Script file.

**NOTE:** Calls to this macro are typically given in the application macros. Users do not need to invoke this macro directly.

### Usage

```
tscg_InsertSPLUSScript(scriptFile);
```

### Required Arguments

#### **scriptFile**

The file path and name of the S-PLUS script. (Default is set via concatenation of several global SAS macro variables.)

### Optional Arguments

None.



# TSCG\_MAKELIST

The `tscg_MakeList` macro converts a space-delimited list of items into a user-specified delimited list of items with optional quotes. The code does not check for the presence of the separating commas beforehand, however, so use with caution!

## Usage

```
tscg_MakeList(inputString,  
              delim=%str(, ),  
              quote=T)
```

## Required Arguments

### inputString

A string to be converted into a comma-separated, quoted list.

## Optional Arguments

### delim

The delimiter to use for output. Default is a comma and a space.

### quote

A logical flag (0/1, F/T or N/Y) indicating whether the output items should be quoted. Default is T.

## EXAMPLE CALLS:

```
%tscg_MakeList("One" "Two" "Three");  
  
%tscg_MakeList('One' 'Two' 'Three');  
  
%tscg_MakeList(One Two Three);  
  
%tscg_MakeList(1 2 3);  
  
%tscg_MakeList(1 2 3, delim=%str( * ), quote=F);  
  
%tscg_MakeList(1 2 3, quote=T);
```

## TSCG\_PARSE

The `tscg_Parse` macro configures the default values needed for the TIBCO Spotfire Clinical Graphics SAS integration package.

**NOTES:** This macro sets the value of a global macro variable, `tscg_Parse`, to the value associated with the specified item in the HTTP response. Example uses of this macro are to find the `jobid` from an initial request or to find the status of an executing request.

Calls to this macro typically are given in the application macros. Users do not need to invoke this macro directly.

### Usage

```
tscg_Parse(dsname=&tscg_http.,  
           item=jobid);
```

### Required Arguments

None.

### Optional Arguments

#### **dsname**

The name of the dataset containing the HTTP response. (The default is the value of the global macro variable `tscg_http`.)

#### **item**

The name of the item in the first line of the HTTP response to be returned. (The default is the `jobid`.)

## TSCG\_PARSEBODY

The macro `tscg_ParseBody` parses the body of an HTTP response for a specific item identified by a string to match and an optional number of rows offset.

**NOTES:** This macro sets the value of a global macro variable, `tscg_ParseBody`, to the value associated with the specified item in the http response. Example uses of this macro are to find the TSCG file path root or other attributes and values contained within an HTTP response body. (See `tscg_Parse` for a macro to parse items from the HTTP response header.)

Calls to this macro are typically given in the application macros. Users do not need to invoke this macro directly.

### Usage

```
tscg_ParseBody (dsname=&tscg_http.,
               tag=,
               offset=0);
```

### Required Arguments

#### tag

A string to be located in the response body.

### Optional Arguments

#### dsname

The name of the dataset containing the http response. (Default is the value of the global macro variable `tscg_http`.)

#### offset

The offset (number of rows after tag location) of the desired data. Default is 0.

## TSCG\_SORT

The Sort Macro, `tscg_Sort`, provides the ability to sort the labels of a character variable by a numeric variable determining the order.

### Usage

```
tscg_Sort(targetVariable=,  
inputOrder=False,  
byVariable=,  
summaryFunction=,  
withinVariable=,  
withinVariableLevel=,  
smallToLarge=);
```

### Required Arguments

#### **targetVariable**

The column name for which levels are to be sorted.

#### **inputOrder**

TRUE if the target variable should be sorted using the input order from the source dataset. The default is FALSE. If this parameter is set to TRUE, any additional sorting requests are ignored.

### Optional Arguments

#### **byVariable**

The column name containing values by which to sort the target column. The default is unspecified NULL, in which case it is the alphabetic order of `targetColumn` that is considered.

#### **summaryFunction**

The summary function for ordering: max, mean, min, max, median and count. This argument is useful for ordering panels, so patient-panels are ordered according to the max of a lab parameter, for example.

#### **withinVariable**

The column name to sort within.

#### **withinVariableLevel**

The value of the `withinVariable` to sort within. (This argument is required if `withinVariable` is set.)

For example, sort `withinVariable=ARMCD`, and  
`wiwithinVariableLevel=Placebo`.

**smallToLarge**

Specify TRUE to sort ascending: small to large (default);  
otherwise FALSE to sort descending: large to small.

**Example**

```
%tscg_Sort(targetVariable=site,  
  byVariable=yield,  
  summaryFunction=max,  
  withinVariable=,  
  withinVariableLevel=,  
  smallToLarge=false);  
  
%tscg_barchart(dataLibrary = examples,  
  dataSet = barley,  
  outputFile=&outputPath.\sort,  
  outputFormat=png,  
  xVariable = year,  
  yVariable = yield,  
  groupVariable = site);
```

## TSCG\_STARTSUBPLOT

Use the macro `tscg_StartSubPlot` to create an SPJ plot using the specified SAS dataset.

### Usage

```
tscg_StartSubPlot(SPJList=&tscg_SPJList.);
```

### Required Arguments

None.

### Optional Arguments

#### SPJList

The name of a SAS dataset to hold the list of the SPJ files to include in the subplot. The default value is defined in the configuration macro.

## TSCG\_STRIPTAG

The strip tag macro `tscg_StripTag` strips the indicated XML opening and closing tags from a string.

**NOTES:** This macro "returns" the value of string with the XML tags removed.

Calls to this macro typically are given in the application macros. Users do not need to invoke this macro directly.

### Usage

```
tscg_StripTag(string=,  
tag=);
```

### Required Arguments

#### **string**

The string containing an item surrounded by an opening and closing set of XML tags.

#### **tag**

The text label of the XML tags to be removed.

### Optional Arguments

None.

# TSCG\_SUBPLOT

Use the macro `tscg_SubPlot` to create an SPJ graph

Usage

```
tscg_SubPlot(OutputFile=,
             GraphType=&tscg_outputFormat.,
             Columns=,
             Rows=,
             Orientation=,
             Size=,
             Units=IN,
             Width=7,
             Height=5.4,
             AsTable=TRUE,
             SPJList=&tscg_SPJList.);
```

Required Arguments:

OutputFile

Fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)

<b>Note</b>
You must ensure the specified output target location exists before you try to write output to a location. The macros do not automatically create a destination folder.

Columns

The value for the number of columns. (Only one of `Columns` and `Rows` is required but both are allowed.)

Rows

The value for number of rows. (Only one of `Columns` and `Rows` is required but both are allowed.)

Optional Arguments

GraphType

Space-delimited list of graph file formats for output.



**NOTE:** If an extension is specified on the `outputFile` parameter, this argument will be ignored. (Valid elements are EPS, JPG, PDF, PNG and WMF. The default is determined by the `tscg_outputFormat` variable defined in macro `tscg_configuration`.)

**Orientation**

Specifies the page orientation. Can be `Portrait` or `Landscape` (The default).

**Size**

Specifies the paper size. Can be `A4` (the default) or `A3`.

**Units**

Specifies the size and margin units. Can be `cm` (the default) or `in`.

**Width**

Specifies the page width. (If `Size` specified, `Width` is ignored.)

**Height**

Specifies the page height. (If `Size` specified, `Height` is ignored.)

**SPJList**

The name of a SAS dataset to hold the list of the SPJ files to include in the subplot. The default value is defined in the configuration macro.

## TSCG\_TESTBOOLEAN

The `tscg_TestBoolean` macro returns a consistent 0 or 1 value for any of various values indicating a boolean value. Can be T/F, Y/N, 0/1.

### Usage

```
tscg_TestBoolean(booleanString);
```

### Required Arguments

#### **booleanString**

The string to be tested as a boolean value. If the first character is Y, T, or 1 then a 1 is returned. If the first character is N, F, or 0, then 0 is returned. For any other value, an empty string is returned.

### Optional Arguments

None.

## TSCG\_VALIDATE

The `tscg_Validate` macro is used to validate that the needed server components are available. Examine the notes in the log for the following components:

- Tomcat.
- WebDAV.
- Spotfire Statistics Services.

**Usage** `tscg_Validate;`

**Required Arguments** None.

**Optional Arguments** None.

## TSCG\_WAIT

The macro `tscg_wait` configures the default values needed for the TIBCO Spotfire Clinical Graphics SAS integration package.

**NOTES:** This macro polls the TSCG server (at an interval set by the global macro variable `tscg_pollInterval`) until the job completes or the timeout value (set by the global macro variable `tscg_timeout`) is exceeded.

Calls to this macro are typically given in the application macros. Users do not need to invoke this macro directly.

### Usage

```
tscg_wait(dsname=&tscg_http.,  
          jobid=);
```

### Required Arguments

#### **jobid**

The `jobid` to wait for.

### Optional Arguments

#### **dsname**

Name of a dataset to contain the HTTP response. (The default is the value of the global macro variable `tscg_http`.)

# TSCG\_WRITEGRAPHJOB

The macro `tscg_WriteGraphJob` writes an XML **GraphJobSpec** file that is used by the TIBCO Spotfire Clinical Graphics package.

**NOTE:** Calls to this macro are typically given in the application macros. Users do not need to invoke this macro directly.

## Usage

```
tscg_WriteGraphJob(jobSpecFQName=,
  outputFQName=,
  graphDocFQName=,
  scriptFQName=,
  parmSet=&tscg_ParmSet.);
```

## Required Arguments

### jobSpecFQName

The fully-qualified name of the XML **GraphJobSpec** file to be created for specifying the job to execute.

### Note

jobSpecFQName **MUST** be a path and file name, NOT a URL.

### outputFQName

The fully-qualified name of the output graphic to be created. (Can be a path and file name or a URL.)

### graphDocFQName

The fully-qualified name of the GraphDocument to be used as the basis of the new graph. (Can be a path and file name or a URL.)

## Optional Arguments

### scriptFQName

The fully-qualified name of the Spotfire S+ script file executed to create the requested graph. (Can be a path and file name or a URL.)

### parmSet

The name of a dataset to hold run-time parameter names and values prior to them being written to the **GraphJobSpec** file. The default name is provided by the global macro variable `tscg_ParmSet` that is set in the `tscg_configuration` macro.



---

<b>Introduction</b>	<b>146</b>
<b>SAS Macro Errors</b>	<b>147</b>
<b>Job Processing Errors</b>	<b>148</b>
Server Not Running	148
Axis Scale Warning	148

## **INTRODUCTION**

This chapter describes possible errors with macros or general processing errors that SAS or TSCG can return.



## SAS MACRO ERRORS

When a call to one of the SAS macros results in an error, the macro displays a detailed message specifying the nature of the problem. For example:

```
[tscg_Legend] Please specify a valid value for the  
LEGENDLOCATION parameter (current value is  
&legendLocation.).
```

This error reports:

- The macro where the error occurred. (tscg\_Legend, in the above example.)
- A description of the error. (You supplied an erroneous parameter value.)
- The parameter where the error occurred. (legendLocation, in the above example.)
- If available, additional information.

For information about the macro and the parameter where the error occurred, see the macro topic in this reference manual.

## JOB PROCESSING ERRORS

This section describes other errors and you might see when trying to process a SAS job.

### Server Not Running

If Spotfire Statistics Services is not running when you submit a SAS job, you see the following error in the log:

```
ERROR: Connection refused.
```

### Axis Scale Warning

If the requested axis scale does not include the full range of data, you see the following warning.

```
Warning: ID-GOM-18W|Scale|Alert|To (3500) is less than max  
scale range 3855. Scale range is shortened. in  
FUN(...X.sub.i....)
```

That is, you have set the axis maximum to be less than the data maximum. The axis range is truncated based on the maximum you set, so you probably have data that is not included in the plot (data values that fall above the maximum you set for the axis).

### Output Error

Because most PDF readers take an exclusive lock on a file when they open it, if the specified PDF output file is already open in another application (for example Acrobat Reader) when you submit a SAS job, you might see the following error:

```
ERROR: [tscg_httpqrst] tscg server unable to process  
request.  
ERROR: [tscg_httpqrst] Contents of HTTP response listed  
below:  
...  
  
Problem in .JavaMethod("com.insightful.graphlet.conv...:  
com.insightful.graphlet.SPJConverter$ConversionException:  
Could not create output file: C:\TSCG\Sample\Graph.PDF (The  
process cannot access the file because it is being used by  
another process)
```

# INDEX

---

## A

area bar chart 24  
area plot 27  
Axis macro 7

## B

bar chart 30  
box plot 33  
bubble plot 35

## C

cumulative incidence plot 42

## D

delta plot 44  
dot plot 47  
double dot plot 51  
duration plot 55

## E

ECDF plot 38

## F

forest plot 58

## G

GraphDocument 143  
GraphJobSpec 143

graph table 13

## H

histogram 61

## I

image plot 64

## K

Kaplan-Meier plot 66

## L

Legend macro 14  
line plot 68

## M

matrix of scatter plots 81  
multiple-page graph option 17

## P

page layout 16  
population pyramid 72

## Q

qq comparison plot 77  
qqmath plot 75

## **R**

Reference Arrows 98  
Reference Curve 101  
Reference Points Element 103  
Reference Text Element 105

## **S**

scatterplot 79  
survival curve 83

## **T**

Themes macro 18  
Titles 19  
tree map plot 85  
tscg\_AddGraphParm 116  
tscg\_Axis 7  
tscg\_Configuration 117  
tscg\_CreateGraph 11  
tscg\_DeleteParmSet 119  
tscg\_FindPath 120  
tscg\_GetProgName 121  
tscg\_GetSourcePath 122  
tscg\_GraphOutput 123  
tscg\_GraphTable 13  
tscg\_HttpPost 124  
tscg\_HttpRqst 125  
tscg\_InsertGraphODS 126  
tscg\_insertgraphparm 116  
tscg\_InsertSPJName 128  
tscg\_InsertSPLUSLog 129  
tscg\_InsertSPLUSScript 130  
tscg\_Legend 14

tscg\_LSFit 92  
tscg\_LTSRobFit 93  
tscg\_MakeList 131  
tscg\_MMRobFit 94  
tscg\_Page 16, 17  
tscg\_Parse 132  
tscg\_ParseBody 133  
tscg\_PolyLSFit 95  
tscg\_RefABLines 96  
tscg\_RefArrow 98  
tscg\_RefCurve 101  
tscg\_RefPoints 103  
tscg\_RefText 105  
tscg\_RefXLines 107  
tscg\_RefYLines 110  
tscg\_ScatterPlot 79  
tscg\_Sort 134  
tscg\_StartSubPlot 136  
tscg\_StripTag 137  
tscg\_SubPlot 138  
tscg\_TestBoolean 140  
tscg\_Themes 18  
tscg\_Titles 19  
tscg\_validate 141  
tscg\_Wait 142  
tscg\_WriteGraphJob 143

## **X**

X-Reference Line 107

## **Y**

Y-Reference Line 110