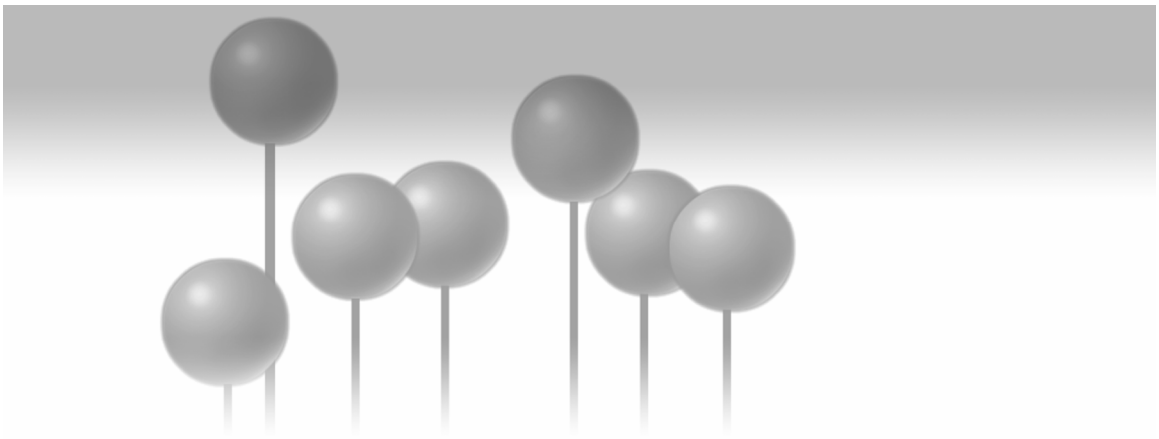


Part 1

SAS AppDev Studio

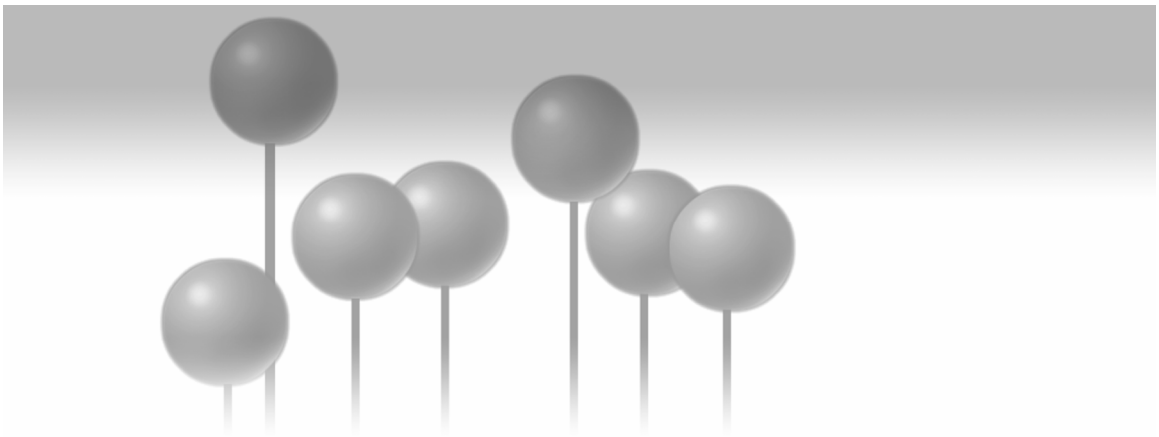
Chapter 1 Getting Started with SAS AppDev Studio 3

Chapter 2 SAS AppDev Studio Custom Tags and Attributes for Basic Graphics 13

Chapter 3 SAS AppDev Studio 3 Graph Model Tags 39

Chapter 4 Nested Tags for Graph Models 99

Chapter 5 Supporting Tags 125



Chapter 1

Getting Started with SAS AppDev Studio

1.1 Tag Libraries for SAS AppDev Studio 3

1.2 Getting Ready to Add Graphs to Your SAS AppDev Studio Project 8

1.3 Your Data 10

1.3.1 Tips and Information 11

1.1 Tag Libraries for SAS AppDev Studio

In this section we are going to look at creating graphs in a JavaServer Page (JSP) using the SAS Custom Tag Library. This tag library comes with the SAS AppDev Studio product, which includes webAF as the development environment tool. You can use webAF or another Java Interactive Development Environment (IDE), such as Eclipse, to utilize the tag libraries and API components that come with SAS AppDev Studio.

SAS AppDev Studio 2.x and SAS AppDev Studio 3 use separate tag libraries. SAS AppDev Studio 3 introduced some major enhancements to the graphics components. We'll look at the most commonly used graphs in each version.

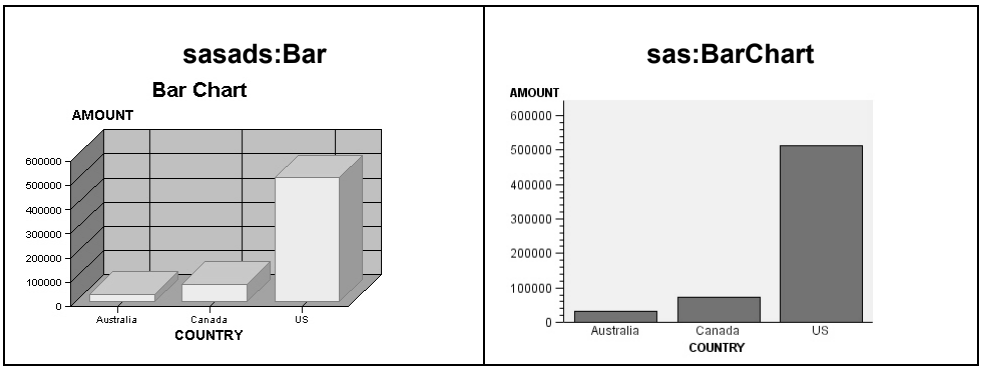
The easiest way to add graphs to your JSP or servlet project in SAS AppDev Studio is to use the Component Palette. The graph components are on the **Graphics** tab. You can change the palette by clicking the down arrow on the toolbar title bar.

We will be working with graphs on the SAS JSP/Servlet (Version 3) and SAS JSP/Servlet (Version 2) palettes.

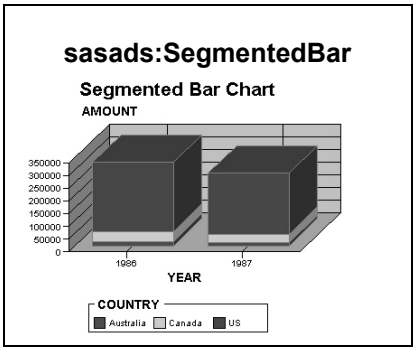
The version 2 tag library is part of SAS AppDev Studio 3. When you create a new SAS AppDev Studio 3 project, you can choose to include the version 2 tag library so that these tags are available in the component palette.

When you upgrade an existing webAF 2 project to use webAF 3 components, your old SAS AppDev Studio 2 tags (which have a sasads prefix) will still be in your project and will not be automatically updated to SAS AppDev Studio 3 tags (with a sas prefix). In other words, if you have a sasads:Bar chart in your project, it will not automatically be converted to a sas:BarChart if you move to webAF 3.

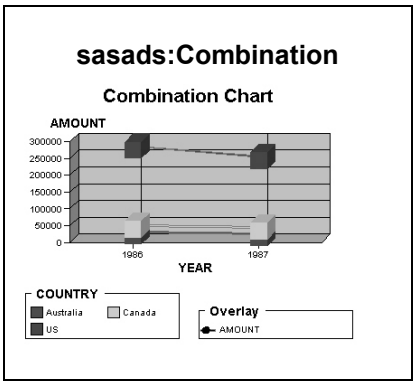
Here is an overview and comparison of the graphs available in both versions. Remember that sasads tags are from SAS AppDev Studio 2, whereas sas tags are from the more recent SAS AppDev Studio 3.



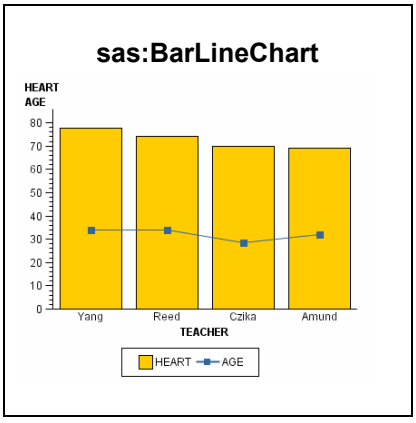
Here is a direct comparison of the default bar charts. Notice that the default for sasads:Bar is a three-dimensional chart, whereas the sas:BarChart is two-dimensional. Both have several options that can enhance the appearance.



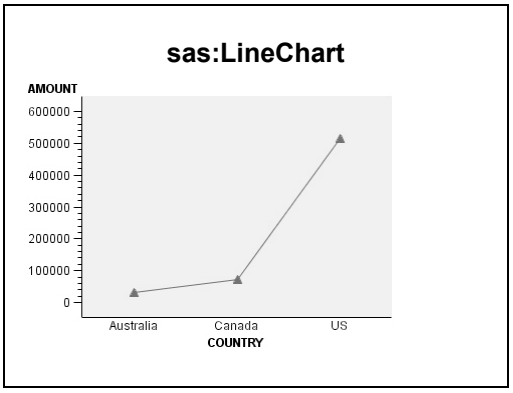
The sasads:SegmentedBar tag really does not have an equivalent in SAS AppDev Studio 3. However, you can mimic this graph by using the various model attributes with the sas:BarChart tag.



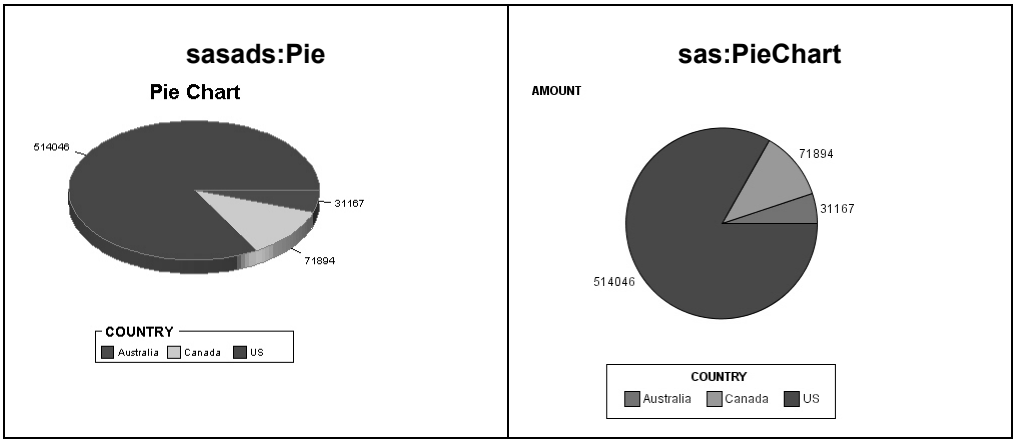
The sasads:Combination tag can be used in several different ways to perform similarly to a sas:BarLineChart, sas:LineChart, or sas:ScatterChart tag.



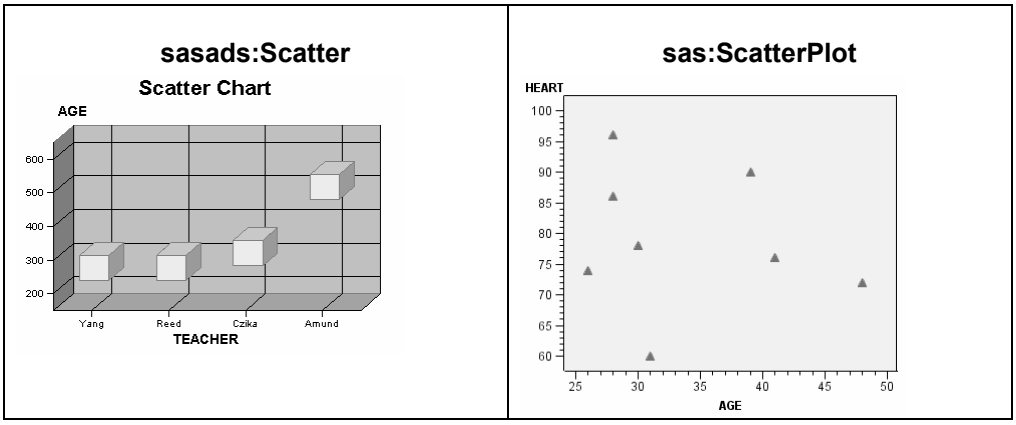
This sas:BarLineChart tag has no direct partner in SAS AppDev Studio 2 tags.



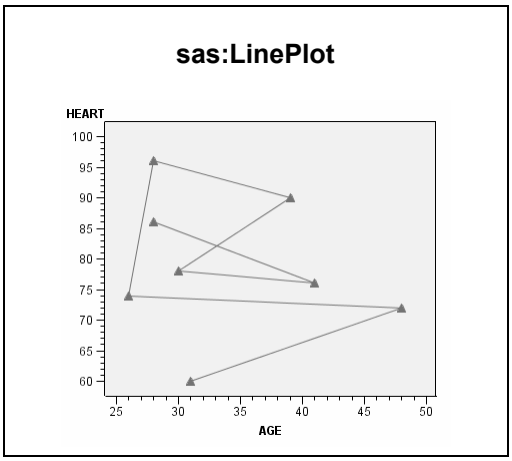
In SAS AppDev Studio 3, line charts can be created using the sas:LineChart tag. In SAS AppDev Studio 2, you might consider using the sasads:Combination tag.



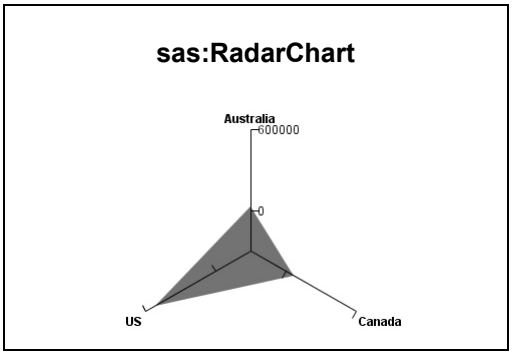
As with the bar charts, sasads:Pie defaults to a three-dimensional chart, and sas:PieChart is two-dimensional. Both charts have loads of options to create a number of different looks. Additionally, the sas:PieChart tag can create a donut and subgroup data into concentric rings.



SAS AppDev Studio 2 includes the sasads:Scatter tag. The equivalent tag in SAS AppDev Studio 3 is sas:ScatterPlot.



The sas:LinePlot tag could be compared to the previously shown SAS AppDev Studio 2 sasads:Combination tag.



The sas:RadarChart tag is a brand-new graph type for SAS AppDev Studio 3.

1.2 Getting Ready to Add Graphs to Your SAS AppDev Studio Project

There are a couple of things you need to consider before you can add graphs to your project. One is how to connect to the data. In your development environment, everything is local and easy to get to, but in normal production environments that usually isn't the case. You need to plan your strategy for accessing your SAS server. Is security an issue? How many people will be accessing this application? How many people at the same time might need to grab the same data? All these questions should be discussed with both the Web server and SAS server administrators. These administrative roles may be filled by the same person or by people from different groups. Either way, you must plan and coordinate with them. There are so many variables and combinations of system setup that we can't cover all of them in this book.

For this book, we've taken a simple approach and used a basic Java Database Connectivity (JDBC) connection for SAS AppDev Studio 3 tags and SAS/CONNECT for all others. If you need something more elaborate for your environment, review the SAS documentation for SAS Integration Technologies. This resource will help you make decisions on setting up security, pooling, connection types, etc.

Now that you have the connection, the next step is to create data models. Data models allow you to shape the data for graphing. For instance, you might need to sort the data alphabetically or by increasing values. Data models can also specify the columns, subset the data, and perform other functions.

In SAS AppDev Studio 2, one common method is to use a sasads:DataSet tag. This allows us to use the connection, specify the data, and shape it. Here's an example:

```
<sasads:DataSet connection="bbuConnection" dataSet="samples.grains"
  id="dsBar" scope="session" displayedColumns="country amount" />
```

In this example, we use the bbuConnection object to connect to our SAS server. Then the dataSet attribute is used to define the data set we want to use. Here we are using the Grains data set from the Samples library. Then, to help shape the data, we use the displayedColumns attribute. This allows us to specify only the columns we will be needing for the graph.

When using the SAS AppDev Studio 3 tags, we need to use different types of models. These may seem more complicated than simply using the sasads:DataSet tag, but they are more flexible. For sas:BarChart you use a com.sas.graphics.components.barchart.BarChartTableDataModel. Here's an example:

```
<jsp:useBean id="barChartTableDataModel1" scope="session"
  class="com.sas.graphics.components.barchart.BarChartTableDataModel">
  <jsp:setProperty name="barChartTableDataModel1" property="model"
   value="<%=jdbcTableModelAdaptor%>" />
</jsp:useBean>

<%
barChartTableDataModel1.setCategoryVariable(
  new com.sas.graphics.components.ClassificationVariable("COUNTRY"));
barChartTableDataModel1.setResponseVariable(
  new com.sas.graphics.components.AnalysisVariable("AMOUNT"));
%>
```

This snippet of code shows the configuration of a data model. It references a JDBCTableModelAdapter and sets properties for the data model, such as the category and response variables to use. See Chapter 11, "Final Reports," for a complete example with SAS AppDev Studio 3 tags.

1.3 Your Data

When you are using SAS data sets in SAS graphs, there are specific terms used to describe the types of variables. For instance, you may think in terms of an X axis and Y axis. However, because we are looking at statistical or computed variable data, most of the time each variable has a role. On the X axis you may actually be charting data by a category variable. Below are some terms that may help you along the way.

Category variable

A variable that determines the number and arrangement of bars, slices, lines, etc.

Response variable

The variable you are trying to understand, explain, or model.

Midpoint

The value associated with a bar on a bar or block chart or the slice on a pie chart. This is the category variable.

Chart variable

The data column to be charted. This variable can be character or numeric.

Chart statistic

Most commonly, the sum of a numeric variable or the frequency (count) of a character variable. Other common statistics are percentages and means. The statistics available vary by type of graph.

Midpoint axis

The axis that shows the categories of data.

Response axis

The axis that shows the range of values for the chart statistic.

Contiguous variables

Variables that contain a range of numeric values that are represented on the chart. For example, dollars or quantities are contiguous variables.

Discrete variables

Variables that contain a finite number of specific values that are represented on the chart. For example, years, geographical areas, and company divisions are discrete variables.

X

Used in plots (vertical) to identify the variable on the horizontal axis.

Y

Used in plots (vertical) to identify the variable on the vertical axis.

Z

Used in plots (vertical) to identify the depth variable.

For more information on terminology and graphing basics, see the SAS OnlineDoc documentation.

1.3.1 Tips and Information

Here are a few tips on how to organize your data for maximum performance.

- Usually, you should presummarize large data sets to improve performance.
- If you have a large number of different categories, you might want to subset or group the data to control the number of bars or pie slices shown in the chart. Depending on the size of your chart, it is easy to get so many bars that you cannot read the labels.
- When there are too many values to represent in a pie chart, the smaller values are automatically grouped into one slice labeled “Other.” By creating your own “Other” grouping, you can prevent a smaller, but important, category from being hidden.
- When you are accessing SAS data, the system often puts a lock on the data set you are using. For this reason, be sure to close all connection to the data once you are done. Or consider making your connection read-only and investigate how to set up workspace pooling or connection sharing techniques.

In short, the saying “Garbage in, garbage out” is so true. You need to be careful of how your data is constructed and formatted before you base critical decisions on your output.