



**Dear Readers,**

If you've ever attended a SAS event you've most likely gone home with a SAS brand item...a badge holder, can cooler, umbrella or t-shirt. Ever wonder how you can get more SAS items? Until now, it's been tough, but it just got easier!

Now you can purchase a number of SAS items through our on-line company store. The company store is operated by BrandFuel Promotions, and all monetary transactions are handled by BrandFuel. If you're hankering for some golf accessories, or a gift is needed for your office or home, check out the [online company store](#). They just might have what you want.

A handwritten signature in black ink that reads 'Shelley Sessoms'. The signature is fluid and cursive, with the first name being more prominent.

**Shelley Sessoms**  
Editor, *SAS Tech Report*

## Try This Demo: The ExcelXP Tagset and Microsoft Excel

---

### Introduction

This topic provides example code that uses the ExcelXP tagset to generate XML output. You can run the code in SAS 9.1 or a later release, and then open the XML files in Microsoft Excel 2000 or later. The output also works with the Calc spreadsheet program from [OpenOffice.org](http://OpenOffice.org).

To run these examples in SAS, first download the latest ExcelXP tagset from the [ODS MARKUP](#) page. That page also provides links to documentation for using and customizing tagsets.

See additional ExcelXP samples in these papers and presentations:

- [ODS MARKUP: The SAS Reports You've Always Dreamed Of \(.pdf\)](#) and the paper's [example code \(.tgz\)](#)
- [Example code for ODS Tagsets & Excel, an Excellent Combination \(.zip\)](#)
- [The Beginner's Guide to ODS MARKUP: Don't Panic! \(.pdf\)](#) and the paper's [example code \(.tgz\)](#).

---

### Skip Spacing

You can adjust the spacing between objects in a report by using the SKIP\_SPACE= option. The value of this option should be a comma-separated list of numbers that correspond to the spacing after tables, bylines, titles, footers, and pagebreaks.

The default values are:

- Table : 1
- Byline : 0
- Title : 1
- Footer : 1
- PageBreak : 1

```
ods tagsets.excelxp file="spacing.xls" style=statistical
  options( skip_space='3,2,0,0,1' sheet_interval='none'
           suppress_bylines='no' );
```

```
proc sort data=sashelp.class out=class;
  by age;
run;
```

```
proc print data=class;
  by age;
run;
```

```
ods tagsets.excelxp close;
```

## Suppress Bylines

You can use the SUPPRESS\_BYLINES= option to turn the bylines on and off.

```
ods tagsets.excelxp file="bylines.xls" style=statistical
  options( suppress_bylines='yes' sheet_interval='none' );

proc sort data=sashelp.class out=class;
  by age;
run;

proc print data=class;
  by age;
run;

ods tagsets.excelxp options( suppress_bylines='no' sheet_interval='none' );

proc print data=class;
  by age;
run;

ods tagsets.excelxp close;
```

---

## Table of Contents and Index of Tables

You can generate a table of contents and an index of tables by using the CONTENTS= and INDEX= options, respectively. Both the table of contents and index are hyperlinked to the corresponding worksheets in the report.

```
ods tagsets.excelxp file="toc.xls" style=statistical
  options( contents='yes' index='yes' );

proc print data=sashelp.class; run;

proc contents data=sashelp.class; run;

proc means data=sashelp.class; run;

ods tagsets.excelxp close;
```

---

## Data-Driven Worksheets

By default, a new worksheet is generated for each table. You can override this and create a new worksheet for each procedure, bygroup, or never. The example here creates a new worksheet for each bygroup.

```
ods tagsets.excelxp file='multisheet.xls' style=statistical
  options( sheet_interval='bygroup' );

proc sort data=sashelp.class out=class;
  by age;
```

```

run;

proc print data=class;
  by age;
run;

ods tagsets.excelxp close;

```

---

## Manual Multiple Worksheets

Use the SHEET\_NAME= and SHEET\_INTERVAL= options to specify the name of a worksheet as well as the interval in which to create new worksheets.

```

data prdsale;
  set sashelp.prdsale;
  difference = actual-predict;
run;

proc sort data=prdsale; by country region division year; run; quit;

title; footnote;

*;
* Create a workbook with multiple tables per worksheet, and specify
* the name for the worksheets. Autofilters will be applied to the
* second table in each worksheet. The SUM statement of PROC PRINT
* will result in a subtotal row, but note that the SUBTOTAL function
* is not used for the DIFFERENCE column because a formula was specified on
* the SUM statement. In this case, the user-specified formula takes
* precedence over the auto subtotal.
*;

ods listing close;

ods tagsets.excelxp file='multitable.xml' style=statistical
  options(auto_subtotals='yes' default_column_width='7, 10, 10, 7, 7'
    frozen_rowheaders='yes' sheet_interval='none'
sheet_name='Canada'
    autofilter='all' autofilter_table='2');

*;
* The output from the following two procs will be in a single worksheet
* with a user-specified name of 'Canada'.
*;

proc tabulate data=prdsale;
  where country eq 'CANADA' and year eq 1993;
  var predict actual;
  class region division prodtype;
  table
    region*(division*prodtype all={label='Division Total'})
all={label='Grand Total'},
  predict={label='Total Predicted Sales'}*f=dollar10.*sum={label='' }
  actual={label='Total Actual Sales'}*f=dollar10.*sum={label='' };
run; quit;

```

```

proc print data=prdsale noobs label split='*';
  where country eq 'CANADA' and year eq 1993;
  id country region division;
  var prodtype product quarter month year;
  sum predict / style={tagattr='format:Currency'};
  sum actual / style={tagattr='format:Currency'};
  sum difference / style={tagattr='format:Currency formula:RC[-1]-RC[-
2]'};
  label prodtype = 'Product*Type'
        predict  = 'Predicted*Sales'
        actual   = 'Actual*Sales';
run; quit;

ods tagsets.excelxp options(sheet_interval='none' sheet_name='Germany');

*;
* The output from the following two procs will be in a single worksheet
* with a user-specified name of 'Germany'.
*;

proc tabulate data=prdsale;
  where country eq 'GERMANY' and year eq 1993;
  var predict actual;
  class region division prodtype;
  table
    region*(division*prodtype all={label='Division Total'})
all={label='Grand Total'},
    predict={label='Total Predicted Sales'}*f=dollar10.*sum={label='' }
    actual={label='Total Actual Sales'}*f=dollar10.*sum={label='' };
run; quit;

proc print data=prdsale noobs label split='*';
  where country eq 'GERMANY' and year eq 1993;
  id country region division;
  var prodtype product quarter month year;
  sum predict / style={tagattr='format:Currency'};
  sum actual / style={tagattr='format:Currency'};
  sum difference / style={tagattr='format:Currency formula:RC[-1]-RC[-
2]'};
  label prodtype = 'Product*Type'
        predict  = 'Predicted*Sales'
        actual   = 'Actual*Sales';
run; quit;

ods tagsets.excelxp options(sheet_interval='none' sheet_name='United
States');

*;
* The output from the following two procs will be in a single worksheet
* with a user-specified name of 'United States'.
*;

proc tabulate data=prdsale;
  where country eq 'U.S.A.' and year eq 1993;
  var predict actual;
  class region division prodtype;

```

```

table
  region*(division*prodtype all={label='Division Total'})
all={label='Grand Total'},
  predict={label='Total Predicted Sales'}*f=dollar10.*sum={label='' }
  actual={label='Total Actual Sales'}*f=dollar10.*sum={label='' };
run; quit;

proc print data=prdsale noobs label split='*';
  where country eq 'U.S.A.' and year eq 1993;
  id country region division;
  var prodtype product quarter month year;
  sum predict / style={tagattr='format:Currency'};
  sum actual / style={tagattr='format:Currency'};
  sum difference / style={tagattr='format:Currency formula:RC[-1]-RC[-
2]'};
  label prodtype = 'Product*Type'
        predict = 'Predicted*Sales'
        actual = 'Actual*Sales';
run; quit;

ods tagsets.excelxp close;

ods listing;

```

---

## Autofilters

Filters enable you to visually subset your data within Excel. The values for AUTOFILTER= are ALL, a column number (e.g., 3), or a range of column numbers (e.g., 3-5).

The example below specifies ALL, which causes Excel to display a drop-down menu that contains all of the values within a column. When you select one of these values, the rows that don't match that value collapse.

```

ods listing close;

ods tagsets.excelxp file='filters.xls' style=statistical
  options(autofilter='all');

proc print data=sashelp.class; run;

ods tagsets.excelxp close;

ods listing;

```

---

## Data-Specified Formulas

If a data point starts with an equal sign, that cell is evaluated as a formula in Excel.

```

ods tagsets.excelxp file="dataeq.xml" options(zoom='75' debug_level='5');

/* Interestingly, this first example doesn't work, although the XML looks
* perfectly fine. Excel doesn't seem to like absolute cell

```

```

* references. If the formulas are fixed, and saved, the
* resulting XML replaces all absolute references with relative
* ones.
*/

data foo;
  length a b 8 c $20;
  input a b c $;
  cards;
  1 2 3
  2 3 =A3+B3
  3 4 =A4+B4
  . . =SUM(C2:C4)
  ;
run;

proc print noobs;
run;

data foobar;
  length a b 8 c $20;
  input a b c $;
  cards;
  1 2 3
  2 3 =RC[-2]+RC[-1]
  3 4 =RC[-2]+RC[-1]
  . . =SUM(R[-3]C:R[-1]C)
  ;
run;

proc print noobs;
  var a b;
  var c / style(head) = {flyover="Hello World"}
  style(data) = {cellwidth=10pt};
run;

ods tagsets.excelxp close;

```

---

## Formats and Formulas

You can use Excel formats and formulas by embedding them in the TAGATTR= style option.

```

data prdsale;
  set sashelp.prdsale;
  difference = actual-predict;
run;

proc sort data=prdsale; by country region division year; run; quit;

title; footnote;

ods listing close;

ods tagsets.excelxp file='formulas.xls' style=statistical;

```

```

proc print data=prdsale noobs label split='*';
  where country eq 'CANADA' and year eq 1993;
  id country region division;
  var prodtype product quarter month year;
  sum predict / style={tagattr='format:Currency'};
  sum actual / style={tagattr='format:Currency'};
  sum difference / style={tagattr='format:Currency formula:RC[-1]-RC[-
2]'};
  label prodtype = 'Product*Type'
        predict = 'Predicted*Sales'
        actual = 'Actual*Sales';
run; quit;

ods tagsets.excelxp close;

ods listing;

```

---

## Embedded Titles and Footnotes

Use the EMBEDDED\_TITLES= and EMBEDDED\_FOOTNOTES= options.

```

ods listing close;

ods tagsets.excelxp file="embed.xls" style=statistical
  options( embedded_titles='yes' embedded_footnotes='yes' );

title "My First Title";
title3 "My Third Title";

proc print data=sashelp.class (obs=5); run;

footnote "My First Footnote";
footnote3 "My Third Footnote";

ods tagsets.excelxp close;

ods listing;

```

---

## Default Column Width and Width Fudge

You can set the default character widths of columns by using the DEFAULT\_COLUMN\_WIDTH= option. The WIDTH\_FUDGE= option is also used to adjust the column width based on font parameters. The default width fudge is 0.75.

```

title; footnote;

*;
* Illustrate the use of split characters and column justification.
* Use WIDTH_FUDGE= to get slightly wider columns.
*;

ods listing close;

```

```

ods tagsets.excelxp file='report.xml' style=statistical
options(width_fudge='0.8');

proc print data=sashelp.class noobs label split='*';
  var name          / style={just=r};
  var age sex       / style={just=c};
  var height weight / style={just=l};
  label name      = 'Student*Name'
        age       = '*Age'
        sex       = '*Gender'
        height    = 'Height*(inches)'
        weight    = 'Weight*(pounds)';
run; quit;

*;
* PROC REPORT is not setting column widths.  So we must force them with
* DEFAULT_COLUMN_WIDTH=.  Reset WIDTH_FUDGE= to the default value of
"0.75".
*;

ods tagsets.excelxp options(default_column_width="7.5, 7.5, 5, 7.5, 7.5"
width_fudge='0.75');

proc report data=sashelp.class nowindows split='*';
  column name sex age height weight;
  define name / display 'Student*Name' left style={just=l};
  define sex / display '*Gender' right style={just=r};
  define age / display '*Age' center style={just=c};
  define height / display 'Height*(inches)' center style={just=c};
  define weight / display 'Weight*(pounds)' center style={just=c};
run; quit;

ods tagsets.excelxp close;

ods listing;

```

---

## Rotated Column Headers

You can use the `vert_excelxp` tagset to create tables with rotated column headers.

```

/*-----*/
/*-- Column widths will be too wide because the tagset thinks --*/
/*-- the headers are horizontal. --*/
/*-----*/

proc template;
  define style styles.mystyle;
    parent=styles.default;
    style vertical_header from header /
      tagattr = 'rotate:45'
  ;
end;
run;

```

```

ods tagsets.excelxp
  style=mystyle
  file="test.xls"
  options(absolute_column_width="4,8,4,3,4,5"
         row_heights="30"
        );

proc print data=sashelp.class;
  var name / style(header) = vertical_header;
  var age sex;
  var weight height / style(header) = vertical_header;
run;

ods _all_ close;

```

---

## Flyovers

You can create a tooltip for a cell by using the FLYOVER= style attribute.

```

proc sort data=sashelp.prdsale out=prdsale;
  by country region division year;
run; quit;

title; footnote;

ods listing close;

ods tagsets.excelxp file='flyover.xls' style=statistical;

proc print data=prdsale noobs label split='*';
  where country eq 'CANADA' and year eq 1993;
  id country region division;
  var prodtype product quarter month year;
  sum predict / style={tagattr='format:Currency' flyover='Predicted
Total'};
  sum actual / style={tagattr='format:Currency' flyover='Actual Total'};
  label prodtype = 'Product*Type'
        predict   = 'Predicted*Sales'
        actual    = 'Actual*Sales';
run; quit;

ods tagsets.excelxp close;

ods listing;

```

## **A Look at Doug Samuelson**

### **SAS Silver Circle Winner**

*How often do you have the chance to earn \$2 billion for your employer?*

Doug Samuelson had that opportunity while working for the US Department of Energy in 1980. The government suspected that a major oil company had broken an energy pricing regulation in the mid-1970s. At the time, the government had price controls on oil pumped from existing fields, but not from new ones.

Using SAS and the Proc PLOT text-based plotting function, Samuelson and a team at the Department of Energy proved when the company had engineered a large field, pushing all of the oil from a series of existing wells near the edge of the oil field toward another series of wells near the middle. It was the same oil and the same field – just a shift in where it was pumped out. Under federal regulation, when a field was “unitized” in this way, it had to be reported as a unit as soon as “a significant change in producing patterns” took place; this is because the way it was reported greatly affected whether the oil was subject to a price ceiling. To prove when this had happened, the government team needed to process and summarize month-by-month production reports for more than a thousand properties that were part of the unit.

#### **SAS® makes quick work of analyzing data**

Samuelson was amazed at how quickly SAS helped him analyze the data – a few days to run the plots, as he recalled. Remember, these were the days of bulky mainframes and programs that took all night to run, and other people’s attempts to analyze the data had consumed weeks without success. The Department of Energy team plotted the results onto an aerial map of the oil field, and it became a key piece of evidence. “We had a ring of red dots near the periphery of the field. About a third of the way from the red ring to the center was a nearly concentric ring of yellow dots,” Samuelson recounted. The trial court imposed a \$2 billion judgment against the oil company, and the US Supreme Court upheld it.

Samuelson believes the opportunity to plot the results visually had a dramatic impact on the case. “We needed a simple picture that anyone could understand,” said Samuelson. “We created the most valuable picture in the history of the world. With all due respect to da Vinci, Rembrandt and van Gogh, what other picture, anywhere, anytime, ever fetched \$2 billion?”

#### **Samuelson’s use of SAS® covers multiple arenas**

The oil case was one of the first times Samuelson used SAS, and he has rarely been without the software since. “It boosted my career. I knew I needed to keep using it,” he said. Since then, the operations research analyst has used SAS products in jobs with numerous government agencies, private-sector companies and his own firm. Operations research, his specialty, involves the analysis of outcomes and determinants of outcomes in complex systems using predictive modeling, statistical pattern recognition and forecasting techniques.

An adjunct faculty member at George Washington University and the University of Pennsylvania, Samuelson is a longtime contributor to *OR/MS Today*. “He is the academic equivalent of a syndicated newspaper columnist. He’s got quite a following,” said John Sneed, President of TESLA Inc., an electric systems load forecasting provider, who has been Samuelson’s colleague for many years. Samuelson is also an inventor and entrepreneur. In 1989, he patented predictive dialing for outbound telephone call centers, for which he was a finalist for the 1998 Wagner Prize for OR practice.

SAS helped Samuelson and his employers root out Medicare fraud and look for possible patterns of fraud and error in vote counting in Ohio following the 2004 election. (Note to conspiracy theorists: He and his colleagues didn’t find evidence of enough wrongdoing to change the outcome of the election.) He’s looked at the health effects of hazardous materials. He’s examined the costs to industry of tighter

regulations – a project he undertook for the Federal Aviation Administration when the government wanted to know how controlling noise around airports would affect the airlines.

He is currently working for the Homeland Security Institute, a privately owned, federally funded research and development center in Arlington, Virginia, where he uses JMP®, statistics software made by SAS, for visualizing and uncovering data patterns. For example, he's used JMP to analyze data on which protective measures appear to deter potential terrorists. He's helped researchers examine the reasons Hispanics stick with post-secondary education. Using SAS, he designed the first computational project for the introductory engineering statistics class at George Washington University.

One of Samuelson's recent favorites was a Medicare project in 2000. It takes a few months for the Social Security Administration to process a death certificate. One of Samuelson's colleagues noticed that hospitals make discharge data, including condition at discharge, readily available well before other government records are updated. By matching the data from Medicare's databases – and analyzing it using SAS and Dataprobe – the government could quickly determine which durable medical goods providers were billing for services well beyond the beneficiary's date of death. As a result, the government recovered more than \$50 million.

The power and ease of use offered by SAS have been particularly fruitful for Samuelson's consulting work. "Most operations research is done within the confines of large companies or large consulting firms. To succeed as a one-person shop is quite an achievement," notes colleague Sneed. "He's done innovative work."

## **A Look at Frank Dilorio**

### **SAS Silver Circle Winner**

*There aren't many people who would tell you that a software program changed their life. Frank Dilorio is one of the few.*

The Chapel Hill, NC, resident was a graduate student in city planning at the University of North Carolina Chapel Hill in 1975 when he earned a fellowship that required computer expertise. All he knew at the time was a bit of the Fortran programming language. His university employer handed him a copy of SAS and a batch of survey data. "The whole document for the software fit into 250 pages," recalls Dilorio. He felt lucky that his employers were using software built just down the road – in nearby Cary, NC. "If you were having a problem you could just bring your plotter over and they'd help you."

#### **His first big break**

Dilorio felt lucky to have learned a marketable skill, since job openings for city planners were scarce when he received his degree. Instead, he used his new SAS programming skills to find other jobs at UNC and then got a big break when a colleague recommended him for a job with the Bank of New Zealand. He had been eager to live overseas, and his knowledge of SAS was critical to making the deal happen. In fact, his first task at the bank was to help figure out whether to use SAS or an IBM product to run some reports. "I'll never forget, we did a benchmark and SAS ran in one-seventeenth the time it took to run the reports using the IBM product."

Back in the United States, he discovered that although there wasn't a well-defined niche for SAS programmers, "it was obvious there was a need," and Dilorio gave up any thoughts of using his city planning degree. "Over the years, SAS programmer evolved from a curiosity into a job category with consistently high employment opportunities."

#### **A career using SAS®**

Today, Dilorio's resume includes 14 jobs where he has used SAS. He is President of [CodeCrafters Inc.](#), of Chapel Hill. His current client is Rho Inc., a contract research organization. A developer with expertise in metadata and tools, Dilorio is helping Rho streamline drug development submissions to the Food and Drug Administration. Every day a submission is delayed costs drug developers thousands of dollars. So when Dilorio wrote a program in SAS to embed and verify hyperlinks that are required in every submission, it saved money. Before that, Rho sent the hyperlink work to a subcontractor at the cost of a two-week delay.

Rho's head of submissions, Jeff Abolafia, brought Dilorio on at Rho – thrilled at the opportunity to work with a man he considers a mentor. "I started going to his talks in 1985 when I was a biostatistics student at UNC. He blew me away. He had a big influence on my philosophy toward programming, and I learned so much from him on how to check my work."

At every turn in his career Dilorio has advocated for SAS because his experience with niche vendors was always so poor. "You might have a vendor that has a more elegant solution for one type of statistical problem, but overall you do better if you use SAS for everything rather than a bunch of 'best of breed' niche players." It's the difference, says Dilorio, of potentially making multiple calls to multiple vendors' tech support lines or just one call to SAS in the rare event of an error.

#### **A steadfast advocate**

Many times in his early years of using SAS, Dilorio's clients and employers needed to take a "leap of faith . . . to select a SAS tool that didn't quite fit into a specific category and (believe) it would get the job done. These experiences have always been exciting and intellectually demanding and the clients have always been happy."

He's never wavered from his own desire to work with SAS, even when colleagues expressed uncertainty. Dilorio has enjoyed opportunities to share his experience with other SAS users. He began presenting papers at SAS Users Group International conferences in the early 1980s, "then found an even bigger outlet for my urge to write." Dilorio is the author of [SAS Applications Programming: A Gentle Introduction](#) and co-authored [Quick Start to Data Analysis with SAS](#). He is a popular mentor on SAS user boards, co-chaired 1994 and 1996 conferences of SouthEast SAS Users Group and traveled throughout the U.S., U.K., Australia and New Zealand presenting papers to users groups. "As with writing the books, this is a part of my professional life that I never even remotely considered when I started out."

"In a real sense, SAS has 'been' my career and has provided job, writing and other opportunities that were as unexpected as they were welcome. Publishing, conferences and job opportunities may be taken for granted by the next generation of SAS programmers, but I will always be impressed with what a varied, fertile professional world I've been able to inhabit."

## Limited Time Training Offers

Pre-purchase training units from SAS and use them for a variety of training services.

[Corporate customers](#) can save up to 34% when you pre-purchase SAS training before Dec 31. For [academic customers](#), your 50% discount still applies, and with prices increasing in 2007, you'll spend less when you pre-purchase training before Dec 31.

Don't need a block of training units? Register and pay for an individual 2007 public course before Dec. 31 and you'll pay the 2006 price, saving you up to \$150. [Save more](#) by also taking advantage of our standard discounts during this limited time offer.

JMP users can take advantage of a special 2-for-1 discount on any JMP public training course scheduled between now and the end of 2006. [Read more](#) about this terrific deal.

## Technical Support Hot-Fix Site Has New Features to Help Identify Problems That Have ALERT Status

*Hot fixes* address software issues that are either discovered by customers as they use their applications or that are found during internal testing in post production. For some customers, a hot fix provides a quick solution to an immediate problem. For others, the hot fix addresses issues that affect system maintenance. Recently, some new features have been added to the [Technical Support Hot Fixes](#) Web site to help you quickly identify hot fixes for issues that have an ALERT status. *ALERT status* is given to problems that you need to be aware of before you install or use the software, such as problems related to security, data integrity, or incorrect output. The hot-fix Web site, which provides downloads for customers, also includes the following features:

- For SAS 9.1.3, the site has a separate download page that lists hot fixes only for problems that have an ALERT status:

[http://ftp.sas.com/techsup/download/hotfix/e9\\_sbc\\_home\\_alert.html](http://ftp.sas.com/techsup/download/hotfix/e9_sbc_home_alert.html)

If you are running SAS 9.1.3 with Asian language support, see [download page for DBCS systems](#).

**Note:** While the new page contains just a list of the issues that have an ALERT status, those issues are also included in the **comprehensive** list of hot fixes that are available for each operating system.

- On the [Additional SAS Products and Solutions](#) page, the symbol **A** appears next to the products/solutions that contain fixes for issues that have an ALERT status.
- The external Web site has an [FAQ](#) with additional, general information about issues that have an ALERT status. (Information about the new features is also available.)

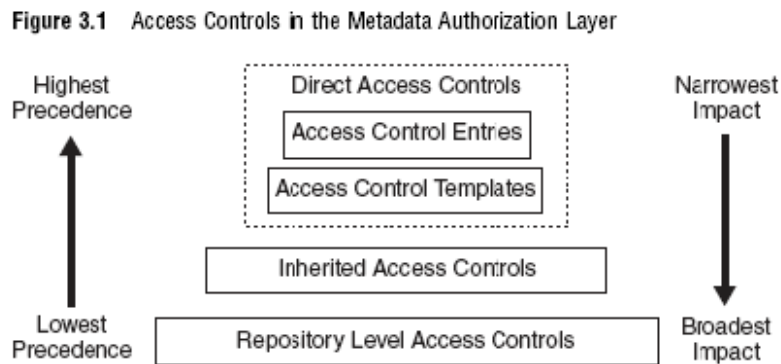
Along with information about the ALERT hot fixes, users frequently want to know such things as how to determine which hot fixes they actually need, whether they really need to install hot fixes at all, and so on. You can find the answer to these and other commonly asked questions in the article [SAS Hot Fixes: An Evolving Process](#). For additional details about applying hot fixes, see the [Hot Fix FAQ](#).

## Metadata Security and the DefaultACT in SAS®9

When designing the metadata architecture several years ago, the goal was to provide a flexible architecture that would not have to be tied to a hierarchical structure. To achieve this, object associations were used to tie different metadata elements together. Some associations provide additional context, and some associations provide security inheritance.

To determine authorizations, we first look for a direct access control entry (ACE) or access control template (ACT) on the object. If an ACE or an ACT does not exist, security associations are searched until an ACE is found. If all associations are traversed without finding explicit authorizations, then the authorizations in the DefaultACT will apply.

The following diagram from the SAS 9.1.3 Intelligence Platform: Security Administration Guide<sup>1</sup> shows the order of precedence.



If authorizations are not placed directly on an object, security inheritance is based on the object's security associations. Objects can be associated to a folder in a content tree and/or associated to another "parent" object. If you update an association that participates in security inheritance, you must have WriteMetadata (WM) authorization to both the original object and the associated object. For example, to register a library to a server context, you must have WM authorization granted on both the library and the server context.

### Metadata Security Implications in SAS 9.1.3

Due to this architecture, you must grant ReadMetadata (RM) and WM authorizations in the repository-level access control template (DefaultACT) for any groups and users that will read and write metadata.

- There are multiple object types, such as access control templates, server contexts, users, and groups, that do not have security associations to a content tree. These "loose" objects have authorizations applied directly from the DefaultACT.
- When new objects are created, authorizations are checked when the object is first created, before associations are added. Therefore, by default, newly created objects have authorizations applied directly from the DefaultACT. This means that any user who creates a new object must have WM authorization granted at the repository level.
- Objects that you see in end-user applications (such as reports, stored processes, ETL jobs, and so on) are called "logical objects" because they are referenced as a single metadata object, but they are actually made up of multiple, individual metadata elements. Some of these elements (such as resource templates for defining libraries and servers) are also "loose" objects, such that their authorizations are determined only by the DefaultACT. This means that WM authorization must be granted in the DefaultACT to update these associations.

<sup>1</sup> <http://support.sas.com/documentation/configuration/bisecag.pdf>

## Troubleshooting SAS Intelligence Servers (Part III)

The following is a list of objects, names, and locations, including description and notes to help you in troubleshooting your environment in SAS Services Application, Web Report Studio, Web Report Viewer, LSF, and the JobScheduler:

### SAS Services Application

- a. login.conf - C:\SAS\Metadata server access information
- b. logging\_config\_svc.xml -  
C:\SAS\Remote Services log4j logging definition file
- c. services.log -  
C:\SAS\Remote Services SAS core log file.

### Web Report Studio

- a. deployment.html - C:\Program Files\SAS\SASWebReportStudio\3.1  
Configuring and deploying SAS Web Report Studio
- b. wrs.config - C:\Program Files\SAS\SASWebReportStudio\3.1  
WRS configuration file populated during the configuration process - running the Configuration Wizard
- c. sas.wrs.config.bat - C:\Program Files\SAS\SASWebReportStudio\3.1  
This will create and populate folders named live, wrspackaging, and remote, and it will create SASWebReportStudio.war. This file is called when the Configurations Wizard is executed on the mid-tier.
- d. sas.wrs.unconfigure.bat -  
C:\Program Files\SAS\SASWebReportStudio\3.1  
Removes the output created by the previously executed sas.wrs.config.bat.
- e. SASWebReportStudio.war -  
C:\Program Files\SAS\SASWebReportStudio\3.1  
Web Report Studio WAR file location of exploded WRS application.
- f. sas.wrs.tomcat.deploy.bat -  
C:\Program Files\SAS\SASWebReportStudio\3.1  
If you are deploying Web Report Studio to run in Tomcat, execute this file.
- g. sas.wrs.weblogic.prepare.bat -  
C:\Program Files\SAS\SASWebReportStudio\3.1  
If you are deploying WebReport Studio to run in BEA WebLogic, execute this file.
- h. WebReportStudio.log -  
C:\SAS\Web Report Studio log file

### Web Report Viewer

- a. deployment.html - C:\Program Files\SAS\SASWebReportViewer\3.1  
Configuring and deploying SAS Web Report Viewer
- b. wrv.config - C:\Program Files\SAS\SASWebReportViewer\3.1  
WRV configuration file populated during the configuration process - running the Configuration Wizard
- c. sas.wrv.config.bat - C:\Program Files\SAS\SASWebReportViewer\3.1  
This will create and populate folders named live, wrspackaging, and remote, and it will create SASWebReportViewer.war. This file is called when the Configurations Wizard is executed on the mid-tier.
- d. sas.wrv.unconfig.bat -  
C:\Program Files\SAS\SASWebReportViewer\3.1  
Removes the output created by the previously executed

- sas.wrv.config.bat.
- e. SASWebReportViewer.war -  
C:\Program Files\SAS\SASWebReportViewer\3.1  
Web Report Viewer WAR file
- f. sas.wrv.tomcat.deploy.bat -  
C:\Program Files\SAS\SASWebReportViewer\3.1  
If you are deploying Web Report Viewer to run in Tomcat, execute this file.
- g. sas.wrv.weblogic.prepare.bat -  
C:\Program Files\SAS\SASWebReportViewer\3.1  
If you are deploying Web Report Viewer to run in BEA WebLogic, execute this file.
- h. WebReportViewer.log -  
C:\SAS\<configdir>\Lev1\web\Deployments\WebReportStudio\logs  
Web Report Studio log file

### LSF

- a. Isf.conf - C:\LSF\_60\etc  
LSF configuration file
- b. passwd.lsfuser - C:\LSF\_60\conf  
File that contains the passwords for users who run jobs in LSF
- c. Isf.cluster.sas\_cluster - C:\LSF\_60\conf  
Cluster definition file. Contains LSF Admin IDs.
- d. Isf.cluster.sas\_cluster - C:\LSF\_60\logs  
Location of LSF log files

### JobScheduler

- a. js.conf - C:\Program Files\Platform Computing\Platform Process  
Manager\conf  
JS configuration file
- b. js.conf - C:\Program Files\Platform Computing\Platform Process  
Manager\logs  
JS Log file location

Also see the following:

- [SN-013078](#) SAS Intelligence environment log and configuration Files to assist you in troubleshooting (part I)  
(SAS Software Navigator, Configuration Wizard, Metadata Server, Object Spawner, Workspace Server, Stored Process Server, and Batch Server)
- [SN-013084](#) SAS Intelligence environment log and configuration files to assist you in troubleshooting (part II)  
(Share, OLAP, and Connect servers and Portal)
- [SN-013087](#) SAS Intelligence environment log and configuration Files to assist you in troubleshooting (part III)  
(SAS Services Application, Web Report Studio, Web Report Viewer, LSF, and JobScheduler)

**Product:** Base SAS  
**Component:** Installation  
**Priority:** N/A  
**Note Type:** Usage Issue  
**Date:** Mon, 14 Aug 2006

## Operating System and Source Fix Information

<b>System</b>	<b>Release Reported</b>	<b>Release Fixed</b>
Windows NT	9.1.3 TS1M3	
Windows 2000 Datacenter Server	9.1.3 TS1M3	
Windows 2000 Professional	9.1.3 TS1M3	
Windows Server 2000 family	9.1.3 TS1M3	
Windows Server 2003 family	9.1.3 TS1M3	
Windows XP	9.1.3 TS1M3	

Unless otherwise stated above, no fixes are available for this issue.

## **Webcasts and Events**

### **M2006**

**Oct. 23-24**

**Las Vegas**

Network with colleagues from all over the world and get the most up-to-date information in the industry from data mining's top thought leaders and practitioners. Be sure to attend the reception sponsored by SAS on Monday evening.

### **Retention and Churn Modeling**

**On-Demand Webcast**

**Available Oct. 11**

What's the most reliable and consistent way to uncover the causes of customer defection? How do you use that information to prevent future loss? Join David Ogden, Analytical Consultant for SAS, as he discusses these and other issues.

### **Inside SAS: Spotlight on Information Quality**

**On-Demand Webcast**

Speaker Larry English, President and Principal of INFORMATION IMPACT International, gives you clearer insights into the process improvements, organizational changes, information governance and stewardship necessary to create a sustainable information quality culture.